Data-Driven Approaches for Emissions-Minimized Paths in Urban Areas

Jan Fabian Ehmke, Ann Melissa Campbell, Barrett W. Thomas

April 12, 2015

Abstract

Concerns about air quality and global warming have led to numerous initiatives to reduce emissions. In general, emissions are proportional to the amount of fuel consumed, and the amount of fuel consumed is a function of speed, distance, acceleration, and weight of the vehicle. In urban areas, vehicles must often travel at the speed of traffic, and congestion can impact this speed particularly at certain times of day. Further, for any given time of day, the observations of speeds on an arc can exhibit significant variability. Because of the nonlinearity of emissions curves, optimizing emissions in an urban area requires explicit consideration of the variability in the speed of traffic on arcs in the network. We introduce a shortest path algorithm that incorporates sampling to both account for variability in travel speeds and to estimate arrival time distributions at nodes on a path. We also suggest a method for transforming speed data into time-dependent emissions values thus converting the problem into a time-dependent, but deterministic shortest path problem. Our results demonstrate the effectiveness of the proposed approaches in reducing emissions relative to the use of minimum distance and time-dependent paths. In this paper, we also identify some of the challenges associated with using large data sets.

Key words: emissions, stochastic shortest path, green logistics, floating car data, big data

1 Introduction

Concerns about air quality and global warming have led to numerous initiatives to reduce emissions. As a significant source of CO_2 emissions, transportation has been an important focus of these efforts. Urban areas, where emissions are exacerbated by congestion, have been particularly active in looking at ways to reduce the environmental impact of transportation, especially freight transportation. For example, Amsterdam and London have introduced low-emission zones that restrict truck traffic near the city centers (see `http://www.milieuzones.nl` and `http://www.tfl.gov.uk/roadusers/lez/default.aspx`). Also, many companies have begun to focus on the CO_2 emissions resulting from the movement of goods, highlighting their efforts in annual reports (TNT Express, 2012) and press releases (UPS, 2013). Generally, the focus is on minimizing the traveled distance. While minimizing the traveled distance contributes to minimizing emissions, there are other factors that impact emissions.

In general, CO_2 emissions are proportional to the amount of fuel consumed, and the amount of fuel consumed is a function of speed, distance, acceleration, and weight of the vehicle (Demir et al. (2014)).

Optimizing for these factors, notably average speed, as is often the case in the academic literature, may not be feasible, particularly in urban areas. In urban areas, vehicles must often travel at the speed of traffic, and congestion can impact this speed particularly at certain times of day. These changes in speed have a significant impact on emissions, as noted by van Woensel et al. (2001). One of the more well-known models of emissions is the MEET model (Hickman et al., 1999), which is illustrated in Figure 1. This graph reveals the highly nonlinear relationship between speed and emissions, with particularly high emissions at slower speeds. At certain times of the day when traffic is at its peak and vehicles' speeds are reduced, a path that may yield low emissions at other times may suddenly yield high emissions. Thus, to optimize expected emissions in an urban area, it is critical to consider the speed of traffic on arcs in the network at different times of the day. This data is now increasingly available thanks to GPS equipment on taxicabs in nearly all major cities around the world (Ehmke et al. (2010)). The collection of speed observations leads to large historical databases containing speeds for a very detailed road network, which allow for a detailed analysis by route planners. It is well-known that congestion has recurring patterns for different times of the day, so we can use this data to estimate time-dependent speeds on different arcs on the network (Ehmke et al., 2012a).

Experimentation with these data sets reveals that even at the same time of day, the observations of speeds on a given arc can exhibit significant variability (Brockfeld et al., 2007). Due to the nonlinear relationship between speed and emissions, we cannot simply use an average speed on an arc at different times of day to choose the expected emissions-minimizing path between origin and destination. The average speed may not capture the average emissions that occur on a particular arc. For example, if the travel speed on an arc often drops far below the average speed, the actual emissions will be much higher than if travel occurs consistently at the average speed. Thus, to optimize emissions in an urban area, we must explicitly consider the variability in the speed of traffic on arcs in the network at different times of the day.

Motivated by the need to account for speed variation in the construction of expected emissions-minimizing paths between customers, this paper explores how to determine such paths. We propose two methods. Each method uses speed values from a historical database to determine time-dependent average emissions on each arc in a chosen path. In this way, we can ensure that our solution methods account for the impact of speed

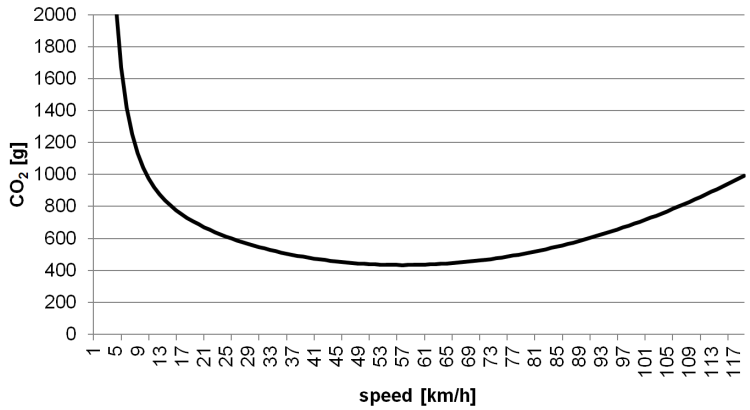

Figure 1: Evolution of emissions for 7.5-16 ton vehicles (MEET model) for a distance of 1 km

variation on emissions.

The first method, referred to as the *path-averaging method*, is a path construction method that uses sampling to evaluate the expected emissions associated with paths under construction. The use of sampling allows us to account for the impact of variability not only on emissions, but also on the arrival times to each node in a path. The expected emissions for a path are estimated based on averaging the emissions across all associated sample paths. To the best of our knowledge, our proposed sampling-based shortest path algorithm is the first to incorporate sampling in a path construction approach. This method can easily be generalized for applications beyond emissions, notably where arc costs are time-dependent and are evaluated via sampling.

Motivated by the computation time associated with the first method, the second approach ignores the variability in arrival times at each node. We refer to this approach as the *arc-averaging method*. In ignoring arrival time variability, we can operate on both expected time-dependent travel time and emissions values for each arc. Importantly, both of these values can be pre-computed. Further, operating on expected values reduces the problem to a deterministic one. Our computational experiments will examine the impact of this change both in the quality of the solutions found and in the runtimes.

We also compare both of our proposed methods with deterministic time-dependent travel time and distance-based shortest path methods commonly used in the literature. Our results demonstrate the value of accounting for speed variability when determining expected emissions-minimizing paths.

This paper makes an important contribution to the literature not only because it addresses how to

minimize emissions in a way that makes sense in urban areas, but also because it discusses some of the challenges associated with using large data sets. When working with large data sets, it is important to derive the appropriate level of aggregation that is both computationally tractable and that preserves the most important characteristics of the problem. We present two ways of aggregation, namely the usage of sampling by retaining individual emissions values in path generation, and, alternatively, the usage of pre-computed, time-dependent emissions estimates. In this context, because it is computationally intractable to seamlessly integrate a real-world database of 230 million speed records based on standard database queries, we discuss how to overcome the technical challenges of using large data sets in optimization. Finally, we discuss the issue that, even when considering such an extensive number of speed records, there are some arcs in the network for which there is very little data at all.

The remainder of the paper is outlined as follows. We review related literature in Section 2. We formally define the problem in Section 3 and introduce our solution approaches in Section 4. We describe our implementation and experimental design in Section 5, which includes a discussion of the issues with using large data sets. Our computational experiments are presented in Section 6, and we discuss future work in Section 7.

2 Literature Review

In the following, we provide an overview of the related literature. While the authors are unaware of any literature that explicitly addresses the question of computing shortest paths with emissions objectives, shortest paths have a long history in the literature, and we first review this literature. Next, we discuss various models of vehicle emissions. Finally, we discuss vehicle routing problems incorporating emissions objectives. Such problems are the original motivation for this work and represent an application that requires shortest paths as a subproblem.

2.1 Shortest Path Computation

Given an arc-dependent cost function, the idea of the shortest path problem is to determine the path of minimum cost between an origin and a destination. Arc costs may be based on travel time, distance, toll, energy consumption, or other considerations. Deterministic shortest path computation is a long and well-

established field of research with the label-setting algorithm of Dijkstra (1959) serving as the basis for nearly all modern shortest path algorithms. Recent computational advances are described by Geisberger et al. (2012). For path planning incorporating real-time information, we refer the reader to Güner et al. (2012).

In shortest path computations for vehicles, the availability of detailed traffic information has led many authors to explore time-dependent shortest path problems (TDSPP). In the TDSPP, arc costs depend on the time that the arc is entered, and as a result, the minimum cost path may differ depending on the start time at the origin and the arrival time to the nodes en route. A key to many efficient solutions methods is the FIFO assumption. The assumption basically disallows overtaking, meaning that it is always better to arrive to a node earlier rather than later. With the FIFO assumption, the problem can be solved by modified variants of any deterministic label-setting or label-correcting shortest path algorithm. A discussion of relevant theoretical issues and a review of early literature can be found in Dean (2004). Discussion of the use of traffic data in developing time-dependent networks and a review of relevant literature can be found in Ehmke et al. (2012b). Of particular interest to the work presented here is Chabini and Lan (2002). They demonstrate the effectiveness of the $A^\star$ shortest path algorithm (discussed more fully in Section 4) to the problem of finding shortest paths in time-dependent FIFO networks. The solution approach proposed in this paper adapts, for the expected emissions objective, the heuristic estimates proposed by Chabini and Lan (2002).

For the purpose of this paper, the most relevant TDSPP research is that which discusses the case where the FIFO property does not hold. If one seeks to minimize emissions, it can indeed be defensible to arrive to a particular node later via a circuitous path than to arrive earlier via a more direct route. Thus, it is difficult to defend the FIFO assumption. Sherali et al. (1998) show that with even a single non-FIFO arc the TDSPP is no longer polynomial but rather NP-Hard. Orda and Rom (1990) present an example where an optimal time-dependent path can have an infinite number of hops. Ziliaskopoulos and Mahmassani (1993) demonstrate that, by discretizing time, backward dynamic programming is capable of finding non-FIFO paths. If time is not discretized, algorithms are challenged by the proliferation of paths and labels. To overcome the proliferation of paths in solving the TDSPP with non-FIFO arcs, many authors seek to reduce the problem size by eliminating less promising paths. Sherali et al. (2003) call this network curtailment. In a

manner analogous to Chabini and Lan (2002), Sherali et al. (2006) incorporate estimates of future cost into heuristics for reducing the runtime of algorithms solving time-dependent, label-constrained shortest path problems.

This paper is particularly concerned with time-dependent and stochastic shortest paths. While the authors are unaware of any literature that addresses such problems with an emissions objective, the problem has been explored in the context of other objectives. Fu and Rilett (1998) study expected shortest paths in time-dependent and stochastic traffic networks. The authors develop a second order approximation that allows them to recursively compute the expected cost of a path. To accurately compute emissions, it is important for us to have a distribution on each arc. As will be discussed in the next section, even a second order approximation of the mean travel time would be insufficient because of the convexity of emissions functions and Jensen's inequality.

To capture the appropriate arc travel-time distribution, an alternative to Fu and Rilett (1998) is to compute the convolutions of the arc distributions to compute the arrival time to each node. For instance, Lecluyse et al. (2009) assume travel times are lognormally distributed. While determining the convolution is computationally tractable for certain distributions (Kao, 1978), such distributions do not generally fit travel time data (Ehmke and Campbell, 2014). In general, the convolutions are difficult to compute and in some cases the closed forms are unknown. Demeyer et al. (2012) overcome the challenge by using percentiles of the arc travel time distributions. The drawback of this method is that tractability comes at the cost of fidelity to the original distribution. As a result, we choose a data-driven approach that is unaffected by the form of the underlying distributions, and with appropriate sample size, faithfully represents the underlying distribution.

2.2 Modeling Emissions

Modeling emissions requires an understanding of the physical relationships between fuel consumption, speed, distance and acceleration. Demir et al. (2014) provide an overview of existing models of emissions as well as applications. Demir et al. (2011) present a comprehensive review of six analytical emission models and compare them with respect to granularity of predicted emissions, required input data, and potential field of usage. Their comparison reveals significant differences with respect to complexity, comprehensibility and

correctness of predicted and actual fuel consumption.

Of particular interest for this work are the emission models that have appeared in recent routing papers related to the minimization of emissions. The first model is the comprehensive emissions model by Barth and Boriboonsomsin (2008). The authors describe the fuel consumption for an arc of a road network as based on speed, weight of a vehicle, and numerous vehicle and arc-specific constants. The advantage of this comprehensive emissions model is that the impact of a vehicle's load on emissions is considered. The disadvantage is that this model is relatively complex, and it may be challenging to correctly set the required variables. This is alleviated by the second model by Hickman et al. (1999). This relatively simple model (MEET model) allows for conversion of speeds into emissions, using typical fuel consumption and emission rates that have been derived from engine test-bed measurements. Within the scope of freight transportation, parameter configurations are available for vehicles with a gross weight of 3.5-7.5 tons, 7.5-16 tons, 16-32 tons, and 32-40 tons.

2.3 Emissions in Vehicle Routing

While the focus of this paper is the determination of expected emissions-minimizing paths, shortest path problems are a subproblem of vehicle routing problems (VRP), and a number of recent VRP papers have addressed emissions-related objectives. However, it is important to note that none of the VRP papers consider variability in the paths connecting two customers. Demir et al. (2014) and Lin et al. (2014) provide an overview of emissions considerations in the VRP literature. Of note is recent work by Bektas and Laporte (2011), who introduce the pollution routing problem. Using the comprehensive emissions model of Barth and Boriboonsomsin (2008), Bektas and Laporte (2011) minimize CO_2 emissions by determining the optimal speed level for the given load considering load induced costs, speed induced costs and driver induced costs. The model only captures speeds faster than 40 km/h, avoiding the high emissions found at low speeds in convex emissions functions. Results show that the traditional objective of distance minimization does not necessarily imply minimization of either fuel cost or driver cost, and that a cost-optimal solution does not imply an emissions-optimal solution. It is also demonstrated that the presence of customer time windows may change the results significantly.

Demir et al. (2012) extend the approach of Bektas and Laporte (2011) and introduce an Adaptive Large Neighborhood Search in order to tackle instances of realistic size, also allowing for speeds below 40 km/h. For each route, they determine the optimal speed on each arc so as to minimize an objective function involving costs of fuel consumption and drivers' wages. Franceschetti et al. (2014) study this problem with consideration of time-dependent cost functions of speeds and fuel consumption. They present an exact solution approach minimizing the cost of fuel and drivers while optimizing speed and departure times. Specifically, they consider a two-period problem where a congested period with relatively low speeds (and high emissions) is followed by an uncongested period with relatively high speeds (and lower emissions). In order to avoid low speeds and high emissions in congested periods, post-service waiting at customers is allowed. Franceschetti et al. (2014) highlight that extension of this simplified timeline to a more practical pattern "is likely to be significantly more complicated to analyze."

Jabali et al. (2012) also account for time-dependent travel times between customers and minimize travel time, fuel, and CO_2 emissions costs. They derive CO_2 emissions from the MEET model. Except for rush hour periods, where they assume that speeds are fixed due to congestion, they determine speeds according to an emissions-optimal level. This time-dependent routing problem is solved by a tabu search procedure. Figliozzi (2010) investigates the impact of congestion on emissions resulting from urban freight operations. He considers time-dependent travel times, hard customer time windows, and capacity constraints, and also derives emissions from the MEET model. Speed and departure time are handled as decision variables. Based on Solomon instances, it is shown that congestion in urban areas has a significant impact on travel times and emissions.

In contrast to the approaches to emissions routing in the VRP literature, in this paper, we assume that, in urban areas in particular, vehicles must travel at the speed of traffic and do not have the ability to control their speed in a way that minimizes emissions. Further, this work accounts for the fact that urban speeds are highly variable and impact emissions through the non-linearity of the emissions curves.

3 Formal Model

In this section, we present a model of the expected emissions-minimizing path problem and demonstrate how to estimate the objective function value via sampling.

3.1 Model

Let $G = (N, E)$ be a graph where N is a set of nodes and E a set of directed arcs in the graph. We make no assumptions on whether or not the network is FIFO. We assume that travel times are random and that the distribution of travel times on the individual arcs are independent, but that the travel time distributions are time-dependent. As noted in Fu and Rilett (1998), the time dependence helps account for correlation between travel times on arcs. Let f_e be a function of emissions on arc e whose domain is speed. We require a separate function for each arc to account for the distance traveled on that arc at any particular speed.

A path is a series of arcs traversed en route from a given origin to a given destination. Let $P_{\alpha,\omega}$ be the set of all paths from an origin α to a destination ω, α, ω in N. We seek $p^\star$ in $P_{\alpha,\omega}$ such that $p^\star$ minimizes, for all p in $P_{\alpha,\omega}$, $\mathbb{E}\left[\sum_{e\in p} f_e(S_e)\right]$, where S_e is a random variable representing the speed on an arc e on path p.

3.2 Criterion Evaluation via Sampling

While the previous section mathematically describes the criterion, actually computing it for any given solution is challenging. As noted in Section 1, computing the expected speeds and then the expected emissions is insufficient for computing the criterion. Most emissions functions are convex, and from Jensen's Inequality $\left(\sum_{e\in p} f_e(\mathbb{E}[S_e]) \leq \mathbb{E}\left[\sum_{e\in p} f_e(S_e)\right]\right)$, computing expected emissions via the expected speed underestimates emissions. To overcome these challenges, we take advantage of the increasing amount of detailed speed data now being captured by many municipalities and employ a sampling-based approach to evaluate the cost of a path.

We begin by rewriting the criterion as $\sum_{e\in p} \mathbb{E}[f_e(S_e)]$. For our purposes, it will be helpful to make the dependence on the arrival time to the tail node of the arc explicit. Thus, for each arc, we have:

$$\mathbb{E}\left[\mathbb{E}\left[f_e(S_e) \mid Y_e^p\right]\right] = \int_0^\infty \mathbb{E}\left[f_e(S_e) \mid y_e^p\right] g_{Y_e^p}(y_e^p)\partial y_e^p, \tag{1}$$

where Y_e is a random variable representing the arrival time to arc e on a given path p and $g_{Y_e^p}(y_e^p)$ is the probability density function for the arrival time to arc e when traversing path p. We note that it is straightforward to make the arrival time distribution itself dependent on a start time T from the origin.

Then, we can evaluate the criterion via sampling by proceeding in an iterative fashion from the origin. Let n be the number of samples used to evaluate the cost of each arc. For the first arc e_1 and a given starting

time T, we sample n speeds for arc e_1 when entering arc e_1 at time t, or really a user-defined interval or bucket around t. The need for the interval or bucket is due to the limited number of speed observations available for any particular time. Let $s_1^k(t)$ be the kth sample associated with starting time t for arc e_1. Our estimate of the expected emissions on arc e_1 is $\hat{\mathbb{E}}\left[f_e(S_1) \mid t\right] = \frac{\sum_{k=1}^{n} f_e\left(s_1^k(t)\right)}{n}$.

To generalize the estimation for arcs $e_j, j > 1$, on a path p, we note that each sampled speed can be converted into travel time because the length of each arc in the graph is known. Let λ_e be a function that maps the speed on arc e to travel time. Then, for example, we can compute an arrival time at the head node of the first arc e_1 and the kth sample as $\lambda_{e_1}(s_1^k(t)) + t = a_1^k$, assuming travel began at time t. Generalizing, the sampled k^{th} arrival time to the head node of arc j is $a_j^k = \lambda_{e_j}(s_k^j(a_{j-1}^k)) + a_{j-1}^k$, where a_{j-1}^k is the arrival time for the kth sample to the tail node of arc e_j. The set of arrival times $\{a_j^k : k = 1, \ldots, n\}$ is a sample of the arrival time distribution at the head node of arc e_j. Thus, for the jth arc on a path p, we estimate $\mathbb{E}\left[\mathbb{E}\left[f_e(S_{e_j}) \mid Y_j^p\right]\right]$ as

$$\hat{\mathbb{E}}\left[\mathbb{E}\left[f_e(S_{e_j}) \mid Y_j^p\right]\right] = \frac{\sum_{k=1}^{n} f_e\left(s_j^k(a_{j-1}^k)\right)}{n}.$$

4 Solution Approaches

In this section, we present solution approaches for expected emissions-minimizing paths in urban areas. First, we introduce a path-based label setting $A^\star$-variant that solves the time-dependent, expected emissions-minimizing path problem. By incorporating sampling to evaluate emissions of a given arc, the $A^\star$-variant can both capture the impact of variation on emissions as well as estimate the arrival time distribution to a given node on a path. In subsequent sections, we will refer to this method as *path-averaging*. Because of the computation time required to create these sample paths, we introduce a second solution method. In the second approach, we first convert each data point in our database of travel speeds into an emissions value. We then average the emissions values for each hour of the day and each arc to generate time-dependent emissions values. Using these values and using expected arrival times at each node on a path, finding the expected emissions-minimizing path becomes a deterministic problem for which we adapt a common Dijkstra-based approach. In subsequent sections, we will refer to this method as *arc-averaging*. We will also subsequently refer to both solution approaches as emissions-based methods.

4.1 Path-Averaging Method

Algorithm 1 presents a path-based label setting algorithm based on the path-search heuristic $A^\star$. $A^\star$ is itself based on Dijkstra's algorithm with the difference being that selection of the node for expansion accounts not only for the cost from the start node to the selected node, but also an estimate of the cost from the selected node to the destination node. This estimate is often called the heuristic estimate. An extensive overview of $A^\star$ can be found in Pearl (1984).

A key challenge with the algorithm is the evaluation of the expected emissions of traveling on a particular arc. To do this, we adapt the sampling method that was previously described for evaluating the expected cost of a given path. Of note, this method requires that our node label includes both the node number and a vector of sampled arrival times to that node. As it is possible to arrive at a node via multiple paths and thus have multiple arrival time distributions to each path, there are cases in which we will create multiple labels for a node to account for the possibility of different arrival time distributions.

The algorithm takes as input a problem instance with a defined origin and destination as well as a start time T from the origin. We also fix the number of samples as n. Finally, the algorithm requires a function f_{ij} that computes emissions for a given speed on arc (i, j), a function λ_{ij} that computes the travel time for a given speed on arc (i, j), and a heuristic estimate h_i of the remaining cost from a node i to the destination.

We pre-compute the heuristic estimate h_i for every node i. The first step in computing the heuristic estimate is to aggregate individual speed observations into what are known in the literature as time buckets (see Ehmke et al. (2012a)). We will use these time buckets to account for the impact of time-dependence in computing our heuristic estimate. To construct the time buckets, for each arc, we organize speed observations into 24 fixed hourly buckets, notably in buckets from 06:00-06:59, 07:00-07:59, ... For each arc and hourly bucket, we convert each speed observation into an emissions value. Specifically, for each speed observation o on arc e in time bucket b, we convert it to emissions as $f_e(o)$. For an arc e and a given bucket b, we compute the time-dependent mean emission as $\eta_e^b = \frac{\sum_k f_e(o_k)}{K_e^b}$, where o_k is the k_b^{th} observation in the bucket b and K_e^b is the number of observations in time bucket b for arc e. We note that our calculation of the expected time-dependent emissions is an expectation of the emissions value rather than the emissions value of the expected speed. As a result, the nonlinear relationship between speeds and emissions is reflected in

the sampled emissions that are averaged for each arc.

Having computed time-dependent expected emissions values for each arc, for every arc e, we choose the minimum average emissions cost over all of the previously described time buckets ($\min_e\{\eta_e^1, \ldots, \eta_e^{24}\}$). Finally, for every node i in the network, we use these minimum values to compute shortest paths to the destination node. The emissions costs of these shortest paths is our heuristic estimate h_i for each node i and thus each label containing the node. By using the lowest expected emissions cost over all of the time buckets for each arc, these heuristic estimates h_i should serve as a lower bound on the expected emissions to reach the destination node, as suggested in Chabini and Lan (2002).

Algorithm 1 operates on labels $(i, \langle a^1 \ldots a^n \rangle)$, where $\langle a^1 \ldots a^n \rangle$ is an n-dimensional sample of arrival times to a node i. Throughout the algorithm, for each label l, we track the sample average of emissions incurred on the path to label l as g_l. We also store the value $D_l = g_l + h_l$, where h_l is the heuristic estimate. We define the set L as set of nodes that have been labeled but not yet expanded, and F as the set of labels that have already been expanded.

The algorithm is initialized by adding the label $l = (\alpha, \langle T \ldots T \rangle)$ to the set L. We also set $g_l = 0$, the expected emissions cost occurred on the path to the start node, and the estimated expected cost of emissions at the start $D_l = g_l + h_l$. To run, the algorithm chooses from L the label l that achieves the minimum value of D_l of the labels in L. Given a label $(i_l, \langle a_l^1 \ldots a_l^n \rangle)$, for all outgoing arcs (i_l, j) from the node i_l associated with the label l, we create labels $(j, \langle a^1 \ldots a^n \rangle)$, where $a^k = a_l^k + f_{ij}(s_i^k(a_l^k))$, $s_{ij}^k(a_l^k)$ is a sample of the speed along arc (i, j) when starting from i at time a_l^k. For practical reasons related to the data and discussed in Section 5, the sample $s_{ij}^k(a_l^k)$ is drawn from a bucket dynamically determined by

$$\left[a_l^k - \delta, a_l^k + \delta\right] \tag{2}$$

for each sampled arrival time a_l^k associated with label l. In Algorithm 1, the function Sample returns n samples for a given label l and a destination node j.

For each newly constructed label $(j, \langle a^1 \ldots a^n \rangle)$, we check to see if the path back to the start node includes any cycles and delete any labels for which the path includes a cycle. While the proposed algorithm allows non-FIFO paths, allowing cycling leads to a proliferation of labels that our experiments showed did not improve solution quality and significantly increased runtime. If no cycle is present on the pointer path of the

new label $(j, \langle a^1 \dots a^n \rangle)$, we add the new label to L if it is not already, and if it is, we update $g_{(j,\langle a^1 \dots a^n \rangle)}$ and $D_{(j,\langle a^1 \dots a^n \rangle)}$ if necessary. If the label is already in F, we add it to L and remove it from F if $D_{(j,\langle a^1 \dots a^n \rangle)}$ is reduced by traveling via label l, updating $g_{(j,\langle a^1 \dots a^n \rangle)}$ and $D_{(j,\langle a^1 \dots a^n \rangle)}$. For all labels that are added to L, we direct a pointer back to the label l so that we can trace the path back to the origin upon reaching the destination. Finally, we add the label l to F.

The algorithm terminates when F is empty or a label containing the destination node ω is removed from L. This termination criterion is the same as is used for the deterministic version of the $A^\star$ algorithm. For the deterministic algorithm, it is well known that $A^\star$ terminates with an optimal path if the heuristic estimate h is a lower bound on the cost-to-go. Because of the need to sample the arc costs and the use of buckets in the construction of the heuristic estimate h, the same guarantee does not hold in the case of Algorithm 1. Nonetheless, our experience suggests that the proposed termination criterion is effective for the problem discussed in this paper. If $L = \emptyset$ before a label containing the destination node ω is removed, there is no feasible path.

4.2 Arc-Averaging Method

In this section, we propose an alternative to the $A^\star$-based approach described above. We are motivated by the fact that the $A^\star$-based approach can be computationally expensive. There are two reasons for the computational challenges. First, every time we expand a label, we must generate samples for every successor of the current label. Second, because we do not assume a FIFO network, it is possible for there to be a large number of labels associated with every node in the network. This label proliferation exists even though we do not allow cycles in our shortest paths.

To reduce this computational burden, we develop our alternative solution approach by making two concessions to model fidelity. First, we eliminate the dynamic bucketing (given in Equation (2)) and map arrival times to one of the 24 buckets (06:00-06:59, 07:00-07:59, ...) used in the computation of the heuristic estimates above. Second, the alternative approach uses the expected arrival times at each node when determining the shortest path. It is straightforward to show that the expected arrival time to the tail node of arc e for a given path p is the sum of the time-dependent expected travel times on each arc on the path p up to e.

Algorithm 1 Path-Based-Sampling $A^\star$-based Algorithm

Input:
Data for a problem instance including an origin α, a destination ω, a departure time T from the origin node, a function f_{ij} measuring emissions as a function of speed for each arc (i, j) in E, a function λ_{ij} measuring travel time as a function of speed for each arc (i, j) in E, and a function h which takes a label, particularly the node associated with the label, as input and returns an estimate of the cost-to-go
A number of samples n
Output: Solution, $p^\star$
Initialization:
$L \leftarrow l_1 = (\alpha, \langle T \dots T \rangle)$
$g_{l_1} = 0, h_{l_1} = 0$
$D_{l_1} = g_{l_1} + h_{l_1}$
while $L \neq \emptyset$ **do**
    $l \leftarrow \text{argmin}_{l \in L}\{D_l\}$ and $L \leftarrow L \setminus \{l\}$.
    if The label l contains the destination node ω **then**
        Terminate algorithm
    else
        for all (i_l, j) such that $(i_l, j) \in E$ **do**
            $s \leftarrow \textsc{Sample}(l, j)$
            $a^k \leftarrow a_l^k + \lambda_{i_l j}(s^k)$ for all $k = 1, \dots, n$
            Label $l' \leftarrow (j, \langle a^1 \dots a^n \rangle)$
            $\hat{g}_{l'} \leftarrow g_l + \frac{\sum_{k=1,\dots n} f_{i_l j}(s^k)}{n}$
            $\hat{D}_{l'} \leftarrow \hat{g}_{l'} + h_{l'}$
            if $l' \notin L$ and $l' \notin F$ **then**
                $L \leftarrow L \cup \{l'\}$
                $D_{l'} \leftarrow \hat{D}_{l'}$ and $g_{l'} \leftarrow \hat{g}_{l'}$
            else if $\hat{D}_{l'} < D_{l'}$ **then**
                $D_{l'} \leftarrow \hat{D}_{l'}$ and $g_{l'} \leftarrow \hat{g}_{l'}$
                $L \leftarrow L \cup \{l'\}$, and if $l' \in F, F \leftarrow F \setminus \{l'\}$
            end if
        end for
        $F \leftarrow F \cup \{l\}$
    end if
end while

Given these concessions, we need only the expected time-dependent travel times on each arc in order to determine the expected arrival times to each node on a path. To compute the time-dependent expected travel times on each arc, we follow an approach analogous to the computation of the expected time-dependent emissions values on each arc. The mean time-dependent travel time for each arc e and time bucket b, denoted τ_e^b, is computed analogously to the η_e^b as $\tau_e^b = \frac{\sum_k \lambda_e(o_k)}{K_e^b}$, where o_k is the k_b^{th} speed observation in the bucket b and K_e^b is the number of observations in time bucket b for arc e. Thus, the expected arrival times can be computed by summing the appropriate τ values for each arc and associated time bucket on a given path. Most importantly, these τ values can be pre-computed.

An important consequence of using expected arrival times is that Equation (1) becomes

$$\mathbb{E}\left[f_e(S_e) \mid \bar{y}_e^p\right], \tag{3}$$

where $\bar{y}_e^p$ is the expected arrival time to the tail node of arc e when following path p. Equation (3) reduces to the η_e^b value corresponding to the bucket b in which $\bar{y}_e^p$ is found. As discussed previously, the η_e^b values are also pre-computed. Consequently, using expected values to approximate the arrival time distributions allows us to avoid sampling altogether and to use pre-computed expected emissions values when computing the cost of the paths, reducing the problem to a time-dependent, but deterministic shortest path problem. As was noted in Section 4.1, we compute the expected time-dependent emissions values for each arc by averaging the emissions that result from each of the speed observations rather than using an average speed, and as a result, we are able to capture the impact of the speed variability on emissions.

We further reduce computation time by employing a label-setting rather than label-correcting algorithm to find the minimum time-dependent expected emission path. In our approach, we maintain one label for each node. In doing this, we control the proliferation of labels that lead to longer runtimes for the $A^\star$-based algorithm. Our label-setting approach is based on Dreyfus (1969), who showed that a straightforward adaption of Dijkstra's algorithm can solve the time-dependent shortest path problem. We specifically base our adaptation on the algorithm presented in Dean (2004). As with the algorithm presented above, our does allow for non-FIFO paths (up to the point that the node is treated), but does not allow for paths that contain cycles. However, our implementation stores a single label for each node, controlling the proliferation of labels.

We present our adaptation in Algorithm 2. For convenience, we use $b(l)$ to refer to the time bucket corresponding to an expected arrival time associated with a label l. Given a start time T, the algorithm is initialized by adding the label (α, T) to the set of temporarily labeled nodes L, setting $D_\alpha = 0$. Note that, because we store only a single label for each node, we can label the D values by node and not label. We also create a temporary label for each node i, $i \neq \alpha$, denoted by $(i, \cdot)$, and for each of these labels l_i, we set $D_i = \infty$. As the algorithm runs, one label for each node will be maintained, and these temporary labels will be updated to include expected arrival time information for the expected emissions-minimizing path from

Algorithm 2 Arc-Averaging Time-Dependent Label Setting Algorithm

1: **Input:**
2: Data for a problem instance including an origin α, a destination ω, a departure time T from the origin node, an average emissions value η_{ij}^{b} and average travel time τ_{ij}^{b} for each (i,j) in E and time bucket b
3: **Output:** Solution, $p^{\star}$
4: **Initialization:**
5: $L \leftarrow l_{\alpha} = (\alpha, T)$ $L \leftarrow l_i = (i, \cdot)$ for all $i \neq \alpha$
6: $D_{alpha} = 0$, $D_i = \infty$ for all labels $l_i = (i, \cdot)$, and $F = \emptyset$
7: **while** $L \neq \emptyset$ **do**
8: $l \leftarrow l_i \mid i = \text{argmin}_{i|l_i \in L}\{D_i\}$, $L \leftarrow L \setminus \{l\}$, and $F \leftarrow l$
9: **if** The label l contains the destination node ω **then**
10: Terminate algorithm
11: **else**
12: **for all** (i_l, j) such that $(i_l, j) \in E$ and l_j is not in F **do**
13: **if** $D_j > D_{i_l} + \eta_{i_l j}^{b(l)}$ **then**
14: $l_j \leftarrow (j, t_l + \tau_{i_l j}^{b(l)})$
15: $D_j \leftarrow D_{i_l} + \eta_{i_l j}^{b(l)}$
16: **end if**
17: **end for**
18: **end if**
19: **end while**

the origin to the node associated with the label.

To run, the algorithm chooses from L the label l achieving the least emissions cost. As in Dijkstra's algorithm, we now consider the node associated with the chosen label as permanently labeled and add it to the set F. For all outgoing arcs (i,j) from i and the expected arrival time associated with node i, we then determine if traveling to node j via i reduces the emissions cost of a path to j using a mean time-dependent emission. If so, we update the label l_j associated with j by updating the expected arrival time to j and also update the emissions cost of the path to j. The algorithm terminates when either the set L is empty or when the end node ω is in the set of permanently labeled nodes F. For all labels that are updated in Step 14 of the algorithm, we direct a pointer back to the node associated with label l so that we can trace the path back to the origin upon reaching the destination.

The goal of the proposed arc-averaging method is to balance solution quality with computation time. However, by maintaining only a single label for each node, the arc-averaging method is a heuristic method. In non-FIFO networks, we cannot guarantee that it is optimal (assuming cycles are not allowed). However, as our computational results demonstrate, the method does deliver promising results for the data tested in this paper.

5 Data Preparation and Implementation

This section provides an overview on the preparation of real speed data for path finding with emissions objectives. We introduce a historical speed database in Section 5.1. Limitations of this large data set and challenges of integrating the large data set in shortest path algorithms are discussed in Section 5.2.

5.1 Data Preparation

We demonstrate and evaluate the generation of expected emissions-minimizing paths based on real speed data from the metropolitan area of Stuttgart, which is a major city in the southern part of Germany. The metropolitan area of Stuttgart is well known for its congestion, especially at peak times (Kröger (2013)). In particular, we refer to a database of approximately 230 million speed observations from the years 2003-2005. These speed observations have been collected by the German Aerospace Center with FCD technology (Lorkowski et al. (2004)) with a fleet of approximately 700 taxis.

An FCD based speed observation captures the speed of a single taxi on a particular arc for a particular time and day. Given a fleet of taxis operating in a metropolitan area, it is thus possible to collect speed observations for the majority of arcs of the road network at low cost. Arcs are the smallest entity in the road network. While collecting speed observations with FCD technology is very cost-efficient, there are also limitations that must be noted. Speeds observed in the same time period may vary significantly because of such factors as the environment, varying traffic flows and traffic lights. Furthermore, the number of speed observations per time period and arc may also vary significantly due to some arcs being used often by taxis, while others are used only rarely. Thus, it is important to ensure a sufficient number of speed observations per arc and time period in order to abstract from singular events to guarantee validity for shortest path computation.

For subsequent computational experiments, we combine all speed observations collected on Tuesdays, Wednesdays and Thursdays in the years 2003-2005 to represent a typical working day. For details regarding filtering of outliers and wrong measurements, see Ehmke et al. (2009). Based on a NAVTEQ digital roadmap, we derive a network that consists of 6832 arcs, 4135 nodes, and approximately 74 million speed observations. In this detailed digital roadmap, arcs have an average length of 97 meters with a median of 72 meters.

5.2 Implementation

Because speed observations are not distributed uniformly across the arcs in the network, we encounter technical challenges and limitations that usually do not occur in optimization. In this section, we summarize the challenges we face and how they affect the implementation of our solution approaches. We discuss how to process the large number of speed observations efficiently, determine the length of dynamic buckets, set the minimum number of samples, and ensure comparability of solution approaches that work on different granularities of buckets.

The processing of historical speed observations requires efficient data storage and loading techniques. Contrasting with a simple lookup table of average speeds and travel times often used in deterministic path generation, working with dynamic buckets implies the availability of all speed observations during path construction. Instead of repeatedly querying the historical database for each specific set of speed observations required, we assign individual speed observations to corresponding arcs, order them according to the particular time that they were collected, and make them quickly accessible by loading them from flat files during initialization of shortest path computation. While the production of the flat files requires several days, the initialization step can be done in about two minutes. With all the speed data being available in RAM, an expected emissions-minimizing path can be computed in 17 seconds on average. For details on the runtime of our approaches, see Section 6.3.

To ensure the validity of the sampling approach for the path-averaging method, we need to define the length of the dynamic buckets as well as the minimum number of samples for the dynamic buckets properly. Note that there is a trade-off between these parameters. While a longer bucket increases the number of speed observations available for sampling, it may decrease the precision of the arrival time distributions' estimates. For instance, estimating arrival time distributions for a 120-minute bucket would smooth well-known variation of speeds at peak times. In contrast, using a 15-minute bucket limits the number of available speed observations, thus reducing the precision of estimates of the arrival time distribution. We experimented with different parameter settings for the minimum number of samples and lengths of dynamic buckets. We found that setting the length of the dynamic buckets to 30 minutes ($\pm$15 minutes around the estimated arrival time at a node) is a good compromise between capturing variability of arrival times and enabling a

reasonably-sized network. We set the number of samples that are drawn from dynamic buckets to $n = 1000$. In case that fewer than 1000 observations are available, we oversample.

It is also important to note that the path-averaging method uses dynamic bucketing while the methods to which we compare (described in Section 6.2) use the static hourly buckets (06:00-06:59, 07:00-07:59, ...). For the same minimum number of samples, it is very likely that the number of arcs available for the fixed one-hour buckets is much higher than for dynamic 30-minute buckets. To ensure the comparability of solutions, we require that each solution method operate on the same network structure. We do this by eliminating any arcs from the considered arc set during any hour where there are less than 20 total observations. For the path-averaging method, if the average arrival time of sample paths falls into a static hourly bucket (e.g., 06:00-06:59) that has fewer than the minimum number of observations for a particular arc, that particular arc is not considered in constructing the path-averaging solution.

6 Computational Experiments

In this section, we discuss our test sets, heuristic benchmarks, and the results of our computational experiments.

6.1 Test Sets and Emissions Model

For the generation of test sets, we carefully select combinations of origin and destination pairs (O-D pairs) from the set of 4135 nodes described in Section 5.1. Our intent is to create typical urban travel patterns that may be affected by congestion. To this end, we choose O-D pairs representing paths from suburban to inner city locations (and vice versa), between inner city locations, and between suburban locations as described in Table 1. Note that we limit the selection of O-D pairs to areas where we expect a sufficient number of speed observations for arcs between origin and destination across the entire day. Thus, we do not determine them at random. However, as detailed speed observations become more readily available, we anticipate that our approach could be used to find paths between any O-D pair for any metropolitan area.

For each O-D pair, we consider 15 different start times. These times are: 06:30, 07:30, 08:30, 09:30, 10:30, 11:30, 12:30, 13:30, 14:30, 15:30, 16:30, 17:30, 18:30, 19:30, 20:30. Four start times, 07:30, 08:30, 17:30, and

18:30, are considered the peak travel or rush hours.

Index	O-D Pair	Direction
1	(124,663)	southern suburbs to northern suburbs via city center
2	(323,408)	city center to western suburbs
3	(408,323)	western suburbs to city center
4	(480,663)	southern suburbs to northern suburbs via city center
5	(663,124)	northern suburbs to southern suburbs via city center
6	(663,480)	northern suburbs to southern suburbs via city center
7	(663,881)	northern suburbs to western suburbs
8	(732,733)	western suburbs to northern suburbs via city center
9	(732,871)	western suburbs to city center
10	(732,943)	western suburbs to eastern suburbs via city center
11	(811,927)	city center to southern suburbs
12	(829,942)	northern suburbs to eastern suburbs via city center
13	(871,899)	city center to southern suburbs
14	(943,871)	city center to eastern suburbs

Table 1: Description of O-D Pairs

6.2 Benchmarks and Solution Evaluation

Our proposed solution approaches are compared with popular methods used to generate paths. Specifically, we consider minimum distance paths and minimum time-dependent travel time paths. We consider minimum distance paths because, as indicated in the introduction, many authors and companies have simplified the minimization of emissions to the minimization of distance, ignoring the impact of congestion. An alternative that incorporates the impact of congestion that has become more popular in recent years is through minimum time-dependent travel time paths. Such paths avoid slow arcs during times of congestion at the cost of longer distances. However, from an emissions perspective, it is not obvious whether replacing slower arcs with longer distances and looking for average higher speeds will always yield the expected emissions-minimizing path.

Each of the four solution methods operates using a different objective. Thus, to compare the solutions resulting from each method, we must evaluate them using the objective given in Equation (1). Each solution from all methods is evaluated with a simulation employing sampling as described in Section 3.2. For each solution being evaluated, we use 2000 sample paths. Each sample in a sample path is generated dynamically by randomly drawing a speed observation from a dynamic bucket 15 minutes before and after the arrival time at each node on the sample path. Travel times and emission values of individual arcs are summed across the path and then averaged to create the reported values.

The benchmark deterministic approaches are implemented as follows. We generate minimum distance paths by a straightforward implementation of Dijkstra's algorithm. Arc distances are derived from a NAVTEQ digital roadmap. For minimum time-dependent travel time paths, we follow a common approach in the literature, such as described in Cooke and Halsey (1966), but use averages based on historical speed data and hourly buckets (as described in Section 4).

For all solution methods and the simulation of solution values, speeds are transformed into emissions according to the MEET model with parameters for a gross weight of 7.5 tons. The MEET model serves here as the f function, but, as mentioned earlier, any emissions model could be used in conjunction with the ideas proposed here.

Our experiments are performed on a Windows 7 64-bit operating system with an Intel Core i5-3470 processor and 8 GB of RAM. Algorithms are coded and executed in Java 64-bit.

6.3 Summary Results

We first report the results for all O-D pairs, averaging the results for the 15 start times. In Section 6.4, we present detailed results for one O-D pair, demonstrating what happens at each considered departure time and showing some of the selected paths. Detailed data for each departure time for the entire set of O-D pairs is included in the electronic supplement to this article.

In Tables 2 and 4, we report the results from the evaluation of the paths chosen by each of the previously identified methods for each O-D pair. In the evaluation of each path by simulation, the following metrics are recorded: the distance of that path in kilometers, the mean travel time in minutes, the standard deviation of the travel time, the mean CO_2 emissions in kilograms, and the standard deviation of the emissions value. The results in Tables 2 and 4 represent an average of these metrics over the 15 start times for each O-D pair. Table 2 presents the results for the minimum emissions paths found by arc-averaging and path-averaging methods. Table 4 presents the results for the minimum distance and the minimum time-dependent travel time paths.

Each table also includes additional columns comparing the relative performance of the methods. Each column labeled "inc%" represents the percentage change in a given metric relative to the value from the method that performs best for that metric. Specifically, for all methods except the minimum distance

method, we compute

$$100\% \times \frac{\text{distance of path from alternative method } - \text{distance of minimum distance path}}{\text{distance of minimum distance path}}.$$

A positive percentage indicates an increase in distance over the minimum distance path. Similarly, for all methods except the minimum time-dependent travel time method, we compute

$$100\% \times \frac{\text{mean travel time of path from alternative method } - \text{mean travel time of minimum time-dependent travel time path}}{\text{mean travel time of minimum time-dependent travel time path}}.$$

A positive percentage indicates an increase in travel time over the minimum time-dependent travel time path. For all methods except the path-averaging emissions method, we compute

$$100\% \times \frac{\text{mean emissions from path from alternative method } - \text{mean emissions from path-averaging emissions path}}{\text{mean emissions from path-averaging emissions path}}.$$

A positive percentage indicates an increase in emissions over the path-averaging emissions path. We also note again that regardless of how a path is constructed, the expected emissions values are computed using the sample method described in Section 3.2.

In looking at Table 2, we can compare the performance of path-averaging and arc-averaging methods. Somewhat surprisingly, the average emissions across all O-D pairs is the same between the two methods (to two decimal places). The average percent increase in emissions from using arc-averaging is only 0.02%, and the largest percent difference in emissions for any O-D pair is 0.39% (for O-D pair 10). This indicates that in an urban setting, such as the one studied here, it is not critical to do the dynamic bucketing used by path-averaging methods. Instead, the one-hour static bucketing used by our arc-averaging methods is sufficient to capture the time-dependency in travel speeds (and emissions). We predict that the results would be less similar if the arcs were longer, where variation would lead to a wider range of arrival times at the next node, or if we used larger bucket sizes than one hour. In many urban environments, arcs tend to be quite short, so we anticipate that both emissions-based approaches would yield similar results in other cities with arcs of similar size.

The two methods, though, have an important difference in computational time, as reflected in Table 3. The runtime column reflects the average number of seconds required to compute the best path for each O-D pair across the different start times for the path-averaging method. Run times vary significantly between

Table 2: Results for Arc-Averaging and Path-Averaging Emissions Methods

		Minimum arc-averaging emissions path (all departures)								Minimum path-averaging emissions path (all departures)						
index	O-D pair	distance		travel time			emissions			distance		travel time			emissions	
		avg	%inc	avg	std	%inc	avg	std	%inc	avg	%inc	avg	std	%inc	avg	std
1	124, 663	18.58	1.76%	46.16	6.29	7.66%	12.64	2.70	0.25%	18.69	2.37%	46.08	6.10	7.49%	12.61	2.59
2	323, 408	10.80	2.71%	29.04	3.71	16.77%	7.59	1.59	0.04%	10.87	3.41%	28.97	3.72	16.50%	7.58	1.59
3	408, 323	9.95	0.28%	23.74	3.95	2.42%	6.50	1.66	0.00%	9.96	0.39%	23.69	3.98	2.23%	6.50	1.68
4	480, 663	20.60	1.40%	47.06	6.36	3.03%	13.39	2.72	0.05%	20.74	2.13%	47.00	6.39	2.90%	13.38	2.71
5	663, 124	20.71	3.39%	38.37	3.40	0.41%	12.16	1.31	-0.01%	20.74	3.56%	38.47	3.45	0.65%	12.17	1.31
6	663, 480	21.24	1.66%	38.59	3.76	0.09%	12.50	1.47	0.00%	21.27	1.83%	38.69	3.80	0.35%	12.50	1.46
7	663, 881	17.10	2.54%	35.46	4.65	-2.64%	10.51	1.91	0.01%	16.98	1.81%	35.47	4.67	-2.61%	10.51	1.93
8	732, 733	21.05	3.81%	41.52	4.40	11.11%	12.74	1.65	0.11%	21.07	3.90%	41.55	4.39	11.19%	12.73	1.62
9	732, 871	12.35	0.23%	27.59	3.96	6.44%	7.77	1.62	-0.08%	12.35	0.29%	27.61	3.98	6.50%	7.77	1.64
10	732, 943	20.94	4.33%	41.71	4.28	6.81%	12.61	1.67	-0.39%	20.91	4.14%	42.46	4.33	8.75%	12.66	1.66
11	811, 927	15.26	8.18%	23.65	3.44	13.47%	8.31	1.25	0.37%	15.35	8.82%	23.90	3.48	14.67%	8.27	1.24
12	829, 942	16.85	2.26%	36.37	4.12	9.11%	10.37	1.69	-0.03%	16.87	2.39%	36.38	4.10	9.13%	10.37	1.68
13	871, 899	18.82	0.98%	23.94	2.74	2.33%	12.89	1.22	-0.07%	18.82	0.97%	24.01	2.77	2.62%	12.90	1.24
14	943, 871	9.61	15.98%	23.05	4.23	3.91%	6.39	1.71	0.06%	9.58	15.67%	22.99	3.82	3.67%	6.39	1.55
	avg	16.70	3.54%	34.02	4.23	5.78%	10.45	1.73	0.02%	16.73	3.69%	34.09	4.21	6.00%	10.45	1.71

Table 3: Average Runtimes for Path-Averaging Emissions Methods

index	O-D pair	runtime (sec)
1	124, 663	67.7
2	323, 408	9.5
3	408, 323	3.2
4	480, 663	58.1
5	663, 124	12.1
6	663, 480	12.3
7	663, 881	4.7
8	732, 733	15.5
9	732, 871	4.0
10	732, 943	20.2
11	811, 927	5.5
12	829, 942	12.7
13	871, 899	12.2
14	943, 871	3.2
	avg	17.2

the instances, depending on the length of the path, data availability and the resulting complexity of the network. While the runtimes of the arc-averaging method are only a fraction of a second and are thus not reported in detail here, the path-averaging method requires about 17 seconds on average. This indicates that the arc-averaging method offers similar solution quality to the path-averaging method at a runtime that is much faster and may be usable in real-time routing situations.

In Table 2, we can also compare how emissions objectives change the solutions relative to the other objectives. We can observe that emissions-based methods choose paths on average just less than 4% longer in distance than the distance-based method. This again is an average based on a wide range of values, with the largest increase in distance percentage-wise clearly being O-D pair (943,871). Emissions-based methods also choose paths on average around 6% longer in terms of travel time than the travel time method. From Table 4, we see that the time-dependent travel time method chooses paths on average 11.53% more distance than the distance method, and the distance method chooses paths that take 11.87% more travel time than the travel time method. This makes it clear that emissions-minimizing paths strike a balance between the two benchmark approaches. On average, they do not increase distance as much as time-dependent travel time paths or increase travel time as much as minimum distance paths. We conjecture that the emissions-based methods often choose paths that highly resemble the distance-based paths, since distance directly impacts emissions, but swap out arcs that would lead to particularly long travel times, since they cause high fuel

Table 4: Results for Minimum Distance and Time-Dependent Travel Time Methods

		Minimum distance path (all departures)							Minimum time-dependent travel time path (all departures)						
index	O-D pair	distance	travel time			emissions			distance		travel time		emissions		
		avg	avg	std	%inc	avg	std	%inc	avg	%inc	avg	std	avg	std	%inc
1	124, 663	18.26	48.10	6.44	12.20%	13.08	2.83	3.74%	21.36	17.02%	42.87	6.29	13.46	2.66	6.70%
2	323, 408	10.51	29.71	3.82	19.48%	7.69	1.64	1.38%	13.73	30.60%	24.87	3.65	8.38	1.50	10.53%
3	408, 323	9.92	24.07	3.94	3.85%	6.55	1.67	0.83%	11.94	20.39%	23.18	3.21	7.38	1.35	13.67%
4	480, 663	20.31	48.41	6.48	5.99%	13.64	2.81	1.98%	21.20	4.37%	45.68	6.39	13.73	2.71	2.62%
5	663, 124	20.03	44.17	4.59	15.59%	13.22	1.93	8.69%	20.91	4.42%	38.22	3.70	12.26	1.45	0.77%
6	663, 480	20.89	40.23	4.12	4.33%	12.74	1.65	1.92%	21.25	1.71%	38.56	3.94	12.51	1.57	0.08%
7	663, 881	16.68	36.85	4.88	1.18%	10.61	2.04	0.98%	17.33	3.93%	36.42	4.65	10.59	1.92	0.78%
8	732, 733	20.28	51.96	5.41	39.04%	13.73	2.17	7.88%	21.87	7.86%	37.37	4.44	12.98	1.73	2.01%
9	732, 871	12.32	27.97	3.96	7.89%	7.84	1.64	0.87%	14.14	14.77%	25.92	3.39	8.49	1.42	9.24%
10	732, 943	20.07	48.79	5.25	24.95%	13.35	2.21	5.44%	22.87	13.91%	39.05	4.38	13.47	1.81	6.42%
11	811, 927	14.10	20.92	2.31	0.34%	8.78	0.89	6.05%	14.19	0.58%	20.84	2.36	8.72	0.91	5.35%
12	829, 942	16.48	39.57	4.56	18.69%	10.91	1.90	5.14%	18.47	12.10%	33.34	4.88	11.42	2.02	10.09%
13	871, 899	18.64	23.53	2.77	0.57%	12.95	1.25	0.41%	18.80	0.86%	23.39	2.68	12.95	1.21	0.40%
14	943, 871	8.28	24.87	4.80	12.12%	6.60	2.19	3.38%	10.68	28.87%	22.18	3.11	6.44	1.29	0.82%
	avg	16.20	36.37	4.52	11.87%	10.84	1.92	3.48%	17.77	11.53%	32.28	4.08	10.91	1.68	4.96%

usage and emissions. We will look at such paths in Section 6.4.

In Table 4, we see that emissions-based methods reduce emissions by roughly 3.5% over the minimum distance path and 5.0% relative to the minimum time-dependent travel time path. This 5.0% is based on an average of values ranging from 0.08% to 13.67% for the different O-D pairs. This wide range of values reflects that using an emissions-based method is particularly important for certain O-D pairs. Both the minimum distance and the minimum time-dependent travel time paths are worse than path-averaging emissions in terms of emissions for all O-D pairs. This shows the value of an approach focused on minimizing emissions.

The O-D pairs with highest improvement in emissions relative to distance based methods are primarily the ones involving traveling from one side of the city to the other via the city center. There are exactly three O-D pairs with less than 1% difference in emissions (index 3, 9, and 13), and each of these represent travel from the suburbs to the city center or the city center to the suburbs. Such paths tend to be shorter and only have traffic in the morning or evening.

As documented in the electronic supplement of the paper, for the majority of instances and departure times (73% of investigated paths), the emissions-based methods identify solutions that are different from and superior to paths provided by traditional path finding objectives. For 25% of investigated paths, emissions-based and traditional path finding objectives lead to a tie, which indicates that there is not much path flexibility in the network for these particular O-D pairs and departure times. We acknowledge that there are a few cases where traditional path finding objectives can provide slightly better results due to the samples chosen for evaluation (2% of investigated paths). We expect this problem to disappear with larger sample sizes and increasing data availability.

6.4 Example O-D pair

In this section, we examine O-D pair (124, 663). The paths for this O-D pair travel from the southern to the northern suburbs via the city center. The results can be found in Table 5. For each of the four methods, we report the distance ("dist"), the mean travel time ("tt"), the standard deviation of the travel time ("std"), the mean emissions ("em"), and the standard deviation of the emissions values ("std em"). All of the units are the same as for the previous tables, and again the performance of each path is simulated 2000 times to compute these means and standard deviations.

In Table 5, the distance values tell us how often the path selected by a different method changes over the course of the day. Clearly, the minimum distance path will be the same over the course of the day, so this value is 18.26. For the time-dependent travel time method, we see the path changes in the morning (7:30, 8:30) due to rush hour traffic conditions, but stays pretty much the same for the rest of the day. Emissions-based methods change paths much more often. From the distances, we can also see that the paths chosen by path-based methods and arc-averaging methods are often different, such at the 6:30 and 7:30 departures times, even though the associated emissions are quite close. This indicates that there can be multiple paths in the graph at a particular departure time that have close to the same total emissions. Since these solution methods are focused only on emissions and not distance, it is not surprising that one emissions-based method may choose a path with higher or lower distance than the other at certain times of day.

It is also interesting to see how the emissions values change over the course of the day for all methods. Emissions are at their worst for the path-averaging method at 16:30 and 17:30 as expected and are at their minimum at 20:30. The peak emission is roughly 31% higher than the minimum emission. Other methods yield similar patterns over the day.

Next, we will look at the specific paths that were chosen over the course of the day for this O-D pair. In the graphs, the minimum distance path is labeled "dist," the minimum time-dependent travel time path is labeled "td-tt", the minimum arc-averaging emissions path is labeled "a-a", and the minimum path-averaging emissions path is labeled "p-a". First, we will look at the graph of all paths chosen at the 7:30 departure time (morning peak) in Figure 2. All paths are similar to the minimum distance path (except for a small detour by path-averaging emissions). After avoiding the highway in the south, all paths use the main through road to reach the destination. At noon, Figure 3 shows that the minimum time-dependent travel time path avoids the city center and uses several highways in the suburbs with a minor road along the border of the city center. Both emissions-minimizing paths follow the minimum distance path through the city center, but then pick an alternative major road in the western part of the inner city. During the afternoon peak, travel times and emissions values generally increase significantly. Figure 4 reflects the paths at 17:30. The minimum time-dependent travel time path again avoids the city center. The emissions-minimizing paths for

Table 5: Results over the day for (124, 663)

	Minimum distance					Minimum t-d travel time					Minimum arc-a emissions					Minimum path-a emissions				
departure	dist	tt	std	em	std	dist	tt	std	em	std	dist	tt	std	em	std	dist	tt	std	em	std
6:30	18.26	43.18	7.99	12.86	3.64	21.73	39.08	7.79	13.41	3.30	18.63	39.67	7.83	12.04	3.48	19.50	39.47	6.87	11.96	2.98
7:30	18.26	49.60	7.26	13.46	3.00	18.33	48.84	7.47	13.42	3.11	18.26	48.91	7.27	13.37	3.07	18.88	48.53	6.56	13.32	2.74
8:30	18.26	50.54	7.86	13.82	3.84	19.57	49.12	7.77	13.87	3.68	18.68	50.09	7.85	13.53	3.70	18.68	50.09	7.85	13.53	3.70
9:30	18.26	47.28	6.16	12.82	2.85	21.73	40.52	5.62	13.14	2.71	18.56	45.03	5.82	12.28	2.63	18.68	44.95	5.66	12.21	2.52
10:30	18.26	43.74	5.18	12.04	2.35	21.73	39.06	5.04	12.76	2.01	18.56	43.19	5.19	11.91	2.23	18.38	43.29	4.76	11.90	2.07
11:30	18.26	43.76	4.89	11.84	1.93	21.73	37.95	3.89	12.44	1.42	18.33	43.23	5.18	11.80	2.24	18.38	42.83	4.59	11.61	1.77
12:30	18.26	44.83	6.11	12.34	2.87	21.73	39.06	5.84	12.96	2.61	18.63	43.50	5.98	12.08	2.70	18.68	43.73	5.84	12.05	2.53
13:30	18.26	43.70	4.63	11.80	1.67	21.73	39.87	4.89	13.00	1.87	18.26	42.84	4.61	11.66	1.66	18.26	42.84	4.61	11.66	1.66
14:30	18.26	47.53	6.59	12.87	3.01	21.73	40.85	6.18	13.25	2.75	18.56	45.57	6.78	12.48	2.93	18.56	45.57	6.78	12.48	2.93
15:30	18.26	54.82	7.27	14.13	3.07	21.73	44.81	7.53	14.24	3.43	18.81	51.21	7.10	13.64	3.05	18.94	51.00	7.31	13.64	3.10
16:30	18.26	59.87	8.18	15.43	3.55	21.73	51.39	8.54	15.23	3.72	18.94	55.44	7.16	14.27	3.06	18.94	55.43	7.01	14.25	3.03
17:30	18.26	58.18	6.95	15.00	2.92	21.73	52.05	7.56	15.06	3.02	18.93	54.25	6.46	13.98	2.57	18.94	54.18	6.55	14.00	2.60
18:30	18.26	52.99	7.17	14.23	3.34	21.73	46.20	6.43	14.04	2.61	18.94	49.84	7.13	13.57	3.23	18.94	49.77	7.23	13.52	3.25
19:30	18.26	43.45	5.62	12.40	2.50	21.73	38.47	5.40	12.89	2.37	18.26	42.57	5.44	12.15	2.38	18.26	42.57	5.44	12.15	2.38
20:30	18.26	38.07	4.77	11.19	1.92	21.73	35.82	4.45	12.14	1.32	18.33	36.99	4.48	10.89	1.62	18.33	36.99	4.48	10.89	1.62
avg	18.26	48.10	6.44	13.08	2.83	21.36	42.87	6.29	13.46	2.66	18.58	46.16	6.29	12.64	2.70	18.69	46.08	6.10	12.61	2.59

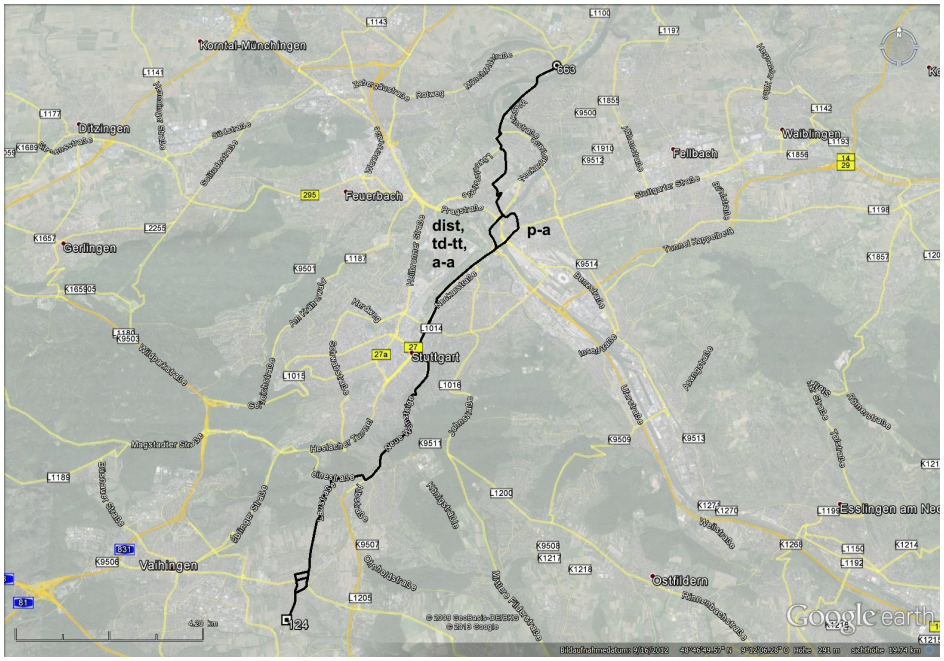

Figure 2: Path variants for O-D pair (124, 663), departure at 7:30 (aerial view provided by Google Earth)

the most part follow the minimum distance path, but access the inner city by a more western alternative major road that avoids some of the worst traffic. These examples graphically illustrate our earlier conjecture that emissions-minimized paths try to strike a balance between distance and travel time. They follow the distance path as much as possible since distance impacts emissions, but make changes where necessary to avoid roads with very low speeds that would lead to exceptionally high fuel usage and emissions values.

7 Conclusions and Future Work

In this paper, we present two data-driven approaches for the construction of expected time-dependent emissions minimizing paths in urban areas and compare their performance with regard to computational efficiency and quality of the solutions. The first approach incorporates sampling into an $A^\star$-based path construction algorithm. The second approach relaxes model fidelity to transform the path construction into a deterministic problem.

A surprising result is that expected emissions are about the same for the proposed arc-averaging and path-averaging methods. While the quality of solutions turned out to be comparable, the computational runtime is not. The arc-averaging method, which builds on a simplified estimation of arrival time distributions, can

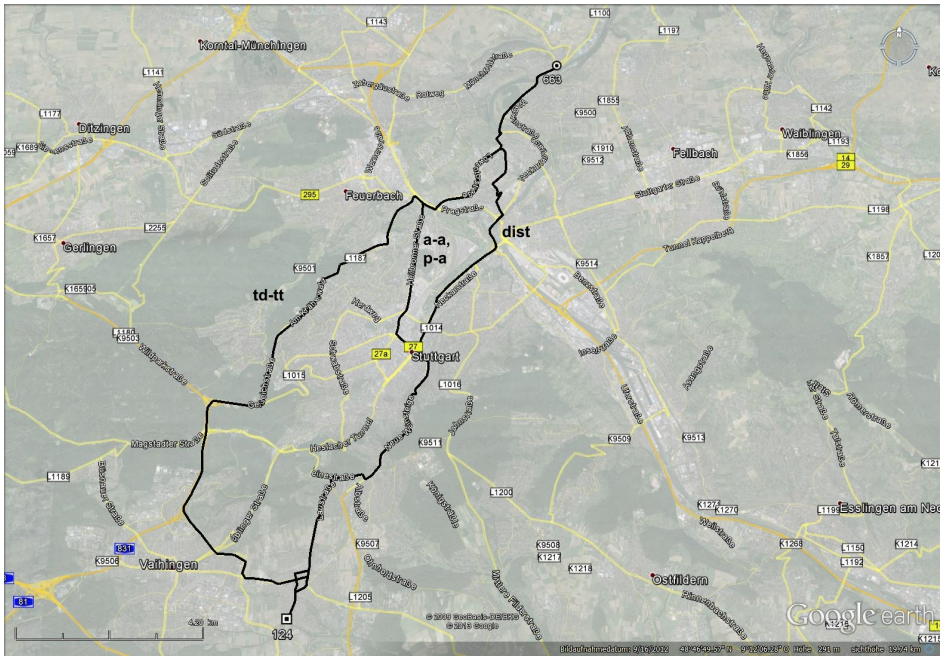

Figure 3: Path variants for O-D pair (124, 663), departure at 12:30 (aerial view provided by Google Earth)

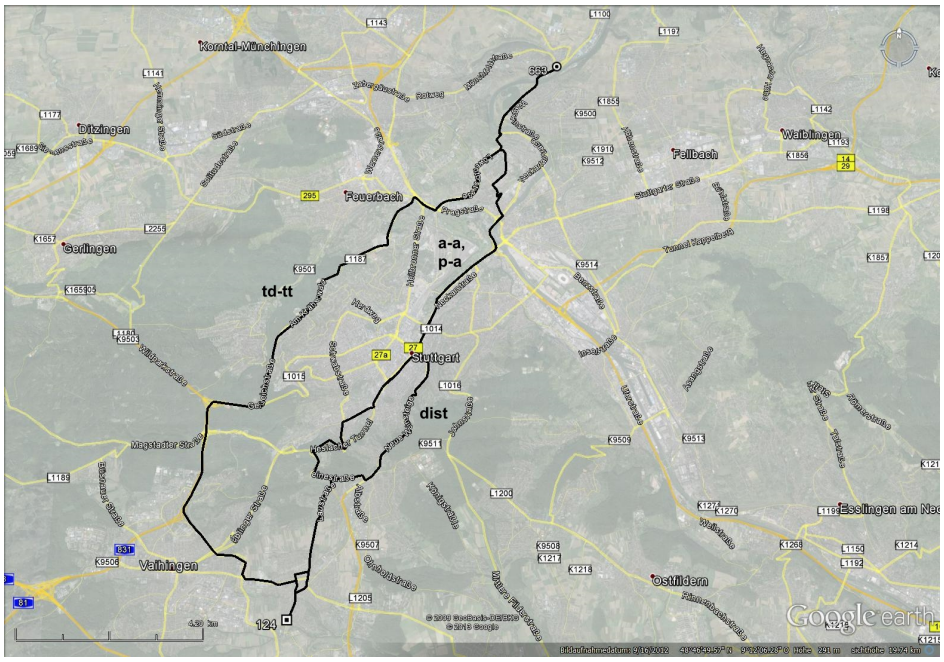

Figure 4: Path variants for O-D pair (124, 663), departure at 17:30 (aerial view provided by Google Earth)

provide solutions in significantly less runtime than the path-averaging method. Computational experiments also provide implications for sustainable urban logistics, especially with regard to the trade-offs between emissions minimization and other objectives. An important result is that emissions-based methods often choose paths that highly resemble the distance-minimizing paths, but swap out arcs that would lead to particularly long travel times. Interestingly, the most significant savings in emissions can be achieved when traveling from one suburb to another suburb via the city center. On average, emissions-minimizing methods can reduce emissions by roughly 3.5% over distance-minimizing paths and 5.0% relative to minimum time-dependent paths. The emissions-optimized path is around 4% longer than the paths generated by distance-based methods and around 6% longer in terms of travel time compared to travel-time optimized paths. These findings highlight the necessity of shortest path algorithms that are able to consider non-traditional objectives and large data sets.

Nevertheless, there are limitations in our approach that arise from the processing of "big data", and at some points of the network, data availability is still not sufficient. While at the moment we do not know a more comprehensive data set that overcomes these limitations, the ever growing collection of data will almost certainly overcome these limitations in the near future. Related to this, a number of opportunities exist for future work. For one, this paper examines only one model of vehicle emissions. While the proposed algorithms are indifferent to the choice of emissions model, it would be interesting to examine how the minimum expected emissions paths change. Second, the current results suggest that the two proposed approaches are comparable. However, running tests on longer paths may reveal more value to the approximation of arrival time distributions, thus demonstrating greater value of the path-averaging method. Such tests would require data for both a larger city as well as enough data to employ the sampling approach on the majority of the city's road network. The proposed solution approaches for expected emissions-minimizing paths also offer an opportunity to revisit vehicle routing problems with emissions objectives. As noted previously, existing literature has assumed deterministic path models that ignore the effects of variability. Our approaches can capture not only the effect of loaded vehicle weight on emissions, but also the impact of variability in travel speeds. Finally, the proposed sampling method can easily be generalized for applications beyond emissions, notably where arc costs are time-dependent and are evaluated via sampling.

8 Acknowledgments

The authors wish to thank the German Aerospace Center for allowing the use of the database of historical travel speeds for Stuttgart and to the German Academic Exchange Service for supporting Jan Fabian Ehmke's post-doctoral studies in the Department of Management Sciences at the University of Iowa. The authors also thank an anonymous referee whose comments helped improve the paper.

References

Barth, M. and Boriboonsomsin, K. (2008). Real-World Carbon Dioxide Impacts of Traffic Congestion. *Transportation Research Record: Journal of the Transportation Research Board*, 2058:163–171.

Bektas, T. and Laporte, G. (2011). The Pollution-Routing Problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250.

Brockfeld, E., Passfeld, B., and Wagner, P. (2007). Validating travel times calculated on the basis of taxi floating car data. In *Proceedings of 14th ITS Conference*, Beijing, China.

Chabini, I. and Lan, S. (2002). Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. *IEEE Transactions on Intelligent Transportation Systems*, 3(1):60–74.

Cooke, K. L. and Halsey, E. (1966). The Shortest Route through a Network with Time-Dependent Internodal Transit Times. *Journal of Mathematical Analysis and Applications*, 14(3):493–498.

Dean, B. C. (2004). Shortest paths in FIFO time-dependent networks: Theory and algorithms. Technical report, Massachusetts Institute of Technology.

Demeyer, S., Audenaert, P., Logghe, S., Pickavet, M., and Demeester, P. (2012). Practical Time-Dependent and Stochastic Routing with Historical Measurements of Travel Times. In *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1638–1643, Anchorage, Alaska, USA.

Demir, E., Bektas, T., and Laporte, G. (2011). A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5):347–357.

Demir, E., Bektas, T., and Laporte, G. (2012). An Adaptive Large Neighborhood Search Heuristic for the Pollution-Routing Problem. *European Journal of Operational Research*, 223(2):346–359.

Demir, E., Bektas, T., and Laporte, G. (2014). A Review of Recent Research on Green Road Freight Transportation. *European Journal of Operational Research*, 237(3):775–793.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Dreyfus, S. E. (1969). An Appraisal of Some Shortest-Path Algorithms. *Operations Research*, 17(3):395–412.

Ehmke, J. and Campbell, A. M. (2014). Customer Acceptance Mechanisms for Attended Home Deliveries in Metropolitan Areas. *European Journal of Operational Research*, 233(1):193–207.

Ehmke, J. F., Meisel, S., and Mattfeld, D. C. (2009). Data Chain Management for Planning in City Logistics. *International Journal of Data Mining, Modelling and Management*, 1(4):335–356.

Ehmke, J. F., Meisel, S., and Mattfeld, D. C. (2010). Floating car data based analysis of urban travel times for the provision of traffic quality. In Barceló, J. and Taniguchi, E., editors, *Traffic Data Collection and its Standardization*. Springer US.

Ehmke, J. F., Meisel, S., and Mattfeld, D. C. (2012a). Floating car based travel times for city logistics. *Transportation Research Part C: Emerging Technologies*, 21(1):338–352.

Ehmke, J. F., Steinert, A., and Mattfeld, D. C. (2012b). Advanced Routing for City Logistics Service Providers based on Time-dependent Travel Times. *Journal of Computational Science*, 3(4):193–205.

Figliozzi, M. (2010). Vehicle Routing Problem for Emissions Minimization. *Transportation Research Record: Journal of the Transportation Research Board*, 2197:1–7.

Franceschetti, A., Honhon, D., van Woensel, T., Bektas, T., and Laporte, G. (2014). The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, 56:265–293.

Fu, L. and Rilett, L. R. (1998). Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research Part B: Methodological*, 32(7):499–516.

Geisberger, R., Sanders, P., Schultes, D., and Vetter, C. (2012). Exact Routing in Large Road Networks Using Contraction Hierarchies. *Transportation Science*, 46(3):388–404.

Güner, A. R., Murat, A., and Chinnam, R. B. (2012). Dynamic routing under recurrent and non-recurrent congestion using real-time ITS information. *Computers & Operations Research*, 39(2):358–373.

Hickman, A. J., Hassel, D., Joumard, R., Samaras, Z., and Sorenson, S. (1999). MEET Methodology for Calculating Transport Emissions and Energy Consumption. Technical Report 3011746, Transport Research Laboratory.

Jabali, O., van Woensel, T., and de Kok, A. G. (2012). Analysis of Travel Times and CO2 Emissions in Time-Dependent Vehicle Routing. *Production and Operations Management*, 21(6):1060–1074.

Kao, E. P. C. (1978). A Preference Order Dynamic Program for a Stochastic Traveling Salesman Problem. *Operations Research*, 26(6):1033–1045.

Kröger, M. (2013). Stau-Analyse: Stuttgart quält seine Pendler am meisten (Analysis of traffic jams: Stuttgart tortures its commuters the most). In *Spiegel Online*.

Lecluyse, C., van Woensel, T., and Peremans, H. (2009). Vehicle routing with stochastic time-dependent travel times. *4OR*, 7(4):363–377.

Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., and Lam, H. Y. (2014). Survey of Green Vehicle Routing Problem: Past and future trends. *Expert Systems with Applications*, 41(4):1118–1138.

Lorkowski, S., Mieth, P., Thiessenhusen, K.-U., Chauhan, D., Passfeld, B., and Schäfer, R.-P. (2004). Towards Area-Wide Monitoring-Applications derived from Probe Vehicle Data. In *AATT 2004*, pages 389–394.

Orda, A. and Rom, R. (1990). Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM*, 37(3):607–625.

Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Publishing Company.

Sherali, D. D., Hobeika, A. G., and Kangwalklai, S. (2003). Time-Dependent, Label-Constrained Shortest Path Problems with Applications. *Transportation Science*, 37(3):278–293.

Sherali, H. D., Jeenanunta, C., and Hobeika, A. G. (2006). The Approach-Dependent, Time-Dependent, Label-Constrained Shortest Path Problem. *Networks*, 48(2):57–67.

Sherali, H. D., Ozbay, K., and Subramanian, S. (1998). The time-dependent shortest pair of disjoint paths problem: Complexity, models, and algorithms. *Networks*, 31(4):259–272.

TNT Express (2012). TNT Express Annual Report.

UPS (2013). Saving Fuel: UPS Saves Fuel and Reduces Emissions the "Right" Way by Avoiding Left Turns.

van Woensel, T., Creten, R., and Vandaele, N. (2001). Managing the Environmental Externalities of Traffic Logistics: the Issue of Emissions. *Production and Operations Management*, 10(2):207–223.

Ziliaskopoulos, A. K. and Mahmassani, H. (1993). Time-dependent shortest path algorithm for real-time intelligent vehicle/highway system. *Transportation Research Record: Journal of the Transportation Research Board*, 1408:94–104.