Physics-guided Energy-efficient Path Selection: A Summary of Results

Yan Li
Shashi Shekhar
lix4266@umn.edu
shekhar@umn.edu

Dept. of Computer Science & Eng
University of Minnesota - Twin Cities
Minneapolis, Minnesota

Pengyue Wang
William Northrop
wang6609@umn.edu
wnorthro@umn.edu

Dept. of Mechanical Engineering
University of Minnesota - Twin Cities
Minneapolis, Minnesota

ABSTRACT

Given a spatial road network, an origin, a destination, and trajectory data of vehicles on the network, the Energy-efficient Path Selection (EPS) problem aims to find the most energy-efficient path (i.e., with least energy consumption) between the origin and the destination. With world energy consumption growing rapidly, estimating and reducing the energy consumption of road transportation is becoming critical. The main challenge of this problem is to adopt energy consumption as the cost metric of paths, which is neglected by the related work in shortest path selection problem whose typical metrics are distance and time. Additionally, negative energy consumption caused by the use of regenerative braking on electrified vehicles prevents classical algorithms like Dijkstra's algorithm from functioning correctly. We introduce a Physics-guided Energy Consumption (PEC) model based on a low-order physics model, which estimates energy consumption as a function of the vehicle parameters (e.g., mass and powertrain system efficiency) and use the estimation in the proposed adaptive dynamic programming algorithm for path selection. Our PEC model treats energy consumption as a unique metric that is determined not only by the path and vehicle's motion along the path, but also on properties of the vehicle itself. Experiments show that the PEC model estimates are more similar to real trajectory data than the estimates represented by the mean or histogram of historical data. Also, the path found by the proposed method is more energy-efficient than both the currently used path and the fastest path found by a commercial routing package. As far as we know, this is the first paper to use a physics-guided method to estimate the vehicle energy consumption and perform path selection.

CCS CONCEPTS

• Information systems → Geographic information systems; Data mining;

KEYWORDS

routing, physics-aware, shortest path, energy efficiency

ACM Reference Format:

Yan Li, Shashi Shekhar, Pengyue Wang, and William Northrop. 2018. Physics-guided Energy-efficient Path Selection: A Summary of Results. In *26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '18)*, November 6–9, 2018, Seattle, WA, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3274895.3274933>

1 INTRODUCTION

Given a spatial road network, an origin, a destination, and trajectory data of vehicles on the network, the Energy-efficient Path Selection (EPS) problem aims to find the most energy-efficient path (i.e., with least energy consumption) between the origin and the destination. The EPS problem, a variant of the Shortest Path Selection (SPS) problem, has two subtasks, namely, estimating the energy consumption of a vehicle along a set of paths, and then selecting the most energy-efficient path for that vehicle.

The goal of this problem is to estimate and reduce the energy cost for road transportation, which is important and necessary both economically and environmentally. In 2015, energy consumption in transportation in the United States cost more than \$507 billion [22], and contributed significantly to air pollution and anthropogenic climate forcing. Supported by U.S. Department of Energy, many efforts are being put into the development of energy-efficient vehicle technologies, such as electric cars, to achieve better fuel economy [8]. Despite this attention, energy use continues to climb. The U.S. Department of Energy predicts world energy consumption for transportation will rise 28% between 2015 and 2040 [21]. On the other hand, much computer science study has been conducted on the SPS problems to improve the ability of navigation systems to select the shortest or fastest routes. However, shorter travel distance or time does not necessarily mean less energy consumption. Algorithms designed around time or distance as the path cost metric have ignored the influence of a vehicle's physical properties on the amount of energy consumed along a path. Our work addresses this limitation by incorporating energy consumption explicitly into the SPS problem to find the most energy-efficient path.

Adopting energy consumption as the path cost metric rather than time or distance in the traditional SPS problem is challenging. Energy consumption is affected not just by the path and a vehicle's motion on the path, but also by vehicle parameters (e.g., cargo weight). In other words, different vehicles traveling along the same path following the same velocity profile will have different

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

SIGSPATIAL '18, November 6–9, 2018, Seattle, WA, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5889-7/18/11...\$15.00

<https://doi.org/10.1145/3274895.3274933>

Figure 1: An example for basic concepts.

rates of energy consumption. The same is true for one vehicle with varying parameters like passenger mass. Thus, the energy cost of traveling along a path cannot be represented by a statistic such as the average or the distribution of historical data, which are the commonly-used representations for distance and time. Therefore, a better integration of trajectory data from different vehicles or from a single vehicle's many trips is necessary. Additionally, the wide use of regenerative braking in electrified vehicles (e.g., hybrids) complicates the problem by possibly making energy consumption negative, which prevents classical algorithms like Dijkstra's algorithm from functioning correctly.

In this work, our primary contributions are three-fold. First, we introduce a Physics-guided Energy Consumption (PEC) model based on a low-order physics model to estimate the energy consumption of a vehicle along a path. The PEC model considers the dependence of energy consumption on variable vehicle parameters. Second, we propose a dynamic programming algorithm based on the estimation to select the most energy-efficient path that handles the possibly negative energy consumption efficiently with an early-stop strategy. The experiments show that the PEC model estimates are more similar to real trajectory data than the state-of-the-art model which represents a path's cost as a distribution [24]. The experiments also indicate that the paths found by the proposed method are more energy-efficient than both the currently used paths and the fastest paths found by a commercial routing package [17]. Third, we present a case study using real-world trajectory data from a package delivery company and a road map from OpenStreetMap. As far as we know, ours is the first study to propose a physics-guided method to estimate the energy consumption and perform path selection.

This paper is organized as follows: In §2, we explain the basic concepts and formally define the physics-guided energy-efficient path selection problem. §3 reviews the related literature. §4 presents our model and algorithm for solving the problem, and an evaluation is given in §5. §6 concludes the paper and presents our future work.

2 BASIC CONCEPTS AND PROBLEM DEFINITION

We introduce the basic concepts in this study, based on which the Energy-efficient Path Selection (EPS) problem is formally defined.

2.1 Basic Concepts

A **road segment** (r) is a portion of a road with uniform characteristics between two road intersections. A **road network** is a system of roads composed of road segments and intersections. Figure 1 shows an example road network consisting of 12 road segments and 12 intersections.

A **path** (P) is a set of road segments linking an ordered sequence of intersections. Its origin and destination are the first and the last intersections respectively. The union of two paths (i.e., $P_i \cup P_j$) is a path composed of all of their road segments. A **subpath** ($sub(P)$) of path P is defined as a path whose road segments belong to P . Given two paths $P_1 = [r_5, r_4, r_6]$ and $P_2 = [r_4, r_6, r_9]$ in Figure 1, $P_1 \cup P_2 = [r_5, r_4, r_6, r_9]$, and $sub(P_1)_1 = [r_5, r_4]$ is a subpath of P_1 .

A **trajectory** (T) is a record of a path traveled by a vehicle and the amount of energy it consumes on each road segment of the path. A trajectory T is said to exist along a path P if P is a subpath of the path recorded by T . For example in Figure 1, there is one trajectory along path $[r_4, r_7, r_{10}]$ whose energy consumption is in the form of a vector $[c_4, c_7, c_{10}]$, where c_i is the energy consumption of the trajectory on r_i . From trajectory data, we can infer each trajectory's **motion properties** as three vectors representing the velocity and acceleration of the logged vehicle and the time it uses on each road segment. A **scenario** on a path is a combination of attributes shared within a group of similar trajectories, including vehicle model and motion properties.

Ideally, the most accurate approach for estimating the energy consumption of a vehicle moving along a path is to employ a sizable set of trajectories along the same path. However, the number of trajectories along a path decreases rapidly as the number of road segments in the path increases. For a more practical method, we classify all paths into three groups (similar to [24]), namely, **trajectory-aware paths** (TAP_β s), **trajectory-union paths** (TUP s), and other paths. A TAP_β is a path along which there are at least β trajectories. A TUP is the union of several TAP_β s. The remaining paths are other paths. Suppose $\beta = 1$. In Figure 1, $[r_5, r_7]$ and $[r_7, r_{10}]$ are two TAP_1 s, and $[r_5, r_7, r_{10}]$ is the TUP obtained by joining them, and there is no trajectory on path $[r_1, r_6, r_{11}]$ which belongs to other paths. In this paper, we introduce a model for estimating the energy consumption of the first two types of paths. Due to a lack of trajectory data, the estimation of other paths is outside the scope of this paper.

2.2 Problem Definition

We formally define the Energy-efficient Path Selection (EPS) problem as follows:

Input:

- A set of historical trajectories.
- A road network.
- A threshold for the minimal number of trajectories along a trajectory-aware path β .
- Two road intersections o and d .

Output: A path between o and d with the smallest estimated energy consumption.

Objective: Avoid a path that is energy-inefficient.

Constraints:

- The resulting path is a trajectory-aware path or a trajectory-union path.
- Information to estimate the vehicle's parameters is provided.

Given the trajectories and the road network shown in Figure 1, a sample task of the proposed problem is to find the most energy-efficient path between r_5 and r_{10} . Suppose $\beta = 1$. There are two TUPs between r_5 and r_{10} , namely, $[r_5, r_7, r_{10}]$ and $[r_5, r_4, r_6, r_9, r_{10}]$, from which the most energy-efficient path is chosen. Since many factors affecting energy consumption are hard to predict before a trip (e.g., accidents), we cannot guarantee that the path selected by our method is actually the most energy-efficient, but we can guarantee that it is not energy-inefficient. This type of strategy is often used in real-world engineering problems, such as query optimization in DBMS.

3 RELATED WORK AND LIMITATIONS

The Energy-efficient Path Selection (EPS) problem is a variant of the Shortest Path Selection (SPS) problem. Based on how path cost is estimated, the related work on SPS problem can be categorized into two groups, i.e., edge-centric, and path-centric methods (the left branch in Figure 2).

Edge-centric methods treat the cost of a path as the sum of the cost of individual road segments. The work dates back to the 1950s and 1960s when Dijkstra's [9] and the Bellman-Ford [15] algorithms were proposed to select the shortest path in a static-weighted graph where the cost on each road segment is a constant. Later study have focused on accelerating computation [1, 3, 6, 20], as well as introducing new constraints (e.g., battery capacity constraint for electric vehicles [2, 11]) and cost metrics (e.g., happiness [18], bi-objectives metric [10]). In order to make the representation of the cost of each road segment more accurate, some work extends the road network as a spatio-temporal network, in which the cost of each road segment varies with time rather than staying the same [5, 13]. Besides time, there are other factors affecting the cost of each road segment that are hard to model directly. Some research represents the cost of each road segment as a stochastic distribution [7, 12]. In all of the aforementioned methods, the cost of each road segment (a constant or a stochastic distribution) is assumed to be prior knowledge. In order to estimate the cost, methods have also been introduced to utilize the growing volumes of trajectory data being collected from vehicles [16, 23, 25, 26]. However, all the edge-centric methods suffer from the problem that when decomposing trajectories to estimate the cost of individual road segments, some information, such as the dependence between the costs of adjacent parts along a trajectory, will be lost.

Rather than thinking of a path as a sequence of individual road segments, path-centric methods treat it as a sequence of overlapping subpaths. The approach in [24] decompose a path into subpaths followed by representing the cost of the path derived from trajectories along subpaths as a stochastic distribution. Since the smallest unit in the path-centric methods to estimate the cost of a path is a subpath, these methods maintain the dependence between the costs of adjacent parts along a trajectory, which is beneficial to energy consumption estimation. Take three connected road segments in our dataset as an example. There are 69 trajectories along a path composed of road segments 803 and 11234, whose average energy

Figure 2: Related work.

Table 1: Path cost varies by vehicle weight

Trip	Cost (KWh)	Weight (kg)
1	3.1	3000
2	5.2	5000
3	5	5000
4	4.9	5000
5	4	4000
6	2.9	3000

consumption on road segment 11234 is 0.076 KWh. Along another path composed of road segments 11233 and 11234 there are 83 trajectories, whose average energy consumption on road segment 11234 is 0.064 KWh. One potential cause for this difference is that the left turn between road segments 803 and 11234 makes the vehicles traveling on them decelerate and accelerate at the intersection and results in relatively higher energy consumption. If we use an edge-centric method, decompose trajectories into road segments, and treat all 152 trajectories on road segment 11234 the same, the estimation of the cost on road segment 11234 cannot reflect this difference. By contrast, in a path-centric method, trajectories on the two paths are dealt with separately, so the difference will be maintained and estimation will be more accurate. Thus, we propose a path-centric model for energy consumption estimation.

However, neither the edge-centric nor the path-centric methods in the related work are able to reflect the dependence of energy consumption on varying vehicle parameters. Vehicle mass is an example of the parameters having large effects on energy use and can vary frequently. For example, suppose a vehicle moves along a path with the same motion properties 6 times. Table 1 shows the energy consumption and the weight of the vehicle on each trip. Answering a question like "What is the typical energy consumption of a vehicle with a weight of 3000 kg on this path?" with a statistic of the cost such as the mean or a random variable distribution (methods commonly used by the previously-mentioned related work) is not accurate, since there are only two trips where the weight of the vehicle is 3000 kg, and any trip where the vehicle's weight is higher will likely consume more energy. Instead, 3 KWh may be a better answer. Because of the meaninglessness of energy consumption estimates using the related work, none are applicable for selecting energy-efficient paths in a road network.

Meanwhile, there also exists a branch of studies on the EPS problem which assumes energy consumption on a road segment

Table 2: Table of symbols in Section 4.1

Symbol	Physical Interpretation
c	energy consumption
$EAIR$	energy consumption for air resistance
VP	vehicle parameter factor of energy consumption for rolling resistance
MP	motion property factor of energy consumption for rolling resistance
$\hat{X}$	estimation of $X \in \{EAIR, VP, MP\}$
$X_i(j)$	$X \in \{c, EAIR, VP, MP\}$ of trajectory i on road segment j

is calculated according to a simplified model and focuses on routing based on the assumed model [11, 19]. However, the simplified model cannot handle spatial heterogeneity and the dependence between adjacent road segments, which may result in incorrect energy consumption estimation and path selection.

4 APPROACH

We first introduce a Physics-guided Energy Consumption (PEC) model to estimate the energy cost of a vehicle moving along a path, which accounts for the dependence of energy consumption on vehicle parameters. Then we present a dynamic programming algorithm to select the most energy-efficient path using the estimates from the PEC model.

4.1 Physics-guided Energy Consumption Model

The PEC model is a path-centric model, which has two parts: (1) a scenario-based model for trajectory-aware paths (TAP_β s); (2) a TAP_β -union model for trajectory-union paths (TUP s). Table 2 shows the symbols used in this section to clarify the discussion.

4.1.1 Scenario-based model for TAP_β s. The scenario-based model is derived from a simplified powertrain energy consumption model [4], in which the energy consumption of a vehicle moving along a path is calculated as follows:

$$c = \frac{t}{\eta}(mav + c_{rr}mgv + \frac{1}{2}c_{air}Apv^3). \quad (1)$$

Briefly, the energy consumption (c) of a vehicle on a path is determined by the vehicle's motion properties (i.e., time (t), acceleration (a), and velocity (v)) as well as its physical parameters (i.e., mass (m), front surface area (A), and powertrain system efficiency (η)). c_{rr} , c_{air} and ρ are physical constants. We denote the part of energy used for air resistance as $EAIR = \frac{A}{2\eta}c_{air}\rho tv^3$, and denote the part of energy used for rolling resistance and acceleration as the product of a vehicle parameter factor $VP = \frac{m}{\eta}$ and a motion property factor $MP = tav + c_{rr}tg$. Equation 1 can be written as

$$c = EAIR + VP \times MP, \quad (2)$$

where c , $EAIR$ and MP are vectors whose elements represent their values on each road segment, while VP is a scalar determined only by the vehicle.

Vehicles of the same model have the same front surface area and powertrain system efficiency, but possibly varying mass due to different cargo loads. Therefore, we cluster the trajectories into groups according to their MP , which is determined by motion properties, and $EAIR$, which is determined by motion properties as well as a vehicle's front area and powertrain system efficiency, so that trajectories in each group serve as a log of vehicles of similar models and motion properties, i.e., similar scenarios as defined in Section 2. The energy consumption of the trajectories in a scenario will be a linear function of VP , whose intersect and slope are the trajectories' shared $EAIR$ and MP respectively. We call $EAIR$ and MP the signature of each scenario, and call the linear function of VP the signature function of each scenario.

Thus, if energy consumption for a group of trajectories can be represented by the same linear function $Work = f(VP)$, we conclude that these trajectories tend to occur in the same scenario. We can estimate the cost of a vehicle traveling along a path if we know: (1) all the scenarios on the path and their signature functions; (2) the scenario the vehicle is in currently, and its VP .

We adopt the K-means algorithm, a popular clustering method, to group the trajectories along a TAP_β into k scenarios according to their $EAIR$ and MP . We assign the passing trajectories to the closest scenario, and then update the signature of each scenario according to the trajectories belonging to it. These two operations are performed iteratively until convergence.

The rules to calculate $EAIR$ and MP are derived from Equation 2. Suppose there are l trajectories belonging to an arbitrary scenario with a signature of $EAIR_s$ and MP_s . If we represent the energy consumption and vehicle parameter factor of each trajectory as c_i and VP_i ($i = 1, 2, \dots, l$), the difference between the energy consumption of two arbitrary trajectories will be

$$c_i - c_j = (VP_i - VP_j) \times MP_s, \quad (3)$$

where $i, j = 1, 2, \dots, l$ and $i \neq j$. In other words, the difference between the energy consumption of two trajectories in a scenario is a multiple of their shared MP . Therefore, we assume that if the pairwise energy consumption differences of more than three trajectories are multiples of each other, the trajectories will share the same MP .

Based on the above analysis, we can derive specific equations for the assignment and update steps of the K-means algorithm. Suppose that the path of the scenario above has n road segments, and each trajectory's cost c_i is a vector of n elements. Its j th element, $c_i^{(j)}$, represents the energy consumption of the i th trajectory on the j th road segment. The algorithm will iterate between the following two steps:

Update step: According to Equation 3, we can represent a multiple of MP_s (denoted as $\hat{MP}$) by the average pairwise differences between energy consumption of the l trajectories:

$$\hat{MP} = \frac{\sum_{i=2}^l (c_i - c_1)}{\sum_{i=2}^l (c_i(1) - c_1(1))}. \quad (4)$$

In this Equation, $\hat{MP}$ is standardized such that the first element is 1. Because we are only interested in a multiple of MP , this standardization does not affect the result. Meanwhile, the energy consumption of every trajectory in a scenario can be represented by the shared

$EAIR$ plus a multiple of the shared MP according to Equation 2. It is also equal to the energy consumption of any other trajectory in the scenario plus a different multiple of the shared MP according to Equation 3. Thus, we can replace $EAIR_s$ with the average energy consumption of all l trajectories in the scenario (denoted as $\hat{EAIR}$):

$$\hat{EAIR} = \frac{\sum_{i=1}^l c_i}{l}. \quad (5)$$

Because $\hat{EAIR}$ is equal to the scenario's $EAIR$ ($EAIR_s$) plus a multiple of the scenario's MP (MP_s), it is clear that the energy consumption of all trajectories in the scenario is equal to $\hat{EAIR}$ plus a multiple of $\hat{MP}$.

Assignment step: Let $\hat{EAIR}_s$ and $\hat{MP}_s$ be the signature of the s th scenario on a path, where $s = 1, 2, \dots, k$. The energy consumption of a trajectory in the scenario can be represented by the scenario's signature function, i.e., $\hat{EAIR}_s + \mu \times \hat{MP}_s$, where $\mu \in \mathbb{R}$ is unknown. We define a metric for evaluating whether a trajectory's cost c_i can be represented by the s th scenario's signature function as their difference

$$Diff_s = \min_{\mu \in \mathbb{R}} \|c_i - \hat{EAIR}_s - \mu \times \hat{MP}_s\|. \quad (6)$$

The smaller the $Diff_s$, the closer the trajectory is to the scenario. Each trajectory is assigned to the closest scenario.

With respect to initialization of the algorithm, we apply a modified Forgery method, which randomly chooses k scenarios from the data [14]. The s th scenario is initialized as

$$\begin{aligned} \hat{EAIR}_s &= \frac{c_{2s-1} + c_{2s}}{2}, \\ \hat{MP}_s &= c_{2s-1} - c_{2s}. \end{aligned}$$

The number of scenarios is determined by an Elbow method. The idea is to run the K-means algorithm with a range of values of k . The performance for each k is evaluated by the sum of the difference between the energy consumption of each trajectory and the scenario it belongs to. If the value of sum or the change of sum between adjacent runs is smaller than a threshold, we choose the corresponding k .

After finding the scenarios on a TAP_β , given the scenario a vehicle is in and the vehicle parameter factor (VP), we can evaluate the cost for the vehicle of moving along the path. The scenario a vehicle is in and the corresponding VP can be predicted using the vehicle's historical data before the trip and then may be modified accordingly after more data is collected during the trip. In this paper, for simplicity, we use a vehicle's energy consumption on the first two road segments of the TAP_β to determine the scenario the vehicle is in and its VP . Given a vehicle's energy consumption on the first two road segments c , which is a vector of two elements, we evaluate the difference of c and a scenario with a signature of $\hat{EAIR}_s$ and $\hat{MP}_s$ as

$$Diff_s = \min_{\mu \in \mathbb{R}} \|c - \hat{EAIR}_s[1:2] - \mu \times \hat{MP}_s[1:2]\|, \quad (7)$$

where $\hat{EAIR}_s[1:2]$ and $\hat{MP}_s[1:2]$ are vectors consisting of the first two elements of $\hat{EAIR}_s$ and $\hat{MP}_s$ respectively. The vehicle is in the scenario whose difference from the vehicle's cost is the smallest. Its VP is the value of μ corresponding to the smallest difference. For example, in Figure 3, 6 trajectories are along a TAP_β composed of 3 road segments. The K-means algorithm clusters the trajectories

Figure 3: An example for the scenario-based model on a TAP_β .

into 2 scenarios such that the energy cost of the trajectory 1, 2, and 3 is equal to $EAIR_1 + \theta_1 \times MP_1$ where $\theta_1 \in \mathbb{R}$, while the energy cost of the trajectory 4, 5, and 6 is equal to $EAIR_2 + \theta_2 \times MP_2$ where $\theta_2 \in \mathbb{R}$. Suppose we are given a vehicle's energy cost on r_1 and r_2 as [5, 9]. We will find that it belongs to Scenario 1, and that its estimated energy cost on this TAP_β is [5, 9, 7].

4.1.2 TAP_β -union model for TUPs. In order to evaluate the energy consumption of a trajectory-union path (TUP), we propose a model based on the path decomposition method introduced in [24]. A decomposition of a path P is defined as a sequence of paths $DE = [P_1, P_2, \dots, P_n]$ ($n > 1$) which satisfies three conditions:

- The union of all paths in the path decomposition is P ;
- A path P_i is not a subpath of another path P_j ($1 \leq i, j \leq n, i \neq j$);
- The first road segment of P_i appears earlier than that of P_j in path P if $i < j$.

A path decomposition DE_1 is defined to be coarser than another DE_2 if each path $P_b \in DE_2$ is the subpath of at least one path $P_a \in DE_1$ and at least one $P_b \neq P_a$. Since a TUP is the union of a set of TAP_β s, we have Lemma 4.1.

LEMMA 4.1. *A TUP has at least one path decomposition where each element is a TAP_β .*

PROOF. Suppose an arbitrary TUP (P) is the union of a sequence of TAP_β s (P_i where $i = 1, 2, \dots, n$). We conduct the following two operations to get a path decomposition of P :

- Order the TAP_β s by their first road segments.
- Eliminate the paths that are subpaths of other paths.

Because the two operations do not affect the union of the paths which is P , and the remaining paths satisfy the second and third conditions, the sequence composed of the remaining paths is a path decomposition of P . $\square$

In order to estimate the energy consumption of a TUP , we add two more conditions to its path decomposition: (1) each element is a TAP_β ; (2) two adjacent TAP_β s in the decomposition must share at least two road segments so that the scenarios on them can be joined.

Next we show how to generate the union of two adjacent TAP_β s, i.e., P_i, P_{i+1} , in a decomposition of a TUP . Suppose P_i has k_i scenarios, specifically the signature of the s th scenario (denoted as $Scenario_i(s)$) are $\hat{EAIR}_i(s)$ and $\hat{MP}_i(s)$ ($s = 1, 2, \dots, k_i$). We will say that a scenario on P_i is similar to a scenario on P_{i+1} when the energy consumption represented by the two scenarios' signature functions are similar on the shared road segments. To make it clear, let $\hat{EAIR}_i(s)'$ and $\hat{MP}_i(s)'$ be vectors composed of the elements of $\hat{EAIR}_i(s)$ and $\hat{MP}_i(s)$ corresponding to the shared road segments. The energy consumption represented by the s th scenario of P_i and t th scenario of P_{i+1} on the shared road segments are $\hat{EAIR}_i(s)' + \mu_i \times \hat{MP}_i(s)'$ and $\hat{EAIR}_{i+1}(t)' + \mu_{i+1} \times \hat{MP}_{i+1}(t)'$ respectively. As we noted in Section 4.1.1, $\hat{MP}$ is a multiple of the true Motion Property factor; for simplicity, we assume that $\hat{MP}_i(s)'$ being a multiple of $\hat{MP}_{i+1}(t)'$ is equivalent to the fact that the Motion Property factors of two scenarios are similar on their shared road segments. When the two conditions are satisfied, that is $\hat{MP}_i(s)'$ is a multiple of $\hat{MP}_{i+1}(t)'$, and $\hat{EAIR}_i(s)'$ is equal to $\hat{EAIR}_{i+1}(t)'$ plus a multiple of $\hat{MP}_{i+1}(t)'$, we say that the energy consumption represented by the two scenarios is similar on the shared road segments. Thus, we define the difference between the energy consumption represented by scenarios as

$$\begin{aligned} Diff_{s,t} = & \min_{\theta \in \mathbb{R}} \|\hat{MP}_i(s)' - \theta \times \hat{MP}_{i+1}(t)'\| + \\ & \min_{\gamma \in \mathbb{R}} \|\hat{EAIR}_i(s)' - \hat{EAIR}_{i+1}(t)' - \\ & \gamma \times \frac{\hat{MP}_i(s)' + \hat{\theta} \times \hat{MP}_{i+1}(t)'}{2}\|, \end{aligned} \quad (8)$$

where $\hat{\theta} = \operatorname{argmin}_{\theta \in \mathbb{R}} \|\hat{MP}_i(s)' - \theta \hat{MP}_{i+1}(t)'\|$. The smaller the difference, the more similar the two scenarios. If a vehicle is in $Scenario_i(s)$ with vehicle parameter factor VP_i , and $Scenario_{i+1}(t)$ is the most similar scenario to $Scenario_i(s)$ on P_{i+1} , then the vehicle parameter factor in $Scenario_{i+1}(t)$ is

$$\begin{aligned} VP_{i+1} = & \operatorname{argmin}_{\theta \in \mathbb{R}} \|\hat{EAIR}_i(s)' + VP_i \times \hat{MP}_i(s)' \\ & - \hat{EAIR}_{i+1}(t)' - \theta \times \hat{MP}_{i+1}(t)'\|. \end{aligned} \quad (9)$$

Since a TUP may have more than one decomposition, we will use the coarsest decomposition in which the number of shared road segments between adjacent TAP_β s is the largest in all decompositions. More shared road segments provide more evidence for determining the difference between scenarios on adjacent TAP_β s. After determining which scenario a vehicle is in on the first TAP_β in a path decomposition and the corresponding vehicle parameter factor (VP), the scenario and the corresponding VP of the latter TAP_β s are determined according to the method of joining adjacent TAP_β s introduced above. Given the energy consumption on each TAP_β , the union of them will be the cost for the whole path, while the cost of each road segment shared by several TAP_β s is the mean of the cost of these TAP_β on the segments. For example, as shown in Figure 4, a $TUP = [r_1, r_2, r_3, r_4]$ has a path decomposition $[TAP_\beta 1, TAP_\beta 2]$, each of which has 2 scenarios. Suppose a vehicle is in Scenario 1 on $TAP_\beta 1$ with a VP of 4. From the calculation we get Scenario 4 is the most similar to Scenario 1 among the scenarios on $TAP_\beta 2$. According to Equation 9, the vehicle parameter factor in Scenario 4 is 10. According to Equation 2, the energy consumption of the vehicle on the two TAP_β s is $[7, 13, 10]$ and $[13, 10, 31]$ respectively, and on TUP , it is $[7, 13, 10, 31]$.

Figure 4: An example for the TAP_β -union model on a TUP .

4.2 Adaptive Dynamic Programming Path Selection Algorithm

The general framework for a path selection algorithm is shown in Algorithm 1. Given a road network, an origin and a destination, as well as a model to estimate a path's cost, the algorithm generates a path satisfying certain criteria. The main steps of the algorithm are as follows. The list of candidate paths CP is initialized in Line 1, typically using the paths consisting of one road segment from the origin. Then in each iteration (Lines 2-8), the most promising path in CP is taken out and extended by adding one road segment to its end, and the result path is added to CP . The iteration ends when the stop criterion is met. The related work implements the steps in this algorithm from different perspectives. For example, in Dijkstra's algorithm the stop criterion is a path found between the origin and the destination, while the candidate paths are evaluated by their cost and the path with the smallest cost is the most promising one. In our paper, we propose a trajectory-aware filter for extending candidate paths, a dynamic programming method for estimating the cost of an extended path based on that of the original path, and an early-stop strategy for the stop criterion.

Algorithm 1 General algorithm framework

Require:

- G : A road network;
- o : The origin;
- d : The destination;
- $model$: A model to estimate a path's cost.

Ensure: The path between o and d satisfying the criteria.

- 1: Candidate paths $CP \leftarrow$ initialization;
- 2: **while** Stop criterion is false **do**
- 3: $p \leftarrow$ the most promising path in CP ;
- 4: **for all** Extensions p' s of p **do**
- 5: Compute the cost of p' ;
- 6: Add p' to CP ;
- 7: **end for**
- 8: **end while**

4.2.1 Trajectory-aware filter for path extension. Since we intend to find a TAP_β or a TUP , when extending a candidate path to new paths by adding one road segment to its end, a filter can be applied so that not all road segments connecting to the path's last node are added. The filter is based on the following lemma.

LEMMA 4.2. *The road segments that can be added to the end of a path to obtain a TAP_β or a TUP should satisfy the condition of there being at least β trajectories along it that are also along the last two road segments of the original path.*

PROOF. Suppose that there are less than β trajectories along a path composed of the last two road segments of the original path and the added road segment. The last two road segments of the original path and the added road segment cannot be in the same TAP_β , because the definition of a TAP_β requires the number of trajectories along the path be at least β . So the result path is neither a TAP_β nor a TUP which is a union of TAP_β s. $\square$

4.2.2 *A dynamic programming method for estimating a path's energy consumption.* The idea of dynamic programming is to divide the problem into subproblems and avoid repeated computation for the subproblems. We adopt this idea when computing the energy consumption of a result path obtained from an extension. By reusing the path decomposition of the original path and modifying the part concerning the added road segment to get the decomposition of the result path, we avoid the computation of searching for the path decomposition of the result path and estimating the energy consumption on each subpath from raw data. We treat the decomposition of a TAP_β as a sequence which only contains the path itself, and find Lemma 4.3.

LEMMA 4.3. *The coarsest decomposition of the result path should contain all but the last subpath in the coarsest decomposition of the original path.*

PROOF. Suppose that P_i is a subpath, but not the last one, in the original path's coarsest decomposition, and that it is not in the result path's coarsest decomposition. Since P_i is a subpath of the original path, and the original path is a subpath of the result path, P_i is a subpath of the result path, which, together with the fact that P_i is not in the result path's coarsest decomposition, indicate that there is a subpath P'_j in the result path's coarsest decomposition such that P_i is a subpath of P'_j and $P_i \neq P'_j$. Because P_i is in the original path's coarsest decomposition, P'_j is not a subpath of the original path. The only road segment in the result path but not in the original path is the added one, so the added road segment is in P'_j . Let P_j be the path obtained by excluding the added road segment from P'_j . P_j is a subpath of the original path. Because P_i is a subpath of P'_j and does not contain the added road segment, P_i is a subpath of P_j . Because both P_i and P_j are in the original path's coarsest decomposition, $P_i = P_j$. P_i must contain the last road segment of the original path, otherwise the union of P_i and the added road segment is not connected, and P'_j is not a path. Let P_n be the last subpath in the original path's coarsest decomposition. Because $P_i \neq P_n$, the first road segment of P_i appears earlier than that of P_n in the original path, and both P_i and P_n end at the last road segment of the original path. That is, P_n is a subpath of P_i , which is a contradiction. $\square$

Therefore, to extend a path we only need to modify the last subpath in its coarsest decomposition, the procedure for which is shown in Algorithm 2. Suppose that we need to obtain the union of a path P composed of $[r_1, \dots, r_n]$ and a road segment r_{n+1} as a new

Figure 5: Extend a path by adding r_{n+1} to its end.

path P' (shown in Figure 5). We denote the last TAP_β in P 's coarsest decomposition as $oldPath$, which consists of $[r_{n-k}, \dots, r_n]$, and get $newPath = oldPath \cup r_{n+1}$. If $newPath$ is a TAP_β , P' 's coarsest decomposition is obtained by replacing $oldPath$ with $newPath$ in P 's coarsest decomposition. If not, we recursively remove the first road segment from $newPath$ (i.e., $r_{n-k}, r_{n-k+1}, \dots$), and add the longest TAP_β composed of the remained road segments to P 's coarsest decomposition to obtain that of P' .

Algorithm 2 Extend a path

Require:

- P : A path;
- r : A road segment connected to P at P 's last node;
- T : A set of trajectories.

Ensure:

 The path composed of the union of P and r .

```

1:  $oldPD \leftarrow P$ 's coarsest decomposition;
2:  $oldPath \leftarrow$  the last  $TAP_\beta$  in  $oldPD$ ;
3:  $newPath \leftarrow$  the union of  $oldPath$  and  $r$ ;
4: if  $newPath$  is a  $TAP_\beta$  then
5:    $newPD \leftarrow$  replace  $oldPath$  with  $newPath$  in  $oldPD$ ;
6:   return  $P' \leftarrow$  union of  $newPD$ ;
7: else
8:   while Number of road segments in  $newPath > 3$  do
9:      $newPath \leftarrow newPath[1 : ]$ ;
10:    if  $newPath$  is a  $TAP_\beta$  then
11:       $newPD \leftarrow oldPD.add(newPath)$ ;
12:      return  $P' \leftarrow$  union of  $newPD$ ;
13:    end if
14:  end while
15: end if
```

4.2.3 *Early-stop strategy.* The early-stop strategy comes into play after the first path between the origin and the destination is found. This accords with our way of searching for candidate paths so that the algorithm can stop before exploring all TAP_β s and TUPs. We use energy consumption as the metric to evaluate the candidate paths and choose the one with smallest energy consumption as the currently most promising path. Since we always extend from the path with the currently known smallest energy consumption, the path we found between the origin and the destination is the currently known most energy-efficient path. However, the possibly negative energy consumption caused by regenerative braking means that the paths extended from a path may cost less energy than the original path. To deal with this issue, before the path selection, we identify the road segments on which there exist trajectories whose energy consumption is negative. The sum of the smallest negative energy consumption values on these road segments indicates the largest amount of energy that may be gained by traveling along these road segments. A value we call regenerative

Figure 6: Experiment design.

energy bound. Therefore, if the sum of the energy consumption of a candidate path and the regenerative energy bound is greater than the energy consumption of the current path between the origin and the destination, it is impossible to extend the candidate path to get a cheaper path between the origin and the destination, so the extension of the candidate path is terminated.

5 EXPERIMENTAL EVALUATION AND CASE STUDIES

We conducted two experiments to evaluate the estimation accuracy of the Physics-guided Energy Consumption (PEC) model and the energy saving resulting from using the adaptive dynamic programming path selection (Physics-guided) algorithm, which answer four questions: (1) Were the estimates of the PEC model more accurate than those of related work? (2) What was the impact of path length on estimation accuracy? (3) What was the impact of β on estimation accuracy? (4) Was the energy consumption of paths selected by the Physics-guided algorithm less than that of the related work? We also conducted a case study on path selection to show our methods can substantially reduce energy consumption.

5.1 Experiments

The experiment design is shown in Figure 6. The data we used included a road network from OpenStreetMap, and trajectory data collected from a vehicle in Fort Worth, TX from 1/1/2017 to 3/1/2018, which contained 10759 trajectories.

5.1.1 Energy consumption estimation accuracy. In the first experiment, we compared the PEC model with two methods which were commonly used as the ground truth in the state-of-the-art related work, one that uses the average of the energy consumption recorded by historical trajectories, and another that uses the distribution of historical data [24]. The experiment was conducted iteratively. In each run, given an input β , we randomly chose a TAP_β composed of an input number of road segments, and estimated the energy consumption on it through the three methods. Then, we represented the path as a TUP by excluding the trajectories along the entire path, and estimated the energy consumption using the

Figure 7: Estimation accuracy with varying path length and β .

PEC model. The energy consumption of each trajectory along the entire path was used as the ground truth, and the absolute value of its difference with each estimation was called the estimation error. The relative estimation error was defined as the ratio between the estimation error and the ground truth. Since the histogram method represents the estimation as a distribution, its estimation error was defined as the weighted average of the absolute value of the difference between the value of each bin in the histogram and the ground truth. The average relative estimation error was used as the metric to evaluate the estimation accuracy of each method. The smaller the metric was, the more accurate the estimation.

Figure 7 shows the average relative estimation error of estimates provided by the histogram method, the mean method, and the PEC model on TAP_β s, as well as the PEC model on TUP s, given different number of road segments and β . Generally, the estimation error of all methods decreased with increasing number of road segments. A potential cause of this was that the impact of random factors not modeled by these methods declined as the path got longer. For example, the impact of stopping for a red traffic light is larger

Figure 8: Estimation accuracy of the PEC model with varying β on TUPs with 45 road segments.

on a short trip than on a long trip. The number of road segments showed the length of a path indirectly. On paths composed of a certain number of road segments, even though greater β meant more trajectories along a path available for estimation, the estimation accuracy of the methods stayed almost unchanged with varying β . Given the number of road segments and β , the estimation error of the PEC model on TAP β s was clearly lower than those of the mean and histogram methods. For example, on paths composed of 40 road segments with $\beta = 20$, the relative estimation error of the PEC model was 11.40%, while those of the mean and histogram methods were 17.25% and 23.08% respectively. In addition, while the estimation accuracy of the PEC model on TUPs was similar to that of the mean method, the PEC model required less data. On paths composed of 45 road segments, neither the histogram nor the mean methods were able to provide estimation, while the estimation error of the PEC model was shown in Figure 8, which was around 16%. Therefore, the estimate of the PEC model was more accurate than the histogram and the mean methods that have been used as the ground truth in related work.

5.1.2 Energy savings. In the second experiment, we compared the energy consumption of the paths selected by the proposed Physics-guided method with that of vehicle drivers' choices, and that of the fastest paths selected by Itinero, an open-source routing library using OpenStreetMap. This experiment was also conducted iteratively. In each run, we randomly chose two road intersections which were on a random trajectory as the origin and the destination. This trajectory recorded a vehicle driver's choice. The proposed Physics-guided algorithm was executed to find the most energy-efficient path, while the Itinero library provided the fastest path. The energy consumption of the three paths was estimated through the PEC model and used as the metric to evaluate whether the paths were energy-efficient.

The results are shown in Figure 9. Paths between 159 randomly-chosen origin and destination (OD) pairs were selected (75 origins and 52 destinations). Each pixel in Figure 9a, 9b, or 9c represents the energy consumption of fastest, driver-chosen, or energy-efficient paths between OD pairs. The lighter the pixel the less energy is consumed along the path, and white pixels represent not chosen pairs. Obviously, the pixels in Figure 9c are much lighter than those in Figure 9a and 9b. Indeed, the experiment showed that the energy-efficient paths selected by the Physics-guided method consumed 58% less energy than the driver-chosen path and 42% less energy than the fastest paths.

Figure 9: Energy consumption of paths selected by candidate methods.

Figure 10: A path selected by the proposed method is more energy-efficient than that from Google Maps.

5.2 Case study

We conducted a case study to compare the paths selected by the proposed method and Google Maps. The data we used in the proposed method was the same as that in the experiments. The Physics-guided method selected a path 1.5 miles long with an estimated time cost of 5 minutes (Figure 10a). Google Maps chose a path 1.6 miles long but with a smaller estimated time cost of 4 minutes (Figure 10b). Nevertheless, the path selected by the proposed method had a lower estimated energy cost (1.11 KWh) than the path from Google Maps (1.60 KWh). A potential cause of this difference was that the part of the route affected by heavy traffic (shown in orange) was longer on the path selected by the proposed method than on the Google Maps path, but its impact on energy consumption was not as large as on time cost. Therefore, the proposed method was able to select the more energy-efficient path than the currently widely-used method.

6 CONCLUSION AND FUTURE WORK

The energy-efficient path selection (EPS) problem aims to find the path with least energy consumption between a given origin and destination. We introduced a physics-guided energy consumption model, which estimates energy consumption as a function of the physics parameters of a vehicle, to solve the main challenge of

the EPS problem, adopting energy consumption as the cost metric of a path. We also proposed an adaptive dynamic programming algorithm for path selection using the estimates from the PEC model. The Experiments and case study showed that the PEC model estimates were more similar to real trajectory data than two state-of-the-art models, and that the paths found by the proposed method was more energy-efficient than the currently-used paths and the fastest paths found by Itinero and Google Maps.

In this study, we only focused on estimating the energy consumption of paths with a certain number of trajectories along them. Our future research will focus on method of estimating energy consumption for paths along which there are few or no trajectories.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grants No. 1541876, 1029711, IIS-1320580, IIS-0940818, and IIS-1218168, the USDOD under Grants No. HM1582-08-1-0017 and HM0210-13-1-0005, the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy under Award No. DE-AR0000795, the NIH under Grant No. UL1 TR002494, KL2 TR002492, and TL1 TR002493, the USDA under Grant No. 2017-51181-27222, and the OVPF Infrastructure Investment Initiative and Minnesota Supercomputing Institute (MSI) at the University of Minnesota. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. The authors would like to thank Kim Koffolt and the University of Minnesota Spatial Computing Research Group for their comments.

REFERENCES

- [1] Sabeur Aridhi, Philippe Lacomme, Libo Ren, and Benjamin Vincent. 2015. A MapReduce-based approach for shortest path problem in large-scale networks. *Engineering Applications of Artificial Intelligence* 41 (May 2015), 151–165.
- [2] Andreas Artmeier, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher. 2010. The Shortest Path Problem Revisited: Optimal Routing for Electric Vehicles. In *KI 2010: Advances in Artificial Intelligence (Lecture Notes in Computer Science)*. Springer, Berlin, Heidelberg, 309–316.
- [3] Jrgen Bang-Jensen and Gregory Z. Gutin. 2008. *Digraphs: Theory, Algorithms and Applications*. Springer.
- [4] A. Cappiello, I. Chabini, E. K. Nam, A. Lue, and M. Abou Zeid. 2002. A statistical model of vehicle emissions and fuel consumption. In *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*. 801–809.
- [5] B. Y. Chen, W. H. K. Lam, Q. Li, A. Sumalee, and K. Yan. 2013. Shortest Path Finding Problem in Stochastic Time-Dependent Road Networks With Stochastic First-In-First-Out Property. *IEEE Transactions on Intelligent Transportation Systems* 14, 4 (Dec. 2013), 1907–1917.
- [6] Daniel Delling, Andrew V. Goldberg, Andreas Nowatzyk, and Renato F. Werneck. 2013. PHAST: Hardware-accelerated shortest path trees. *J. Parallel and Distrib. Comput.* 73, 7 (July 2013), 940–952.
- [7] Yong Deng, Yuxin Chen, Yajuan Zhang, and Sankaran Mahadevan. 2012. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing* 12, 3 (March 2012), 1231–1237.
- [8] Department of Energy. 2016. Energy Department Announces \$58 Million to Advance Fuel-Efficient Vehicle Technologies. <https://www.energy.gov/articles/energy-department-announces-58-million-advance-fuel-efficient-vehicle-technologies>
- [9] E. W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1, 1 (Dec. 1959), 269–271.
- [10] Daniel Duque, Leonardo Lozano, and Andrs L. Medaglia. 2015. An exact method for the biobjective shortest path problem for large-scale road networks. *European Journal of Operational Research* 242, 3 (May 2015), 788–797.
- [11] Jochen Eisner, Stefan Funke, and Sabine Storandt. 2011. Optimal Route Planning for Electric Vehicles in Large Networks. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI'11)*. AAAI Press, San Francisco, California, 1108–1113. <http://dl.acm.org/citation.cfm?id=2900423.2900599>
- [12] Yuan Gao. 2011. Shortest path problem with uncertain arc lengths. *Computers & Mathematics with Applications* 62, 6 (Sept. 2011), 2591–2600.
- [13] V. M. V. Gunturi, S. Shekhar, and K. Yang. 2015. A Critical-Time-Point Approach to All-Departure-Time Lagrangian Shortest Paths. *IEEE Transactions on Knowledge and Data Engineering* 27, 10 (Oct. 2015), 2591–2603.
- [14] Greg Hamerly and Charles Elkan. 2002. Alternatives to the K-means Algorithm That Find Better Clusterings. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM '02)*. ACM, New York, NY, USA, 600–607.
- [15] P. E. Hart, N. J. Nilsson, and B. Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4, 2 (July 1968), 100–107.
- [16] Tsuyoshi Id and Masashi Sugiyama. 2011. Trajectory Regression on Road Networks.. In *AAAI*.
- [17] Itinero. 2018. Route planning for .NET. <http://www.itinero.tech/>
- [18] Daniele Quercia, Rossano Schifanella, and Luca Maria Aiello. 2014. The Shortest Path to Happiness: Recommending Beautiful, Quiet, and Happy Routes in the City. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media (HT '14)*. ACM, New York, NY, USA, 116–125.
- [19] Martin Sachenbacher, Martin Leucker, Andreas Artmeier, and Julian Haselmayr. 2011. Efficient Energy-optimal Routing for Electric Vehicles. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI'11)*. AAAI Press, San Francisco, California, 1402–1407. <http://dl.acm.org/citation.cfm?id=2900423.2900645>
- [20] Christian Sommer. 2014. Shortest-path Queries in Static Networks. *ACM Comput. Surv.* 46, 4 (March 2014), 45:1–45:31.
- [21] U.S. Energy Information Administration. 2017. *International Energy Outlook 2017*. Technical Report DOE/EIA-0484(2017). Washington, DC.
- [22] U.S. Energy Information Administration. 2017. Total U.S. energy expenditures in 2015 were the lowest in more than a decade. <https://www.eia.gov/todayinenergy/detail.php?id=32432>
- [23] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel Time Estimation of a Path Using Sparse Trajectories. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, New York, NY, USA, 25–34.
- [24] Bin Yang, Jian Dai, Chenjuan Guo, Christian S. Jensen, and Jilin Hu. 2018. PACE: a PATH-Centric paradigm for stochastic path finding. *The VLDB Journal* 27, 2 (April 2018), 153–178.
- [25] B. Yang, M. Kaul, and C. S. Jensen. 2014. Using Incomplete Information for Complete Weight Annotation of Road Networks. *IEEE Transactions on Knowledge and Data Engineering* 26, 5 (May 2014), 1267–1279.
- [26] Jiangchuan Zheng and Lionel M. Ni. 2013. Time-Dependent Trajectory Regression on Road Networks via Multi-Task Learning.. In *AAAI*.