

Physics-guided Energy-efficient Path Selection Using On-board Diagnostics Data

YAN LI, PRATIK KOTWAL, PENGYUE WANG, YIQUN XIE, SHASHI SHEKHAR, and WILLIAM NORTHPROP, University of Minnesota - Twin Cities

Given a spatial graph, an origin and a destination, and on-board diagnostics (OBD) data, the energy-efficient path selection problem aims to find the path with the least expected energy consumption (EEC). Two main objectives of smart cities are sustainability and prosperity, both of which benefit from reducing the energy consumption of transportation. The challenges of the problem include the dependence of EEC on the physical parameters of vehicles, the autocorrelation of the EEC on segments of paths, the high computational cost of EEC estimation, and potential negative EEC. However, the current cost estimation models for the path selection problem do not consider vehicles' physical parameters. Moreover, the current path selection algorithms follow the "path + edge" pattern when exploring candidate paths, resulting in redundant computation. Our preliminary work introduced a physics-guided energy consumption model and proposed a maximal-frequented-path-graph shortest-path algorithm using the model. In this work, we propose an informed algorithm using an admissible heuristic and propose an algorithm to handle negative EEC. We analyze the proposed algorithms theoretically and evaluate the proposed algorithms via experiments with real-world and synthetic data. We also conduct two case studies using real-world data and a road test to validate the proposed method.

CCS Concepts: • **Information systems** → **Geographic information systems**; **Data mining**;

Additional Key Words and Phrases: On-board diagnostics data, energy-efficient path, eco-routing, physics-guided, shortest path

ACM Reference format:

Yan Li, Pratik Kotwal, Pengyue Wang, Yiqun Xie, Shashi Shekhar, and William Northrop. 2020. Physics-guided Energy-efficient Path Selection Using On-board Diagnostics Data. *ACM/IMS Trans. Data Sci.* 1, 3, Article 22 (September 2020), 28 pages.

<https://doi.org/10.1145/3406596>

This material is based upon work supported by the National Science Foundation under Grant Nos. 1737633, 1901099, 1541876, 1029711, IIS-1320580, IIS-0940818, and IIS-1218168; the USDOD under Grant Nos. 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 Nos. UL1 TR002494, KL2 TR002492, and TL1 TR002493; the USDA under Grant No. 2017-51181-27222; and the OVPR Infrastructure Investment Initiative and Minnesota Supercomputing Institute (MSI) at the University of Minnesota.

Authors' addresses: Y. Li, P. Kotwal, Y. Xie, and S. Shekhar, Dept. of Computer Science & Eng, University of Minnesota - Twin Cities, 200 Union Street SE, Minneapolis, MN, 55455; emails: {lix4266, kotwa007, xiexx347, shekhar}@umn.edu; P. Wang and W. Northrop, Dept. of Mechanical Engineering, University of Minnesota - Twin Cities, Mechanical Engineering Building, 111 Church St SE, Minneapolis, Minnesota, 55455; emails: {wang6609, wnorthro}@umn.edu.

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.

© 2020 Association for Computing Machinery.

2577-3224/2020/09-ART22 \$15.00

<https://doi.org/10.1145/3406596>

1 INTRODUCTION

Given a spatial graph, two nodes in the graph as the origin and destination, and historical on-board diagnostics (OBD) data, the energy-efficient path selection (EPS) problem aims to find the most energy-efficient path between the origin and the destination. Figure 1 illustrates a sample input of the EPS problem consisting of a spatial graph with 11 nodes (i.e., $n1, n2, \dots, n11$) and 12 edges (i.e., $e1, e2, \dots, e12$), 6 traces of OBD data (i.e., $t1, t2, \dots, t6$), and 2 nodes $n1$ and $n5$ as the origin and the destination. Figure 2 shows 6 traces in the OBD data in Figure 1. Every trace is in the form of an ordered sequence of records, and each record is composed of an edge and the status of the vehicle on the edge. Energy consumption and average speed are two examples of the status attributes. Suppose that the expected energy consumption of the vehicle on path $[e1, e5, e8, e11, e12, e10, e7, e4]$ is the lowest among those on all possible paths linking $n1$ and $n5$; the path is the energy-efficient path for this example.

Monitoring and managing traffic and transportation systems using the OBD data collected from telematics devices on connected vehicles is an important component of a smart city. This article describes our work on leveraging OBD data to provide energy-efficient routing suggestions. Two main objectives of the vision for smart cities are sustainability and prosperity, both of which benefit from the ability to estimate and reduce the energy consumption for transportation. Reports show that car exhaust emissions contribute significantly to air pollution and anthropogenic climate forcing [12], and energy consumption for transportation in the United States cost more than \$507B in 2015 [30]. Despite the efforts to reduce energy consumption for transportation, such as electric car research supported by the U.S. Department of Energy [11], the expected energy use continues to climb [29].

Geo-referenced OBD data facilitate accurate travel cost estimation and novel path selection algorithms using the estimation. Previous research on energy-efficient path selection has demonstrated that the potential energy saving is about 10% by taking energy-efficient routes instead of fastest routes [1, 21]. A McKinsey Digital report also estimates that personal geo-referenced data could help save consumers about \$600B by 2020, by providing routes to vehicles that avoid traffic congestion through next-generation routing algorithms [22]. Equation (1) is a simplified powertrain energy consumption model commonly used in mechanical engineering, where the meaning of the symbols are in Table 1 [7]. Briefly, the energy consumption of a vehicle is determined by the vehicle's motion properties (i.e., t, a, v , and v_h) as well as its physical parameters (i.e., m, A, c_{air} , and η). Between two places there are often multiple possible paths, and different paths have different spatiotemporal features such as speed limit, traffic, and road conditions, which affect the motion properties of a vehicle and its energy consumption in turn. For example, Figure 3(a) shows that between two places there are a fast but long path through highways and two slow but short paths through local roads. The high speed on highways reduces the time cost, but may make the thermal energy due to air resistance, which is represented by the term with v^3 in Equation (1), dominate in energy consumption. The road test we conducted in Cincinnati, Ohio (detailed in Section 7), indicated that the expected energy consumption on the energy-efficient path composed of local roads is about 38% lower than that on the fastest path composed of highways. In addition, the existence of up/down hill roads (e.g., in San Francisco (Figure 3(b))) also affects vehicles' energy consumption. Therefore, in this article, we propose a method to leverage OBD data for energy-efficient path selection.

$$W = \frac{1}{\eta} \left[\int (mav)dt + \int (mc_{rr}gv)dt + \int \left(\frac{A}{2} c_{air} \rho v^3 \right) dt + \int mgv_h dt \right]. \quad (1)$$

The EPS problem, which is a variant of the shortest path selection problem, has two sub-tasks, namely, the prediction of expected energy consumption (EEC) of a path and the selection of the

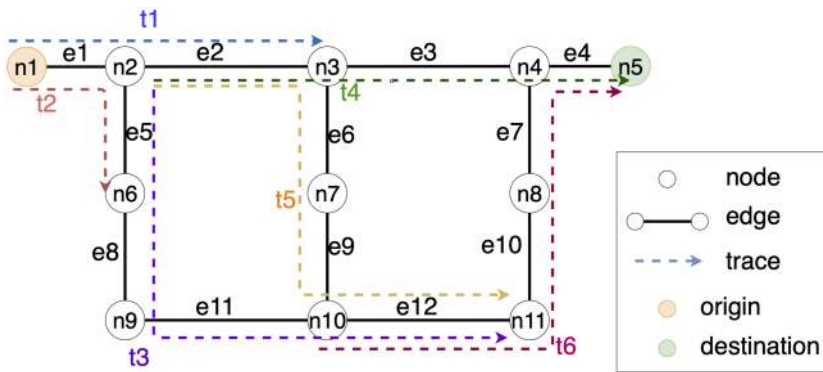


Fig. 1. A spatial graph with six traces of on-board diagnostics data.

t1			t2			t3		
edge	Energy consumption	Average speed	edge	Energy consumption	Average speed	edge	Energy consumption	Average speed
e1	2	9	e1	1	5	e5	3	8
e2	9	30	e5	1	6	e8	2	5
t4			t5			t6		
edge	Energy consumption	Average speed	edge	Energy consumption	Average speed	edge	Energy consumption	Average speed
e2	7	15	e2	9	40	e12	3	15
e3	9	35	e6	2	30	e10	2	10
e4	2	7	e9	2	20	e7	2	8
			e12	7	30	e4	2	4

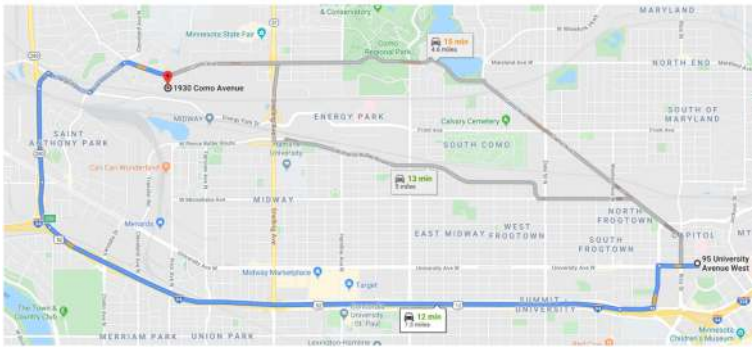
Fig. 2. Sample OBD data with six traces.

most energy-efficient path. The challenges of predicting EEC of a path are two-fold. The first challenge is the dependence of EEC on physical parameters of vehicles, which is different from the cost metrics (i.e., distance, time) for the shortest or fastest path selection problems. For example, a construction truck consumes more energy than a sedan when traveling along a path following the same velocity profile, even though the distance and the time cost are the same. Moreover, the autocorrelation of the energy consumption on different segments of a path prevents edge-centric travel cost estimation models from estimating accurately. In other words, the EEC of a path is a property of the entire path, but not the sum of the EEC of individual edges along the path. The selection of an energy-efficient path given an EEC estimation model also has two challenges: The first is the high computational cost of EEC estimation, which is needed for every candidate path in currently existing path selection algorithms. The second challenge is that the EEC on a path may be negative because of regenerative braking, which makes popular path selection algorithms like Dijkstra’s algorithm inapplicable, since they assume that travel cost is non-negative.

Most of the methods for predicting EEC are composed of two steps [6, 21, 23, 32, 34]: (1) predict or obtain the velocity profile along the objective path, and (2) estimate EEC using the velocity profile via a vehicle simulation model. However, accurate velocity profile prediction is challenging,

Table 1. Physics Model Symbols

Symbol	Physical Interpretation
W	work (energy consumption)
η	vehicle's powertrain system efficiency
m	vehicle's mass
A	vehicle's front surface area
c_{air}	air resistance coefficient
c_{rr}	rolling resistance coefficient
a	acceleration
v	velocity
v_h	vertical velocity
t	time
g	gravity acceleration
ρ	air density



(a) Highway vs. local roads.



(b) Up/down hill roads.

Fig. 3. There are often multiple paths between two places.

since a large number of factors may affect it, some of which (e.g., the schedule of vehicles and pedestrians) are unpredictable in real-world road systems. A novel EEC estimation model using OBD data without velocity profile is needed. In addition, the commonly used path selection algorithms are based on Dijkstra's or the Bellman-Ford algorithm [13, 19] and apply a "path + edge" pattern to explore candidate paths. These algorithms evaluate the travel cost of a candidate path once the path is explored. However, the estimation of travel cost is computationally expensive, and some estimation is redundant in cases where the OBD data on an edge are a subset of the OBD data on a path containing the edge. Therefore, a novel path selection algorithm is needed that adopts a "path + another path" pattern when exploring candidate paths.

Our preliminary work [24, 25] proposed a physics-guided energy consumption (PEC) model for the prediction of EEC and a maximal-frequented-path-graph shortest-path (MFPG-SP) algorithm using the PEC model for the path selection.

This article extends our previous work by proposing two algorithms for the sub-task of path selection given an EEC estimation model, analyzing their correctness and completeness, and validating the algorithms through experiments and two case studies.

Our new contributions in the article are as follows:

- (1) We propose an A^* -like admissible heuristic and an informed maximal-frequented-path-graph shortest-path (IN-MFPG-SP) algorithm.

- (2) We propose a maximal-frequented-path-graph label-correcting (MFPG-LC) algorithm that can handle negative energy consumption in a maximal frequented path graph.
- (3) We analyze the proposed algorithms for correctness, completeness, and computational time complexity.
- (4) We evaluate the proposed algorithms via controlled experiments with real-world and synthetic data.
- (5) We conduct two case studies using the OBD data collected from three UPS trucks in 18 months to evaluate the potential energy saving of adopting the proposed method compared to using the path recorded in the data, and to show that the proposed method can suggest paths that are more energy-efficient than the paths suggested by the commonly used path selection tools.
- (6) We conduct one road test using one UPS truck to validate that the proposed method can suggest paths that are more energy-efficient than the paths suggested by the commonly used path selection tools.

The scope of the article: The goal of this article is to introduce a method of suggesting more energy-efficient paths according to our estimation, compared with the previously used paths and the fastest paths suggested by the currently available tools. Current initiatives such as USDOE ARPA-E NEXTCAR [31] and EU optiTruck [16] consider path selection to be an important approach to reduce energy consumption by vehicles. Previous research has also shown that path selection can save up to 10% of energy [1, 21], and it can complement other energy-saving methods related to the refinements of vehicle characteristics, driving behaviors, road surface, traffic flow, and so on. Identifying the main factors to energy consumption is out of the scope of this article.

The rest of this article is organized as follows: In Section 2, we introduce the basic concepts and formally define the problem of energy-efficient path selection. This is followed by a review of the related work and our preliminary work and their limitations in Section 3. Section 4 discusses our new algorithms for solving the problem, and an experimental evaluation is presented in Section 5. The results of two case studies using the proposed method are shown in Section 6. The results of a real-world road test are shown in Section 7. Section 8 concludes the article and presents our future work.

2 BASIC CONCEPTS AND PROBLEM DEFINITION

In this section, we introduce the basic concepts in this study, based on which the energy-efficient path selection problem is formally defined.

2.1 Basic Concepts

A **spatial graph** $G_s = (N_s, E_s)$ consists of a set of **spatial-nodes** N_s and a set of **spatial-edges** E_s , where each element $n \in N_s$ is a geo-referenced point, and each element $e = (n_i, n_j) \in E_s$ is an edge that joins spatial-node n_i and spatial-node n_j . Figure 1 shows an example of a spatial graph where circles represent nodes (e.g., n_1, n_2) and lines represent edges (e.g., e_1, e_2). A road system is an example of a spatial graph where nodes are road intersections and edges are road segments.

A **path** in a graph is a sequence of edges linking an ordered sequence of nodes. The first and the last nodes are the origin and the destination of the path, respectively. Given two paths ϕ_1 and ϕ_2 , ϕ_1 is a **sub-path** of ϕ_2 if the edges of ϕ_1 are all along ϕ_2 . In Figure 1, path $[e_1, e_2]$ is a sub-path of path $[e_1, e_2, e_3]$. The union (\cup) of two paths $\phi_1 \cup \phi_2$ at a node shared by them is composed of the edges of ϕ_1 before the node and those of ϕ_2 after the node. For example, in Figure 1, $[e_2, e_3, e_4] \cup [e_3, e_7]$ at n_4 is $[e_2, e_3, e_7]$.

A **trace** in an on-board diagnostics (OBD) dataset is a map-matched multi-attribute trajectory of a vehicle that consists of a list of records. Each record is composed of an edge and a set of vehicle status on the edge, such as the total energy consumption, the average speed, and the state of charge. Figure 1 shows six traces of OBD data as dashed arrows whose detailed records are in Figure 2. For example, trace t_4 has three records, indicating that the vehicle travels along edges e_2, e_3, e_4 and its energy consumption and average speed on these edges are 7,9,2 and 15,35,7, respectively.

A **frequented path** (FP) is a path along which there are at least a certain number of traces of OBD data in the same direction. A **union of frequented paths** (UFP) is a path composed of a union of two or more FPs such that it is not the sub-path of any other FP. The FPs are paths where we have most historical OBD data, and UFPs are formed by FPs, so we assume the energy consumption estimation on them would be more accurate compared to that on other paths. In addition, FPs and UFPs also imply people's traveling preference. If we set the minimum number of traces of OBD data along an FP as 1 in Figure 1, path $[e_1, e_2]$ is an FP along which there is a trace t_1 . A sample of UFP is $[e_1, e_2, e_3, e_4]$, which is formed by the union of two FPs $[e_1, e_2]$ and $[e_2, e_3, e_4]$ at n_3 .

Expected energy consumption (EEC) of a path in a spatial graph is the amount of energy expected to be consumed by a vehicle traveling along the path. For the sake of simplicity, in the examples in the article, the EEC of an FP is calculated as the average energy consumption of the traces along it. For example, in Figure 1 there is only one trace t_1 on the entire path $[e_1, e_2]$, so the EEC of each edge along path $[e_1, e_2]$ is $[2, 9]$ according to t_1 , and the EEC of the path is $2 + 9 = 11$. The EEC of a UFP is the summation of the EEC of the FP forming it, and the EEC of the overlapping edges of multiple FPs is the average EEC of the FPs on the edges. For example, in Figure 1 the UFP $[e_1, e_2, e_3, e_4]$ is formed by two FPs $[e_1, e_2]$ and $[e_2, e_3, e_4]$, and the two FPs overlap on e_2 . Since the EEC of the edges on $[e_1, e_2]$ is $[2, 9]$ and that on $[e_2, e_3, e_4]$ is $[7, 9, 2]$, the EEC of the edges on $[e_1, e_2, e_3, e_4]$ is $[2, 8, 9, 2]$, and that of the entire path is $2 + 8 + 9 + 2 = 21$. This method of estimating EEC is path-centric, because only traces along the entire path are considered when estimating the EEC of an FP. For example, when estimating the EEC of path $[e_1, e_2]$, the path-centric methods only use t_1 , while the traditional edge-centric methods will first estimate the EEC on e_1 according to t_1 and t_2 as well as the EEC on e_2 according to t_1, t_4 , and t_5 . However, part of the energy consumption of t_2, t_4 , and t_5 is for factors other than traveling on the path, such as the right turn of t_2 at node n_2 and the start of t_4 on edge e_2 .

An **energy-efficient path** between two locations is the path with the least EEC according to our estimation using historical OBD data among all possible paths between the locations. For example, the energy-efficient path between n_1 and n_5 is the path $[e_1, e_5, e_8, e_{11}, e_{12}, e_{10}, e_7, e_4]$ whose EEC is 18, since the other two possible paths $[e_1, e_2, e_3, e_4]$ and $[e_1, e_2, e_6, e_9, e_{12}, e_{10}, e_7, e_4]$ have a total cost of 21 and 25, respectively.

In our preliminary work [24], we defined two other key terms for the maximal-frequented-path-graph shortest-path algorithm.

A **maximal frequented path** (MFP) is an FP that is not the sub-path of any other FP. Since any sub-path of an FP is an FP, only estimating the EEC of MFPs eliminates the redundant computation of estimating the EEC of an FP and its sub-paths repeatedly.

A **maximal frequented path graph** (MFPG) of a spatial graph is a directed graph representing the MFPs and the relationship between each pair of them. Thus, the nodes of an MFPG (**MFP-nodes**) are MFPs. An edge of an MFPG (**MFP-edge**) exists between two MFP-nodes (ϕ_1 and ϕ_2) if $\phi_1 \cup \phi_2$ form a UFP at a spatial-node. Suppose that the minimum number of OBD traces along an FP is 1. Each OBD trace in Figure 1 defines an MFP. The spatial graph and the OBD data in Figure 1 can be transformed into an MFPG, as shown in Figure 4. Each square is an MFP-node representing an MFP in the spatial graph, while the directed arrows are the MFP-edges between

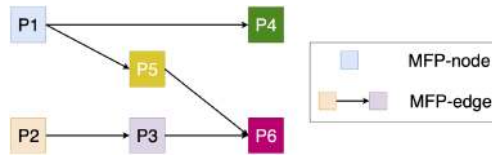


Fig. 4. The MFPG for the spatial graph and OBD data in Figure 1 (the color of each MFP matches the trace on it).

them. MFP-node P_i is the MFP defined by the trace t_i . For example, P_1 is the path $[e_1, e_2]$ where t_1 is along.

We say a path in a spatial graph is represented by a path in an MFPG if the path in the spatial graph is a sub-path of the UFP formed by the MFP-nodes along the path in the MFPG, and the origin and the destination of the path in the spatial graph are on the first and the last MFP-nodes, respectively. Since an MFP contains several spatial-nodes in it, multiple paths in a spatial graph could be represented by a path in an MFPG. For example, the MFPG-nodes in path $[P_5, P_6]$ in the MFPG form a UFP $[e_2, e_6, e_9, e_{12}, e_{10}, e_7, e_4]$ in the spatial graph, so UFPs such as $[e_2, e_6, e_9, e_{12}, e_{10}]$, $[e_2, e_6, e_9, e_{12}, e_{10}, e_7]$, and $[e_6, e_9, e_{12}, e_{10}]$ are represented by it. The cost of a path in an MFPG is defined as the cost of the UFP made up of all MFP-nodes in the path and excluding the spatial-edges before the origin. For example, let n_3 be the origin, the cost of path $[P_5, P_6]$ in the MFPG is the cost of UFP $[e_6, e_9, e_{12}, e_{10}, e_7, e_4]$, which is $[e_6, e_9, e_{12}] \cup [e_{12}, e_{10}, e_7, e_4]$, in the spatial graph. We prove that, given a spatial graph and its MFPG, every FP is a sub-path of at least one MFP, and every UFP can be formed by the union of at least one collection of MFPs. Therefore, we can model the FPs and UFPs in a spatial graph without losing any information via an MFPG.

2.2 Problem Definition

We formally define the energy-efficient path selection problem as follows:

Input:

- A spatial graph.
- Historical OBD data of vehicles in the graph.
- A minimum threshold for the number of OBD traces along an FP.
- Two spatial nodes o and d .

Output: An energy-efficient path between o and d .

Objective: Avoid a path that is energy-inefficient.

Constraints: The resulting path is an FP or a UFP.

An example of the problem we are solving in this article is in the following form: We are given the spatial graph shown in Figure 1, six traces on it with details shown in Figure 2, the minimum number of traces along an FP as 1, and two spatial nodes n_1 and n_5 as the origin and destination. The output of the problem would be the path $[e_1, e_5, e_8, e_{11}, e_{12}, e_{10}, e_7, e_4]$ with a total cost of 18. In this article, we only focus on finding energy-efficient paths among FPs and UFPs. Energy consumption estimation on paths without enough OBD data is outside the scope of this article.

3 RELATED WORK AND PRELIMINARY RESULTS

3.1 Related Work

Based on the basic spatial unit where travel cost is estimated, the existing path selection methods can be categorized into two groups, i.e., edge-centric and path-centric methods (left branch in Figure 5).

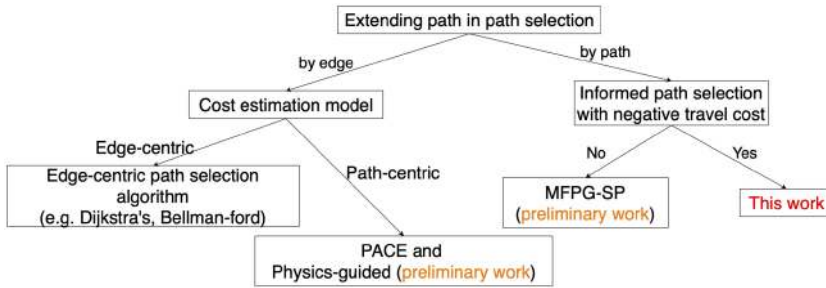


Fig. 5. A tree of related works.

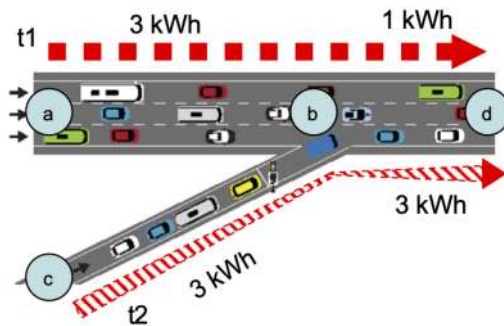


Fig. 6. Difference between edge-centric and path-centric view at a highway ramp.

Edge-centric methods assume the travel cost on individual edges is independent, and the travel cost of a path is the sum of the costs on the edges along the path. Dijkstra's [13] and the Bellman-Ford [19] algorithms are widely applied with an assumption that the cost of traveling on each edge is a constant. Other studies based on them have focused on accelerating computation [2, 5, 9, 28] or introducing new constraints [3, 15], new cost metrics [14, 27], and new cost representation [8, 10, 17, 18]. Most of the currently available energy-efficient path selection methods belong to this group [1, 21], which suggest path with the lowest energy consumption according to the estimated energy consumption on individual edges. However, all the edge-centric methods suffer from the fact they ignore the dependence between the costs of different parts along a path.

Rather than thinking of a path as a sequence of individual edges, path-centric methods treat it as a sequence of overlapping sub-paths [25, 33]. Since the basic unit to estimate the cost of a path is a sub-path, these path-centric methods maintain the dependence between the costs of different parts along a path. This is beneficial to energy consumption estimation. For example, Figure 6 shows a road intersection b on a highway from a to d , where there is an entrance ramp from c . There are two OBD traces ($t1$ and $t2$). $t1$ is along the highway, while $t2$ is from the entrance to the highway at b . The energy consumption of the traces on each edge is annotated. To estimate the expected energy consumption (EEC) of the path from a to b then to d , an example of edge-centric solutions is to sum up the average energy consumption on edges a to b and b to d individually. The average energy consumption on edge a to b is 3 kWh according to trace $t1$, and that on edge b to d is 2 kWh according to traces $t1$ and $t2$. Thus, the EEC of the path from a to b then to d is $3 + 2 = 5$ kWh according to the edge-centric solution. Instead, if applying a path-centric solution that only uses traces along the whole path (i.e., $t1$), we will get the result of 4 kWh. Intuitively, a part of the energy consumed by a vehicle after entering a highway is for acceleration, which

should not be included in the EEC of a vehicle traveling on a highway. If we use the traces that do not lie along the whole path (e.g., $t2$ in this case), we may mistakenly include energy consumption caused by factors not on the path (e.g., the traffic light at the entrance ramp). Therefore, our path selection algorithm uses a path-centric travel cost estimation model.

Despite their advantages of using path-centric cost estimation model, the PACE and Physics-guided methods in References [25, 33] require a lot of redundant computation. The general framework for a path selection algorithm is shown in Algorithm 1. Given a spatial graph, an origin, and a destination, as well as a travel cost estimation model, the algorithm generates a path satisfying certain criteria. The main steps of the algorithm are as follows: A set of candidate paths CP is initialized in Line 1, typically using the paths consisting of one edge from the origin. Then in each iteration (Lines 2–8), the most promising path in CP is extended, and the result path is added to CP . The iteration ends when the stop criterion is met. The related work implements these steps in different ways. For example, the Dijkstra algorithm’s stop criterion is that a path is found between the origin and the destination, while the most promising path in CP is the one with the smallest cost.

All the existing path selection algorithms explore the candidate paths in Line 4 following the pattern of “path + edge,” including the PACE and Physics-guided methods. In other words, a candidate path is generated by adding an edge to the end of an old path. These algorithms estimate the EEC of each candidate path after it is explored, which results in redundant computation. Take Figure 1 as an example. Exploring candidate paths at $n6$ given the current path $[e1, e5]$, a path $[e1, e5, e8]$ would be the candidate path following the pattern of “path + edge.” However, according to the OBD data, all traces passing $n9$ are on the same path from $n2$ to $n11$, which means the estimated cost of the path $[e5, e8]$ would be the same as the corresponding part of the path $[e5, e8, e11, e12]$. The travel cost of the path $[e5, e8]$ is estimated repeatedly when the paths $[e5, e8]$, $[e5, e8, e11]$, and $[e5, e8, e11, e12]$ are explored. Therefore, in our preliminary work, we proposed an algorithm that applies a “path + another path” pattern [24].

ALGORITHM 1: General path selection algorithm framework

Require:

G : A spatial graph;
 o and d : Two nodes;
 $model$: A cost estimation model.

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

```

1: candidate paths  $CP \leftarrow$  initialization;
2: while stop criteria are not met 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

```

3.2 Preliminary Work

Our preliminary work included a physics-guided energy consumption (physics-guided) model [25] and an algorithm based on a maximal frequented path graph [24].

3.2.1 Scenario-based Physics-guided Energy Consumption Model. We proposed two energy consumption models to estimate the expected energy consumption (EEC) of traveling along paths with

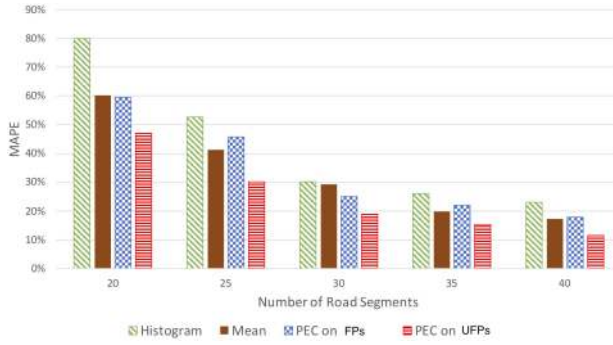


Fig. 7. MAPE of candidate methods with varying path length.

enough data, namely, a scenario-based model for frequented paths (FPs) and an FP-union model for union of frequented paths (UFPs).

The physics-guided model is a path-centric model based on the simplified powertrain energy consumption model in Equation (1). We denote the part of energy used for air resistance as $AIR = \int (\frac{A}{2\eta} c_{air} \rho v^3) dt$ and denote the part of energy used for rolling resistance and acceleration as the product of a vehicle parameter factor $V = \frac{m}{\eta}$ and a motion property factor $M = \int (av + c_{rr}gv) dt$. Equation (1) can be written as

$$W = AIR + V \times M, \quad (2)$$

where W , AIR , and M are vectors whose elements represent their values on each edge, while V is a scalar determined by the vehicle.

We clustered the traces along each FP into k scenarios using the K-means algorithm, a popular clustering method, according to their M , which is determined by motion properties, and AIR , which is determined by motion properties as well as a vehicle's front area and powertrain system efficiency. The traces in each group record vehicles of similar motion properties, front surface area, and powertrain system efficiency, but varied mass due to cargo loads. The energy consumption of the traces in a scenario will be a linear function of V , whose intersect and slope are the traces' shared AIR and M , respectively. The detailed initialization, update, and assignment steps of the K-means algorithm are discussed in Reference [25].

We also propose an FP-union model based on the path decomposition method introduced in Reference [33] to evaluate the energy consumption of a trace-union path (UFP). The key to estimating energy consumption along a UFP is to join the scenarios of adjacent FPs in the decomposition of the UFP according to the M and AIR of each scenario on the shared edges of the adjacent FPs. A scenario on an FP is joined with the scenario with the most similar M and AIR on the next FP in the path decomposition. The similarity of M and AIR and the method to calculate V are discussed in detail in Reference [25].

We experimentally compared the accuracy of the physics-guided model, including the scenario-based model for FPs and the FP-union model for UFPs with two statistics that are commonly used as ground truth in state-of-the-art related work: mean of the historical energy consumption and the distribution of historical energy consumption represented by a histogram. The data used included a road system from OpenStreetMap and OBD data collected from a vehicle in Fort Worth, Texas, from 1/1/2017 to 3/1/2018, which contained 10,759 traces.

Figure 7 shows the mean absolute percentage error (MAPE) of estimates provided by the histogram method, the mean method, and the physics-guided model on FPs, as well as the physics-guided model on UFPs with the minimum number of traces on an FP as 10. As can be seen, all

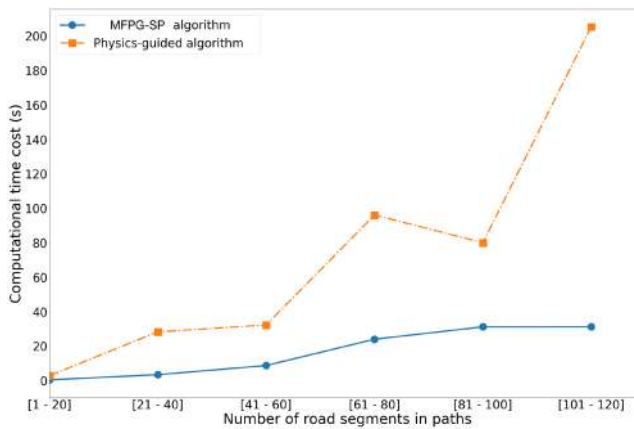


Fig. 8. Computational time cost of the physics-guided and the MFGP-SP algorithm with varying number of edges in the objective paths.

methods became more accurate as the number of edges increases. The physics-guided model was clearly the most accurate on FPs. On UFPs, it achieved similar accuracy as the mean method, even though it requires much less data.

3.2.2 Maximal-Frequented-Path-Graph Shortest-Path (MFGP-SP) Algorithm. The MFGP-SP algorithm explores new candidate paths following a “path + another path” pattern in Line 4 of Algorithm 1 using the maximal frequented path graph (MFGP) of the input spatial graph. Because given a spatial graph, the path to be found in this problem is either an FP or a UFP, and all FPs and UFPs are represented by an MFP-node or a path in the MFGP of the input data, the MFGP-SP algorithm is composed of two steps: (1) find the MFP-node or the path in the MFGP that links the origin and destination, and (2) get the objective path in the spatial graph represented by the found MFP-node or the path in the MFGP. Since the exploration of candidate paths is conducted in the MFGP, and each MFP-node is an MFP, the paths in the spatial graph represented by the candidate paths are explored following the pattern of “path + another path.” In this way, we avoid the redundant computation for estimating the expected energy consumption of the sub-paths of MFPs. The details of the MFGP-SP algorithm following the general framework Algorithm 1 are as follows: In Line 1, the set of candidate paths CP is initialized with the MFPs where the origin lies. The stop criterion in Line 2 is that there is no candidate path with cost lower than that of the found path from the origin to the destination. The most promising path in CP is the path with the lowest cost. In Line 4, as candidate paths are searched, one MFP is added to the currently most promising path if the two can form a UFP. Once a path is extended to the destination, we estimate its cost and remove it from the candidate path set. If the estimated cost is lower than the current lowest cost, the result path and the lowest cost are updated. The cost estimation method used in Line 5 is provided as an input.

We experimentally compare the performance of the MFGP-SP algorithm against the physics-guided algorithm in Reference [25] on a real dataset containing 10,129 traces of OBD data collected from three UPS trucks in Fort Worth, Texas, 1/1/2017–6/30/2018. Each trace logs the status of a truck when it moves between two delivery stops. The road system is from OpenStreetMap. The origin-destination (OD) pairs of each energy-efficient path query in the experiments were the OD pairs of the traces of the OBD data. Figure 8 shows the results when the the minimum number of traces on an FP is 20. As can be seen, the MFGP-SP algorithm always has a smaller time cost than

the physics-guided algorithm. Furthermore, the gap increases and indeed becomes overwhelming with increasing result path length.

Even though the MFPG-SP algorithm realizes “path + another path” pattern when exploring candidate paths, it applies an uninformed search strategy to find the objective path, which does not make any use of the information we have about the destination to help in the search process. An example of the information is the direction of the destination. If the destination is to the east of the origin, a path heading east may be more likely to be the energy-efficient path compared with a path heading west. The ignorance of the information may result in redundant computation to explore paths that are impossible to reach the destination. Furthermore, the MFPG-SP algorithm assumes travel cost is always positive, which is not applicable for vehicles with regenerative breaking.

4 PROPOSED APPROACHES

We design an admissible heuristic in a maximal frequented path graph (MFPG) that can guide the search space towards the destination and still guarantee finding the correct path. Then, we apply it in an informed maximal-frequented-path-graph shortest-path (IN-MFPG-SP) algorithm. In addition, we propose a maximal-frequented-path-graph label-correcting (MFPG-LC) algorithm to handle possibly negative energy consumption in the energy-efficient path selection problem.

4.1 MFPG Heuristic

Since an MFPG is a representation of a spatial graph and the OBD data on it, we take an admissible heuristic in the spatial graph as an input to compute the heuristic in the MFPG, which we call MFPG heuristic. We name the spatial-nodes on an MFP where the MFP form a UFP with other MFPs as the transfer spatial-nodes of the MFP. Then, we define the heuristic travel cost from an MFP-node P to the destination as:

$$H(P) = \min_{n \in N_{P(\text{transfer})}} \{h(n)\}, \quad (3)$$

where $N_{P(\text{transfer})}$ is the set of transfer spatial-nodes of an MFPG-node P , and $h(n)$ is an admissible heuristic travel cost from a spatial-node n to the destination. When the destination is on MFP-node P , then $H(P) = 0$. If an MFP-node does not contain any transfer spatial-nodes or the destination, its heuristic travel cost would be ∞ , since there would be no path from it to the destination.

To use the MFPG heuristic in the EPS problem, we propose an admissible heuristic for the energy consumption of a vehicle from a spatial-node to the destination in a spatial graph based on the physics model shown in Equation (1). In this model, the energy is consumed for three purposes, namely, accelerating $\frac{m}{\eta} \int (av)dt$, working against rolling resistance $\frac{mc_{rr}g}{\eta} \int (v)dt$, and working against air resistance $\frac{Ac_{air}\rho}{2\eta} \int (v^3)dt$. A heuristic travel cost is admissible if it always underestimates the actual travel cost in a path selection algorithm, so we find the lower bound of the energy consumption as an admissible heuristic. Suppose that the physical parameters of the vehicle are given. The minimal energy for acceleration is reached when the vehicle keeps a constant speed, in which case the energy for acceleration is 0. The energy for working against rolling resistance is linearly correlated to the travel distance, so its minimum is reached when the distance is equal to the Euclidean distance between the spatial-node and the destination. The energy for working against air resistance increases with the velocity of the vehicle, so its minimum is reached when the velocity is the minimum speed by law in the road system. To ensure that this heuristic is admissible, we make use of the lowest values of c_{rr} , c_{air} , and ρ from the literature and assume the powertrain system η to be 1. Because this heuristic for energy consumption is admissible, we can get the MFPG heuristic for energy consumption in an MFPG by using this heuristic as $h(n)$.

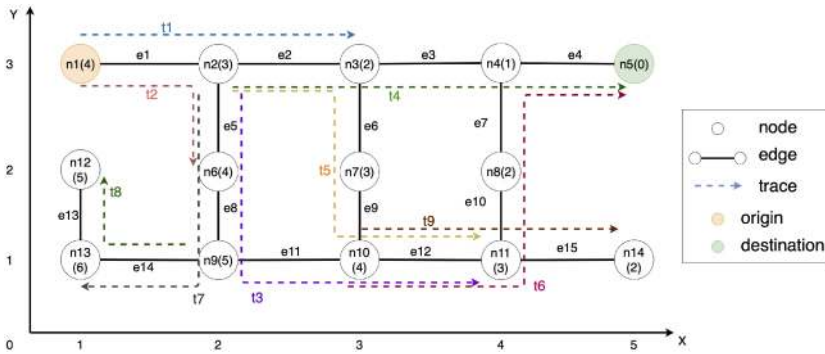


Fig. 9. Sample data in Figure 1 with additional spatial-nodes, spatial-edges, and traces of OBD data.

t7			t8			t9		
edge	Energy consumption	Average speed	edge	Energy consumption	Average speed	edge	Energy consumption	Average speed
e5	1	6	e14	1	5	e12	1	8
e8	1	5	e13	1	6	e15	1	7
e14	1	7						

Fig. 10. Three new traces of OBD data in Figure 9 in addition to those in Figure 2.

4.2 Informed MFPG-SP (IN-MFPG-SP) Algorithm

We adjust the MFPG-SP algorithm by including the MFPG heuristic in the cost of a path in an MFPG to develop the informed maximal-frequented-path-graph shortest-path (IN-MFPG-SP) algorithm. Similar to the MFPG-SP algorithm, the IN-MFPG-SP algorithm has two steps as well. The second step is the same in both algorithms, but the first step of the IN-MFPG-SP algorithm uses the A* algorithm to efficiently find the MFP-nodes or the paths in an MFPG linking the origin and destination. That is to say, in the first step, the IN-MFPG-SP algorithm orders the candidate paths in the MFPG not by the actual travel cost of the path ($C(path)$), but by the full travel cost $F(path) = C(path) + H(P)$, where P is the last MFP of the path, and $H(P)$ is the heuristic travel cost from P to the destination.

To show how the MFPG heuristic helps guide the search space towards the destination, we modify the spatial graph from Figure 1 to include three more traces, namely, $t7$, $t8$, and $t9$ (Figure 9), whose details are shown in Figure 10. Suppose we set the minimum number of traces along an FP as 1. All of the added traces lie along new MFPs, since none is along a sub-path of any other FP. The data in Figure 9 can be represented by the MFPG shown in Figure 11. For simplicity, we assume that the energy consumption is only determined by travel distance, and that the edge length in the spatial graph is 1. We find that the least energy consumption per unit distance is 1 in the OBD data, so we use 1 times the Manhattan distance from a spatial-node to the destination as the admissible heuristic of the spatial graph. The heuristic energy consumption of each spatial-node to the destination $n5$ is annotated in parentheses (Figure 9). We calculate the heuristic travel cost of all the MFP-nodes according to Equation (3) (Table 2). For example, $n3$ is the only transfer spatial-node on $P1$ ($[e1, e2]$) whose heuristic travel cost is 2, so the heuristic travel cost of $P1$ is 2. The destination is on $P4$ and $P6$, so their heuristic travel cost is 0. Since there is no transfer spatial-node on $P8$ and $P9$, their heuristic travel cost is ∞ .

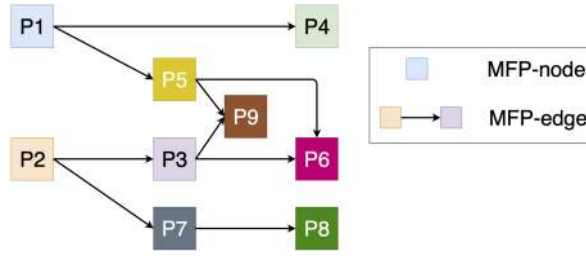


Fig. 11. The MFPG for the spatial graph and OBD data in Figure 9 (the color of each MFP matches the trace on it).

Table 2. Heuristic Travel Cost of MFP-nodes in Figure 11

MFP-node	Transfer Spatial-nodes	$H(o)$
P1	n3	2
P2	n6	4
P3	n11	3
P4	-	0
P5	n11	3
P6	-	0
P7	n13	6
P8	-	∞
P9	-	∞

Table 3. Execution Trace of the MFPG-SP Algorithm

Step	CP	$C(o)$	p	Result Path (Cost)
1	[P1],[P2]	0,0	[P1]	
2	[P1, P4], [P1, P5],[P2]	10,11,0	[P2]	[P1, P4] (21)
3	[P1, P5], [P2, P3], [P2, P7]	11,3,2	[P2, P7]	[P1, P4] (21)
4	[P1, P5], [P2, P3], [P2, P7, P8]	11,3,4	[P2, P3]	[P1, P4] (21)
5	[P1, P5], [P2, P3, P6], [P2, P3, P9], [P2, P7, P8]	11,12,11,4	[P2, P7, P8]	[P2, P3, P6] (18)
6	[P1, P5], [P2, P3, P9]	11,11	[P2, P3, P9]	[P2, P3, P6] (18)
7	[P1, P5]	11	[P1, P5]	[P2, P3, P6] (18)
8	[P1, P5, P6], [P1, P5, P9]	20,19		[P2, P3, P6] (18)

First, we refer to the execution trace in Table 3 to show how the original MFPG-SP algorithm solves the problem with the new trace data shown in Figure 9. In step 1, the set of candidate paths is initialized to include [P1] and [P2], the two MFPGs where the origin lies. Since both paths have the same cost, we can extend either one. [P1] is extended, which results in [P1, P4] and [P1, P5]. [P1, P4] already reaches the origin, so the cost along it is estimated to be 21, and it is removed from the set of candidate paths. At every step the process continues extending the most promising paths one MFP at a time, so in step 2 [P2] is extended, and so on. It takes eight steps for the algorithm to finally find the path.

Table 4. Execution Trace of the Informed MFPG-SP Algorithm

Step	CP	$C(\circ)$	$H(\circ)$	$F(\circ)$	ρ	Result Path (Cost)
1	[P1], [P2]	0, 0	2, 4	2, 4	[P1]	
2	[P1, P4], [P1, P5], [P2]	10, 11, 0	0, 3, 4	10, 14, 4	[P2]	[P1, P4] (21)
3	[P1, P5], [P2, P3], [P2, P7]	11, 3, 2	3, 3, 6	14, 6, 8	[P2, P3]	[P1, P4] (21)
4	[P1, P5], [P2, P3, P6], [P2, P3, P9], [P2, P7]	11, 12, 11, 2	3, 0, ∞ , 6	14, 12, ∞ , 8	[P2, P7]	[P2, P3, P6] (18)
5	[P1, P5], [P2, P3, P9], [P2, P7, P8]	11, 11, 4	3, ∞ , ∞	14, ∞ , ∞	[P1, P5]	[P2, P3, P6] (18)
6	[P1, P5, P6], [P1, P5, P9], [P2, P3, P9], [P2, P7, P8]	20, 19, 11, 14	0, ∞ , ∞ , ∞	20, ∞ , ∞ , ∞		[P2, P3, P6] (18)

The execution trace of the proposed IN-MFPG-SP algorithm is shown in Table 4. Again, step 1 initializes the set of candidate paths to include [P1] and [P2]. In this case, however, it is [P1] that should be extended, since its full travel cost ($F(\circ)$) is lower than that of [P2]. Sometimes the rank of the candidate paths according to their $F(\circ)$ is different from that according to their $C(\circ)$, when the MFPG heuristic works. For example, in step 3, the original MFPG-SP algorithm extends [P2, P7] while the informed IN-MFPG-SP algorithm extends [P2, P3], since [P2, P3] heads to the direction of the destination. It takes the IN-MFPG-SP algorithm only six steps to find the path. Therefore, the MFPG heuristic can reduce the computational cost.

4.2.1 Analysis of the IN-MFPG-SP Algorithm. Because the A^* algorithm is complete and correct if the heuristic used is admissible, and the first step of the IN-MFPG-SP algorithm uses it to find an MFP-node or a path in an MFPG representing the energy-efficient path, if the MFPG heuristic is admissible, the IN-MFPG-SP algorithm is complete and correct.

LEMMA 4.1. *If $h(n)$ is an admissible heuristic for a spatial-node n , $H(P)$ is an admissible heuristic for MFP-node P , where $H(P)$ is defined in Equation (3).*

PROOF. We prove this lemma by contradiction. Assume that $H(P)$ is not admissible when $h(n)$ is admissible. Let \hat{n} be the spatial-node with the smallest actual travel cost to the destination on P , so \hat{n} must be a transfer spatial-node of P . We denote the actual and heuristic travel cost from \hat{n} to the destination as $C(\hat{n})$ and $h(\hat{n})$, respectively. Because $h(n)$ is admissible, $C(\hat{n}) > h(\hat{n})$. Since $H(P)$ is not admissible, the actual travel cost from at least one spatial-node in P to the destination is less than $H(P)$. Then, $H(P) > C(\hat{n})$, since we have proved that $C(\hat{n}) > h(\hat{n})$, $H(P) > h(\hat{n})$, which results in a contradiction with the definition of $H(P)$. \square

The worst-case complexity of the IN-MFPG-SP algorithm is the same as that of the MFPG-SP algorithm, and is $O(|E_{MFP}||V_{MFP}|)$, where E_{MFP} and V_{MFP} are the sets of MFP-nodes and MFP-edges, respectively.

4.3 Maximal-Frequented-Path-Graph Label-Correcting (MFPG-LC) Algorithm

As regenerative braking is installed on a growing number of vehicles, the cases where the expected energy consumption is negative on an edge or a path becomes common, especially along downhill slopes in mountainous areas. To enable the algorithm to deal with edges or paths with negative costs, we propose an algorithm using a label-correcting strategy, called MFPG-LC algorithm. This approach does not assume that the first path that reaches an MFP-node is the path with the lowest cost to the MFP-node from the origin. To account for this, a visited node is allowed to be extended. Moreover, the MFPG-SP algorithm terminates once all the candidate paths have a cost greater than

t1		t2		t3	
edge	Energy consumption	edge	Energy consumption	edge	Energy consumption
e1	2	e1	3	e5	5
e2	3	e5	5	e8	2
t4		t5		t6	
edge	Energy consumption	edge	Energy consumption	edge	Energy consumption
e2	1	e2	9	e12	-3
e3	2	e6	2	e10	-2
e4	1	e9	2	e7	-2
		e12	-5	e4	1

Fig. 12. Sample OBD Data with Negative Edge Costs.

Table 5. Execution Trace of the MFPG-SP Algorithm with Negative Edge Costs

Step	CP	Cost	p	Result Path (Cost)
1	[P1], [P2]	0,0	[P1]	
2	[P1, P4], [P1, P5], [P2]	4,8,0	[P2]	[P1, P4] (7)
3	[P1, P5], [P2, P3]	8,8	[P2, P3]	[P1, P4] (7)

the minimum cost of the path found. This would not work if travel cost could be negative, since a candidate path may eventually lead to the destination via paths that have negative costs and have a cost lower than the current minimum. Therefore, the termination condition needs to be adjusted for the negative cost.

To overcome the issue of negative travel cost, we use a technique as follows: Before the first step of the MFPG-SP algorithm, that is, finding the MFP-node or the path in the MFPG linking the origin and destination, all the spatial-edges that have negative costs are identified. The sum of the smallest costs on all these edges indicates the maximum cost that could be regained in the graph. This is called the maximal regain bound. Now, if the costs of all the candidate paths are greater than the cost of the current result path minus the maximal regain bound, it would be impossible for any of these paths to be extended and eventually reach the destination with a cost lower than the currently found path. Only in this case would the algorithm terminate.

To illustrate the advantage of MFPG-LC algorithm, we modify some of the traces from Figure 2 to include negative travel costs and then compare how the MFPG-LG and the original MFPG-SP algorithms handle this case. The modified OBD data are shown in Figure 12. The MFPG for the data would stay unchanged, as shown in Figure 4.

First, the execution trace of the MFPG-SP algorithm is shown in Table 5. The algorithm terminates in step 3, since the cost of the found result path is lower than those of both candidate paths. The path with the lowest cost is $[P1, P4]$ whose cost is 7.

The MFPG-LC algorithm solves this problem as follows: The algorithm estimates the maximal regain bound in the graph first. It identifies all the spatial-edges with negative costs, namely, $e7$,

Table 6. Execution Trace of the MFPG-LC Algorithm with Negative Edge Costs

Step	CP	Cost	p	Result Path (Cost)
1	[P1], [P2]	0,0	[P1]	
2	[P1, P4], [P1, P5], [P2]	4,8,0	[P2]	[P1, P4] (7)
3	[P1, P5], [P2, P3]	8,8	[P1, P5]	[P1, P4] (7)
4	[P1, P5, P6], [P2, P3]	10,8	[P2, P3]	[P1, P5, P6] (5)
5	[P2, P3, P6]	11	-	[P1, P5, P6] (5)

e_{10} , and e_{12} . Then, it calculates the sum of the minimal costs on these edges, which is -9 . Table 6 shows the execution trace of the MFPG-LC algorithm. In step 2, the algorithm finds that $[P1, P4]$ already reaches the destination, so it removes the path from the set of candidate paths and estimates the cost of the objective path in the spatial graph. Since the sum of the costs of the candidate paths is less than the cost of the current result path minus the maximal regain bound, the algorithm continues to execute. The algorithm terminates in step 5 when there is no candidate path left to explore.

The result path found by the MFPG-LC algorithm linking the origin and destination in the MFPG is $[P1, P5, P6]$, which corresponds $[e1, e2, e6, e9, e12, e10, e7, e4]$. Its expected energy consumption is 5, which is lower than that of the path found by the MFPG-SP algorithm, so the MFPG-SP algorithm terminates before finding the correct result. Hence, using the MFPG-LC algorithm would help us find the correct results when travel cost may be negative.

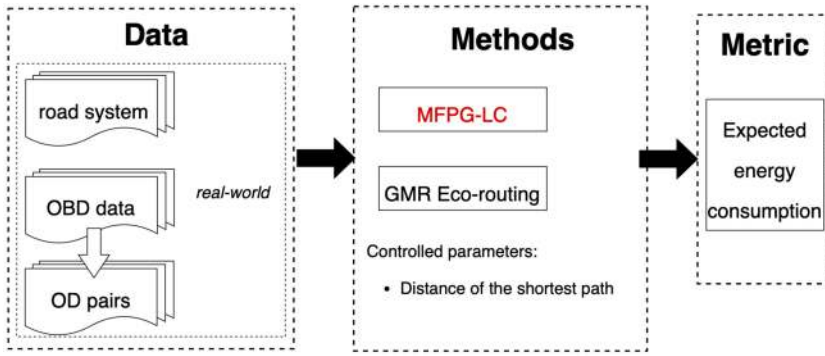
The complexity of the MFPG-LC algorithm is the same as that of the MFPG-SP algorithm, that is, $O(|E_{MFP}| |V_{MFP}|)$, where E_{MFP} and V_{MFP} are the sets of MFP-nodes and MFP-edges, respectively.

5 EXPERIMENTS

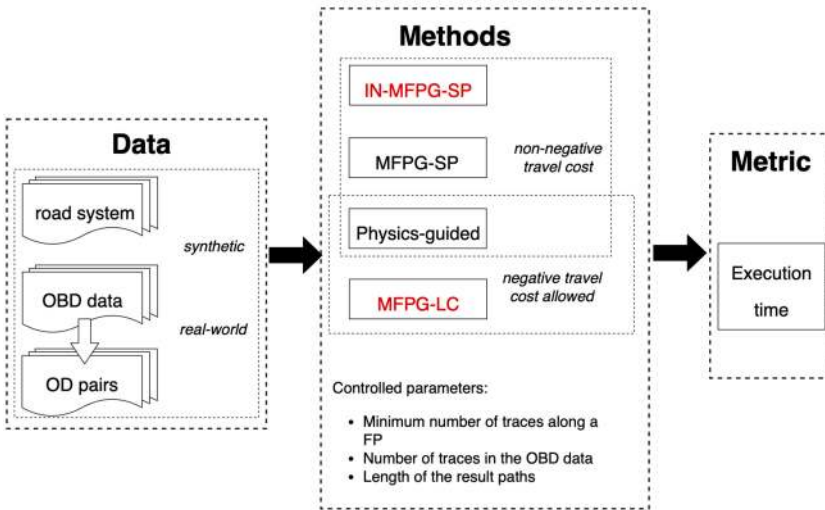
We conducted experiments: (1) to compare the expected energy consumption of the paths suggested by the proposed method (MFPG-LC) and the state-of-the-art energy-efficient path selection method (GMR Eco-routing) [21] and (2) to compare the computational performance of the proposed IN-MFPG-SP and MFPG-LC algorithms against the physics-guided [25] and MFPG-SP [24] algorithms introduced in our preliminary work.

5.1 Experiment Settings

We designed the control experiments shown in Figure 13. In energy-consumption-saving experiments the candidate methods were the proposed method (MFPG-LC) and the state-of-the-art energy-efficient path selection method (GMR Eco-routing) [21]. The GMR Eco-routing method is an edge-centric method, which estimates the expected energy consumption (EEC) on individual road segments and selects paths according to the estimates. This method estimates EEC by categorizing road segments based on their speed limit and fits a Gaussian mixture regression model in each category for EEC according to the average speed, speed change, average elevation change, road segment length, and speed limit. In the computational performance experiments, the candidate algorithms for the case where the travel cost was non-negative were the physics-guided [25], the MFPG-SP [24], and the IN-MFPG-SP algorithms. The ones for the case where the travel cost could be negative were the physics-guided [25] and the MFPG-LC algorithms. The metric measuring energy consumption saving was expected energy consumption, which was estimated using the physics-guided energy consumption model, and the metric for computational performance was execution time.



(a) Energy consumption saving experiments



(b) Computational performance experiments

Fig. 13. Experiment Design.

The experiments were designed to answer the following questions:

- Comparative analysis:
 - Is the EEC of the paths selected by the proposed method lower than that of the paths selected by the state-of-the-art methods?
 - Are the proposed algorithms more computationally efficient than the algorithms in preliminary work?
- Sensitivity analysis:
 - How are the proposed methods affected by the number of input traces?
 - How are the proposed methods affected by the minimum number of traces along an FP?
 - How are the proposed methods affected by the length of the result path?

The real-world dataset used in the experiments was the OBD data collected from three UPS trucks in Fort Worth, Texas, between 1/1/2017 and 6/30/2018. There were 10,129 traces in the OBD data, each of which logged 250 engine measurements (e.g., energy consumption and stop

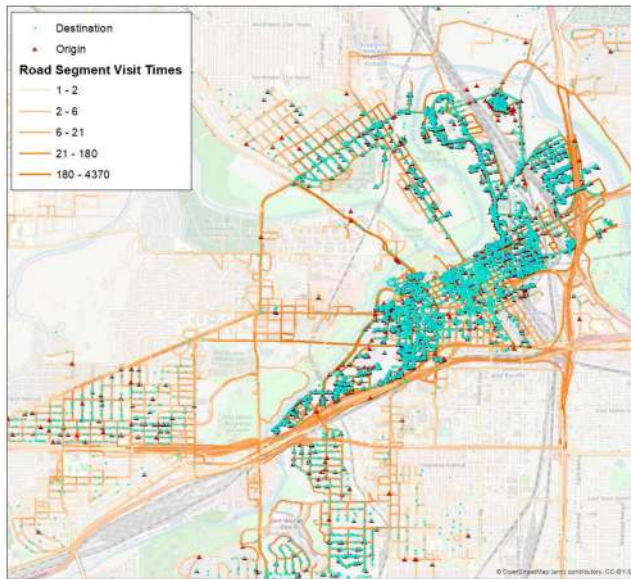


Fig. 14. A map of the road segments visited by traces in the OBD data and OD pairs.

count) along with the geographic location of a vehicle varying with time when it moved between two delivery stops. A map-matching algorithm from Reference [26] was used to align the data with a digital map from the OpenStreetMaps, which contained 9,084 road segments and 6,193 intersections. Figure 14 shows the map of Fort Worth, Texas, with the OBD data. The orange lines show the distribution of OBD data on the map. The darker shades of orange show those paths that had a larger number of traces along them. The origin-destination (OD) pairs of each energy-efficient path query in the experiments were the OD pairs of the traces, so there were 10,129 OD pairs in total. In Figure 14, origins are the red triangles, while destinations are the blue circles. To calculate the MFPG heuristic in the experiments, the parameters affecting the heuristic were set as $m = 3000kg$, $g = 9.8m/s^2$, $A = 2m^2$, $s = 20miles/hour$, $\rho = 1.14kg/m^3$, $c_{air} = 0.4$ [4], and $c_{rr} = 0.4$ [20].

The spatial graph in the synthetic data consisted of 5,929 spatial-nodes, and 9,560 spatial-edges between these spatial-nodes in a grid pattern. Spatial-edges were between adjacent spatial nodes (horizontally and vertically) with a probability. The degree of each spatial-node is 3.2 on average. 15,000 traces were randomly created on the graph. The length of the traces was between 1 to 120. For the sake of simplicity, the travel cost metric was time. The speed of the vehicle was assumed to be between 30 and 50 miles per hour. Left turns were given a higher time penalty compared with right turns to reflect the actual road conditions. Out of the 9,560 spatial edges in the graph, 3,968 were traversed by at least 1 trace. The OD pairs of each fastest path query in the experiments were the OD pairs of the synthetic traces and 10,000 randomly generated OD pairs.

The experiments were conducted on a machine with Intel(R) Core™i5-7500 CPU @ 3.40 GHz and 64 GB memory. The operating system used was Windows 10. The algorithms were implemented using C# .NET Framework 4.7.

5.2 Experiment Results

5.2.1 Energy-consumption-saving Experiments. Among 10,129 OD pairs of the traces in the OBD data, the proposed MFPG-LC method found energy-efficient paths between 4,300 (about 42.45%)



Fig. 15. The average EEC of the paths selected by the GMR Eco-routing method and the MFPG-LC method.

Table 7. Number of Frequented Spatial-edges Varying with the Number of Traces in OBD Data

(a) Spatial-edges with more than 20 traces along it in the real-world data

Number of Traces	5,065	7,597	10,129
Number of Road Segments	864	961	1,112

(b) Spatial-edges with more than 25 traces along it in the synthetic data

Number of Traces	7,500	11,250	15,000
Number of Road Segments	3,029	3,388	3,538

of them. Then, we queried the paths between these 4,300 OD pairs using the GMR Eco-routing method and estimated the EEC of the result paths using the physics-guided model. Figure 15 shows the average EEC of the paths selected by the GMR Eco-routing method and the MFPG-LC method along with the length of the shortest paths between the OD pairs. For example, between the OD pairs between which the shortest paths are of length from 1 km to 2 km, the average EEC of the paths selected by the GMR Eco-routing method is 1.07 kWh, and that of the paths selected by the MFPG-LC method is 0.61 kWh. The average EEC of the paths selected by both methods increased as the OD pairs grew away from each other, but the average EEC of the paths selected by the GMR Eco-routing method was always higher than that of the MFPG-LC method. Therefore, the proposed method was able to find paths with lower EEC than the GMR method.

5.2.2 Computational Performance Experiments.

- Are the proposed algorithms more efficient than the algorithms in preliminary work?

The number of spatial-edges on all FPs varies with the number of traces (T) in OBD data (Table 7) and with the minimum number of traces (β) on an FP (Table 8). Hence, the number of FPs and UFPs depends on these two factors as well, which affects the search space of the path selection algorithms in turn. The number of edges in the result paths also affects the number of iterations the algorithms need to find the path. Therefore, we compared the performance of the algorithms with a fixed β (20 for the real-world data and 25 for the synthetic data) and a fixed T (all traces in both datasets) on result paths with varying length.

Figure 16 shows the execution time of the algorithms in the case where the travel cost is non-negative. In each sub-figure, the Y axis is the execution time, while the X axis is the result path

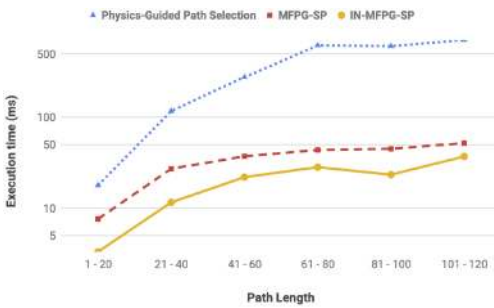
Table 8. Number of Frequented Spatial-edges Varying with the Minimum Number of Traces on It

(a) Real-world data

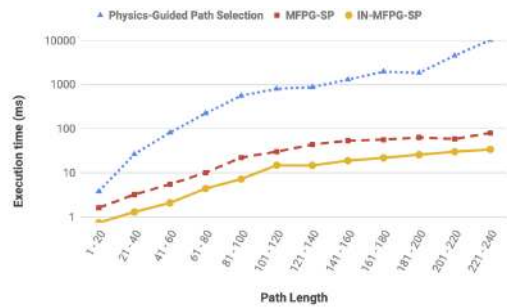
Minimum number of traces on an FP	20	35	50
Number of Road Segments	1,112	858	731

(b) Synthetic data

Minimum number of traces on an FP	25	50	75
Number of Road Segments	3,538	2,988	2,166



(a) Real-world data.



(b) Synthetic data.

Fig. 16. Is the proposed IN-MFPG-SP algorithm more efficient than the physics-guided MFPG-SP algorithms?

length. We can see that in all cases the MFPG-SP and the IN-MFPG-SP algorithms are faster than the physics-guided method. The difference in their performance becomes more evident as the path length increases. This is because as the path length increases, the physics-guided method requires more iterations to traverse a given path compared with the MFPG-SP and the IN-MFPG-SP algorithms, which extend paths by appending paths rather than edges. Also, we see that the admissible heuristic reduces the time required by the IN-MFPG-SP algorithm, as the search space is guided towards the destination.

Figure 17 shows the execution time of the algorithms in the case where the negative travel cost is allowed. Again, the Y axis is the execution time, while the X axis is the result path length. We see that the execution times for both the MFPG-LC and the physics-guided algorithms are higher than the case where the travel cost is non-negative in Figure 16(a). This is because it took these algorithms longer to terminate once a path is found to deal with the negative edge costs. In this case, too, we find that the MFPG-LC algorithm outperforms the physics-guided algorithm, greatly reducing the amount of time taken by the algorithm.

- *How are the proposed methods affected by the number of input traces?*

To illustrate the sensitivity of the proposed algorithms on the number of traces in the input data, we generated two subsets of both the real-world and the synthetic data using random sampling without replacement. One had 75% original traces, and the other one had 50% original traces. To avoid the effect of the result path length, the effect of the number of traces in the input data is shown in groups of similar result path length in Figure 18, where the average execution time of

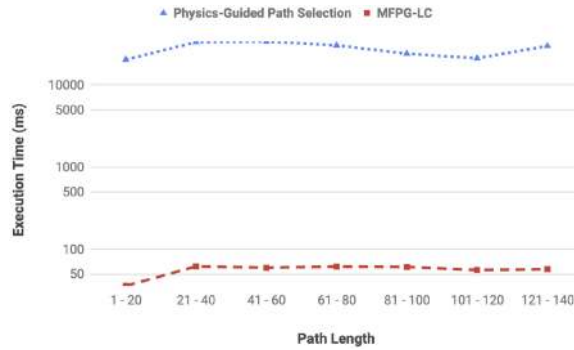
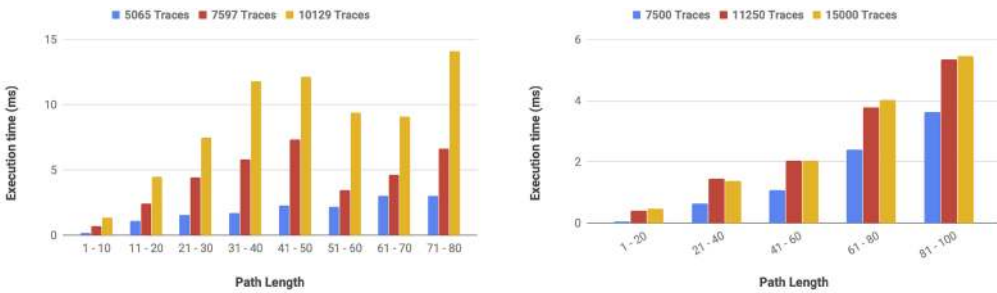
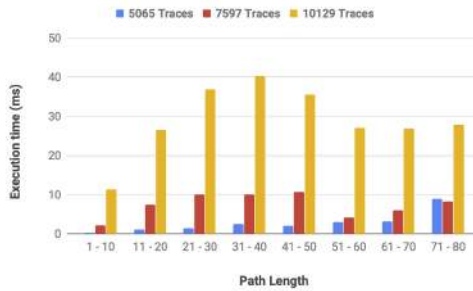


Fig. 17. Is the proposed MFPG-LC algorithm more efficient than the physics-guided algorithm?



(a) IN-MFPG-SP algorithm with the real-world data. (b) IN-MFPG-SP algorithm with the synthetic data.



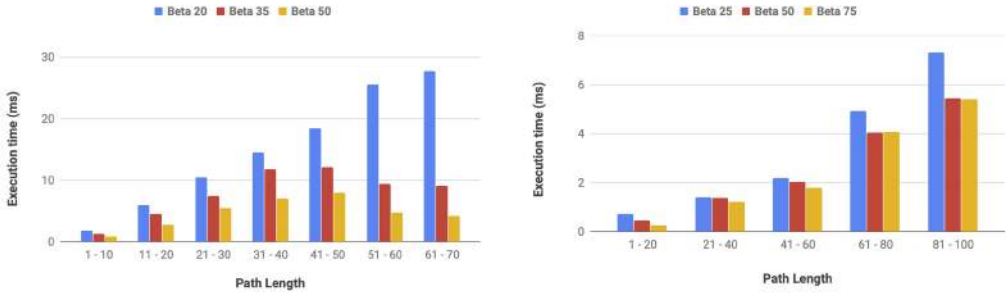
(c) MFPG-LC algorithm with the real-world data.

Fig. 18. How are the proposed methods affected by the number of input traces and the result path length?

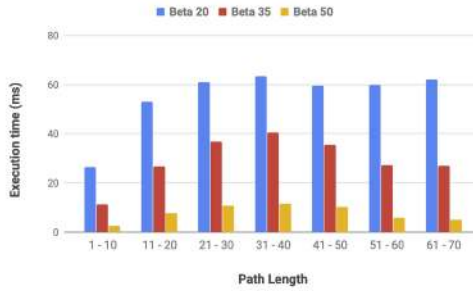
each path selection query is the Y axis, and the result path length is the X axis. The execution times on the 50%, 75%, and entire dataset are shown in blue, red, and orange, respectively. As we can see, the execution time of all the proposed algorithms increases with the number of input traces. The reason for this phenomenon is that increasing the number of input traces increases the number of FPs and UFPs and increases the search space of the path selection algorithms in turn.

- How are the proposed methods affected by the minimum number of traces along an FP (β)?

To show the effect of the minimum number (β) of traces along a frequented path (FP), we executed the proposed algorithms with varying β . $\beta = 20, 35, 50$ for real-world data and $\beta = 25, 50, 75$ for synthetic data. To avoid the effect of the result path length, the effect of β is shown in groups



(a) IN-MFPG-SP algorithm with the real-world data. (b) IN-MFPG-SP algorithm with the synthetic data.



(c) MFPG-LC algorithm with the real-world data.

Fig. 19. How are the proposed methods affected by the minimum number of traces along an FP and the result path length?

of similar result path length in Figure 19, where the average execution time of each path selection query is the Y axis, and the result path length is the X axis. The execution time with different β is shown in different colors. The results indicate that with β increasing, the execution time of the proposed algorithms stays the same or decreases. As analyzed before, as β increases, the number of road segments that have at least β traces along them decreases, which shrinks the search space of the algorithms in turn. Moreover, the effect of β increases with the length of the result paths. The reason for this is that the value of β affects the number of new candidate paths that can be explored at an MFP-node. The difference in the number of new candidate paths with different value of β accumulates with the length of the result path. Hence, choosing an appropriate value of β is essential to make these algorithms run efficiently.

- How are the proposed methods affected by the length of the result path ?

The length of the result path affects the number of iterations the path selection algorithm needs to go through to find a path. According to both Figures 18 and 19, in most cases, increasing length of the result path should increase the execution time. Since the input traces are not evenly distributed throughout the spatial graph, some longer paths may have smaller execution time, as seen in Figure 18(a), between road segment lengths 50 and 70, and in Figure 18(c) between road segment lengths 50 and 80. This is due to some longer paths going along areas that have fewer FPs, resulting in a smaller number of nodes being expanded along these paths. Figure 20 shows two such paths from the UPS truck data. The lines in blue indicate the road segments that have at least 35 traces along them, where FPs exist. The red lines indicate the path found for given origin-destination pairs. As can be seen, a lot of the FPs concentrate in a few regions in the road

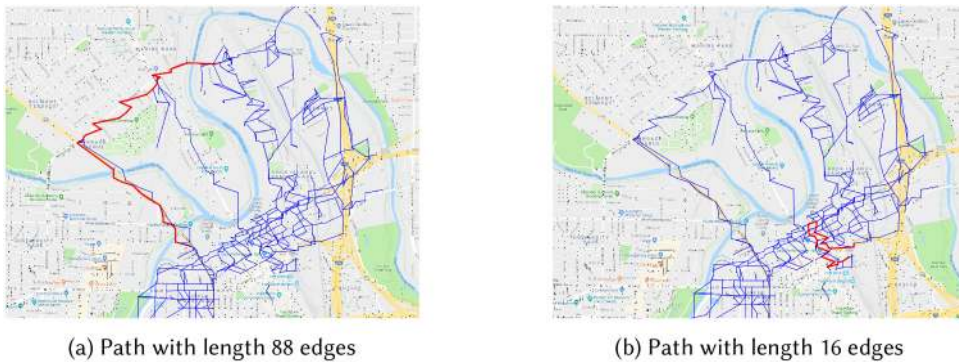


Fig. 20. Paths in the real-world data with abnormal computational time.

network. Figure 20(a) shows a path composed of 88 edges. This path does not pass through a region where FPs concentrate, hence it expands a relatively fewer number of nodes. To find this path, the IN-MFPG-SP algorithm visited 123 spatial-nodes in total, which cost only 45 ms. Figure 20(b), however, shows a path that passes through a region where FPs concentrate. Although this path is composed of 16 edges, it took the IN-MFPG-SP algorithm 246 ms to find it, and 527 spatial-nodes are visited in the searching process.

6 CASE STUDIES

We conducted two case studies: (1) to estimate the potential energy saving resulting from leveraging the proposed energy-efficient path selection method and (2) to illustrate that the proposed method can select paths that are more energy-efficient than the paths selected by the currently widely used path selection methods. The data used in case studies were the same real-world OBD data and road system used in the experiments.

6.1 Energy Saving Resulting from the Proposed Method

We queried the energy-efficient paths between 10,129 origin and destination (OD) pairs of the traces in the OBD data using the proposed method and compared their energy consumption with that of the historical paths in the data. Energy-efficient paths are found between 4,300 (about 42.45%) OD pairs. The energy-efficient paths between 2,510 (about 24.78%) OD pairs have lower expected energy consumption than the historical paths, and the others have the same as the historical paths. Figure 21 shows the frequency distribution of the 2,510 OD pairs between which the energy-efficient paths have lower expected energy consumption than the historical paths according to the relative difference between the expected energy consumption. For example, the energy-efficient paths between 680 OD pairs save 0–20% energy compared with the historical paths between the OD pairs. In this case study, the energy saving resulted from leveraging the proposed method was about 12.10% of the total energy consumption of the traces in the OBD data.

6.2 Comparison between the Proposed Method and Google Maps

We searched for a path between two road intersections in Fort Worth, Texas, using the proposed method and Google Maps, which was an example of the popular tools for routing. The proposed method selected a path 1.5 miles long with an estimated time cost of 5 minutes (Figure 22(a)). Google Maps chose a path 1.6 miles long but with a smaller estimated time cost of 4 minutes (Figure 22(b)). 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

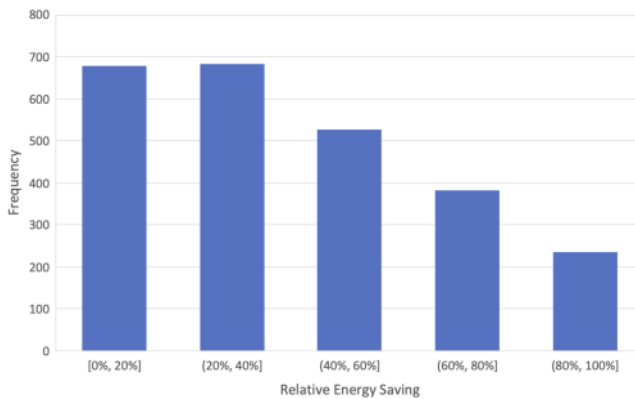
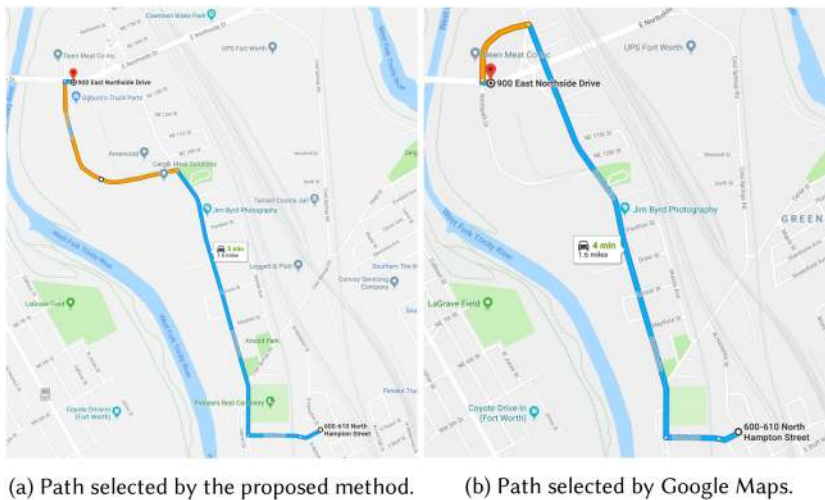


Fig. 21. Frequency distribution of the OD pairs according to the relative difference between the expected energy consumption on the energy-efficient paths and that on the historical paths between them.



(a) Path selected by the proposed method. (b) Path selected by Google Maps.

Fig. 22. A path selected by the proposed method is more energy-efficient than that from Google Maps.

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.

7 A ROAD TEST IN CINCINNATI, OH

We also conducted a real-world road test using one UPS delivery truck and one driver. Figure 23 shows the overview of the study area. The task was to find a path between 100 Commerce Dr. in Loveland, Ohio, and 8063 Montgomery Road in Cincinnati, Ohio. Google Maps suggested the path highlighted in blue. Our proposed method, instead, suggested the path highlighted in green, and estimated that the expected energy consumption of the blue path found by Google Maps was 8.54 kWh, and that of the green path suggested by the proposed method was 5.43 kWh. The expected time costs of the blue and green paths are 14 and 17 minutes, respectively. An explanation of this result is that the main segment of the blue path is a U.S. interstate, while that of the green path

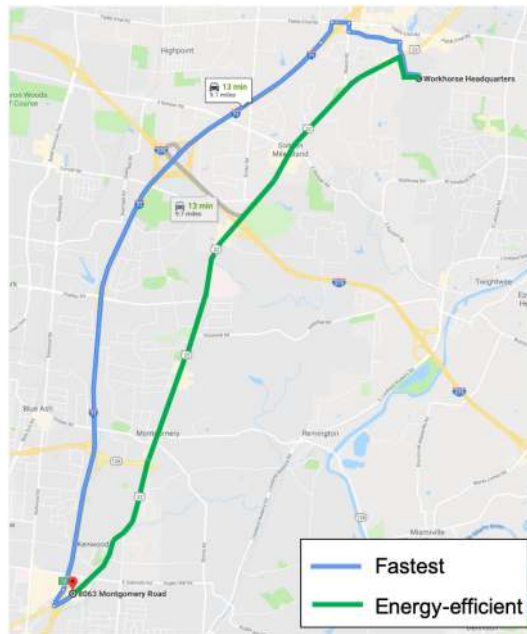


Fig. 23. Paths suggested by Google Maps and the proposed method.

is a U.S. state highway, so the blue path has a greater speed limit than the green path. In addition, the lengths of the blue and green paths are 9.2 and 7.6 miles, respectively. Even though the high speed on the interstate reduces the travel time, it causes high energy consumption. Then, the test truck was driven on both paths 12 times to collect data for validation. According to the validation data, the average energy consumption on the two paths are 8.27 kWh and 5.09 kWh. Therefore, the proposed method was able to select the more energy-efficient path than the currently widely used method.

8 CONCLUSION AND FUTURE WORK

Today's increasing volume of OBD data facilitates monitoring and managing traffic and transportation systems using the data from connected vehicles, which is an important component of a smart city. In this article, we explored the energy-efficient path selection problem, whose challenges included the dependence of the energy consumption on the physical parameters of vehicles, the dependence of the energy consumption on different edges along a path, the high computational cost of estimating expected energy consumption on a path, and the potentially negative energy consumption. We proposed the MFPG heuristic that can guide the search space of the path selection algorithms towards the destination and still guarantee finding the correct path. We also introduced an IN-MFPG-SP algorithm that uses the MFPG heuristic. In addition, we proposed the MFPG-LC algorithm to handle possibly negative energy consumption using a label-correcting strategy. We analyzed the proposed algorithms for correctness, completeness, and computational time complexity. The experiments we conducted on both real-world data and synthetic data showed that the proposed algorithms yielded substantial computational savings compared to the algorithms in our preliminary work. Then, we conducted two case studies, which illustrated that leveraging the proposed method would save 12.10% energy consumption and that the path selected by the proposed method was more energy-efficient than the path found by Google Maps and the

state-of-the-art eco-routing method. Last, we conducted a real-world road test to validate that the proposed method can help to save energy compared with Google Maps.

There are several avenues we can pursue in future work, including the precomputation and the storage of the maximal frequented path graph (MFPG), updating the MFPG given changing OBD data rather than regenerating the MFPG every time the OBD data are updated.

ACKNOWLEDGMENTS

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] Kyoungho Ahn and Hesham A. Rakha. 2013. Network-wide impacts of eco-routing strategies: A large-scale case study. *Transport. Res. Part D: Transport Environ.* 25, 0 (Dec. 2013). Retrieved from <https://trid.trb.org/view/1284524>.
- [2] Sabeur Aridhi, Philippe Lacomme, Libo Ren, and Benjamin Vincent. 2015. A MapReduce-based approach for shortest path problem in large-scale networks. *Eng. Applic. Artif. Intell.* 41 (May 2015), 151–165.
- [3] 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, 309–316.
- [4] Andreas Artmeier, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher. 2010. The shortest path problem revisited: Optimal routing for electric vehicles. In *Proceedings of the Annual Conference on Artificial Intelligence*. Springer, 309–316.
- [5] Jürgen Bang-Jensen and Gregory Z. Gutin. 2008. *Digraphs: Theory, Algorithms and Applications*. Springer.
- [6] Aaron Brooker, Jeffrey Gonder, Lijuan Wang, Eric Wood, Sean Lopp, and Laurie Ramroth. 2015. *FASTSim: A Model to Estimate Vehicle Efficiency, Cost and Performance*. SAE Technical Paper 2015-01-0973. SAE International, Warrendale, PA.
- [7] A. Capiello, I. Chabini, E. K. Nam, A. Lue, and M. Abou Zeid. 2002. A statistical model of vehicle emissions and fuel consumption. In *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*. 801–809.
- [8] 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 Trans. Intell. Transport. Syst.* 14, 4 (Dec. 2013), 1907–1917.
- [9] Daniel Dellling, Andrew V. Goldberg, Andreas Nowatzyk, and Renato F. Werneck. 2013. PHAST: Hardware-accelerated shortest path trees. *J. Parallel Distrib. Comput.* 73, 7 (July 2013), 940–952.
- [10] Yong Deng, Yuxin Chen, Yajuan Zhang, and Sankaran Mahadevan. 2012. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Appl. Soft Comput.* 12, 3 (Mar. 2012), 1231–1237.
- [11] Department of Energy. 2016. Energy Department Announces \$58 Million to Advance Fuel-Efficient Vehicle Technologies. Retrieved from <https://www.energy.gov/articles/energy-department-announces-58-million-advance-fuel-efficient-vehicle-technologies>.
- [12] Diane Cook and Len Jenshel. 2017. Buying Guide—Cars and Their Environmental Impact. Retrieved from <https://www.nationalgeographic.com/environment/green-guide/buying-guides/car/environmental-impact/>.
- [13] E. W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1, 1 (Dec. 1959), 269–271.
- [14] Daniel Duque, Leonardo Lozano, and Andrés L. Medaglia. 2015. An exact method for the biobjective shortest path problem for large-scale road networks. *Eur. J. Oper. Res.* 242, 3 (May 2015), 788–797.
- [15] Jochen Eisner, Stefan Funke, and Sabine Storandt. 2011. Optimal route planning for electric vehicles in large networks. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*. AAAI Press, 1108–1113.
- [16] EU Horizon 2020 Research and Innovation Programme. 2020. optiTruck. Retrieved from <https://optitruck.eu/>.
- [17] Yuan Gao. 2011. Shortest path problem with uncertain arc lengths. *Comput. Math. Applic.* 62, 6 (Sept. 2011), 2591–2600.
- [18] V. M. V. Gunturi, S. Shekhar, and K. Yang. 2015. A critical-time-point approach to all-departure-time Lagrangian shortest paths. *IEEE Trans. Knowl. Data Eng.* 27, 10 (Oct. 2015), 2591–2603.
- [19] P. E. Hart, N. J. Nilsson, and B. Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cyber.* 4, 2 (July 1968), 100–107.
- [20] Heinz Heisler. 2002. 14 - Vehicle body aerodynamics. In *Advanced Vehicle Technology (Second Edition)*, Heinz Heisler (Ed.). Butterworth-Heinemann, Oxford, 584–634. DOI: <https://doi.org/10.1016/B978-075065131-8/50015-4>
- [21] Xianan Huang and Hui Peng. 2018. Eco-routing based on a data driven fuel consumption model. *arXiv:1801.08602 [stat]* (Jan. 2018).

- [22] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. 2011. Big data: The next frontier for innovation, competition, and productivity. Retrieved from <https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation>.
- [23] J. Kwon, A. Rousseau, and P. Sharer. 2007. *Analyzing the Uncertainty in the Fuel Economy Prediction for the EPA MOVES Binning Methodology*. SAE Technical Paper 2007-01-0280. SAE International, Warrendale, PA.
- [24] Yan Li, Pratik Kotwal, Pengyue Wang, Shashi Shekhar, and William Northrop. 2019. Trajectory-aware lowest-cost path selection: A summary of results. In *Proceedings of the 16th International Symposium on Spatial and Temporal Databases (SSTD'19)*. ACM. DOI : <https://doi.org/10.1145/3340964.3340971>
- [25] Yan Li, Shashi Shekhar, Pengyue Wang, and William Northrop. 2018. Physics-guided energy-efficient path selection: A summary of results. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'18)*. ACM, New York, NY, 99–108. DOI : <https://doi.org/10.1145/3274895.3274933>
- [26] Paul Newson and John Krumm. 2009. Hidden Markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS'09)*. ACM, New York, NY, 336–343. DOI : <https://doi.org/10.1145/1653771.1653818>
- [27] 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, 116–125.
- [28] Christian Sommer. 2014. Shortest-path queries in static networks. *ACM Comput. Surv.* 46, 4 (Mar. 2014), 45:1–45:31.
- [29] U.S. Energy Information Administration. 2017. *International Energy Outlook 2017*. Technical Report DOE/EIA-0484(2017). Washington, DC.
- [30] U.S. Energy Information Administration. 2017. Total U.S. energy expenditures in 2015 were the lowest in more than a decade. Retrieved from <https://www.eia.gov/todayinenergy/detail.php?id=32432>.
- [31] USDOE. 2016. ARPA-E | NEXTCAR. Retrieved from <https://arpa-e.energy.gov/?q=arpa-e-programs/nextcar>.
- [32] Ram Vijayagopal, Larry Michaels, Aymeric P. Rousseau, Shane Halbach, and Neeraj Shidore. 2010. *Automated Model Based Design Process to Evaluate Advanced Component Technologies*. SAE Technical Paper 2010-01-0936. SAE International, Warrendale, PA.
- [33] Bin Yang, Jian Dai, Chenjuan Guo, Christian S. Jensen, and Jilin Hu. 2018. PACE: A PAth-Centric paradigm for stochastic path finding. *VLDB J.* 27, 2 (Apr. 2018), 153–178.
- [34] L. Zhu, J. Holden, E. Wood, and J. Gonder. 2017. Green routing fuel saving opportunity assessment: A case study using large-scale real-world travel data. In *Proceedings of the IEEE Intelligent Vehicles Symposium*. 1242–1248.

Received June 2019; revised February 2020; accepted June 2020