

Combined Eco-Routing and Power-Train Control of Plug-In Hybrid Electric Vehicles in Transportation Networks

Arian Houshmand¹, Christos G. Cassandras², *Life Fellow, IEEE*, Nan Zhou¹,
Nasser Hashemi, Boqi Li³, and Huei Peng³

Abstract—We study the problem of eco-routing for Plug-In Hybrid Electric Vehicles (PHEVs) to minimize the overall energy consumption cost. We propose an algorithm which can simultaneously calculate an energy-optimal route (eco-route) for a PHEV and an optimal power-train control strategy over this route. In order to show the effectiveness of our method in practice, we use a HERE Maps API to apply our algorithms based on traffic data in the city of Boston with more than 110,000 links. Moreover, we validate the performance of our eco-routing algorithm using speed profiles collected from a traffic simulator (SUMO) as input to a high-fidelity energy model to calculate energy consumption costs. Our results show significant energy savings (around 12%) for PHEVs with a near real-time execution time for the algorithm.

Index Terms—Vehicle routing, traffic control, energy management, intelligent vehicles.

I. INTRODUCTION

DUE to environmental concerns and the high cost of gas, there has been an increasing interest in vehicles using alternative energy sources such as Electric Vehicles (EV). However, given battery capacity levels in current EVs, their adoption is limited by the All-Electric Range (AER). In this respect, Plug-In Hybrid Electric Vehicles (PHEVs) offer a suitable alternative, as they can overcome range limitations by using both gas and electricity. Depending on battery size, PHEVs can be driven 10-40 miles on electricity, which is roughly the average daily commuting distance in the US [1]. Moreover, it is possible to decrease the energy consumption cost and the carbon footprint of PHEVs using smart eco-routing and power-train control strategies.

Traditional vehicle routing algorithms seek to find the minimum time (fastest) or shortest path routes [2]–[4], whereas

eco-routing algorithms find the paths that minimize the total energy consumption cost. Several eco-routing algorithms have been studied in the literature for conventional vehicles that are capable of finding the energy-optimal routes using historical and online traffic data [5]–[10]. Kubicka *et al.* [11] performed a case study to compare the objective values proposed in the eco-routing literature and showed that the performance of eco-routing algorithms is highly dependent on the energy model used to calculate the traveling cost of each link. Pourazarm *et al.* studied optimal routing of electric vehicles considering recharging at charging stations [12]. De Nunzio *et al.* [13] studied the eco-routing problem for EVs considering road grade and speed changes on each road link. Although eco-routing of conventional vehicles is well studied, there is little research that addresses the case of PHEVs [14]. Cela *et al.* [15] studied the problem of eco-routing for HEVs considering the vehicle longitudinal dynamics. Sun and Zhou [16] and Qiao and Karabasoglu [17] proposed the Charge Depleting First (CDF) approach to address eco-routing for PHEVs. Furthermore, it is shown in [16] that energy-optimal paths typically take more time compared to the time-optimal routes. More recently, De Nunzio *et al.* [18] proposed a semi-analytical solution of the power-train energy management based on Pontryagin's minimum principle to address the eco-routing of HEVs, and in [19] the eco-routing problem for PHEVs is solved by minimizing a combination of time and energy.

The contributions of this paper are summarized as follows. Based on the work introduced in [20], we first review a CDF strategy for finding the energy optimal route for PHEVs and propose two methods for solving this problem: a modified version of Dijkstra's algorithm [21], and a Hybrid-LP Relaxation algorithm. We then propose a Combined Routing and Power-Train Control (CRPTC) eco-routing algorithm for PHEVs that can simultaneously find an energy optimal route as well as an optimal power-train control strategy along the route. In contrast to existing methods in the literature where the power-train control strategy is considered fixed [16], [17], we allow the optimizer to find the optimal PT control strategy. We formulate the problem as a Mixed Integer Linear Program (MILP) and later relax it into a bi-level optimization problem where the upper level problem finds the eco-route and the lower level problem determines the optimal

Manuscript received February 3, 2020; revised April 23, 2021; accepted July 9, 2021. This work was supported in part by the NSF under Grant ECCS-1509084, Grant DMS-1664644, and Grant CNS-1645681; in part by AFOSR under Grant FA9550-19-1-0158; in part by ARPA-E's NEXTCAR Program under Grant DEAR0000796; and in part by MathWorks. The Associate Editor for this article was S. Olariu. (*Corresponding author: Arian Houshmand.*)

Arian Houshmand, Christos G. Cassandras, Nan Zhou, and Nasser Hashemi are with the Division of Systems Engineering, Boston University, Brookline, Boston, MA 02446 USA (e-mail: arianh@bu.edu; cgc@bu.edu; nanzhou@bu.edu; nhashemi@bu.edu).

Boqi Li and Huei Peng are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: boqili@umich.edu; hpeng@umich.edu).

Digital Object Identifier 10.1109/TITS.2021.3102496

1558-0016 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

PT switching control strategy between electricity and gas using a Linear Programming (LP) problem formulation. We show that the bi-level eco-routing algorithm is computationally more efficient than the CRPTC approach and its results are very close to the optimal values calculated using the CRPTC algorithm. Using a HERE Maps API [22], we developed a publicly available web-based tool in which we can request and download the geographical map of a region alongside its traffic information. Using this platform, we applied our eco-routing algorithms to large urban traffic networks, including the city of Boston (110,000 links, 50,000 nodes). As an alternative to such traffic data, we also use the Simulation of Urban MObility (SUMO) [23] to investigate traffic outcomes and also collect speed traces of vehicles following eco-routes and fastest routes. We then use the Vehicle-Engine SIMulation (VESIM) model [24], a high fidelity power-train energy modelling software package, to calculate the actual energy consumption of travelling through an eco-route and fastest route for each individual origin-destination (O-D) pair and compare them against each other. This approach is used to validate the performance of our eco-routing algorithm with results suggesting energy savings of about 12% compared to the fastest route. We show the trade-off between saving energy and time in Section IV.

The remainder of the paper is organized as follows. The PHEV energy consumption model is presented in Section II. A modified Dijkstra's algorithm as well as MILP problem formulation are proposed in Section III to solve the eco-routing problem. In Section IV, we explain our traffic data platform and by using it, we apply our eco-routing algorithms to the urban area of Boston. In Section V, using SUMO and VESIM we introduce a framework to validate the performance of our eco-routing algorithm in real-world scenarios. Finally, conclusions and further research directions are outlined in Section VI.

II. PHEV ENERGY CONSUMPTION MODELING

The first step in developing an efficient eco-routing algorithm is to understand how the PHEV power-train works, and how one can model its energy consumption cost. Unlike conventional vehicles where it is possible to analytically estimate fuel consumption costs as functions of the velocity and acceleration of the vehicle [25], estimating a PHEV's fuel consumption is a more involved process [26]. This is mainly due to the complexity of the PHEV power-train's architecture. A PHEV can run on electricity, gas or as a hybrid. Moreover, when PHEVs use electricity, the battery can be recharged using the regenerative brake and/or other mechanisms [27]. As such, we need a comprehensive model which takes into account the effect of motor/generator units and the Internal Combustion Engine (ICE) to calculate the fuel rate and the electrical power demand from the battery pack.

A PHEV power-train has several different components that work together to drive the vehicle including the engine, motor/generator, inverter, etc. The interactions between these components should be considered to estimate the vehicle's energy consumption. The energy consumption cost of

a PHEV over a finite time horizon can be expressed as follows:

$$\int_{t_0}^{t_f} (C_{gas} \dot{m}_{gas}(t) + C_{ele} P_{batt}(t)) dt \quad (1)$$

where $\dot{m}_{gas}(t)$ is the instantaneous fuel consumption rate, and P_{batt} is the total electrical power used/generated by the motor/generator units. Moreover, C_{gas} (\$/gallon) and C_{ele} (\$/kWh) are the cost of gas and electricity, respectively. We discuss two possible approaches to calculate $\dot{m}_{gas}(t)$ and $P_{batt}(t)$ at any operating condition: a direct method and an indirect method.

A. Direct Method

One can calculate the vehicle's energy consumption at any given time, knowing the details of a vehicle's power-train architecture, efficiency maps of engine and motor/generator units, and physical parameters of the vehicle [28]. To do so, we need to have the torque and speed demand from the engine and motor/generators at any given time. Considering the vehicle's specifications, we can translate its speed and acceleration to torque and rotational speed demand from the engine and motor/generator units [29]. We can then use these values to extract $\dot{m}_{gas}(t)$ and $P_{batt}(t)$ from the efficiency maps. We can either use commercially available software such as *Autonomie/PSAT* [30], or develop our own functions by knowing the details of a specific vehicle. There are two functions through which we can calculate the fuel and electrical energy consumptions for any given vehicle as follows:

$$\begin{aligned} \dot{m}_{gas}(t) &= f(v(t), a(t)) \\ P_{batt}(t) &= g(v(t), a(t)) \end{aligned}$$

where $v(t)$ and $a(t)$ are the speed and acceleration of the vehicle at any given time respectively. Even though this approach may lead to accurate estimates of energy consumption values, it is an elaborate method which is not suitable for the purpose of our higher-level eco-routing framework. Hence, we use a computationally more efficient approach which we call "indirect method".

B. Indirect Method

We use a simplified energy model which was first proposed by Qiao and Karabasoglu [17] to calculate the energy consumption cost of PHEVs. Instead of using real time driving data for a targeted vehicle, we calculate the average $\dot{m}_{gas}$ and P_{batt} per mile for different drive cycles (Table I) using a modified version of the Vehicle-Engine SIMulation (VESIM) model reported in [24] and references therein. In this method, we consider two driving modes for a PHEV: Charge-Depleting (CD) and Charge Sustaining (CS). The CD mode refers to the phase where the PHEV acts like an EV and consumes all of its propulsion energy from the battery pack. Once the State Of Charge (SOC) of the battery reaches a target value, it switches to the CS mode in which the vehicle starts using the internal combustion engine as the main propulsion system and the battery and electric motors are only used to improve the fuel economy as in HEVs [31].

TABLE I
DRIVE CYCLE ASSIGNMENT OF EACH LINK

Traffic Intensity on the link	Assigned Drive Cycle
Low Traffic	HWFET: Highway Fuel Economy Test
Medium Traffic	UDDS: Urban Dynamometer Driving Schedule
High Traffic	NYC: New York City

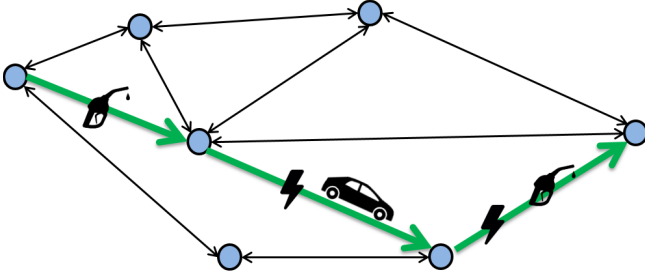


Fig. 1. Traffic network representation as a directed graph (blue dots represent intersections and black arcs denote road-links). In green the optimal eco-route is shown together with the optimal power-train control strategy for switching between charge depleting and charge sustaining modes (a hypothetical CRPTC eco-routing strategy).

Let us consider the traffic network as a directed graph (Fig. 1). Based on their traffic intensity we can categorize the links into three modes: low, medium, and high traffic intensity links, and assign different standard drive cycles to them [17] (Table I). For any target vehicle, we can then use a high-fidelity energy model to calculate the average electrical energy (μ_{CD}) and gas (μ_{CS}) used to drive one mile under CD and CS modes respectively under each of these drive cycles (Table II):

$$\mu_{CDij} = \frac{d_{ij}}{P_{battij}}, \quad \mu_{CSij} = \frac{d_{ij}}{m_{gasij}},$$

where d_{ij} is the length of link (i, j) . By knowing μ_{CDij} and μ_{CSij} on each link, as well as the network topology (length of each link), we can determine the average fuel consumption (m_{gasij}) and electrical power demand from the battery (P_{battij}) on each link (i, j) . We can then use (1) to calculate the total energy cost for each trip. In this paper we use VESIM as the high-fidelity energy model, and it is calibrated for a PHEV Audi A3 e-tron. Since each drive cycle includes acceleration and deceleration periods and the fact that VESIM has built-in regenerative braking capabilities, the resulting average energy consumption values have implicitly considered the effect of regenerative braking. In Section V we validate our results using a highly accurate energy model that has been calibrated for the Audi A3 e-tron considering all the power-train details including but not limited to the efficiency and BSFC maps. Note that this method can easily be extended to conventional vehicles, HEVs and EVs by just considering one of the operational modes.

III. SINGLE VEHICLE ECO-ROUTING

In view of the two driving modes (CD and CS), coming up with an eco-routing algorithm requires knowledge of how the PT controller switches between the two modes on each link a priori; alternatively, we can let the algorithm decide the PT control strategy while finding the optimal route.

TABLE II
ENERGY CONSUMPTION OF VARIOUS DRIVING CYCLES

Vehicle Type	Symbol	Unit	HWFET	UDDS	NYC
Audi A3	μ_{CD}	mi/kWh	4.14	4.39	3.14
	μ_{CS}	mi/gal	47.11	49.03	28.88

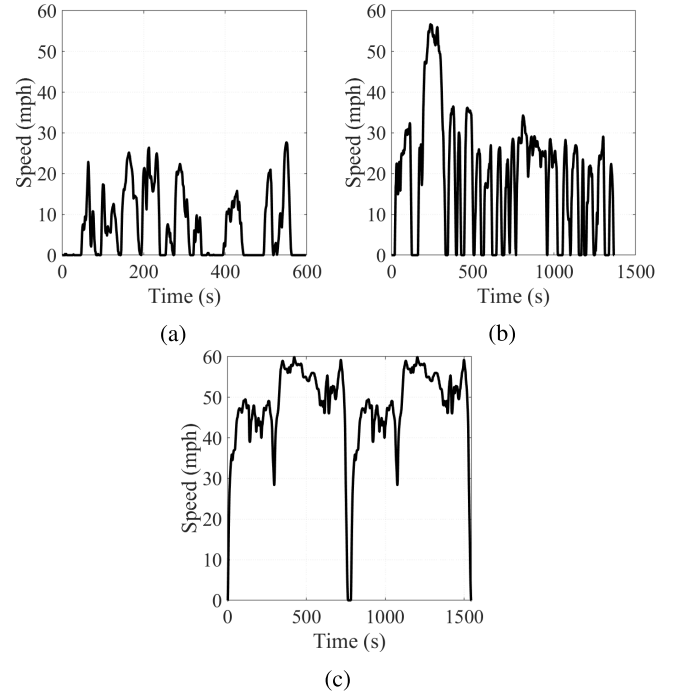


Fig. 2. (a) NYC drive cycle: high traffic links; (b) UDDS drive cycle: medium traffic links; (c) HWFET drive cycle: low traffic links.

In this section, we first review the *Charge Depleting First* (CDF) eco-routing approach [16], [17] and propose two methods to solve it. Next, we propose a *Combined Routing and Power-Train Control* (CRPTC) algorithm to solve the eco-routing problem.

A. Problem Formulation

We model the traffic network as a directed graph $G = (\mathcal{N}, \mathcal{A})$ with $\mathcal{N} = 1, \dots, n$ and $|\mathcal{A}| = m$ with the arc (link) connecting node i to j denoted by $(i, j) \in \mathcal{A}$. The set of nodes that are incoming/outgoing to node i are defined as: $\mathcal{I}(i) = \{j \in \mathcal{N} | (j, i) \in \mathcal{A}\}$ and $\mathcal{O}(i) = \{j \in \mathcal{N} | (i, j) \in \mathcal{A}\}$, respectively. We consider the single-origin-single-destination eco-routing problem where origin and destination nodes are denoted by o and d respectively. The energy cost associated with the vehicle on link (i, j) is denoted by C_{ij} . We use E_i to represent the vehicle's residual battery energy at node i . Moreover, we denote the selection of arc (i, j) by $x_{ij} \in \{0, 1\}$. The problem objective is to determine a path from node o to d so as to minimize the total energy cost consumed by the vehicle to reach the destination. We consider two approaches to solve this problem as follows.

B. Charge Depleting First (CDF)

In this approach [16], [17], we assume that the PHEV always starts every trip in the CD mode and uses electricity to

drive the vehicle until it drains all the energy out of the battery pack. Afterwards, it switches to the CS mode and starts using gas to drive the vehicle. Even though this approach is generally sub-optimal, it is motivated by the fact that it eliminates the need for complicated PHEV power-train control strategies to switch between ICE and electric motors [17]. As a result, we can formulate the eco-routing problem using a Mixed-Integer Nonlinear Programming (MINLP) framework as follows:

$$\min_{x_{ij}, (i,j) \in \mathcal{A}} \sum_{(i,j) \in \mathcal{A}} C_{ij} x_{ij} \quad (2a)$$

$$s.t. \quad C_{ij} = \begin{cases} C_{gas} \frac{d_{ij}}{\mu_{CS_{ij}}}; & E_i \leq 0 \\ C_{ele} \frac{d_{ij}}{\mu_{CD_{ij}}}; & E_i \geq P_{batt_{ij}} \\ C_{ele} E_i + C_{gas} \frac{d_{ij} - \mu_{CD_{ij}} E_i}{\mu_{CS_{ij}}}; & \text{otherwise} \end{cases} \quad (2b)$$

$$E_j = \sum_{i \in \mathcal{I}(j)} (E_i - \frac{d_{ij}}{\mu_{CD_{ij}}}) x_{ij}, \quad \forall j \in \mathcal{N} \quad (2c)$$

$$\sum_{i: (i,j) \in \mathcal{A}} x_{ij} + \mathbb{1}_{j=o} = \sum_{k: (j,k) \in \mathcal{A}} x_{jk} + \mathbb{1}_{j=d} \quad \forall j \in \mathcal{N} \quad (2d)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A} \quad (2e)$$

where C_{gas} and C_{ele} are the price of gas (\$/gallon) and electricity (\$/kWh) respectively, E_i is the remaining electrical energy at node i , and $P_{batt_{ij}} = \frac{d_{ij}}{\mu_{CD_{ij}}}$ is the electrical power needed to travel through link (i, j) . The conversion factors $\mu_{CD_{ij}}$ and $\mu_{CS_{ij}}$, taken from Table II, are functions of the traffic intensity on each link (i, j) . Note that (2d) is the flow conservation constraint [32], and $\mathbb{1}_{i=j}$ is a Boolean indicator function which equals to 1 when $i = j$ and 0 otherwise. We assume that the vehicle has enough gas and electrical power to complete the trip and that $E_o \geq 0$ (initial energy at the origin).

The nonlinearities in problem (2) arise from the dependency of C_{ij} in (2b) on the remaining energy in the battery, which means C_{ij} is route-dependent. If C_{ij} were a priori known, the problem would have been reduced to a shortest path problem and we could solve it using one of the highly studied algorithms such as Dijkstra's algorithm [21]. To address this issue, in what follows we propose two methods to solve this problem: CDF-Dijkstra and Hybrid-LP relaxation.

1) *Charge Depleting First (CDF) - CDF-Dijkstra*: Dijkstra's algorithm finds the shortest path between two points if the weight of each link is a known value. In (2) the energy cost of each link (2b) is a function of remaining energy in the battery pack. As a result, if we keep track of the remaining energy in the battery at the end node of each candidate path while searching the graph using Dijkstra's algorithm, the energy cost of links would be known to us and we can find the eco-route using a modified version of Dijkstra's algorithm. This modified version, which can find the CDF eco-route by keeping track of remaining energy in the battery, is given in Algorithm 1 (see Appendix).

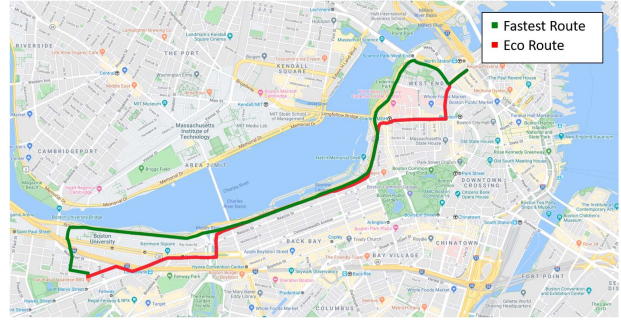


Fig. 3. Visualization of eco-route (CRPTC) and fastest route on a map.

2) *Charge Depleting First (CDF) - A Hybrid-LP Relaxation Approach*: We propose an alternative solution to this problem by reducing the MINLP problem (2) to a simpler one which can be solved using a combination of Linear Programming (LP) and a simple dynamic programming-like algorithm, in order to guarantee global convergence. The nonlinearities of the problem arise in (2b) where C_{ij} is a function of x_{ij} . We show that we can reduce this piece-wise constant function to a constant function, and the MINLP can be converted to a LP by using the properties of the minimum cost flow problem [33]. The proposed algorithm is as follows:

- 1) Find the shortest path and calculate the energy cost on this path using (2b) and set it to ρ . We are going to use ρ as a reference in the next step.
- 2) From the origin, construct all paths reaching node p such that $E_p \leq 0$ for the first time and stop constructing the path at this node. Disregard the paths with a total energy cost greater than ρ and save the remaining paths in a matrix.
- 3) Since $E_p \leq 0$, we can only use gas (CS mode) to travel through links which belong to the paths outgoing from node p .
- 4) Assuming knowledge of traffic modes on each link, the least energy cost path from node p in step 2 to the destination node can be found from:

$$\min_{x_{ij}, (i,j) \in \mathcal{A}} \sum_{(i,j) \in \mathcal{A}} C_{ij} x_{ij} \quad (3a)$$

$$s.t. \quad C_{ij} = C_{gas} \frac{d_{ij}}{\mu_{CS_{ij}}} \quad (3b)$$

$$\sum_{i: (i,j) \in \mathcal{A}} x_{ij} + \mathbb{1}_{j=p} = \sum_{k: (j,k) \in \mathcal{A}} x_{jk} + \mathbb{1}_{j=d} \quad \forall j \in \mathcal{N} \quad (3c)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A} \quad (3d)$$

Note that constraint (3c) ensures that by solving (3), we are finding the optimal path from p to d .

- 5) Using the property of the minimum cost flow problem [33], problem (3a) is equivalent to an LP problem with the integer restriction of x_{ij} relaxed: $0 \leq x_{ij} \leq 1$.
- 6) Find the path from node o to p with the least energy cost. By the principle of optimality, the optimal path from o to d is the one determined in this manner followed by the path selected by steps 4 and 5 from node p to d .

- 7) Find the paths in step 6 for all nodes p such that $E_p \leq 0$, then choose the one with the minimum energy cost. The selected path is the minimum energy cost path.
- 8) If there are paths without any node such that $E_p \leq 0$ (generated at step 2), compare their cost function values with the cost functions in step 6. The optimal route is the minimum among them.

C. Combined Routing and Power-Train Control (CRPTC)

Based on Table II, the CD mode has the best efficiency on medium traffic links. As such, if we always consider using the CD mode at the beginning of each trip and then switch to the CS mode when we run out of battery, we miss the opportunity to harness the effectiveness of the CD mode on medium traffic links towards the end of the route. With this motivation, we propose a new algorithm which finds the energy-optimal routing decisions as well as the PT controller decision to switch between CD and CS modes. Let $y_{ij} \in [0, 1]$ be an additional decision variable on link (i, j) which represents the fraction of the link's length over which we use the CD mode (thus, if we only use the CD mode over link (i, j) , then $y_{ij} = 1$). Considering the new decision variable, we can formulate the CRPTC problem as follows:

$$\min_{x_{ij}, y_{ij}, (i,j) \in \mathcal{A}} \sum_{(i,j) \in \mathcal{A}} [C_{gas} \frac{d_{ij}}{\mu_{CSij}} (1 - y_{ij}) + C_{ele} \frac{d_{ij}}{\mu_{CDij}} y_{ij}] x_{ij} \quad (4a)$$

$$s.t. \sum_{i:(i,j) \in \mathcal{A}} x_{ij} + \mathbb{1}_{j=o} = \sum_{k:(j,k) \in \mathcal{A}} x_{jk} + \mathbb{1}_{j=d} \quad \forall j \in \mathcal{N} \quad (4b)$$

$$\sum_{(i,j) \in \mathcal{A}} \frac{d_{ij}}{\mu_{CDij}} y_{ij} x_{ij} \leq E_o \quad (4c)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A} \quad (4d)$$

$$y_{ij} \in [0, 1] \quad \forall (i, j) \in \mathcal{A} \quad (4e)$$

Note that constraint (4c) ensures that the total electrical energy used in the CD mode would be less than the initial available energy in the battery (E_o). Since we have the term $x_{ij} y_{ij}$ in the problem formulation, this is a MINLP and we may not be able to determine a global optimum. Hence, we transform (4) into a MILP by introducing an intermediate decision variable $z_{ij} = x_{ij} y_{ij}$. We can then make use of the following inequalities to transform the existing MINLP problem into a MILP problem:

$$z_{ij} \geq 0, \quad z_{ij} \leq y_{ij}, \quad z_{ij} \geq y_{ij} - (1 - x_{ij}), \quad z_{ij} \leq x_{ij} \quad (5)$$

Considering z_{ij} and (5), we can reformulate problem (4) as follows:

$$\min_{x_{ij}, y_{ij}, z_{ij}, (i,j) \in \mathcal{A}} \sum_{(i,j) \in \mathcal{A}} (C_{gas} \frac{d_{ij}}{\mu_{CSij}} x_{ij} + (C_{ele} \frac{d_{ij}}{\mu_{CDij}} - C_{gas} \frac{d_{ij}}{\mu_{CSij}}) z_{ij}) \quad (6a)$$

$$s.t. \sum_{i:(i,j) \in \mathcal{A}} x_{ij} + \mathbb{1}_{j=o} = \sum_{k:(j,k) \in \mathcal{A}} x_{jk} + \mathbb{1}_{j=d} \quad \forall j \in \mathcal{N} \quad (6b)$$

$$\sum_{(i,j) \in \mathcal{A}} \frac{d_{ij}}{\mu_{CDij}} z_{ij} \leq E_1 \quad (6c)$$

$$z_{ij} \geq 0 \quad (6d)$$

$$z_{ij} \leq y_{ij} \quad (6e)$$

$$z_{ij} \geq y_{ij} - (1 - x_{ij}) \quad (6f)$$

$$z_{ij} \leq x_{ij} \quad (6g)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{N} \quad (6h)$$

$$y_{ij} \in [0, 1] \quad \forall (i, j) \in \mathcal{N} \quad (6i)$$

This is a MILP problem which can be solved to determine a global optimum.

D. Combined Routing and Power-Train Control: A Bi-Level Approach

Since MILP problems, such as the one above, are typically NP-hard, we now investigate a bi-level optimization approach in which the upper-level problem finds the energy-optimal route considering the CDF approach and the lower-level problem calculates the optimal PT control strategy by solving an LP problem. As a result, we can formulate the bi-level optimization problem as follows:

- 1) Using Algorithm 1, solve problem (2) and find the optimal routing decision vector $\mathbf{x}^* = [x_{ij}, (i, j) \in \mathcal{A}]$. Note that for finding the eco-route, we solve the CDF problem which is the baseline for PT control strategy.
- 2) Fix the routing decision vector $\mathbf{x}^*$ calculated in step 1 using CDF and find the optimal switching strategy $\mathbf{y}^* = [y_{ij}^*, (i, j) \in \mathcal{A}]$ by solving the following LP problem:

$$\min_{y_{ij}, (i,j) \in \mathcal{A}} \sum_{(i,j) \in \mathcal{A}} [C_{gas} \frac{d_{ij}}{\mu_{CSij}} (1 - y_{ij}) + C_{ele} \frac{d_{ij}}{\mu_{CDij}} y_{ij}] x_{ij} \quad (7a)$$

$$\sum_{(i,j) \in \mathcal{A}} \frac{d_{ij}}{\mu_{CDij}} y_{ij} x_{ij} \leq E_1 \quad (7b)$$

$$x_{ij} = x_{i,j}^* \quad \forall (i, j) \in \mathcal{A} \quad (7c)$$

$$y_{ij} \in [0, 1] \quad \forall (i, j) \in \mathcal{A} \quad (7d)$$

Note that constraint (7c) enforces routing variables to be equal to the ones calculated in step 1.

- 3) The optimal route is $\mathbf{x}^*$ calculated in step 1 and the optimal PT switching strategy is $\mathbf{y}^*$ found by solving problem (7).

Note that this solution is sub-optimal, but the computational time is orders of magnitudes faster than solving problem (6). This is due to the fact that the upper-level problem finds the optimal route using the computationally efficient CDF-Dijkstra algorithm, while the lower-level problem solves an LP on a small set of decision variables (only links selected by the router in step 1). We will further discuss the execution time of each algorithm in subsequent sections.

E. Eco-Routing With Time Consideration

There is typically a trade-off between energy savings and time savings in choosing between eco-routes and fastest routes

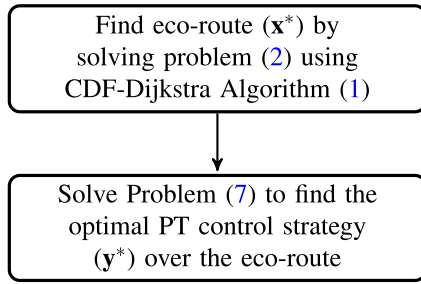


Fig. 4. Bi-level eco-routing structure.

and, as shown in [20], eco-routes can be up to 20% slower than fastest routes (Fig. 3). As a result, in order to consider a balance between time and energy we can introduce a time component into the objective function of our eco-routing problem. Considering time in the eco-routing problem, we can re-write problem (2) as follows:

$$\begin{aligned} \min_{x_{ij}, (i,j) \in \mathcal{A}} \quad & \sum_{(i,j) \in \mathcal{A}} \left(\alpha \frac{t_{ij}}{\beta_1} + (1 - \alpha) \frac{C_{ij}}{\beta_2} \right) x_{ij} \\ \text{s.t.} \quad & (2b) - (2e) \end{aligned} \quad (8)$$

where $\alpha \in [0, 1]$ is a time-to-energy weighting factor, and $\beta_1 > 0$ and $\beta_2 > 0$ are normalization factors for time and energy respectively. Note that if we select $\beta_1 = t_{ij}^{max}$ and $\beta_2 = C_{ij}^{max}$, then the two terms are ensured to be in $[0, 1]$. The two max constants are upper bounds selected based on the topology of the network and the pricing structure. We can use the historical traffic information on each link and based on that calculate the maximum/upper bound of travel time on that link and also estimate the maximum energy consumption on the link. We can use the same analogy and modify the CRPTC problem (6) to include time as follows:

$$\begin{aligned} \min_{x_{ij}, y_{ij}, z_{ij}, (i,j) \in \mathcal{A}} \quad & \sum_{(i,j) \in \mathcal{A}} \left(\frac{1 - \alpha}{\beta_2} \right) \left(C_{gas} \frac{d_{ij}}{\mu C S_{ij}} x_{ij} \right. \\ & \left. + \left(C_{ele} \frac{d_{ij}}{\mu C D_{ij}} - C_{gas} \frac{d_{ij}}{\mu C S_{ij}} \right) z_{ij} \right) + \frac{\alpha}{\beta_1} t_{ij} x_{ij} \\ \text{s.t.} \quad & (6b) - (6i) \end{aligned} \quad (9)$$

IV. NUMERICAL RESULTS

An eco-routing case study was presented in [20] using traffic data from the Eastern Massachusetts highway sub-network collected by *INRIX* [34]. In this paper, we extend our analysis to a larger network which includes the entire Boston urban area.

A. Traffic Data Platform

Using a HERE Maps API, we developed a web-based tool in which we can request and download the geographical map of a region alongside its traffic information (free flow and average speeds) (http://www.bu.edu/codes/simulations/traffic_downloader/). The traffic data includes average speeds of all roads for every 15 minutes of a typical week and their free flow speeds. Moreover, we have topological data of the selected region including how the links

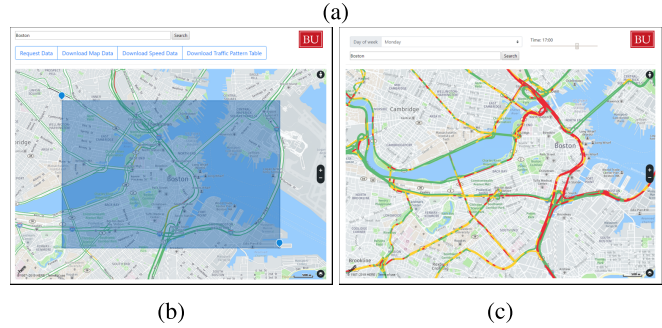
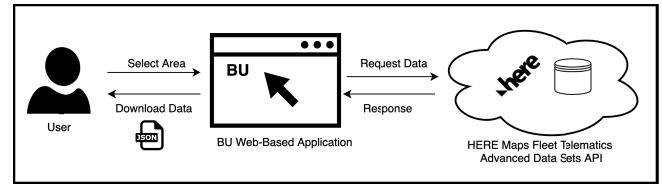


Fig. 5. (a) BU traffic data platform architecture; (b) BU traffic data platform user interface: Downloading tool; (c) Platform user interface: Average speed visualization.

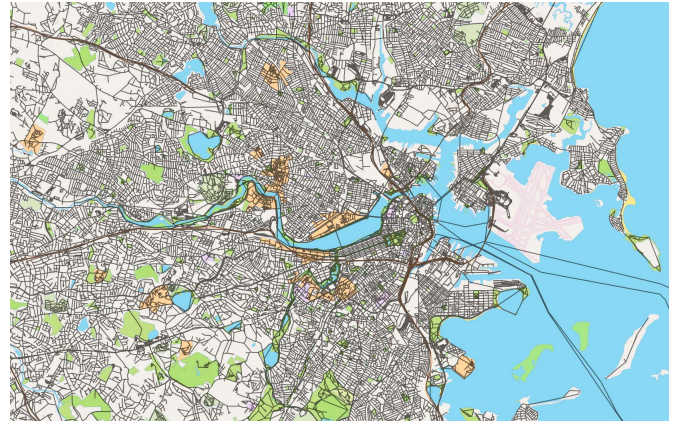


Fig. 6. Available links in our case study network of Boston.

are connected to each other, positions of nodes and links, link lengths, road grade, etc. The structure of this platform is shown in Fig. 5a.

B. Data Preprocessing

Using this platform, we downloaded traffic data of the Boston urban area (Fig. 6) which includes more than 110,000 links and 50,000 nodes. Since the energy model in our eco-routing algorithm depends on the traffic intensity of each link (Table II), we need to first categorize links in terms of traffic intensity: low, medium, high. Using the average speed of every link and its corresponding free flow speed data, we introduce a new variable called *speed factor* (S_{ij}) for each link (i, j) as follows:

$$S_{ij} = \frac{\bar{v}_{ij}}{f_{ij}}$$

where $\bar{v}_{ij}$ and f_{ij} are the average speed and free flow speed respectively of link (i, j) . Note that the speed factor is a normalized speed value on each link indicating how congested the link is. As a result, we can use S_{ij} to categorize links into three modes: (i) If $S_{ij} \leq 0.5$, link (i, j) is categorized as

a high traffic intensity link and values of the NYC drive cycle are assigned to it (Table II), (ii) If $0.5 < S_{ij} < 0.75$, link (i, j) is categorized as a medium traffic intensity link and values of UDDS are assigned to it, (iii) If $S_{ij} \geq 0.75$, link (i, j) is categorized as a low traffic intensity link and values of HWFET are assigned to it.

C. Performance Measurement Baseline

In order to measure the performance of our eco-routing algorithms, we consider the time-optimal path (fastest-route) as the baseline. In this respect, for each O-D pair we find both the eco-route and fastest route and then compare the energy cost of travelling through both. The fastest route can be determined by solving the following problem:

$$\min_{x_{ij}, (i,j) \in \mathcal{A}} \sum_{(i,j) \in \mathcal{A}} t_{ij} x_{ij} \quad (10a)$$

$$s.t. \quad (2d) \quad (10b)$$

$$t_{ij} = \frac{d_{ij}}{v_{ij}} \quad (10c)$$

$$x_{ij} \in \{0, 1\} \quad (10d)$$

where t_{ij} and d_{ij} are the traveling time and length of link (i, j) respectively. In order to calculate the energy cost of travelling through the fastest route, we consider the CDF policy for the car and calculate its energy cost using (2b).

D. Energy Cost and Travel Time Comparison Results

We use the urban area of Boston (Fig. 6) as our case-study network. In order to show the impact of traffic intensity and distance between O-D pairs on the performance of the eco-routing algorithms, we randomly select 100 O-D pairs in this network and calculate eco-routes (CDF, CRPTC, and Bi-level) as well as the fastest route between each of these O-D pairs over different hours of a day (8:00 am, 12:00 pm, 3:00 pm, 5:00 pm, and 9:00 pm). As in [17], we assume that the initial available energy in the battery is $E_{ini} = 5.57kWh$ and the cost of gas and electricity are $C_{gas} = 2.75\$/gal$ and $C_{ele} = 0.114\$/kWh$ respectively [31]. Since the amount of allowable electrical energy depletion ΔE affects the efficiency of eco-routing algorithms, five different values for ΔE are selected: $0kWh$, $0.5kWh$, $1kWh$, $2.5kWh$, and $5.7kWh$. The average energy and time comparison plots over the selected O-D pairs for different allowable ΔE values are shown in Figs. 7 and 8 respectively.

We have also compared the energy cost of different eco-routing algorithms against the fastest route and compared them to each other showing their energy saving distributions in Fig. 9 as box-plots where red line is the median and green triangle is the mean. Note that in these plots we are not showing the outlier data points.

As we can see in Figs. 7 and 8, there is a trade-off between energy and time while travelling through the eco-route and fastest route. To better quantify this trade-off, we compare the travelling time of each eco-routing algorithm (CRPTC, CDF, and bi-level) against that of the fastest route and show their box-plot distribution in Fig. 10. Note that the eco-route that is

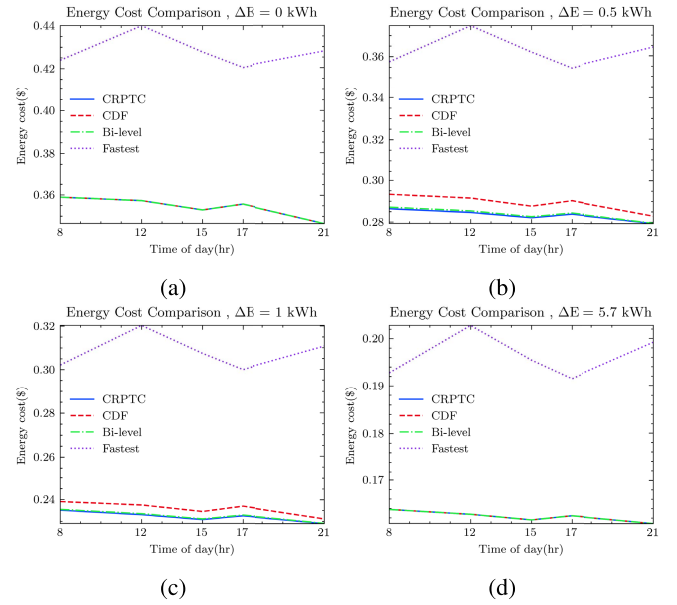


Fig. 7. Average energy cost comparison for the selected O-D pairs with different allowable ΔE values.

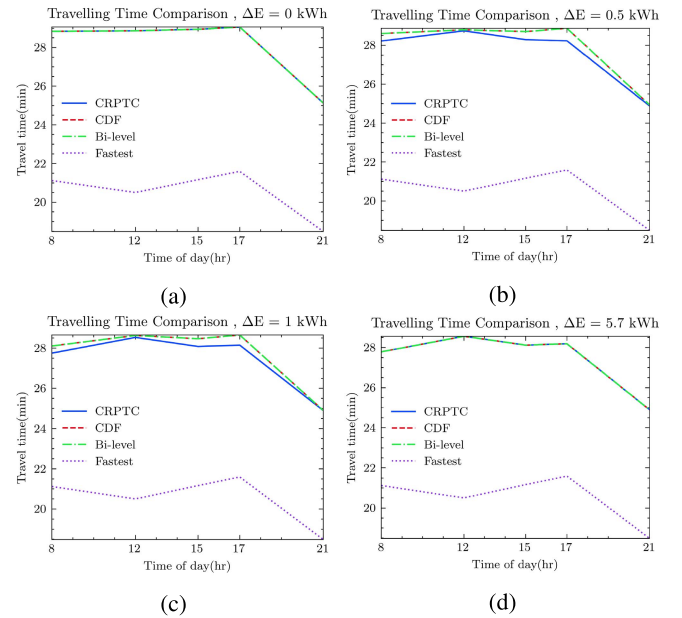


Fig. 8. Average travel time comparison for the selected O-D pairs with different allowable ΔE values.

chosen by the CDF algorithm is the same as that of the bi-level approach; as a result the traveling time results of these two approaches are the same.

To show the interdependence of eco-routing performance and the distance between O-D pairs, we organized O-D pairs based on the shortest distance between them and reported the average energy saving and travel time saving values based on the distance between origin and destination in Figs. 11 and 12.

In summary, the average energy and travel time savings of different routing strategies are shown in Tables III and IV respectively. It can be seen that the CRPTC algorithm always

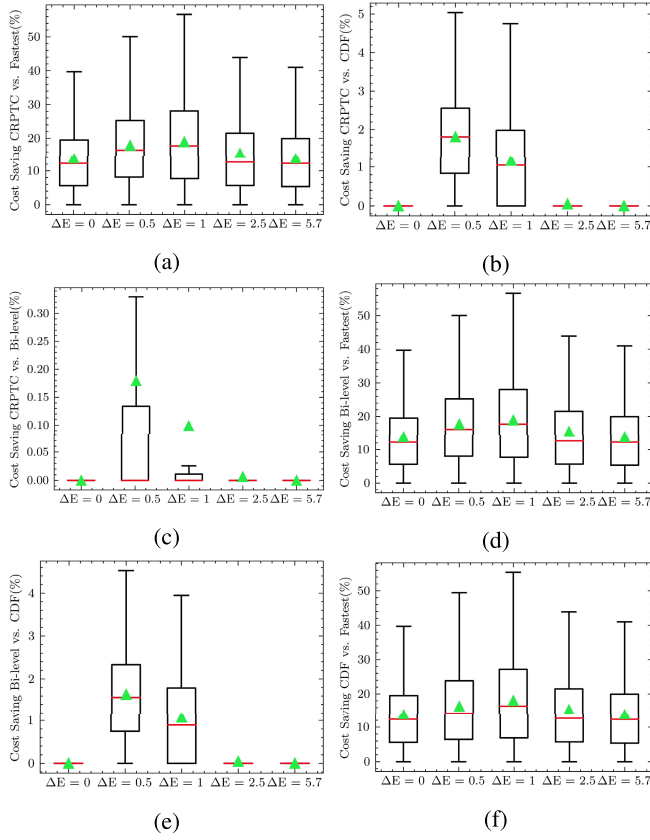


Fig. 9. Average energy cost saving distribution for the selected O-D pairs.

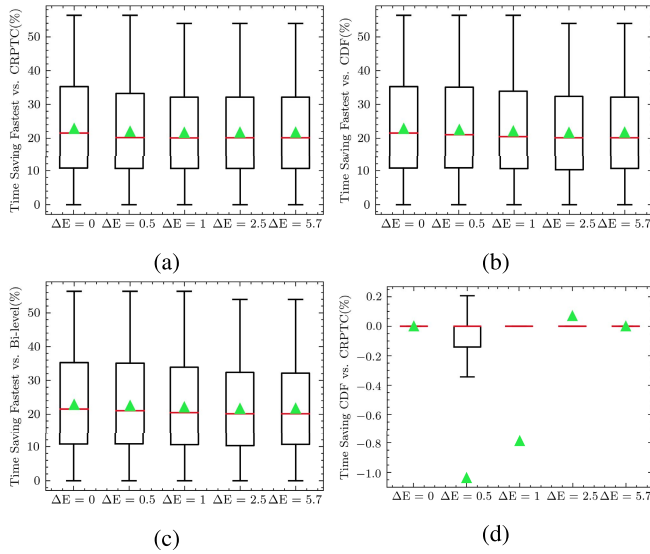


Fig. 10. Average travel time saving distribution for the selected O-D pairs.

has the best performance with energy savings of up to 19% compared to the fastest route. The bi-level eco-routing approach is within 0.2% of the CRPTC approach; even though it does not provide the global optimal solution, its much faster execution time makes it particularly attractive. Note that when the car has enough energy to travel the entire path with electricity, all the eco-routing algorithms give us the same results. This is because in this case the optimal PT control strategy is

TABLE III
AVERAGE ENERGY COST SAVINGS OF THE PROPOSED ECO-ROUTING ALGORITHMS (ΔE VALUES ARE IN kWh)

	Energy Cost Saving (%)					
	CRPTC	Bi-level	CDF	CRPTC	CRPTC	Bi-level
	vs.	vs.	vs.	vs.	vs.	vs.
	Fastest	Fastest	Fastest	CDF	Bilevel	CDF
$\Delta E = 0$	13.8	13.8	13.8	0	0	0
$\Delta E = 0.5$	17.9	17.7	16.4	1.8	0.2	1.6
$\Delta E = 1$	19	18.9	18.1	1.2	0.1	1.1
$\Delta E = 2.5$	15.4	15.4	15.4	0.1	0	0
$\Delta E = 5.7$	13.8	13.8	13.8	0	0	0

TABLE IV
AVERAGE TRAVEL TIME SAVINGS RESULTS UNDER DIFFERENT ROUTING SCENARIOS. (E.G. AT $\Delta E = 0$ FASTEST ROUTE SAVES 22.8% IN TERMS OF TRAVEL TIME COMPARED TO THE CRPTC ROUTE.)
NOTE THAT ΔE VALUES ARE IN kWh

	Travel Time Saving (%)				
	Fastest	Fastest	Fastest	CRPTC	CRPTC
	vs.	vs.	vs.	vs.	vs.
	CRPTC	CDF	Bi-level	CDF	Bi-level
$\Delta E = 0$	22.8	22.8	22.8	0	0
$\Delta E = 0.5$	22	22.5	22.5	1	1
$\Delta E = 1$	21.6	22.1	22.1	0.8	0.8
$\Delta E = 2.5$	21.7	21.6	21.6	-0.1	-0.1
$\Delta E = 5.7$	21.7	21.7	21.7	0	0

to use electricity (CD mode) on all links since it is cheaper than using gas; as a result, CDF and CRPTC give us the same solutions, as does the bi-level optimization approach. By the same token, when we do not let the car deplete electricity along the route ($\Delta E = 0$), the optimal solutions of CRPTC, CDF, and bi-level become the same, since the car can only use gas (CS mode) to travel through the links. In general, the CRPTC eco-routing algorithm has the best performance when the distance between origin and destination is relatively high (> 30 miles), since in those cases we have more options for choosing when and where to use the CD mode on the path and solving the combined problem can give us better results. When we solved the eco-routing problem for the Eastern Massachusetts highway sub-network [20] the distances were more than 30 miles with the CRPTC approach outperforming CDF by an average of more than 2.1%. As expected, there is a trade-off between energy and time savings, and time-optimal routes can be more than 20% faster than the energy-optimal routes. In order to consider time when solving eco-routing problems, we can solve problem (9) or (8) with different α values to establish a desired balance between time and energy.

E. Algorithm Execution Time Comparison Results

An important factor in assessing the performance of eco-routing algorithms, aside from their energy improvement, is their execution time (runtime). It is essential that an algorithm can compute the eco-route quickly and is able to re-calculate the energy-optimal route in case of sudden changes in traffic patterns in the network. As a result,

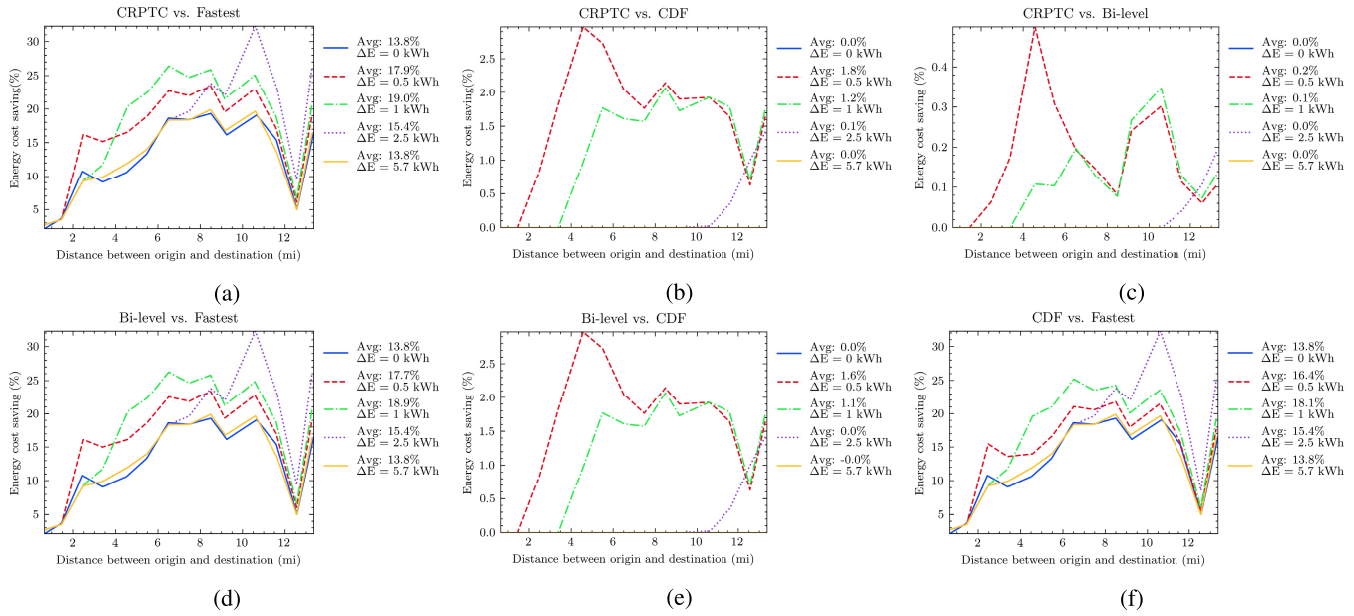


Fig. 11. Average energy cost savings for the selected O-D pairs based on the shortest distance between them.

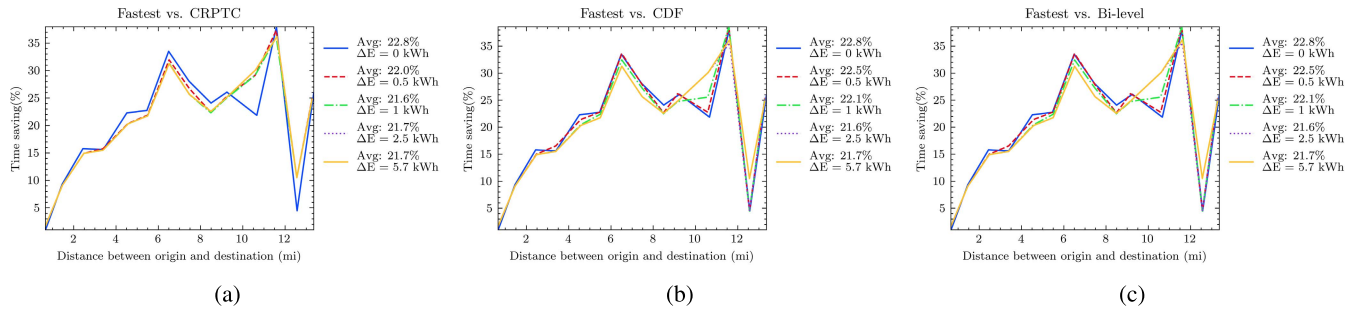


Fig. 12. Average travel time savings for the selected O-D pairs based on the shortest distance between them.

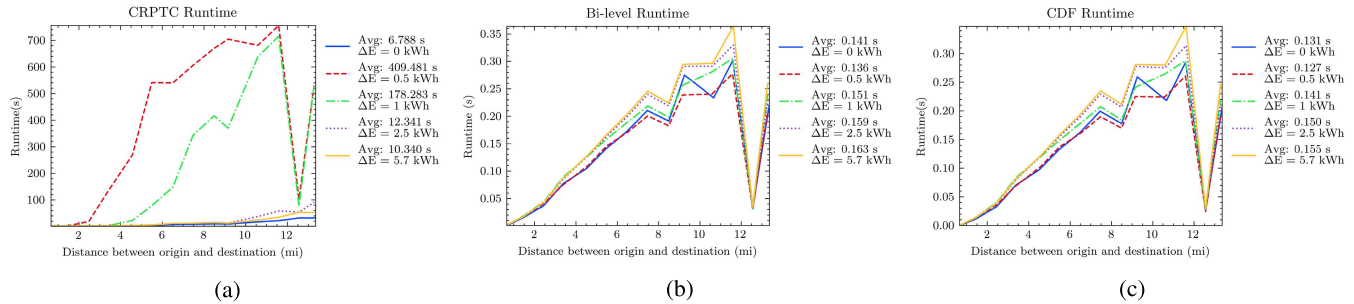


Fig. 13. Average runtime distribution of eco-routing algorithms based on the shortest distance between O-D pairs.

we calculated the execution time of the three proposed eco-routing algorithms and reported the corresponding averages in Table V. We have also included the runtime distribution box-plot of each algorithm in Fig. 14 and showed the execution time dependency on the distance between O-D pairs in Fig. 13. All these algorithms have been coded in Python 3.7.1 and executed on a desktop computer with a 4.2GHZ Core i7 CPU and 16 GB of RAM. We used Gurobi [35] as the MILP solver in this setting. As we can see in Table V, the CRPTC algorithm runtime is on average 10.34s when the battery is full ($\Delta E = 5.7kWh$). This is an interesting observation since

CRPTC is a MILP problem which is NP-hard. However, as we decrease the allowable energy depletion from the battery, we see that the runtime of the CRPTC algorithm starts to increase. This is due to the fact that we are imposing a tighter constraint (6c) to problem (6) which forces the optimizer to explore more options in seeking the optimal solution. Recalling that the CRPTC problem has two sets of decision variables, the routing decision vector $\mathbf{x}$ and PT control strategy vector $\mathbf{y}$, when the car has enough energy to travel the entire route with electricity, the optimal PT control strategy is to set $y_{ij} = 1$ for all $(i, j) \in \mathcal{A}$, and the optimizer can find

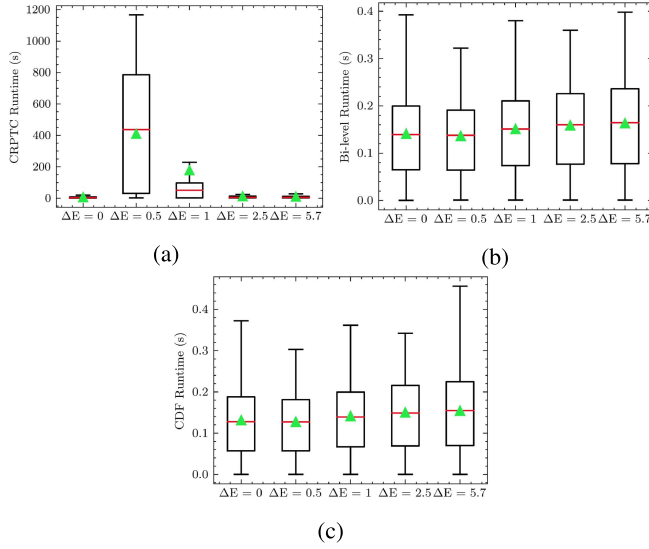


Fig. 14. Average runtime distribution of eco-routing algorithms.

TABLE V
ECO-ROUTING ALGORITHMS RUNTIME COMPARISON

$\Delta E (kWh)$	Runtime (s)		
	CRPTC	Bi-level	CDF
$\Delta E = 0$	6.79	0.14	0.13
$\Delta E = 0.5$	409.48	0.14	0.13
$\Delta E = 1$	178.28	0.15	0.14
$\Delta E = 2.5$	12.34	0.16	0.15
$\Delta E = 5.7$	10.34	0.16	0.15

the optimal solution easily. As we decrease the allowable ΔE value, we increase the search space for the optimization problem, consequently the runtime increases.

As expected, both CDF and bi-level eco-routing algorithms have near real-time execution times ($\sim 150ms$). This is due to the fact that both of these algorithms use a modified version of Dijkstra's algorithm to find the energy-optimal paths (Algorithm 1) which has a complexity of $O(n \log n)$ where n is the number of nodes in the graph.

Another interesting observation is that as the shortest distance between O-D pairs increases, the runtime of the eco-routing algorithms typically increases (Fig. 13). In particular, CRPTC's runtime is the most sensitive to the distance between O-D pairs and the runtime can increase more than 700 times as the distance between O-D pairs increases. Considering the difference between the runtimes of CRPTC and bi-level, while their respective performance is virtually indistinguishable (Table III), the use of the bi-level eco-routing algorithm is practically attractive in urban settings where the O-D pairs are relatively close to each other.

V. VALIDATION USING SIMULATION MODELS

Throughout this paper we have used a simplified energy model to estimate the energy consumption of PHEVs only based on the traffic intensity on each road-link. In this section, we investigate the accuracy of this energy model by using the



Fig. 15. The Audi A3 e-tron which was used in this study.

traffic simulator Simulation of Urban MObility (SUMO) [23] along with a modified version of the Vehicle-Engine SIMulation (VESIM) model [24] which is a high fidelity energy modelling software tool calibrated for the Audi A3 e-tron in Simulink. We use an Audi A3 e-tron since our proposed algorithms have been extensively tested on this vehicle (Fig. 15) at the University of Michigan's M-City and also using Chassis Dyno by Bosch. We start by briefly reviewing our simulation modeling frameworks in SUMO and VESIM, and then explaining how using them allows us to validate the accuracy of our eco-routing algorithm.

SUMO: We use the SUMO to evaluate the performance of the eco-routing algorithm. SUMO is an open source traffic simulation package which can generate speed trajectories of each individual vehicle. In order to have realistic traffic scenarios, we use the calibrated SUMO model for the Ann Arbor network which we previously built in [36]. To briefly summarize, the Ann Arbor traffic model consists of 11,265 road segments (links) and 8,660 traffic junctions. There are 327 traffic lights and 11,857 stop signs, all of which are in accordance with real-world information. The travel demands used in the SUMO model are generated according to a calibrated POLARIS model [37] which is an agent-based mesoscopic traffic simulation package developed by the Argonne National Lab. The focus of the POLARIS model is on realistic generation of travel demands based on travel activities of individual agents within each household using ADAPTS (Agent-based Dynamic Activity Planning and Travel Scheduling) [38]. The travel demands are calibrated with the dataset from the Safety Pilot Model Deployment (SPMD) [39] with records of 321,945 trips in Ann Arbor between 2013 and 2015 with 2,800 passenger cars, trucks, and buses. The route choices of individual vehicles in SUMO are then calibrated again to ensure that the average speed of each road in the simulation match the observed average speeds from the SPMD dataset.

VESIM: This is a Simulink based power-train modeling framework to calculate energy costs for any given speed profile. Our VESIM model is calibrated for an Audi A3 e-tron which is a PHEV, and the engine and electric motor efficiency maps are modified to match that of the Audi. A schematic of the VESIM model in Simulink is shown in Fig. 17.

A. Eco-Routing Performance Validation

We consider the time-optimal path as the baseline for measuring the performance of eco-routing algorithms. In order

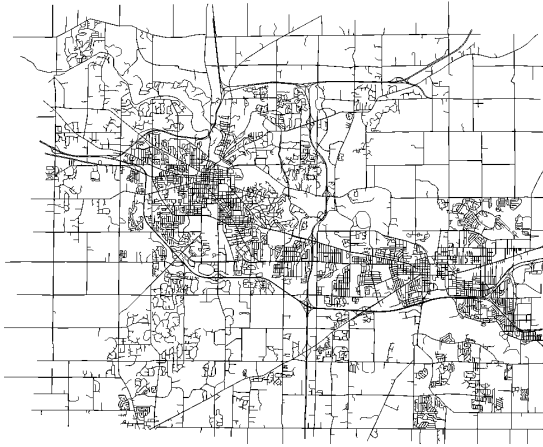


Fig. 16. Ann Arbor's traffic network in SUMO.

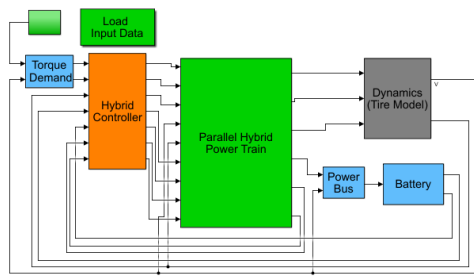


Fig. 17. VESIM model for modelling the power-train of the plug-in hybrid electric Audi A3 e-tron [40].

to validate the effectiveness of our eco-routing algorithm we need to compare the energy cost of travelling through the time-optimal path to that of the energy-optimal path. Since it is very difficult and costly to conduct such experiments in a real-world setting, we use computer simulation software to perform this task. The procedure of using SUMO and VESIM to validate the energy saving results is as follows:

- 1) Use the calibrated SUMO model of Ann Arbor to simulate traffic in the network.
- 2) Choose an O-D pair and find the eco-route and fastest route between the selected origin and destination.
- 3) Send two vehicles in SUMO to follow the fastest route and eco-route and collect their respective speed trajectories.
- 4) Import the speed trajectories from SUMO to VESIM and calculate the energy costs of both the fastest route and the eco-route.
- 5) Compare the energy costs of the eco-route and fastest route and report the energy savings.

Since our CRPTC eco-routing algorithm (6) simultaneously finds both the energy-optimal route and the optimal switching strategy between CD and CS modes, accurate energy results require us to consider the optimal PT control strategy commands from CRPTC while finding energy values using VESIM. So far, we have not incorporated the CRPTC control decisions for the PT controller into VESIM; as a result, to simplify this process, we consider a hybrid electric vehicle (HEV)

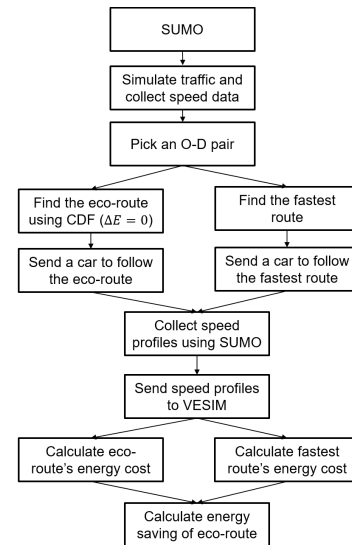


Fig. 18. Procedure for calculating energy costs using SUMO and VESIM.

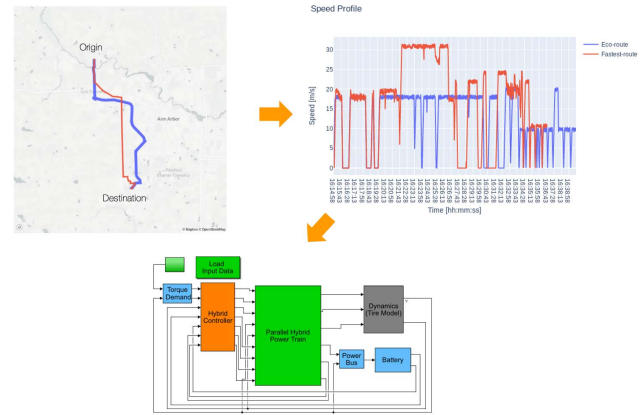


Fig. 19. Procedure of collecting speed profiles from SUMO and sending them to VESIM to validate the performance of the eco-routing algorithm.

instead of a PHEV, which only operates in CS mode (setting $\Delta E = 0$ while solving the eco-routing problem). Hence, we use Algorithm (1) with $E_0 = 0$ to find the eco-route, and in VESIM we choose the CS mode and let the SOC change $\pm 5\%$ throughout the route (Figs. 18 and 19).

1) Results Validation: Using the aforementioned framework (Fig. 18) we randomly select 2200 O-D pairs in the Ann Arbor SUMO network and calculate eco-routes and fastest routes for each of these pairs. We then import the collected speed profiles from SUMO into VESIM and calculate the energy costs savings. We repeat this procedure under two different traffic conditions in the SUMO network: medium and high traffic. The results for the medium traffic network are summarized as follows:

- The average actual energy saving of eco-routes vs. fastest routes calculated by VESIM over the selected O-D pairs is **12.59%**.
- The average predicted energy saving using our simplified energy model (2b) is **12.52%**.

TABLE VI

VESIM ECO-ROUTING VALIDATION RESULTS FOR THE 2200 O-D PAIRS
SELECTED IN SUMO UNDER MEDIUM AND
HIGH TRAFFIC NETWORK TRAFFIC

	Actual average energy savings (VESIM Results)	True Positive rate	True Positive energy savings	False Positive rate	False Positive energy saving
Medium Traffic	12.59%	80.38%	18.46%	19.62%	-11.46%
High Traffic	5.35%	69.00%	17.47%	31.00%	-21.64%

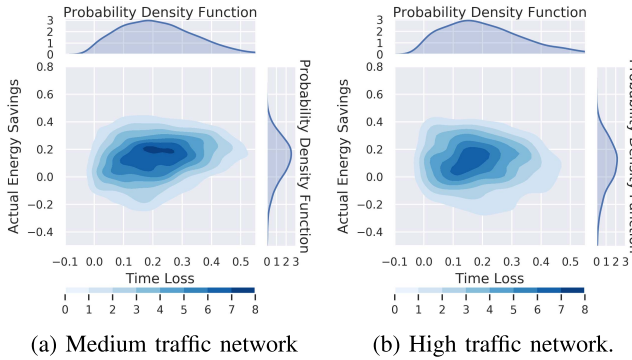


Fig. 20. SUMO + VESIM validation results: Trade-off between energy and time saving in eco-routes (colors represent the probability density values).

- In 80.38% of cases, we correctly predicted the energy optimal routes and the average savings of these cases is 18.46%.
- In 19.62% of cases, the predicted eco-routes consumed more energy than the fastest route, which we refer to as “false positives”. The average energy loss of false positive cases is -11.46%.

As we can see in the medium traffic network, the expected actual energy saving of eco-route vs. fastest route is almost the same as the predicted energy saving using our simplified energy model. However, the energy model does not always predict the eco-route correctly and sometimes finds a route that in reality consumes more energy than the fastest route (Table VI). One of the reasons is that we classified links based on their traffic intensity into only three categories: low, medium, and high traffic intensity links. We also assign standard speed profiles (Tables I and II) to each of these links and use their average energy consumption as the predicted energy consumption over the link. In order to increase the accuracy of the energy model, we can increase the number of link categories and assign more suitable speed profiles to each link based on different factors such as free flow speed, location of the link (urban/highway), road grade, traffic lights/stop signs, etc. Moreover, we update the average speed data in SUMO every 15 minutes. However, since the Ann Arbor network includes many traffic lights and there is often high traffic in the network, traffic conditions may significantly deviate within 15 minutes. This behavior is more evident in the high traffic network case (Table VI). As a result, we may need to extend our analysis to dynamic re-routing whenever traffic changes occur in the network, expecting to improve the accuracy of our results.

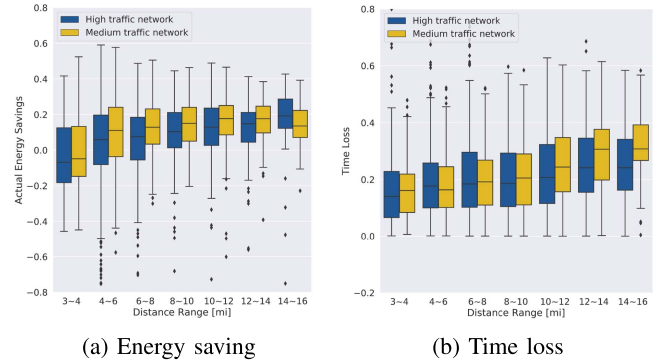


Fig. 21. SUMO + VESIM validation results: Energy saving and time loss of eco-route vs. fastest route as a function of distance between O-D pairs.

The relationship between energy saving and time loss is shown in Fig. 20 where the colored contours show the probability density values. We use kernel density estimation to estimate the probability density function. A Kernel density estimator can be viewed as a special case of the Gaussian mixture model with the weight of each component set to n^{-1} . Moreover, Fig. 21 shows the distribution of energy savings of the eco-route and time savings of the fastest route as a function of the shortest distance between O-D pairs.

VI. CONCLUSION AND FUTURE WORK

We have proposed two methods to solve the minimum-energy cost problem for a single vehicle routing: the CRPTC and bi-level optimization algorithms. The proposed methods are capable of finding both the optimal path and the optimal switching strategy between CD and CS modes on each link, and can be implemented in real time. An open source framework for downloading traffic data was also developed for this work and was used to implement the eco-routing algorithms on a large urban transportation network. The performance of the eco-routing algorithm was validated by using SUMO and VESIM and the results show energy savings of more than 12%. We have also shown that there is a trade-off between energy saving and time saving. It is worth mentioning that our eco-routing algorithms can be applied to any other PHEV, since car manufacturers have access to accurate energy models for their vehicles (such as VESIM) and can use them to calculate energy consumption conversion factors (μ_{CD} and μ_{CS}).

So far, we have not considered dynamically updating routing decisions at network nodes to account for sudden changes in traffic conditions (e.g., due to accidents). We have also limited our analysis to a single vehicle scenario with a known origin and destination. As a next step, we will consider connectivity among vehicles and determine the social optimum for the network considering 100% penetration rate of connected automated vehicle [41]. Moreover, we plan to include multiple vehicle architectures with different fuel consumption models and different initial energies to the problem, as well as adding charging stations into the network to let vehicles recharge their batteries if necessary.

APPENDIX A

See Algorithm 1.

Algorithm 1 CDF-Dijkstra

```

procedure CDF( $G$ , origin, destination,  $E_0$ )
  create node set  $Q$ 
  for all  $v \in \mathcal{N}$  do
    cost[origin]  $\leftarrow \infty$ 
    prev[v]  $\leftarrow$  UNDEFINED
    add  $v$  to  $Q$ 
  end for
  cost[origin]  $\leftarrow 0$ 
   $E[\text{origin}] \leftarrow E_0$ 
  while  $Q$  is not empty do
     $u \leftarrow$  vertex in  $Q$  with min cost[ $u$ ]
    remove  $u$  from  $Q$ 
    if  $u = \text{destination}$  then
       $S \leftarrow$  empty sequence
       $u \leftarrow \text{destination}$ 
      if prev[ $u$ ] is defined or  $u = \text{origin}$  then
        while  $u$  is defined do
          insert  $u$  at the beginning of  $S$ 
           $u \leftarrow \text{prev}[u]$ 
        end while
      end if
      break the while loop
    end if
    for all neighbor  $v$  of  $u$  do
      if energy[ $u$ ]  $\geq \frac{\text{dist}[u,v]}{\mu_{CD}[u,v]}$  then
        cost[ $u, v$ ]  $\leftarrow C_{ele} \frac{\text{dist}[u,v]}{\mu_{CD}[u,v]}$ 
         $E_{temp} \leftarrow E[u] - \frac{\text{dist}[u,v]}{\mu_{CD}[u,v]}$ 
      else
        cost[ $u, v$ ]  $\leftarrow C_{gas} \frac{\text{dist}[u,v] - \mu_{CD}[u,v]E[u]}{\mu_{CS}[u,v]} + C_{ele}E[u]$ 
         $E_{temp} \leftarrow 0$ 
      end if
      alt  $\leftarrow \text{cost}[u] + \text{cost}(u, v)$ 
      if alt  $< \text{cost}[v]$  then
        cost[ $v$ ]  $\leftarrow$  alt
         $E[v] \leftarrow E_{temp}$ 
        prev[ $v$ ]  $\leftarrow u$ 
      end if
    end for
  end while
  return cost[ ],  $S$  [ ]
end procedure

```

REFERENCES

- [1] D. V. Collia, J. Sharp, and L. Giesbrecht, "The 2001 national household travel survey: A look into the travel patterns of older Americans," *J. Saf. Res.*, vol. 34, no. 4, pp. 461–470, Jan. 2003.
- [2] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1, no. 2. Belmont, MA, USA: Athena Scientific, 1995.
- [3] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuysse, "The vehicle routing problem: State of the art classification and review," *Comput. Ind. Eng.*, vol. 99, pp. 300–313, Sep. 2016.
- [4] P. Toth and D. Vigo, *The Vehicle Routing Problem*. Philadelphia, PA, USA: SIAM, 2002.
- [5] M. Barth, K. Boriboonsomsin, and A. Vu, "Environmentally-friendly navigation," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2007, pp. 684–689.
- [6] K. Boriboonsomsin, M. J. Barth, W. Zhu, and A. Vu, "Eco-routing navigation system based on multisource historical and real-time traffic information," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1694–1704, Dec. 2012.
- [7] O. Andersen, C. S. Jensen, K. Torp, and B. Yang, "Ecotour: Reducing the environmental footprint of vehicles using eco-routes," in *Proc. IEEE 14th Int. Conf. Mobile Data Manage. (ICMD)*, vol. 1, Jun. 2013, pp. 338–340.
- [8] E. Yao and Y. Song, "Study on eco-route planning algorithm and environmental impact assessment," *J. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 42–53, 2013.
- [9] B. Yang, C. Guo, C. S. Jensen, M. Kaul, and S. Shang, "Stochastic skyline route planning under time-varying uncertainty," in *Proc. IEEE 30th Int. Conf. Data Eng.*, Mar. 2014, pp. 136–147.
- [10] A. Sciarretta, G. De Nunzio, and L. L. Ojeda, "Optimal ecodriving control: Energy-efficient driving of road vehicles as an optimal control problem," *IEEE Control Syst.*, vol. 35, no. 5, pp. 71–90, Oct. 2015.
- [11] M. Kubicka *et al.*, "Performance of current eco-routing methods," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2016, pp. 472–477.
- [12] S. Pourazarm, C. G. Cassandras, and T. Wang, "Optimal routing and charging of energy-limited vehicles in traffic networks," *Int. J. Robust Nonlinear Control*, vol. 26, no. 6, pp. 1325–1350, Apr. 2016.
- [13] G. De Nunzio, L. Thibault, and A. Sciarretta, "Model-based eco-routing strategy for electric vehicles in large urban networks," in *Comprehensive Energy Management—Eco Routing & Velocity Profiles*. New York, NY, USA: Springer, 2017, pp. 81–99.
- [14] J. Guanetti, Y. Kim, and F. Borrelli, "Control of connected and automated vehicles: State of the art and future challenges," *Annu. Rev. Control*, vol. 45, pp. 18–40, May 2018.
- [15] A. Cela *et al.*, "Energy optimal real-time navigation system," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 3, pp. 66–79, Oct. 2014.
- [16] Z. Sun and X. Zhou, "To save money or to save time: Intelligent routing design for plug-in hybrid electric vehicle," *Transp. Res. D, Transp. Environ.*, vol. 43, pp. 238–250, Mar. 2016.
- [17] Z. Qiao and O. Karabasoglu, "Vehicle powertrain connected route optimization for conventional, hybrid and plug-in electric vehicles," 2016, *arXiv:1612.01243*. [Online]. Available: <http://arxiv.org/abs/1612.01243>
- [18] G. De Nunzio, A. Sciarretta, I. Ben Gharbia, and L. L. Ojeda, "A constrained eco-routing strategy for hybrid electric vehicles based on semi-analytical energy management," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 355–361.
- [19] M. Salazar, A. Houshmand, C. G. Cassandras, and M. Pavone, "Optimal routing and energy management strategies for plug-in hybrid electric vehicles," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 733–739.
- [20] A. Houshmand and C. G. Cassandras, "Eco-routing of plug-in hybrid electric vehicles in transportation networks," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1508–1513.
- [21] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
- [22] HERE Technologies. (2019). *HERE Maps Developer API*. [Online]. Available: <https://developer.here.com/>
- [23] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO-Simulation of Urban MObility," *Int. J. On Adv. Syst. Meas.*, vol. 5, nos. 3–4, pp. 128–138, 2012.
- [24] A. Malikopoulos, Z. Filipi, and D. N. Assanis, "Simulation of an integrated starter alternator (ISA) system for the HMMWV," SAE, Warrendale, PA, USA, Tech. Rep. 2006-01-0442, 2006.
- [25] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Model predictive control of vehicles on urban roads for improved fuel economy," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 831–841, May 2013.
- [26] K. Ahn and H. Rakha, "A simple hybrid electric vehicle fuel consumption model for transportation applications," Virginia Polytechnic Inst. State Univ., Blacksburg, VA, USA, Tech. Rep. DOE-VT-0008209-BC01, 2019.
- [27] Y. Gao, L. Chen, and M. Ehsani, "Investigation of the effectiveness of regenerative braking for EV and HEV," SAE, Warrendale, PA, USA, Tech. Rep. 3184-3190, 1999.

- [28] J. Liu and H. Peng, "Modeling and control of a power-split hybrid vehicle," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1242–1251, Nov. 2008.
- [29] S. J. Moura, D. S. Callaway, H. K. Fathy, and J. L. Stein, "Tradeoffs between battery energy capacity and stochastic optimal power management in plug-in hybrid electric vehicles," *J. Power Sources*, vol. 195, no. 9, pp. 2979–2988, 2010.
- [30] Argonne National Laboratory. (2019). *Autonomie*. [Online]. Available: <https://www.autonomie.net>
- [31] O. Karabasoglu and J. Michalek, "Influence of driving patterns on life cycle cost and emissions of hybrid and plug-in electric vehicle powertrains," *Energy Policy*, vol. 60, pp. 445–461, Sep. 2013.
- [32] D. Bertsimas, J. N. Tsitsiklis, and J. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, MA, USA: Athena Scientific, 1997.
- [33] F. S. Hillier, *Introduction to Operations Research*. New York, NY, USA: McGraw-Hill, 2012.
- [34] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis, "The price of anarchy in transportation networks: Data-driven evaluation and reduction strategies," *Proc. IEEE*, vol. 106, no. 4, pp. 538–553, Apr. 2018.
- [35] Gurobi Optimization. (2019). *Gurobi Optimizer Reference Manual*. [Online]. Available: <http://www.gurobi.com>
- [36] X. Huang, B. Li, H. Peng, J. A. Auld, and V. O. Sokolov, "Eco-mobility-on-demand fleet control with ride-sharing," 2019, *arXiv:1908.09828*. [Online]. Available: <http://arxiv.org/abs/1908.09828>
- [37] J. Auld, M. Hope, H. Ley, V. Sokolov, B. Xu, and K. L. Zhang, "POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations," *Transp. Res. C, Emerg. Technol.*, vol. 64, pp. 101–116, Mar. 2016.
- [38] J. Auld and A. Mohammadian, "Activity planning processes in the agent-based dynamic activity planning and travel scheduling (ADAPTS) model," *Transp. Res. A, Policy Pract.*, vol. 46, no. 8, pp. 1386–1403, Oct. 2012.
- [39] D. Bezzina and J. Sayer, "Safety pilot model deployment: Test conductor team report," US Dept. Transp., Intell. Transp. Syst., Joint Program Office, Tech. Rep. DOT HS 812 171, 2014.
- [40] A. M. Ishtiaque Mahbub and A. A. Malikopoulos, "Concurrent optimization of vehicle dynamics and powertrain operation using connectivity and automation," 2019, *arXiv:1911.03475*. [Online]. Available: <http://arxiv.org/abs/1911.03475>
- [41] A. Houshmand, S. Wollenstein-Betech, and C. G. Cassandras, "The penetration rate effect of connected and automated vehicles in mixed traffic routing," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1755–1760.



Arian Houshmand received the B.S. degree in mechanical engineering from Sharif University of Technology, Tehran, Iran, in 2014, the M.S. degree in mechanical engineering from the University of Cincinnati, Cincinnati, OH, USA, in 2016, and the Ph.D. degree in systems engineering from Boston University, Boston, MA, USA, in 2020. He is currently working as a Motion Planning and Control Software Engineer with Zoox, Foster City, CA, USA. His research interests include motion planning and collaborative teleguidance of autonomous vehicles.



Christos G. Cassandras (Life Fellow, IEEE) received the degrees from Yale University, Stanford University, and Harvard University. He is currently a Distinguished Professor of engineering with Boston University. He is also the Head of the Division of Systems Engineering, a Professor of electrical and computer engineering, and the Co-Founder of the Boston University's Center for Information and Systems Engineering (CISE). From 1982 to 1984, he was with ITP Boston, Inc., where he worked on the design of automated manufacturing systems.

From 1984 to 1996, he was a Faculty Member with the Department of Electrical and Computer Engineering, University of Massachusetts Amherst. He also specializes in the areas of discrete event and hybrid systems, cooperative control, stochastic optimization, and computer simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. In addition to his academic activities, he has worked extensively with industrial organizations on various systems integration projects and the development of decision support software. He has most recently collaborated

with The MathWorks, Inc., in the development of the discrete event and hybrid system simulator SimEvents. He has published over 450 refereed articles in these areas and six books. He is a member of the Phi Beta Kappa and the Tau Beta Pi. He is a Fellow of IFAC. He was a recipient of several awards, including the 1991 Lilly Fellowship, the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook) for *Discrete Event Systems: Modeling and Performance Analysis*, the Distinguished Member Award of the IEEE Control Systems Society in 2006, the 2011 IEEE Control Systems Technology Award, the 2011 and 2014 Prize for IBM/IEEE Smarter Planet Challenge Competition, the 2012 Kern Fellowship, the 2014 Engineering Distinguished Scholar Award at Boston University, and several honorary professorships. He was the 2012 President of the IEEE Control Systems Society (CSS). He served as the Vice President for Publications. He served on the Board of Governors for the CSS and several IEEE committees, and chaired several conferences. He has been a Plenary/Keynote Speaker at numerous international conferences, including the 2017 IFAC World Congress, American Control Conference in 2001, and the IEEE Conference on Decision and Control in 2002 and 2016, and has been an IEEE Distinguished Lecturer. He guest-edited several technical journal issues and also serves on several journal Editorial Boards, including an Editor of *Automatica*. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1998 to 2009, and served as an Editor for Technical Notes and Correspondence and an Associate Editor.



Nan Zhou received the B.S. degree in control engineering from Zhejiang University, Hangzhou, China, in 2011, the M.S. degree in control engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2014, and the Ph.D. degree in systems engineering from Boston University, Boston, MA, USA, in 2019. He is currently a Senior Software Engineer with MathWorks Inc., Natick, MA, USA. His research interests include control modeling, system optimization, and automated code generation.



Nasser Hashemi received the B.S. degree in mechanical engineering from the University of Tehran, Tehran, Iran, in 2012, and the M.S. degree in biomechanical engineering from Sharif University of Technology, Tehran, in 2015. He is currently pursuing the Ph.D. degree in systems engineering with Boston University, Boston, MA, USA. His research interest is applying machine/deep learning methods to predict structure/function of protein complex and protein design.



Boqi Li received the B.S. degree in mechanical engineering from the University of Illinois at Urbana-Champaign and the M.S. degree from Stanford University. He is currently pursuing the Ph.D. degree in mechanical engineering with the University of Michigan, Ann Arbor. His research interests include cooperative routing, ride sharing, and cooperative driving with connected and automated vehicles.



Huei Peng received the Ph.D. degree from the University of California, Berkeley, CA, USA, in 1992. He is currently a Professor with the Department of Mechanical Engineering, University of Michigan, Ann Arbor. He also serves as the Director of Mcity, which studies connected and autonomous vehicle technologies and promotes their deployment. His research interests include adaptive control and optimal control, with emphasis on their applications to vehicular and transportation systems. His current research focuses include design and control of electrified vehicles, and connected/automated vehicles. an SAE Fellow and an ASME Fellow.