Bi-level Optimal Edge Computing Model for On-ramp Merging in Connected Vehicle Environment

Ye, Fei; Guo, Jianlin; Kim, Kyeong Jin; Orlik, Philip V.; Ahn, Heejin; Di Cairano, Stefano

TR2019-163 June 25, 2020

Abstract

The coordinated on-ramp merging is one of the most common but critical vehicular applications that require complex data transmission and low-latency communication in the Connected and Automated Vehicles (CAVs) environment. An effective way to address on-ramp merging is to leverage the edge computing to optimize the coordination among vehicles to achieve overall minimum vehicle travel time and energy consumption. In this study, we propose an Bi-level Optimal Edge Computing (BOEC) model for on-ramp merging in the CAVs environment to optimize both merge time and vehicle trajectory. The simulation results show that the proposed BOEC model achieves great benefits in vehicle mobility, energy saving and air pollutant emission reduction by providing an energy-efficient trajectory following the optimal merge time without compromising safety.

IEEE Intelligent Vehicle Symposium

Bi-level Optimal Edge Computing Model for On-ramp Merging in Connected Vehicle Environment

Fei Ye, Jianlin Guo, Kyeong Jin Kim, Philip V. Orlik, Heejin Ahn, Stefano Di Cairano, and Matthew J.Barth

***Abstract*—The coordinated on-ramp merging is one of the most common but critical vehicular applications that require complex data transmission and low-latency communication in the Connected and Automated Vehicles (CAVs) environment. An effective way to address on-ramp merging is to leverage the edge computing to optimize the coordination among vehicles to achieve overall minimum vehicle travel time and energy consumption. In this study, we propose an Bi-level Optimal Edge Computing (BOEC) model for on-ramp merging in the CAVs environment to optimize both merge time and vehicle trajectory. The simulation results show that the proposed BOEC model achieves great benefits in vehicle mobility, energy saving and air pollutant emission reduction by providing an energy-efficient trajectory following the optimal merge time without compromising safety.**

***Index Terms*—Cooperative vehicle control, edge computing, optimized vehicle scheduling, optimal trajectory planing, ITS.**

I. Introduction

The increasing transportation activities and traffic jam have led to significant effect on social and economic issue. The bottlenecks in the transportation system not only lead to huge economic and mobility cost, but also have side effect on increasing air pollutant emissions and the risk of collision. The emerge of connected and automated technology offers the opportunity to address the aforementioned issues. With the capability of transmitting real-time information between vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), more advanced and efficient management system can be developed to reduce the congestion and air pollutant emissions as well as improve the safety perspective. Connected and Automated Vehicles (CAVs) can improve the highway capacity and eliminate the potential hazard from driver fatigue, irritation and reaction delay with shorter headway and instant response.

On-ramp merging attracts significant research attention. It has been approached by designing cooperative ramp metering [1], [2] and local cooperative maneuver when vehicles approach the merge zone [6], [7]. Ramp metering has been widely used in California to regulate the traffic flow of on-ramp vehicles when they merge into the highway. Other than the traditional pre-timed ramp metering algorithm based on historic traffic data [3], some advanced ramp metering algorithms have been developed using the vehicle connectivity. A cooperative ramp metering is proposed in [1] to take the advantage of the enabled cooperation among vehicles to form a platoon in the CAVs environment. Yang et al. [2] presents a ramp metering control based on reinforcement Q-learning to enhance the capacity of merging section. However, with the ramp metering approaches, the undesirable idling at on-ramp merge point is still not fully avoidable. The stop-and-go pattern at the ramp metering results in large air pollutant emissions and energy consumption and have the side effect on vehicle safety due to the speed gap between the highway vehicle and on-ramp vehicle when on-ramp vehicle starts off from the ramp metering. The existing traffic management is mainly cloud-based system [4]. However, cloud computing may not always be the best strategy for real-time applications such as cooperative on-ramp merging. To address the real time on-ramp merging control, we leverage the edge computing approach [5] that can efficiently access and process resources at the particular point/location in the vehicular network, e.g., roadside unit (RSU).

In this context, we propose a Bi-level Optimal Edge Computing (BOEC) methodology to maximize both the vehicle mobility benefit and energy saving without compromising safety perspective. Instead of considering a bunch of vehicles one time when vehicles are approaching the merging point, our approach can deal with streaming traffic and strategically takes advantage of the large communication range between vehicles and infrastructure to realize the coordinated on-ramp merging. We first develop an optimal merge sequence and merge time scheduling model by crowd sourcing initial state of vehicles, clustering vehicles based on their potential conflict at the merge point and solving a Mixed-Integer Linear Programming (MILP) problem. We evaluate the effectiveness of the proposed MILP-based optimal scheduling model by comparing it with the existing rule-based first-in-first-out (FIFO) model in terms of mobility and sustainability. We then determine the optimal vehicle trajectories to guarantee vehicles meet the assigned merge time with the lowest energy cost. For the vehicle trajectory planing, we propose three different approaches (a closed-form heuristic model, a quadratic programming (QP) based model and a graph based model) on energy consumption map based on the trade off between computation efficiency and optimality.

Fei Ye and Matthew J.Barth are with University of California, Riverside, CA, 92521 USA (email:{fye001@ucr.edu, barth@ece.ucr.edu}).

Jianlin Guo, Kyeong Jin Kim, Philip V. Orlik, Heejin Ahn, and Stefano Di Cairano are with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, 02139 USA (e-mail: {guo, kkim, porlik, hahn, dicairano}@merl.com).

This work was done while Fei Ye was working at Mitsubishi Electric Research Laboratories (MERL).

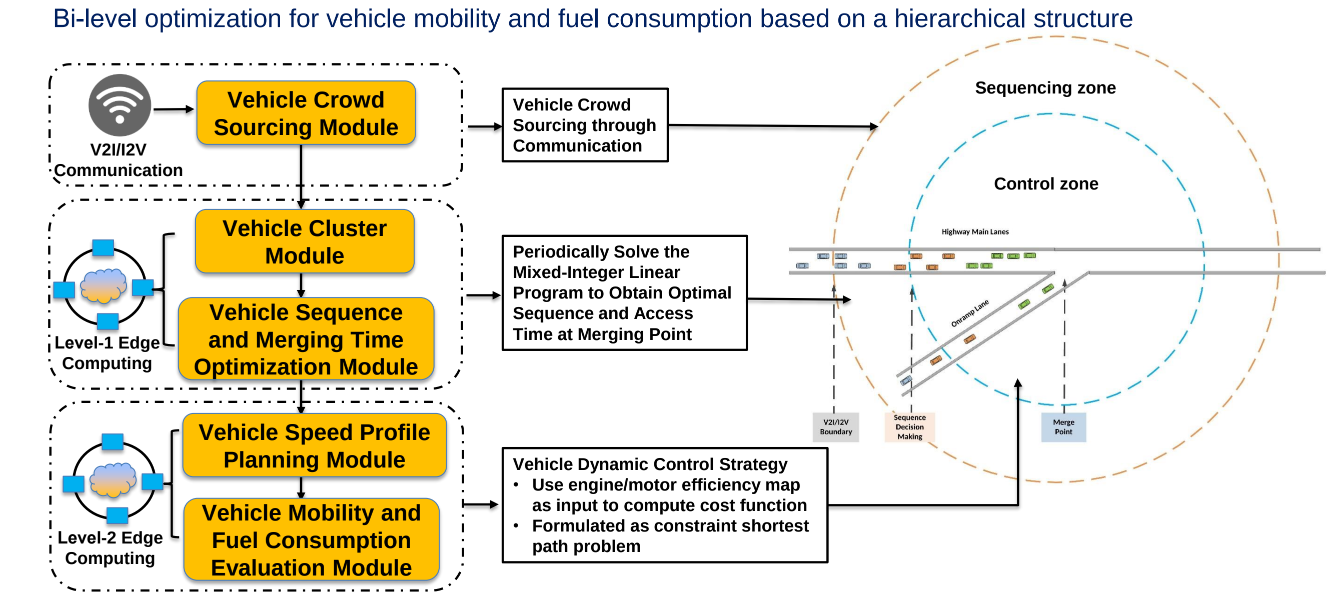
Bi-level optimization for vehicle mobility and fuel consumption based on a hierarchical structure
V2I/I2V Communication
Vehicle Crowd Sourcing Module
Vehicle Crowd Sourcing through Communication
Vehicle Cluster Module
Vehicle Sequence and Merging Time Optimization Module
Level-1 Edge Computing
Periodically Solve the Mixed-Integer Linear Program to Obtain Optimal Sequence and Access Time at Merging Point
Vehicle Speed Profile Planning Module
Vehicle Mobility and Fuel Consumption Evaluation Module
Level-2 Edge Computing
Vehicle Dynamic Control Strategy
• Use engine/motor efficiency map as input to compute cost function
• Formulated as constraint shortest path problem
Sequencing zone
Control zone
Highway Main Lanes
Onramp Lane
V2I/I2V Boundary
Sequence Decision Making
Merge Point

Fig. 1. Flow Diagram and Framework of Bi-level Optimal Edge Computing for On-ramp Merging

The rest of this paper is organized as follows. Section II introduces the framework and overall structure of the proposed BOEC model. Section III presents optimal merge sequence and merge time scheduling. In Section IV, the optimal vehicle speed trajectory planning approaches are presented. Section V describes the simulation setup and test scenario. In Section VI, optimal vehicle merging scheduling is evaluated and compared with the existing FIFO-based strategy, followed by a comprehensive analysis and discussion of the optimal vehicle speed trajectory planning simulation results. Finally, we conclude our paper in Section VII.

II. Bi-level Optimal Edge Computing Architecture

In this section, we present the proposed BOEC architecture for on-ramp merging coordination and eco-driving. Fig. 1 illustrates the framework and overall structure of the proposed BOEC model, which consists of a vehicle crowd sourcing module and a bi-level edge computing module. We partition section of the road within V2I communication range into two zones: the sequencing zone and the control zone.

The first-level edge computing, i.e., the first level optimization, takes place in the sequencing zone to cluster the connected vehicles into groups and to obtain optimal merge sequence and merge time. First, we use RSU to collect vehicle information including position, speed, heading, lane information upstream with respect to the merge point. We then assign each vehicle into an associated cluster group based on its state and potential conflict at the merge point. We formulate the on-ramp merging scheduling as a MILP problem [8]. The formulated optimization problem is periodically solved for the clustered vehicles in the sequencing zone using Gurobi solver. The corresponding outputs (vehicle merge sequence and assigned merge time) are used in the second-level edge computing.

The second-level edge computing, i.e., the second level optimization, is realized in the control zone following merge sequence and merge time assignments to compute the optimal speed trajectories for the connected vehicles in terms of energy efficiency and air pollutant emission reduction. We propose a closed-form heuristic model, a quadratic programming model and graph-based shortest path model. In the control zone, vehicles travel with the optimal energy-efficient trajectory following the merge sequence and merge time.

III. Optimal Vehicle Merge Sequence and Merge Time Scheduling

In this section, we describe the proposed MILP-based model for optimal scheduling and a simple reservation-based FIFO policy approach for performance comparison.

The reservation-based FIFO policy approach takes initial state of the connected vehicles when they enter the sequencing zone and calculates the earliest estimated arrival time at the merge point. This approach then dynamically adjusts vehicle merge time by filling up a reservation table scaling by the safety headway before vehicle enters the control zone. The reservation management is based on FIFO policy. Although this approach can reduce the conflicts at the merge point with low computational cost, it limits the potential of maximizing the vehicle throughput at the merge point and enlarges the transmission burden in the V2I communication network.

To obtain optimal merge sequence and merge time for the clustered vehicles to further improve the travel throughput and maximize the mobility benefit, we formulate the problem as a MILP [8] to minimize total travel time for both mainline vehicles and on-ramp vehicles. As described in Eq.1, the attributes of each connected vehicle CV_i $(1 \leq i \leq n)$ contains vehicle ID, vehicle position, vehicle speed, speed limit, distance to the merge point, lane number, the time vehicle entering to the sequencing zone and the earliest merge time at the merge point calculated based on the speed limit.

1) Vehicle Attributes:

$$CV_i = \{i, x_i, v_i, v_{max}, d_i, l_i, t_{0,i}, t_{merge,min,i}\} \quad (1)$$

2) Decision Variables: Optimal merge time of the vehicle at merging point

$$\begin{aligned} &t_{merge,i},\ i \in ID_M = \{ID_1^M, ID_2^M, ..., ID_{n_1}^M\} \\ &t_{merge,j},\ j \in ID_R = \{ID_1^R, ID_2^R, ..., ID_{n_2}^R\} \end{aligned} \quad (2)$$

3) Objective function and optimization problem:

$$\min J = \min \Sigma_{i=1}^{n_1}(t_{merge,i} - t_{0,i}) + \Sigma_{j=1}^{n_2}(t_{merge,j} - t_{0,j}) \quad (3)$$

4) Constraints:

$$t_{merge,k} > t_{merge,min,k} \quad (4)$$

$$t_{merge,k1} - t_{merge,k2} \geq t_{headway1} \quad (5)$$

$$\begin{aligned} &t_{merge,i} - t_{merge,j} \geq t_{headway2} \\ \text{or}\ \ &t_{merge,j} - t_{merge,i} \geq t_{headway2} \end{aligned} \quad (6)$$

where n_1 and n_2 are the number of the involved mainlane vehicles and the number of the involved on-ramp vehicles, respectively; $t_{0,i}$ is the time mainline vehicle i entering to the sequencing zone; $t_{0,j}$ is the time on-ramp vehicle j entering to the sequencing zone; $k \in ID_M$ or $k \in ID_R$; $k1, k2 \in ID_M$ or $k1, k2 \in ID_R$; $t_{headway1}$ is the time headway between the adjacent merging vehicles on same lane and can be different for the mainlane vehicles and the on-ramp vehicles; $t_{headway2}$ is the time headway between adjacent merging vehicles on different lanes at the merge point. The constraints assure the safety and the traffic rule as well as vehicle acceleration/deceleration capability. The first constraint (Eq.4) ensures that vehicle will not violate the speed limit through the merge zone. The other two constraints (Eq.5 and Eq.6) are based on the assumption that no overtaking is allowed for vehicles in the same lane and ensure the safety headway for both vehicles on the same lane and vehicles between the mainlane and the on-ramp. We choose different headway based on whether vehicles are traveling on the same lane or not. Vehicles on same lane can potentially formulate a platoon with much shorter headway, which is taken into consideration in our optimization problem.

The formulated problem intends to provide the optimal merge time $t_{merge,i}$ for the mainline vehicle i and the optimal merge time $t_{merge,j}$ for the on-ramp vehicle j to minimize their travel time through the merge zone. Therefore, the objective of the MILP problem is to minimize the total travel time for all the involved vehicles through the merge zone. The optimal merge sequence and merge time of the connected vehicles at the merge point guarantee the overall difference between the sequencing zone entering time and the assigned merge time is minimized without compromising any safety perspective or the traffic rule.

To take advantage the relatively long communication range of V2I network, we can further improve vehicle mobility and energy savings by planing the scheduling and timing far ahead before they approaching the merge point.

To mathematically interpret and solve the discontinuity in the last constraint (Eq.6), additional binary variables have been introduced, which we refer to the big-M method [9]. We can transform the last constraint into a 0-1 binary linear programming problem such that if one of these two inequalities is true the other one is always redundant. For this purpose, we introduce new binary variable $B_{i,j}$ and a constant M. Then, the Eq.6 can be converted to Eq.7:

$$\begin{aligned} &t_{merge,i} - t_{merge,j} + MB_{i,j} \geq t_{headway2} \\ &t_{merge,j} - t_{merge,i} + M(1 - B_{i,j}) \geq t_{headway2} \end{aligned} \quad (7)$$

where $B_{i,j}$ is either 0 or 1, and M is a large enough constant to make the $t_{headway2} + |t_{merge,i} - t_{merge,j}|$ negligible compared to it. We set M to 2000 in our formulated optimization problem.

IV. Optimal Vehicle Trajectory Planning

Once the optimal merge sequence and merge time become available, we efficiently plan vehicle speed trajectory online in order to assure the assigned merging time at the merge point while minimizing the overall energy consumption.

In this paper, we propose three different planning approaches to obtain vehicle trajectory through the control zone. Different approaches can be applied under different scenarios or for different goals according to the trade off between computation efficiency and optimality. The first vehicle trajectory planner is based on a closed-form heuristic model combined with the Gipps' car-following model (see Eq.8) to constrain the vehicle headway relative to its preceding vehicle. The second trajectory planner is formulated as a QP optimization problem to minimize the L^2-norm of the control input (acceleration/deceleration rate). Finally, we propose a graph-based optimization approach using Dijkstra's algorithm to find the minimum energy cost path with constraints that directly optimize energy consumption while following the assigned merge sequence and merging time from the first-level edge computing without compromising safety.

A. Heuristic Vehicle Trajectory Planning

Once the merging time is assigned to each vehicle, we determine whether a vehicle can keep the current speed v_i or needs a acceleration/deceleration to follow the assignment based on its current state attributes. If the arrival time at the merge point using current speed is larger than the remaining time to the assigned merging time, vehicle needs to accelerate to meet the goal and vice versa. We assume vehicles have the constant acceleration/deceleration rate till they reach their desired cruise speed that enables their remaining travel time to the merge point equal to the time difference between their merging time to the merge point $t_{merge,i}$ and the current time t_0 as shown in Fig.2.

A closed-form analytical vehicle trajectory solution can be obtained as follows:

Vehicle attribute:

$CV_i = \{i, v_i, v_{lim}, d_i, t_{0,i}, a_{min}, a_{max}, t_{merge,i}\}$

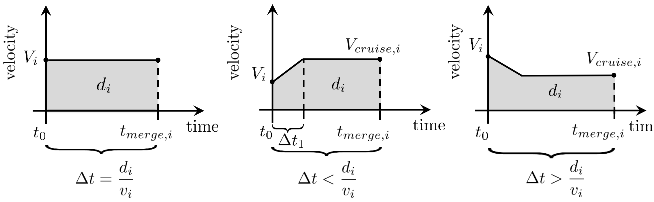

Fig. 2. Heuristic Vehicle Speed Trajectory Planning

Using Fig.2, we can obtain

$$\begin{cases} v_{cruise,i} = v_i + a\Delta t_1 \\ v_i\Delta t_1 + \dfrac{1}{2}a\Delta t_1^2 = d_{in} \\ v_{cruise,i}(\Delta t - \Delta t_1) = d_i - d_{in} \end{cases} \tag{9}$$

which can further derive

$$\Delta t_1 = \frac{a\Delta t \pm \sqrt{a^2\Delta t^2 - 2a(-v_i\Delta t + d_i)}}{a} \tag{10}$$

where $\Delta t = t_{merge,i} - t_{0,i}$; Δt_1 is the acceleration time from the initial speed v_i to the cruise speed $v_{cruise,i}$; d_{in} is the travel distance within the acceleration period Δt_1; $a = a_{max}$ or a_{min} is the constant acceleration rate or deceleration rate.

Therefore, the cruise speed for both acceleration-cruise pattern and deceleration-cruise pattern can be obtained from Eq.9 and Eq.10 as follows:

Acceleration-cruise pattern:

if $t_{merge,i} < t_0 + \dfrac{d_i}{v_i}$

$$v_{cruise,i} = v_i + a\Delta t - \sqrt{a^2\Delta t^2 - 2a(-v_i\Delta t + d_i)} \tag{11}$$

Deceleration-cruise pattern:

if $t_{merge,i} > t_0 + \dfrac{d_i}{v_i}$

$$v_{cruise,i} = v_i + a\Delta t + \sqrt{a^2\Delta t^2 - 2a(-v_i\Delta t + d_i)} \tag{12}$$

The flow diagram of the heuristic vehicle trajectory planner is shown in Fig.3. Given the optimal sequence and merge time from the first-level edge computing and vehicle's current attribute, vehicle can keep its current speed v_i as cruise speed to the merge point if $t_{merge,i} = t_0 + \dfrac{d_i}{v_i}$. Otherwise, we determine vehicle's driving pattern and compute the vehicle speed trajectory based on Eq.11 or Eq.12.

The Gipps' car following model is applied as a subsystem to guarantee safety perspective as described in Eq.8, where τ is the reaction time (can be very small); $v_n(t)$ and $v_{n-1}(t)$ are the speed of the following vehicle n and the leading vehicle $n-1$ at time step t, respectively; v_n^d and a_n are the desired speed and the maximum acceleration of vehicle n; b_n and b are the most severe braking that the driver of vehicle n wishes to undertake and the expected leading vehicle maximum deceleration, respectively.

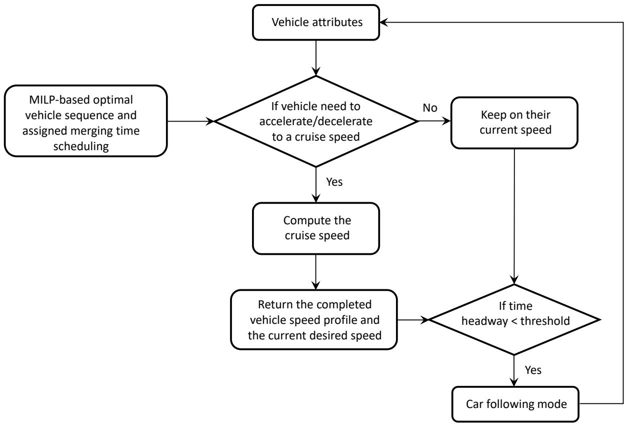
Vehicle attributes
MILP-based optimal vehicle sequence and assigned merging time scheduling
If vehicle need to accelerate/decelerate to a cruise speed
No
Keep on their current speed
Yes
Compute the cruise speed
Return the completed vehicle speed profile and the current desired speed
If time headway < threshold
Yes
Car following mode

Fig. 3. Flow Diagram of Heuristic Vehicle Trajectory Planner

B. QP-Based Optimal Trajectory Planning

It has been proven that the vehicle energy consumption highly relies on the acceleration/deceleration and speed profile [10]. The optimization problem becomes a constrained nonlinear programming problem. Solving that constrained nonlinear programming problem using energy consumption model as the objective function is quite challenge. The nonconvexity and high nonlinearity usually lead to great computational cost and hardly feasible for real-time microscopic vehicle maneuver application. Therefore, in this study, we formulate the problem as minimizing the L^2-norm of the control input (acceleration/deceleration rate) to provide a feasible trajectory based on quadratic programming method.

Vehicle dynamic equations are as follows:

$$\begin{cases} x_i(t_{k+1}) = x_i(t_k) + \dfrac{v_i(t_{k+1}) + v_i(t_k)}{2}\Delta t \\ v_i(t_{k+1}) = v_i(t_k) + a_i(t_k)\Delta t \end{cases} \tag{13}$$

The convex objective function can be expressed by setting H as positive definite matrix and f as zero vector in general form:

$$\min\ \Sigma a_i^T H a_i + f a_i \tag{14}$$

subject to

$$x_i(t_{merge,i}) = d_i \tag{15}$$

$$0 \leq v_i(t_k) \leq v_{lim} \tag{16}$$

$$a_{min} \leq a_i(t_k) \leq a_{max} \tag{17}$$

$$v_n(t+\tau) = \min\left\{ v_n(t) + 2.5a_n\tau\left(1 - \frac{v_n(t)}{v_n^d}\right)\sqrt{0.025 + \frac{v_n(t)}{v_n^d}},\right. \\ \left. b_n\tau + \sqrt{b_n^2\tau^2 - b_n[2(x_{n-1}(t) - s_{n-1} - x_n(t)) - v_n(t)\tau - v_{n-1}^2(t)]} \right\} \tag{8}$$

$$|x_i(t_k) - x_j(t_k)| \geq d_{headway} \tag{18}$$

$$x_i(t_0) = x_{ini,i} \tag{19}$$

$$v_i(t_0) = v_{ini,i} \tag{20}$$

where a_i is the control input (acceleration/deceleration rate); d_i is the current distance to the merging point; $t_{merge,i}$ is the optimal merge time at the merge point; Δt is time step; $x_{ini,i}$ and $v_{ini,i}$ represent the vehicle's initial state; $d_{headway}$ is the safety distance between vehicles; v_{lim} is the road speed limit; a_{min} and a_{max} are vehicle minimum acceleration rate and maximum acceleration rate, respectively.

C. Graph-Based Optimal Trajectory Planning

As aforementioned, the object of the proposed second-level edge computing is to minimize vehicle energy consumption through control zone till they reach the merge point with their assigned merge time. The energy consumption is a nonlinear function that mainly depends on vehicle type, speed, acceleration/deceleration and road grade. Solving that constrained nonlinear programming problem using energy consumption model as the objective function is quite challenge and hardly feasible for a real-time vehicle microscopic maneuver application.

To enable optimization of the energy consumption model while enhancing the computation efficiency, we propose a graph-based optimal trajectory planning approach with constraints on total travel time, total travel distance, maximum capable acceleration/deceleration rate and final speed at the merge point. To formulate this graph-based model, we first discretize the system states, e.g., 0.5 sec time step and 0.5 m/s speed resolution, and balance the data resolution against the computation cost. The state transition diagram is illustrated in Fig.4. At each node of the proposed directed graph $G = (V, E)$, we assign a unique 3-D coordinate (t, x, v) that describes the dynamic state of the vehicle, where $t \in [t_0, t_{merge}]$ is the time (in second), $x \in [0, d]$ is the distance to the merge point (in meter) and $v \in [0, v_l]$ is the speed (in m/s), where v_l is the speed limit in this graph. For each time step, the feasible nodes for the next state is determined by the current state and constraints from speed limit, maximum power and capability of braking system. There is an edge transit from the state (t_i, x_i, v_i) to state $(t_{i+1}, x_{i+1}, v_{i+1})$ with a cost as the energy consumption during this state transition process. We assume the road grade satisfies a predefined function of distance $g(d_{i:i+1})$ as shown in Eq. 21.

$$\theta = g(d_{i:i+1}) = \arcsin \frac{2 * (d_{i+1} - d_i)}{(v_{i+1} + v_i) * \delta t} \tag{21}$$

where d_i is the distance to the merge point at time step i; δt is the time step scaling factor. At this point, the fuel consumption minimization problem is converted into a problem to find the shortest path from the source node $N(0, X, V_s)$ to the destination node $N(t_{merge}, 0, V_f)$ in the directed graph G=(V, E). We apply Dijkstra's algorithm to solve this single-source shortest path problem with non-negative cost. Fig.4 illustrates an example of the optimal speed transit trajectory in terms of energy consumption through the control zone while satisfying all exogenous and endogenous constraints. The proposed algorithm is able to deal with more complicated problem with longer time period/distance and higher time/location resolution efficiently, as the time complexity of Dijkstra's algorithm is O(log(N)*E), where N is the source node number and E is the number of edges.

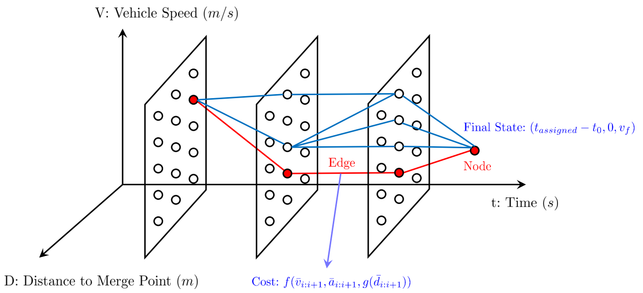
V: Vehicle Speed (m/s)
Final State: $(t_{assigned} - t_0, 0, v_f)$
Edge
Node
t: Time (s)
D: Distance to Merge Point (m)
Cost: $f(\bar{v}_{i:i+1}, \bar{a}_{i:i+1}, g(\bar{d}_{i:i+1}))$

Fig. 4. Graph-based optimal trajectory planner illustration

V. Simulation Setup and Test Scenario

This section describes simulation tools, simulation setup and test scenarios.

A. Simulation Setup

In this paper, the microscopic traffic simulation SUMO (Simulation of Urban Mobility) [11] is used to evaluate the performance of the proposed BOEC model by interfacing with the movement of each individual vehicle. Fig. 5 illustrates the interaction among the traffic simulator SUMO, the developed BOEC model and the energy consumption simulator MOVES [12]. We assume that the penetration rate of connectivity is 100%. We access the connected vehicle information in the network and develop three advanced API modules (vehicle clustering, vehicle optimal scheduling and vehicle optimal trajectory planning) to achieve optimal on-ramp merge coordination in connected vehicle environment through the Traffic Control Interface (TraCI) in SUMO. These API modules acquire vehicle information such as speed, position, acceleration/deceleration and entering time to determine the clustered group for coordination maneuver. Then, the optimal merge sequence and merge time for the involved vehicles are periodically computed using Gurobi optimization solver before vehicles enter the control zone. The API for motion planning provides the energy efficient vehicle trajectory in the control zone following the assigned merge sequence and merge time without compromising safety. These control inputs are sent back to SUMO to model each individual vehicle movement through on-ramp merging. In the simulation, we assume the speed limit on the highway is 108 km/h and on the on-ramp is 72 km/h. The maximum acceleration or deceleration are $2.5 m/s^2$, $-2.5 m/s^2$, respectively. In addition, vehicle trajectories and road elevation map generated by SUMO are used as inputs to MOVES model to evaluate the energy consumption and air pollutant emissions. The mobility benefits

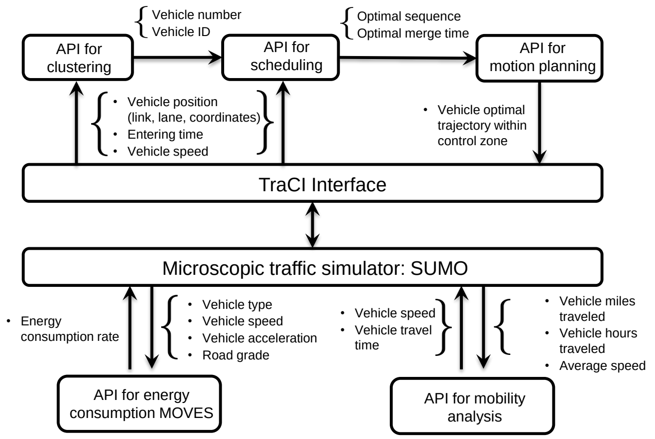

Fig. 5. Simulation Software Components and Interface Block Diagram

are quantified by average travel time and average speed of the vehicles provided in Eq.22

$$\bar{v} = \frac{\Sigma_{i=1}^{n}\Sigma_{t=1}^{T_i} VMT_{i,t}}{\Sigma_{i=1}^{n}\Sigma_{t=1}^{T_i} VHT_{i,t}} \tag{22}$$

where $VMT_{i,t}$ is the vehicle miles traveled for vehicle i at time step t and $VHT_{i,t}$ is the vehicle hours traveled for vehicle i at time step t.

All experiments are carried out using a computer with Intel i7 CPU with 2.80 GHz and 16 GB RAM.

B. Energy Consumption Model and Evaluation Metrics

To quantify the effectiveness of the proposed BOEC model in terms of energy saving and air pollutant emission reduction, the U.S. Environmental Protection Agencys MOtor Vehicle Emission Simulator (MOVES) is applied. The MOVES model is the state of art energy consumption and emission simulator developed by the U.S. Environmental Protection Agency (USEPA). The model is designed to estimate energy consumption and emissions for mobile sources on a macroscale, mesoscale or microscale. The second-by-second Vehicle Specific Power (VSP) can be first calculated based on the vehicles speed trajectory and road grade information. Then, the operating mode (OpMode) distribution over 23 bins for running exhaust emissions can be derived from a function of VSP, speed and acceleration. Finally, with the OpMode distribution, the energy consumption and emissions of all the vehicle trajectories are estimated based on the emission factors from MOVES database. The evaluation metrics chosen for effectiveness analysis on the environmental influences includes emissions of HC, CO, CO_2, NO_X, $PM_{2.5}$ and energy consumption. The energy consumption factor (EF, energy consumption in unit distance, KJ/mile) can be obtained by:

$$EF = \frac{\Sigma_{i=1}^{n}\Sigma_{t=1}^{T_i} energy_{i,t}}{\Sigma_{i=1}^{n}\Sigma_{t=1}^{T_i} VHT_{i,t}} \tag{23}$$

where $energy_{i,t}$ is the energy consumption rate for vehicle i at time step t measured in KJ and $VHT_{i,t}$ is the vehicle hours traveled for vehicle i at time step t.

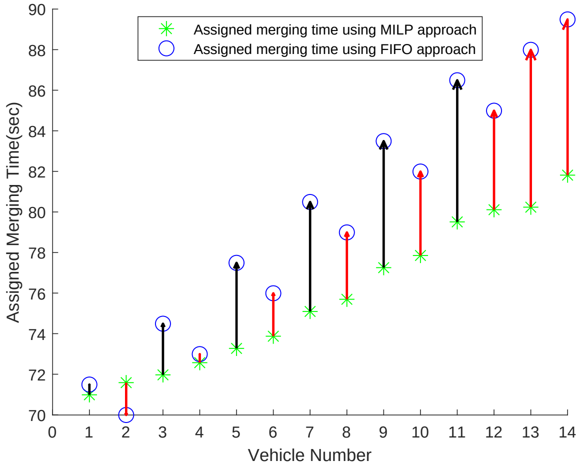

Fig. 6. An example of the Assigned Merge Time Difference Using MILP-based Model and FIFO-based Model

VI. Simulation Results and Performance Comparison

In this Section, we first compare our proposed MILP-based optimal scheduling algorithm with the traditional FIFO-based approach in terms of vehicle mobility benefit. We then evaluate the overall energy consumption and air pollutant emissions by using three different vehicle trajectory planning algorithms. Total 290 vehicles and 207 vehicles have been released from west bound of the highway and the on-ramp, respectively.

A. Simulation Validation of the Proposed MILP-based Optimal Scheduling

Fig.6 shows an example of a clustered vehicles with the merge time and merge sequence differences assigned by the MILP-based model and the FIFO-based model. The blue dots indicate the assigned merge time for each individual vehicle using FIFO-based reservation table and the green stars show the assigned optimal merge time by the proposed MILP-based method. The arrows indicate the time assignment difference with red for mainline vehicles and black for on-ramp vehicles. It is obviously observed that the merge sequence of vehicles using MILP-based model is different from that using FIFO-based model with some vehicles on the mainline sacrifice their merge time to achieve the group travel time saving and mobility benefits. The statistical comparison analysis of the MILP-based optimal scheduling model and the FIFO-based model is shown in Table I. The average travel time on the mainline by the MILP-based model is 10.27 sec that achieves 21.2% improvement compared with FIFO-based reservation model. On-ramp vehicles can also improve their throughput time by 20.4% using the proposed MILP-based optimal scheduling model. In addition, the standard deviation of the MILP-based model is much smaller compared with FIFO-based model, which leads to more reliable performance.

B. Simulation Validation of the Proposed Vehicle Trajectory Planner

Based on the MOVES model, Table II shows the energy and environmental benefits of the total 497 vehicle trajectories

TABLE I

PERFORMANCE COMPARISON BETWEEN OPTIMAL MILP-BASED SCHEDULING MODEL AND FIFO-BASED APPROACH

	Average through time on mainline (sec)	Standard Deviation	Average through time on-ramp (sec)	Standard Deviation
FIFO-based	13.03	1.12	12.67	0.94
MILP-based	10.27	0.48	10.09	0.13
Relative improvement (%)	21.2%		20.4%	

TABLE II

FUEL CONSUMPTION AND POLLUTANT EMISSION COMPARISON
EVALUATION CONDUCTED USING MOTOR VEHICLE EMISSION SIMULATOR (MOVES MODEL)

Vehicle Trajectory Planning	HC (g/mile)	CO (g/mile)	NO_X (g/mile)	CO_2 (g/mile)	Energy (KJ/mile)	PM2.5 (mg/mile)	Ave Time cost (sec)
Graph Based Method	0.029	0.86	0.11	53.4	743.34	2.9	4.6
Quadratic Programming Based Method	0.025	0.54	0.13	58.8	818.33	1.6	0.01
Analytical Solution (Baseline)	0.079	2.43	0.38	106.88	2238.42	7.7	0.00014
Improvement (%) of Graph Based Method	62.6%	64.4%	71.7%	66.8%	66.8%	61.5%	
Improvement (%) of QP Based Method	68.7%	77.6%	65.3%	63.4%	63.4%	79.4%	

generated by three trajectory planning approaches proposed. It is obvious that both QP-based and graph-based optimal trajectory methods can significantly improve the energy saving and air pollutant emissions. Compared with the baseline analytical solution, QP-based optimal solution and graph-based optimal solution improve average energy saving by 63.4% and 66.8%, respectively. In addition, significant air pollutant emission reduction can be observed from Table II. The emissions of HC, CO, NO_x, CO_2 and $PM_{2.5}$ per mile using QP-based model are 68.7%, 77.6%, 65.3%, 63.4% and 79.4% less than the baseline model, respectively. Table II also shows that the proposed graph-based optimal trajectory model can reduce 62.6% of HC, 64.4% of CO, 71.7% of NO_x, 66.8% of CO_2 and 61.5% of $PM_{2.5}$ per mile compared with the baseline analytical model. The average computation cost for a 400 meter trajectory planning using Graph-based model, QP-based model and a heuristic solution are 4.6 sec, 0.01 sec and $1e^{(-4)}$ sec, respectively.

VII. CONCLUSION

In this paper, we propose a Bi-level Optimal Edge Computing (BOEC) model for on-ramp merging to maximize overall vehicle mobility benefits and energy saving while optimizing air pollutant emissions. Our key contributions are: 1) developing an edge computing scheme based on V2I/I2V communication; 2) improving vehicle throughput and average speed by grouping vehicles and periodic mixed-integer linear program optimization; 3) updating the heuristic vehicle trajectory planning approach by introducing a sub-system of car-following model; 4) developing a quadratic programming based optimal trajectory model that guides vehicle merging coordination with the minimum air pollutant emissions; and 5) developing a graph based optimal trajectory model that guides vehicle merging coordination with the minimum energy consumption. Microscopic simulation results demonstrate that the proposed MILP-based optimal merge sequence and merge time scheduling can save the travel time on both mainline and on-ramp by 21.2% and 20.4%, respectively, compared to the FIFO-based reservation approach. The comparative validation results of vehicle trajectory planning algorithms indicate that the proposed QP-based trajectory model and graph-based optimal trajectory model outperform the heuristic baseline model in terms of energy saving by 63.4% and 66.8%, respectively. Both QP-based and graph-based trajectory planning models can reduce air pollutant emissions by 61.5% - 79.4% compared to the baseline approach. It turns out that the computational cost of these three vehicle trajectory planning approaches can satisfy the objective of the real time performance.

REFERENCES

[1] R. Scarinci, B. Heydecker, and A. Hegyi, “Analysis of traffic performance of a ramp metering strategy using cooperative vehicles”, in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst.*, The Hague, The Netherlands, 2013, pp. 324–329.

[2] H. Yang, and H. Rakha, “Reinforcement learning ramp metering control for weaving sections in a connected vehicle environment,” in *Proc. Transp. Res. Board 96th Annu. Mtg.*, Washington, DC., 2017.

[3] Scariza, J. R., “Evaluation of Coordinated and Local Ramp Metering Algorithms using 35 Microscopic Traffic Simulation,” in *MIT*, Boston, MA, 2003.

[4] P. Jaworski, ”Cloud computing based adaptive traffic control and management”, in *PhD Thesis*, 2013, Coventry: Coventry University in collaboration with MIRA Ltd.

[5] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick and D. S. Nikolopoulos, “Challenges and opportunities in edge computing,” in *Proc. IEEE Int. Conf. Smart Cloud (SmartCloud)*, New York, NY, 2016, pp. 20–26.

[6] J. Rios-Torres, A. A. Malikopoulos, and P. Pisu, “Online optimal control of connected vehicles for efficient traffic flow at merging roads”, in *Proc. 18th Int. IEEE Conf. Intell. Transp. Syst.*, Washington, DC., 2015, pp. 2432–2437.

[7] D. Miculescu and S. Karaman, “Polling-systems-based control of high-performance provably-safe autonomous intersections”, in *Proc. IEEE 53rd Annu. Conf. Decision Control*, Los Angeles, CA, 2014, pp. 1417–1423.

[8] S. A. Fayazi, A. Vahidi, and A. Luckow, “Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via MILP”, in *Proc. IEEE Amer. Control Conf.*, Seattle, WA, 2017, pp. 4920–4925.

[9] W. L. Winston, *Introduction to Mathematical Programming: Applications and Algorithms*, 4th ed. Duxbury, MA: Duxbury Resource Center, 2002.

[10] F. Ye, P. Hao, X. Qi, G. Wu, K. Boriboonsomsin, and M. J. Barth, Prediction-Based Eco-Approach and Departure at Signalized Intersections With Speed Forecasting on Preceding Vehicles, IEEE Transactions on Intelligent Transportation Systems, pp. 112, 2018.

[11] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO–Simulation of urban mobility: An overview,” in *Proc. 3rd Int. Conf. Adv. Syst. SIMUL*, 2011, pp. 63–68.

[12] U.S. Environmental Protection Agency, “MOVES (motor vehicle emission simulator)”. Nov. 2014, [online] Available: http://www.epa.gov/otaq/models/moves/.