



Research Article

Centralized and Decentralized Signal Control with Short-Term Origin-Destination Demand for Network Traffic

Can Zhang ¹, Tony Z. Qiu ^{1,2} and Amy Kim¹

¹Department of Civil and Environmental Engineering, University of Alberta, Edmonton T6G 2H5, Canada

²Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430063, China

Correspondence should be addressed to Tony Z. Qiu; zhijunqiu@ualberta.ca

Received 8 June 2021; Revised 17 January 2022; Accepted 3 February 2022; Published 28 April 2022

Academic Editor: Jing Dong

Copyright © 2022 Can Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We develop and assess centralized and decentralized signal control systems with short-term origin-destination (OD) demands as inputs. Considering each intersection turning movement as a virtual link, we assign traffic demand to paths based on minimal instantaneous travel time. Then, the optimal control is formulated using a G/G/n/FIFO open queueing network model (QNM). We also solve the issue of optimal control using a three-step naïve method for the centralized system with the new inputs. Because the optimization of large-scale network traffic signals can involve sizeable numbers of decision variables and nonlinear constraints, making it a nondeterministic polynomial time (NP) complete problem, we further decompose the centralized system into a decentralized system where the network is divided into subnetworks. Each subnetwork has a dedicated agent that optimizes signals within it. Furthermore, traffic demand for the entire network is decomposed into demands for subnetworks via path decomposition index (PDI). The proposed control systems are applied to test scenarios constructed using different demand profiles in grid networks. We also investigate the impact of network decomposition strategy on signal control system performance. Results show that network decomposition with smaller subnetworks results in less computational time (CT) but increased average travel time (ATT) and total travel delay (TTD).

1. Introduction

This paper develops and compares the performance of centralized and decentralized systems for optimal traffic signal control with short-term origin-destination (OD) demand as inputs. To add greater realism to the signal control optimization model, a standard ring-and-barrier diagram with phase plans and all-red intervals are used for intersections in the test grid network. The control variables are cycle length, green times of phases, and offset for each signalized intersection of the network. We formulate the optimization problem with a simulation-based queueing network model (QNM) to minimize average travel time (ATT) and total travel delay (TTD). The optimization of network signal control is nondeterministic polynomial time (NP) complete [1]. Thus, computational time (CT) is the point of comparison for the performances of the centralized signal control (CSC) and decentralized signal control (DSC)

systems. In addition, studies have shown network decomposition to have a significant impact on system performance [2, 3]. Therefore, we investigate the performance of CSC and DSC systems under different decomposition setups, using test scenarios constructed with varying demand profiles and grid networks.

We have constructed the framework of our DSC system within a connected vehicle (CV) environment using mobile edge computing (MEC) to ensure a low impact of data transmission latency and enable distributed computing for network signal control. Centralized systems are designed to solve the optimal control problem by processing all data inside a central traffic management center (TMC). However, without simplifying variables or constraints or relaxing the control objective, the optimal control CT is high, causing high computational latency for real-time control. Data transmission latency worsens as more devices are connected to the central TMC. By adding MEC technology, data can be

analyzed at the location where it is collected. Thus, traffic data can be transmitted through the network with low latency. In the DSC system, the network is divided into subnetworks, and each subnetwork has a dedicated agent to control signals inside it. Thus, the DSC system decomposes the network signal control problem in the CSC system, making it a distributed structure allowing reduced computational latency.

In addition, more types of data will soon be readily available thanks to the rapid development of CV and communication technologies. For example, origin-destination (OD) demand data can be generated from a variety of sources such as navigation systems (e.g., Google Maps), ride-sharing applications (e.g., Uber), and GPS-equipped vehicles; these data can include vehicle origin, destination, and start time. However, many recent studies for network signal control still use data from fixed sensors as inputs, such as loop detector vehicle counts. Traffic data collected by induction loop detectors and other standard sensors is limited because it can only be used to understand traffic states in a short period locally. However, OD demand, a widely-used input for network modeling, can help predict network-level traffic dynamics over a given period.

Our research objective is to present an analysis of centralized and decentralized signal control systems with short-term OD demand, investigating the effect of different network decompositions on the performance of both systems. This research contributes to understanding a simulation-based framework of DSC system within the MEC-enabled CV environment along with a scalable and extendable decomposition method for grid networks.

2. Background

Signal timing plans in urban transportation networks are optimized to increase network capacity utilization, improve mobility (including reducing traveler delays), and mitigate vehicle-based emissions, amongst other aims. Adaptive signal control (ASC) systems are widely used in many modern cities, from the earliest SCATS [4] and SCOOT [5] systems, typical intermediate systems as OPAC [6], RHODES [7], UTOPIA [8], and ASC Lite systems [9], to more recent adaptive fine-tuning (AFT) systems [10] and deep learning-based system [11]. Model predictive control (MPC) is developed by many researchers to solve network signal control in ASC systems. For a decentralized control design, a multiagent MPC system was developed for linear dynamic networks [12]. In addition, a distributed MPC had the advantage of flexibility in large networks over centralized strategies, but fared worse in computational performance [13]. These approaches all rely on traffic volume data from upstream links or queue lengths on each leg of an intersection. Traffic condition predictions from these types of data only reflect the local traffic state over a short period. Recently, Chow et al. [14] considered drivers' route choices under a user-optimal (UO) assumption in centralized and decentralized systems, showing the potential benefits of the knowledge we gain from combining traffic assignment and traffic signal control modeling.

Rapidly developing CV technologies that produce numerous types of vehicle, personal, travel, and geometric data offer the potential to resolve this data quality issue. For example, the 2017 SAE J2735 dedicated short range communications (DSRC) report summarizes 17 messages, 156 data frames, 230 data elements, and 58 external data element definition references [15]. In addition, Mobile Edge Computing (MEC) adds to these capabilities. Currently, the vehicle is not only the edge of data reception but also data collection. Researchers have already introduced a deep learning method for vehicular platoon control based on MEC analysis [16]. A recent study applied MEC to a new Internet of Vehicles (IoV) framework and developed a resource allocation algorithm to process high-dimensional data [17]. Finally, the introduction of 5G will help MEC to perform efficiently in these contexts. The inherent features of 5G, such as high connectivity, low latency, and large bandwidth, will facilitate the demanding computational and communication requirements of MEC [18]. The combination of CV, MEC and 5G technologies will provide high-quality data, stronger computational hardware, low-latency, and high-bandwidth communication technology. On this basis, we can develop more efficient signal control systems for network-level traffic and real-time control. Consequently, this study seeks to understand the performance of different network decompositions when fed OD demands as inputs, motivated by the possibilities of these new technologies.

Our two forms of signal control for comparison are centralized signal control (CSC) and decentralized signal control (DSC). CSC optimizes the parameters of each intersection in the network simultaneously to find an optimal control solution. Much of the research in this area has focused on the algorithms used in different networks towards a range of objectives. For instance, genetic algorithms (GA) and approximate dynamic programming (ADP) approaches were proposed for traffic signal control in oversaturated networks [19]. Another algorithmic solution for oversaturated networks was the ant colony optimization algorithm (ACO) [20]. Other researchers have used heuristic algorithms; for example, Beard and et al. [21] formulated a mixed-integer linear programming (MILP) model and used a heuristic algorithm to solve traffic signal control optimization.

Researchers studied these CSC cases for small networks. Escalating to even a medium-size network means that the complexity (NP-complete) and increased time and topological scale cause more significant difficulties [1]. Thus, strong assumptions are made to avoid complex variables and traffic dynamics. For example, Gregoire et al. [22] applied a backpressure algorithm (BP) to solve the network traffic signal control problem, ignoring travel times for each link and turning movement to keep the problem tractable. Prashanth and Bhatnagar [23] used reinforcement learning (RL) to generate and train data towards minimizing average cost for ASC. However, the application of the method results in the curse of dimensionality, where network signal control with RL remains an NP-complete problem. Although recent advances in computing offer some resources to solve

real-time, large-scale data collection and processing, the NP-complete problem of network-wide real-time signal control remains unresolved.

Prompted by the difficulties involved in CSC, research attention turned to distributed systems that decompose large networks into smaller subnetworks and optimize the parameters of all intersections in the subnetwork simultaneously and cooperatively. This form of distributed control is also referred to as decentralized signal control (DSC). Gokulan et al. [24] developed a distributed, multiagent-based approach for a traffic-responsive signal control system. Multiagents achieved the distributed signal control at each subnetwork. This method significantly reduced the complexity issue of network-level signal control. Chow et al. [3] recently developed and compared centralized and decentralized signal control systems with a BP model for a road network in Central London, UK. Their results showed that while the total travel delay was lower with a centralized system, a decentralized system reduced the computational time by 40%.

Reinforcement Learning (RL) also plays an essential role in DSC systems. Van der Pol and Oliehoek [25], for instance, consider explicit coordination mechanisms between learning agents using coordination graphs. Other researchers have used individual RL agents to control traffic signals in a multi-intersection network from two angles: without communication between RL agents as “game theory” [26], and with communication between agents as “neighborhoods” [27, 28]. The RL method is a powerful tool applied in real-world scenarios, such as the 2,510 traffic lights in Manhattan studied by [29]. However, Wei et al. [30] identified key challenges, two of which are critical. The first is that learning costs are too high for complex problems; the second is that risk management is not widely adopted, causing potential traffic safety issues in a real-world setting.

Network decomposition topology is an important consideration when constructing a framework and optimizing DSC performance in some research. Cell-based decomposition [31] and intersection-based decomposition [1, 32] are two common approaches, but both result in subnetworks with only one intersection each. Adacher and Tiriolo [33] addressed this by working with subnetworks containing more than one intersection. Their recent research using a clustering algorithm (CA) to solve the DSC problem based on a Cell Transmission (CTM) model highlighted the importance of spatial decompositions [2]. However, they tested one network that includes a specific group of subnetworks with a simplified signal phase, thereby offering limited results. A grouping method was developed to decrease delay and number of stops while minimizing traffic operators’ subjective decisions [34] and was applied to a one-way corridor network with 21 intersections in Montgomery County, Maryland. A principal component analysis (PCA) method was developed to dynamically group controllers into clusters, but only a few signalized intersections were considered within their tested traffic networks [35]. The question of how to best decompose a network to optimize signal control remains to be solved.

The above literature is summarized in Table 1.

Except for RL and CA, most methods use networks with up to 100 control variables. For model inputs, link flows are widely used, as in ASC systems. System-level objectives include delay, throughput, travel time, and queue length.

In summary, the components of information in the literature presented above inform our study. Considering that local counts are inputs for most existing ASC systems, new technologies such as CV and MEC will enable the collection of high-quality data and local data processing, thereby offering new possibilities for enriched data inputs. Yet, despite excellent algorithms and machine learning methods, centralized signal control optimization problems remain NP-complete without simplifying signal timing plans and their representation in models. Furthermore, both signal plan representation and model constraints must be kept relatively simple in order to apply algorithms such as BP and ACO. As such, we address the evident need to explore a new framework of DSC in large-scale networks considering greater realism of signal control, new forms of data inputs, and the optimization of network decomposition.

Our research used a standard ring-and-barrier signal timing plan and can consider up to 216 control variables in a 6×6 grid network. Unlike most previous work using existing sensor data sources as inputs for optimizing network signals in a current technological paradigm, we use short-term OD demand as inputs given that the technological paradigm we assume is within the connected, MEC-enabled environment. The decomposition method we use is scalable based on subnetwork size rather than set for a specific network. As such, our method is flexible for expansion to larger networks.

3. Network Description

Short-term origin-destination (OD) demand is assumed to be accessible given that the technological paradigm assumed is within the CV environment. We describe the network signal control problem with short-term OD demand input. A standard ring-and-barrier phase plan will be applied for each intersection inside the network, and each turning movement of a vehicle will be considered a virtual link with periodic travel cost function with respect to signal timing. The cycle length is the same for all signalized intersections. In addition, each traveler will choose the minimum instantaneous path while entering the network. Finally, a queueing process will describe the dynamics of traffic flow.

Table 2 contains notations and variables.

Three types of short-term demand—to be used in the remainder of this paper—are described below:

(1) Short-term demand for path

Short-term demand for path is defined as time-dependent flow entering a path, i.e., number of vehicles entering a path during one timestamp

It is notated as $d_p(1, t)$, where p is the path id, t is timestamp and 1 means it is the first link of the path

(2) Short-term demand for links on the path

TABLE 1: List of methods for network signal control.

Method	Author	Network size	Control variables	# of var (estimated)	Inputs	CSC/DSC	Objectives
GA	Hajbabaie [19]	4×4	Cycle, phase split, and offset	96	Upstream volume from gate signals	CSC	Min delay, travel time; max throughput, trips
ACO	Putha et al. [20]	4×5 one-way	Green time for throughput traffic	40	Link flows	CSC	Max throughput
MILP	Beard et al. [21]	1×2	Cycle, phase split, and offset	12	OD demand	CSC	Min total travel time
BP (CSC)	Gregoire et al. [22]	8×8	Phase type	64	Queue length	CSC	Min total queue length
BP (DSC)	Chow et al. [3]	Bloomsbury network (15 intersections)	Green time for throughput traffic	30	Link flows	CSC and DSC	Min total delay
RL	Chen et al. [29]	Manhattan network (2510 intersections)	Phase type	2510	Link flows	DSC	Min average travel time and max total throughput
CA	Adacher and Tiriolo [2]	Rome network (39 intersections)	Cycle, phase split, and offset	234	Path demand	DSC	Min total delay

TABLE 2: Notation and variables.

Variables	Description
ij	Directed link from node i to node j ; $ij \in A$
A	Set of links
$\{T_{ij}(t)\}_{ij \in A}$	Travel cost matrix at timestamp t
$st(ij, t)$	Signal waiting time for signalized link ij at timestamp t
$gt(ij, t)$	Green time remaining for signalized link ij at timestamp t
$ft(ij)$	Free-flow travel time for link ij
FFT	Matrix of free-flow travel time (FFT)
M	Set of intersections
x_m	Control variable vector for intersection m , where $x_m = (cl_m, g_m^1, g_m^2, g_m^3, g_m^4, of_m)$ consists of cycle length, green times, offset
X	Set of control variables for all intersections
od	Original-destination index, where o is origin node, and d is destination
OD	Set of od pairs
$d(od, t)$	Short-term demand for OD pair od at timestamp t
Δ	Set of short-term OD demand
T	Time horizon
P_{od}, P_s, P	Set of paths for od , for subnetwork s , and the whole network respectively
p	Path index, $p = (p(1), p(2), p(3), \dots, p(l_p))$ is a vector with ordered nodes on the path
l_p	Length of path p , i.e. the total number of nodes in path p
$tc(p, t)$	Instantaneous travel cost for path $p \in P$ at timestamp t
$d_p(k, t)$	Short-term demand at node $p(k)$ for path $p \in P$ at timestamp t
$d_p^s(k, t)$	Short-term demand at node $p(k)$ for path $p \in P_s$ in subnetwork s at timestamp t
w	Maximum discharging flow rate for each lane of the link
$L_{queue}(ij, t)$	Queue length for signalized link ij at timestamp t
$TA(\cdot, \cdot, \cdot)$	Function for the traffic assignment, inputs are short-term OD demand, signal control variables, and FFT, output is short-term demand for the first link of all paths
$Q(\cdot, \cdot)$	Function for queueing process, inputs are short-term demand for the first link of all the path and signal control variables, output is short-term demand for all paths.
s	Subnetwork index
S	Set of subnetwork indexes
$\Gamma_s(p, p_s)$	Path decomposition index

Suppose path p has l links, short-term demand for the k^{th} link on path p is defined as time-dependent flow on path p entering the k^{th} link, i.e., number of vehicles on path p enter k^{th} link during one timestamp

It is notated as $d_p(k, t)$, where p is the path id, t is timestamp and k means it is the k^{th} link of the path

Note: one link can be shared by many paths, but short-term demand for links on path p only counts flow on p .

Short-term demand for the same link on other paths may be different.

Short-term OD demand is defined as time-dependent flow entering from origin node o and heading to destination node d , i.e., number of vehicles traveling from origin node o to destination node d during one timestamp.

It is notated as $d(od, t)$ where $o d$ is the OD pair and t is the timestamp.

Three types of short-term demand are differed by their objectives, such as paths, links, and OD pairs. Since short-term demand for path is equal to short-term demand to enter the first link on the path, it is a special case of short-term demand for links on path. Set of short-term demand for all paths is contained in set of short-term demand for links on all paths. In addition, short-term OD demand is the input for DTA model to predict short-term demand for links on all paths in traffic network.

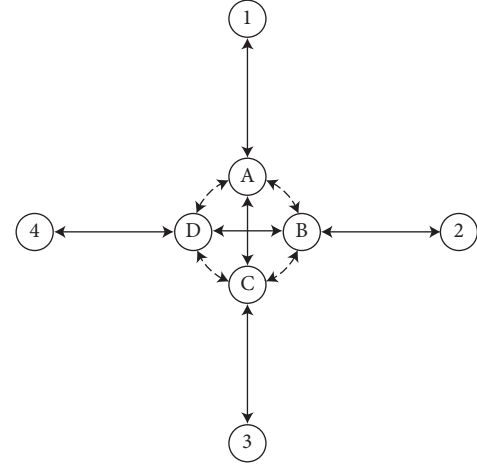


FIGURE 1: Sample of an intersection.

3.1. Travel Cost Matrix. The topology of the network is described using the travel cost matrix $\{T_{ij}(t)\}_{ij \in A}$, where ij is the link and t is a timestamp index. There are two types of links shown in Figure 1: signal links (dotted) and general links (solid). All links are two-way as per the arrow indications.

The travel time $T_{ij}(t)$ for a signalized link consists of signal wait time $st(ij, t)$ and turning time (also considered free-flow travel time $ft(ij)$ for signalized link), while that of general links consists of only free-flow travel time $ft(ij)$. $FFT = \{ft(ij)\}_{ij \in A}$ is the matrix of free-flow travel times for all links. If node i is not directly connected to node j , then ij is not a link, and the travel cost is infinite. Equation (1) describes the travel cost matrix for all links and the network topology.

$$T_{ij}(t) = \begin{cases} 0 & i = j, \\ \infty & ij \text{ is not a link,} \\ st(ij, t) + ft(ij) & ij \text{ is an intersection link,} \\ ft(ij) & \text{otherwise.} \end{cases} \quad (1)$$

3.2. Control Variables. Figure 2 shows a standard ring-and-barrier diagram for a 4-leg intersection signal control strategy. our numerical simulations use this phase plan with control variables $X = \{x_m = (cl_m, g_m^1, g_m^2, g_m^3, g_m^4, of_m)\}_{m \in M}$, where x_m is a vector consisting of cycle length, green times for the four phases and offset for signalized intersection m . Phase 1 is a protected left-turn phase for northbound and southbound movements; phase 2 is protected for northbound and southbound through and right movements; phases 3 and 4 repeat phases 1 and 2 for the eastbound and westbound directions. The bars between phases represent all-red intervals. Offset is between intersections on a corridor for the matching through-phase, based on a reference point or master reference.

Link ij is the signalized link for intersection m . Signal waiting time $st(ij, t)$ in equation (1) is computed according to the control variable x_m , which is a periodic function as per Figure 3. Similarly, we can get the green time remaining function $gt(ij, t)$ according to the control variable x_m .

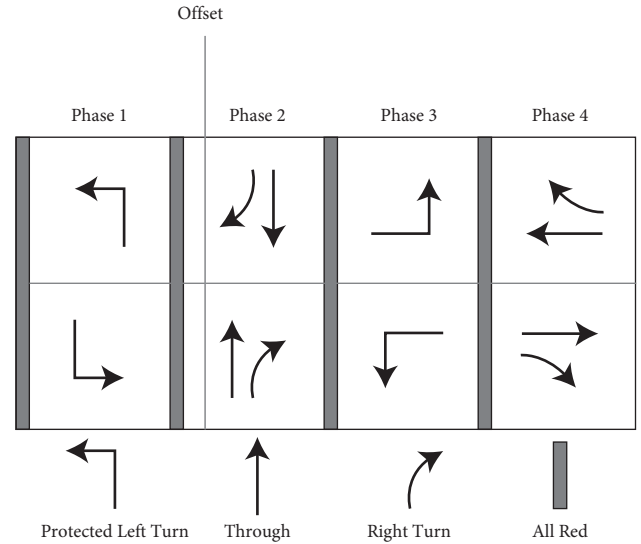


FIGURE 2: Standard ring-and-barrier diagram.

3.3. Origin-Destination (OD) Demand and Traffic Assignment. Short-term OD demand represents the traffic volume between each origin and destination pair entering the network in each timestamp. We use $\Delta = \{d(o d, t)\}_{od \in OD, t \in \mathbb{Z}in[1, T]}$ to represent short-term OD demand for the network during the period $[0, T]$. T is the total time horizon.

We use the “1-0” principle to assign short-term OD demand, i.e., a traveler will choose the path with the minimum instantaneous travel time. We also assume travelers will not change routes.

A path is defined as a vector with ordered nodes along the path:

$$p = (p(1), p(2), p(3), \dots, p(l_p)). \quad (2)$$

The path length, l_p , is quantified as the number of nodes on the path. The instantaneous travel time for path p at timestamp t is represented as follows:

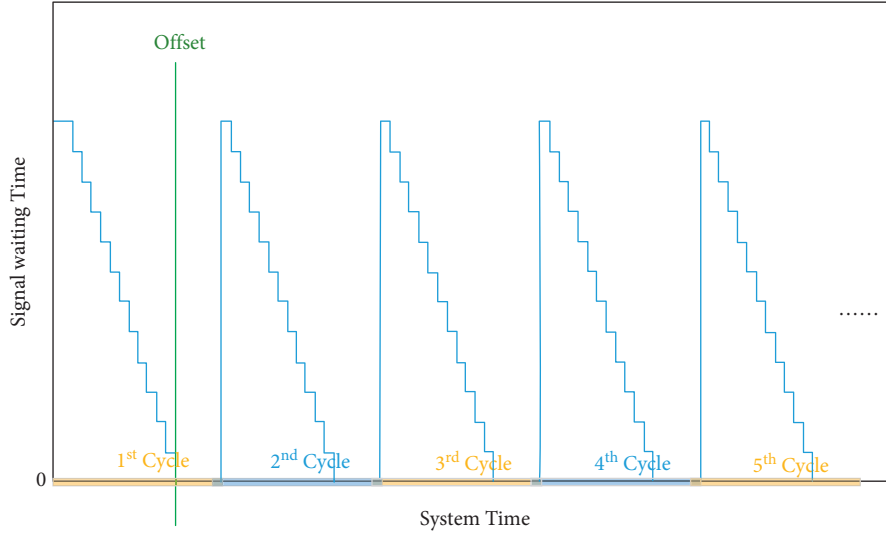


FIGURE 3: Sample of signal waiting time function.

$$tc(p, t) = \sum_{i=1}^{l(p)-1} T_{p(i)p(i+1)}(t). \quad (3)$$

The path with minimum instantaneous travel time for short-term demand $d(o, d, t)$ is expressed as

$$p_{od}^* = \arg_{p \in P_{od}} \min tc(p, t) := \{p_{od}^* \in P_{od} : tc(p_{od}^*, t) \leq tc(p, t), \forall p \in P_{od}\}. \quad (4)$$

Dijkstra's Shortest Path First (SPF) algorithm is used to search for equation (4).

Now we introduce demand for path $d_p(k, t)$ to describe the traffic flow dynamics. $d_p(k, t)$ represents traffic volume at node $p(k)$ for path p at timestamp t . We assign OD demand $d(o, d, t)$ to the path p_{od}^* with minimal instantaneous travel time (note that we use a small random error to make sure the cost of each path distinct):

$$d_{p_{od}^*}(1, t) = d(o, d, t). \quad (5)$$

We use $TA(\Delta, X, FFT)$ in equation (6) to describe the traffic assignment process that takes short-term OD demand, signal control variables, and matrix of FFT as inputs to estimate short-term demand for the first link of all paths.

$$\{d_p(1, t)\}_{p \in P = \cup_{o, d \in O, D} P_{o, d}, t \in Z \text{ in } [1, T]} = TA(\Delta, X, FFT). \quad (6)$$

3.4. Queuing Process. We use two basic assumptions for the queuing process.

- (1) Point queue: we assume that the upstream link holds the queue for each signalized link. The link has infinite vehicle storage. Vehicles exit the queue with a constant discharge rate w .
- (2) First in first out: the first vehicle that enters a link will also be the first to leave the link.

These two assumptions describe link dynamics in Dynamic Traffic Assignment (DTA) [36]. We use these assumptions here to generate traffic dynamics equations in the Queuing Network Model (QNM). The input for the queuing process is short-term demand for the first link of all paths $\{d_p(1, t)\}_{p \in P, t \in Z \text{ in } [1, T]}$ plus signal control variables X during the time horizon.

Figure 4 shows a sample link. The link has three lanes for left, right, and through movements, respectively. It is a general link followed by a signalized intersection. We assume that general links have three lanes (per direction) for each turning movement, while the downstream signalized links have one lane per direction.

Suppose the queue length on a signalized link at timestamp t is $L_{\text{queue}}(ij, t)$. Considering the figure describes the movement for short-term OD demand $d_p(k, t)$, then node $n_{\text{in}} = p(k)$, $n_{\text{out}} = p(k+1)$, and $n_{\text{next}} = p(k+2)$. In addition, after the time required to pass the link $ft(n_{\text{in}}, n_{\text{out}})$, vehicles will join the queue such that $L_{\text{queue}}(n_{\text{out}}, n_{\text{next}}, t + ft(n_{\text{in}}, n_{\text{out}}))$. $n_{\text{out}}, n_{\text{next}}$ represents the signalized link.

Suppose the signalized link $n_{\text{out}}, n_{\text{next}}$ belongs to the h^{th} phase for intersection m . According to the control variable $x_m = (cl_m, g_m^1, g_m^2, g_m^3, g_m^4, of_m)$, we obtain the following information:

- (1) Cycle length for the signalized link $n_{\text{out}}, n_{\text{next}}$ is cl_m
- (2) Green time of the phase for the signalized link $n_{\text{out}}, n_{\text{next}}$ is g_m^h

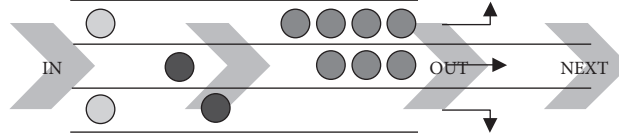


FIGURE 4: Sample link.

(3) Signal waiting time for the signalized link $n_{out}n_{next}$ is $st(n_{out}n_{next}, t)$

(4) Signal green time remaining for the signalized link $n_{out}n_{next}$ is $gt(n_{out}n_{next}, t)$

We then generate traffic dynamics via a close loop:

Step 0 (initialization): $t = 0, k = 1$, input demand profile for all paths

$$\{d_p(1, t)\}_{p \in P = \cup_{o,d \in O, D} P_{o,d}, t \in Z \text{ in } [1, T]} \quad (7)$$

Step 1 (demand update): $\forall p \in P = \cup_{o,d} P_{o,d}$,

$$\begin{aligned} n_{in} &= p(k), \\ n_{out} &= p(k+1), \\ n_{next} &= p(k+2). \end{aligned} \quad (8)$$

Find intersection m , which contains the signalized link $n_{out}n_{next}$.

if $\alpha = d_p(k, t) > 0$, then

$$L_{queue}(n_{out}n_{next}, t + \beta) = L_{queue}(n_{out}n_{next}, t + \beta - 1) + \alpha, \quad (9)$$

$$\beta = ft(n_{in}n_{out}). \quad (10)$$

Equation (8) selects the k^{th} link on path p . Link layout is the same as Figure 4. Equation (9) updates the queue length for each signalized link. β in equation (10) is the first part of the total queueing time—the free-flow travel time spent on link $n_{in}n_{out}$.

Calculate queueing time

Case 1: when the signal is green, $gt(n_{out}n_{next}, t + \beta) > 0$

$$\Delta t = \beta + \frac{L_{queue}(n_{out}n_{next}, t + \beta) - w \cdot gt(n_{out}n_{next}, t + \beta)}{w \cdot g_m^h} \times cl_m + mo d \left(\frac{L_{queue}(n_{out}n_{next}, t + \beta)}{w} - 1, g_m^h \right) + 1. \quad (11)$$

If a vehicle cannot pass the intersection during the green time in the current cycle of the signalized link, it will wait for another cycle for the next green phase. The time for the last vehicle in the queue to reach the front is $L_{queue}(n_{out}n_{next}, t + \beta)/w$. The maximum discharging flow rate w describes the maximum number of vehicles that can traverse the signalized link in one timestamp.

$\lceil \cdot \rceil$ is an operator calculating the ceiling of the inside value, which is a minimal integer greater than the value. For example, if $L_{queue}(n_{out}n_{next}, t + \beta) = 3$ and $w = 2$, then $L_{queue}(n_{out}n_{next}, t + \beta)/w = 3/2 = 2$. $mo d(a, b)$ calculates the remainder of a divided by b . For example, $mo d(7, 4) = 3$.

Case 2: when the signal is red, $gt(n_{out}n_{next}, t + \beta) = 0$

$$\Delta t = \beta + st(n_{out}n_{next}, t + \beta) + \frac{L_{queue}(n_{out}n_{next}, t + \beta)}{w \cdot g_m^h} - 1 \times cl_m + mod \left(\frac{L_{queue}(n_{out}n_{next}, t + \beta)}{w} - 1, g_m^h \right) + 1. \quad (12)$$

In equation (12), the additional term $st(n_{out}n_{next}, t + \beta)$ is the wait time (for the red signal to turn green).

Updating path demand function

$$d_p(k+1, t + \Delta t) = d_p(k+1, t + \Delta t) + \alpha. \quad (13)$$

After exiting the k^{th} link on path p , traffic will enter the $(k+1)^{th}$ link on path p . Since we use constant maximum discharging rate w , $\alpha \leq w$ for each iteration.

Step 2 (queue length update):

After the demand for all paths has been assigned, the queue length is updated.

$$L_{queue}(n_{out}n_{next}, t + \beta) = \begin{cases} L_{queue}(n_{out}n_{next}, t + \beta) & \text{if } T_{green}(n_{out}n_{next}, t + \beta) = 0, \\ L_{queue}(n_{out}n_{next}, t + \beta) - w & \text{if } T_{green}(n_{out}n_{next}, t + \beta) > 0. \end{cases} \quad (14)$$

From Step 1, we increased the length of the queue for each positive demand unit. When the signal is green, the queue length is reduced by the number of discharging vehicles; when the signal is red, the queue length will be the same as Step 1.

If $k < l(p) - 1$, $k = k + 1$, go to Step 1; otherwise, $k = 1$, go to Step 3;

Step 3 (stop condition):

If $t < T + \delta$, $t = t + 1$, go to Step 1; otherwise, Stop.

After all the path demand functions are updated for the current timestamp t , we start the next iteration for

timestamp $t + 1$. δ is extra time for all the vehicles exiting the network.

In summary, equations (8)–(14) describe the dynamic queueing process as $Q(\{d_p(1, t)\}_{p \in P, t \in Z \text{ in } [1, T]}, X)$ as shown in equation (15), which takes short-term demand on the first link for all paths and signal control variables as inputs, and short-term demand of each link for all paths as outputs. The short-term demand for a given path consists of the demand for each link for the path and timestamp.

$$\{d_p(k_p, t)\}_{k_p \in Z \text{ in } [1, l(p)], p \in P, t \in Z \text{ in } [1, T+\delta]} = Q(\{d_p(1, t)\}_{p \in P, t \in Z \text{ in } [1, T]}, X). \quad (15)$$

Since we use general distributions for the demand (the arrival rate) and service rate, finite servers (signalized links), and FIFO assumption, network traffic dynamic model in this work can be regarded as a G/G/n/FIFO Queueing Network Model (QNM). The QNM is also open since the number of customers (vehicles) is not fixed.

3.5. Summary. Network dynamics can be estimated using inputs including short-term OD demand, signal control variables (cycle length, phase split, and offsets) for all intersections, and link free-flow travel times. The data flow is shown in Figure 5.

The travel cost matrix is estimated at the first step with inputs free-flow travel time (FFT) and signal timing. Traffic assignments are based on the “1–0” principle that assumes travelers will choose the minimal instantaneous path. The SPF algorithm is used to generate the shortest path. Thus, OD demand is transferred into volumes for the first link of all paths. Then, the traffic dynamic is estimated within QNM, combined with signal control variables used in the initial step. This estimation is completed via simulation in MATLAB in Section 6.

4. Centralized Signal Control System

The previous section outlines how to predict traffic dynamics using signal control variables, short-term OD demand, and the FFT matrix within the G/G/n/FIFO open QNM. We continue to develop a centralized signal control (CSC) system to optimize network-wide signal control. We assume low data transmission latency within the MEC-enabled CV environment in both the CSC and DSC systems. We will count data transmission latency in our network signal control system in a future study after completing the device test in our Edmonton, Canada testbed. We formulate the objective functions using minimal average travel time (ATT) and total travel delay (TTD) and apply a three-step naive method to optimize the network traffic.

4.1. Problem Formulation

4.1.1. Minimal Average Travel Time. Average travel time (ATT) has been used as an objective or evaluation index in many network signal control studies, as shown in Table 1. ATT is the mean time spent by each vehicle traveling inside a network and is computed using the difference between a vehicle’s network entry and exit times, as in equation (16). However, we cannot generate trajectory data (time points of vehicles entering and exiting network) from short-term OD demand to calculate ATT.

$$ATT = \text{mean}(\Delta t_{veh}) = \frac{\sum_{i=1}^n (t_{veh_i}^{\text{exit}} - t_{veh_i}^{\text{enter}})}{n}. \quad (16)$$

Equation (17) simply rearranges the RHS of equation (16) to represent all vehicles’ mean exit and entrance time. Thus, ATT can be calculated by short-term demand of links for all paths as equation (18).

$$\frac{\sum_{i=1}^n (t_{veh_i}^{\text{exit}} - t_{veh_i}^{\text{enter}})}{n} = \frac{\sum_{i=1}^n t_{veh_i}^{\text{exit}}}{n} - \frac{\sum_{i=1}^n t_{veh_i}^{\text{enter}}}{n}, \quad (17)$$

$$ATT = \frac{\sum_{t=1}^{T+\delta} \sum_{p \in P} d_p(l(p), t) \times t}{n} - \frac{\sum_{t=1}^{T+\delta} \sum_{p \in P} d_p(1, t) \times t}{n}. \quad (18)$$

δ is the additional time required for all demand in T to exit the network. Equation (19) is the total number of vehicles—the sum of short-term OD demand over the time horizon.

$$n = \sum_{t=1}^T d(o, d, t). \quad (19)$$

From Section 3, the short-term demand for a path is generated from short-term OD demand Δ , control variable X , and FFT. We can write one function—equation (20)—to compute ATT from these three inputs.

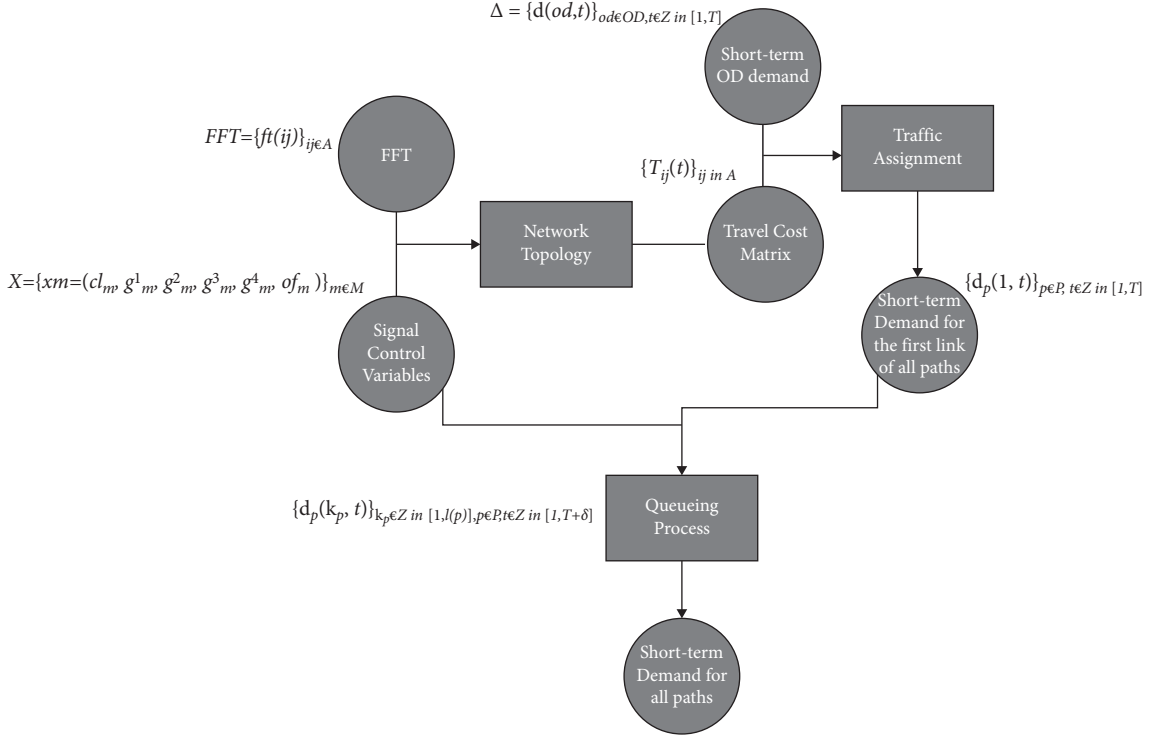


FIGURE 5: Data flow of network traffic dynamics prediction.

$$ATT = \varphi(\Delta, X, FFT). \quad (20)$$

$$\min_{X \in \Omega} ATT \text{ or } TTD$$

$$\text{Subject to : } ATT = \varphi(\Delta, X, FFT) \text{ or } TTD = \phi(\Delta, X, FFT), \quad (23)$$

4.1.2. Total Travel Delay. Total delay is the additional time all vehicles spend in the network due to signal waiting time, idling time in queues, and decelerating and accelerating. In this work, only signal waiting time and idling time in queue are considered due to the macroscopic representation of vehicle behavior.

We calculate total delay based on total queueing time with equations (9)–(13), while total queueing time is calculated as equation (21). Like equation (22), we have one function to compute TTD from the original inputs as equation (20).

$$TTD = \sum_{k \in Z \text{ in } [1, l(p)]} \sum_{p \in P} \sum_{t \in Z \text{ in } [1, T + \delta]} \Delta t \times \alpha, \quad (21)$$

where Δt and α were previously defined.

$$TTD = \phi(\Delta, X, FFT). \quad (22)$$

4.2. Centralized Signal Control. Suppose the cycle length is the same for all intersections, while phases split and offset for different intersections can vary with control objectives. In addition, assuming that an intersection signal's cycle length is the sum of green times, and offset is a positive integer less than the cycle length, the CSC problem can be formulated as follows:

where Ω is the feasible set for X ,

$$\Omega = \{X: cl_m = cl_n \in \Omega_{\text{cycle}}, (g_m^1, g_m^2, g_m^3, g_m^4) \in \Omega_{\text{phase}}, of_m \in \Omega_{\text{offset}}, \forall m, n \in M\} \Omega_{\text{cycle}} \subset Z_+ \text{ and } \Omega_{\text{offset}} \subset Z_+ \text{ are finite subsets of } Z_+. \Omega_{\text{phase}} \subset Z_+^4 \text{ is a finite subset of } Z_+^4.$$

Ω_{cycle} , Ω_{phase} , and Ω_{offset} are the feasible sets of cycle length, phase split, and offset for a signalized intersection. If the number of elements in each feasible set is n_{cycle} , n_{phase} , and n_{offset} respectively, then the size of the feasible set Ω is $|\Omega| = n_{\text{cycle}} \times n_{\text{phase}}^{4|M|} \times n_{\text{offset}}^{4|M|}$. The total number of intersections inside the network is $|M|$.

The control variables are combinations of cycle length, phase splits and offset, which do not have convex relationships. So, the feasible set is not convex, and constraint (21) is nonlinear since it is derived from equations (17)–(22) to describe the network level index of complicated traffic dynamics. Network signal control is known to be an NP-complete problem [1] such that no method currently exists to solve the problem without simplifying variables or

constraints or relaxing the control objective when the network size is relatively large. We now attempt to address this issue.

4.3. Three-Step Naïve Method. As there is no suitable algorithm available to solve the optimization problem with short-term OD demand as inputs, we propose a simple Three-Step Naïve Method. This method is transformed from the basic widely used traffic signal timing coordination [37]. It generates optimal solution for each subproblem which is very close to the overall optimal solution for Problem (A).

After choosing ATT or TTD as objectives, this method decomposes the original Problem (A) into three subproblems A1, A2, and A3. The simple method is described as follows:

Step 1: solve problem (27), optimal cycle length: fix phase splits and offsets for all intersections and find the optimal cycle length as per the optimization objective.

Step 2: solve problem (32), optimal phase split: based on the optimal cycle length in Step 1, fix the offset and generate all combinations of phase splits, then find the optimal phase split as per the optimization objective.

Step 3: solve problem (40), optimal offset: based on the first two steps, generate all the combinations of offsets for all intersections and find the optimal offsets plan as per the optimization objective.

Problem (27, 32, 40) is listed as follows:

$$\min_{X \in \Omega_1} ATT \text{ or } TTD, \quad (27)$$

Subject to: $ATT = \varphi(\Delta, X, FFT)$ or TTD

$$= \phi(\Delta, X, FFT), \quad (28)$$

$$g_m^k = \frac{cl_m}{4}, \forall m \in M, \forall k \in \{1, 2, 3, 4\}, \quad (29)$$

$$cl_m = cl_n, \forall m, n \in M, \quad (30)$$

$$of_m = 0, \forall m \in M, \quad (31)$$

where $\Omega_1 = \{X: cl_m = cl_n \in \Omega_{\text{cycle}}, g_m^h = cl_m/4, of_m = 0, \forall m, n \in M, \forall h \in \{1, 2, 3, 4\}\}$.

ATT or TTD is a convex curve with respect to cycle length, and we can solve problem (27) via the Bisection Algorithm (BA). If cl^* is the optimal cycle length from problem (27), then we formulate problem (32) as follows:

$$\min_{X \in \Omega_2} ATT \text{ or } TTD, \quad (32)$$

Subject to: $ATT = \varphi(\Delta, X, FFT)$ or TTD

$$= \phi(\Delta, X, FFT), \quad (33)$$

$$\sum_{k=1}^4 g_m^k = cl^*, \forall m \in M, \quad (34)$$

$$of_m = 0, \forall m \in M, \quad (35)$$

where $\Omega_2 = \{X: c, l_m = cl^*, (g_m^1, g_m^2, g_m^3, g_m^4) \in \Omega_{\text{phase}}, of_m = 0, \forall m \in M, \forall h \in \{1, 2, 3, 4\}\}$.

Although problem (32) is simpler than Problem (A), the feasible set of the problem is still large and challenging to solve. Although some Genetic Algorithms (GA) may be applied to the problem, they cannot guarantee the solution is optimal, and therefore we use a global search.

Suppose $\{(g_m^{1*}, g_m^{2*}, g_m^{3*}, g_m^{4*})\}_{m \in M}$ is the optimal phase split for all intersections from problem (32), then problem (40) is formulated as follows:

$$\min_{X \in \Omega_3} ATT \text{ or } TTD, \quad (36)$$

Subject to: $ATT = \varphi(\Delta, X, FFT)$ or TTD

$$= \phi(\Delta, X, FFT), \quad (37)$$

$$0 \leq of_m < cl^*, \forall m \in M, \quad (38)$$

where $\Omega_3 = \{X: cl_m = cl^*, (g_m^1, g_m^2, g_m^3, g_m^4) = (g_m^{1*}, g_m^{2*}, g_m^{3*}, g_m^{4*}), of_m \in \Omega_{\text{offset}}, \forall m \in M\}$.

The feasible set of this problem is convex, and thus we use a gradient search approach with a constant step size to update offset vector value.

If there are n intersections, we generate the control variable as a $n \times 6$ matrix \tilde{X} . Each row represents the signal control variable x_m , $m = 1, 2, 3, \dots, n$. The last column \tilde{X}_{of} is a vector of offsets for all intersections, which is the control variable for problem (40).

We update the control variable according to equations (38) and (39).

$$\tilde{X}_{of}^{(i+1)} = \tilde{X}_{of}^{(i)} + \alpha d^{(i)}, \quad (39)$$

$$d^{(i)} = \arg_{d^{(i)} \in D} \min \left\{ \tilde{X}_{of}^{(i)} + \alpha d^{(i)} \right\}, \quad (40)$$

where $D = \{e_1, e_2, \dots, e_n\}$ is the standard basis for vector space R^n , $\alpha = cl^*/\gamma$, and γ is a constant value which is a divisor of cl^* .

Finally, we will get the solution for the network-wide signal optimization as

$$X^* = \{x_m^* = (cl^*, g_m^{1*}, g_m^{2*}, g_m^{3*}, g_m^{4*}, of_m^*)\}_{m \in M}. \quad (41)$$

4.4. Summary. This section formulated the optimization problem for centralized signal control (CSC) with short-term OD demand for a network. We first introduced ATT and TTD as the control objectives with inputs as short-term demands for all paths. Because this optimization problem is NP-complete due to nonconvex variables and nonlinear constraints, we decomposed the problem with a three-step Naïve Method whereby we solve smaller subproblems one by one. However, this problem remains costly to solve (which we demonstrate in Section 6) without developing a new algorithm for the problem.

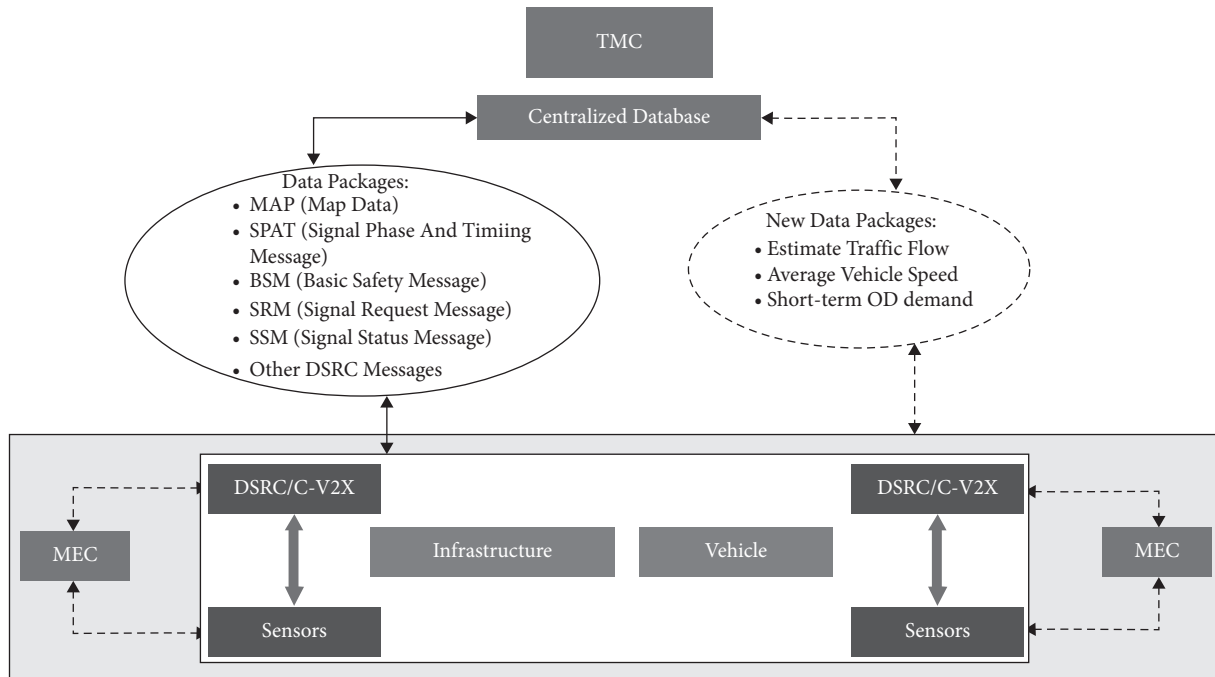


FIGURE 6: CV environment with/without MEC.

5. Decentralized Signal Control System

Decentralized signal control (DSC) is also referred to as distributed control, and its development is motivated by the difficulty of solving CSC. Thus, we develop a DSC system with short-term OD demand as inputs to control network traffic. Section 5.1 introduces the CV environment with Mobile Edge Computing (MEC), Section 5.2 proposes the decomposition method, and Section 5.3 describes the two-layer process for our DSC system.

5.1. Connected Vehicle (CV) Environment. A central traffic management center manages traditional signal control systems. Although data can be transferred through each end of the traffic network in the CV environment, the traditional framework has limitations resulting from the latency and bandwidth during data processing. With the help of MEC, optimization calculations can be completed at the location where data is generated. This benefits network signal control systems by reducing data transmission latency and enables distributed computing for DSC. Like the CSC system, we assume data transmission latency is too low to be counted in a MEC-enabled CV environment in the DSC system. In addition, we focus more on computational efficiency and network performance when comparing CSC and DSC.

Figure 6 compares CV environments with and without MEC for traffic control.

Communication in a CV environment, both with MEC and without MEC, includes three major parts: the TMC, infrastructure and vehicles. In traditional ITS architecture without MEC, all data is sent to and processed at a centralized TMC. However, the limited bandwidth of the backhaul connection between the centralized TMC

(Central-TMC) and local infrastructure will be rapidly exhausted by the volume of data gathered by the increasing number of sensors in the ITS. Likewise, the latency will be unacceptable for CV applications that require large amounts of data.

After introducing MEC into the framework, data can be processed, and applications can be run on any local edge unit, regardless of whether it is an infrastructure edge or vehicle edge. Thus, the heavy computation workload for the central TMC can be evenly distributed to the lower-level mobile edge computers, increasing computation capacity and proximity for real-time data processing.

This CV environment was implemented in a testbed in Edmonton, Canada, with all settings to be applied gradually within the next four years. MEC devices are located at each intersection of the network, collecting and delivering data and controlling the traffic signal. These MEC devices are called local MEC and include a computing server, roadside equipment (RSE), and a controller. Each vehicle has onboard equipment (OBE), and the Central TMC has a cloud computing server.

In addition, we decompose the whole network into subnetworks with several intersections. Each subnetwork has a regional MEC controller. Each regional MEC controller is designed to receive messages from surrounding regional MEC controllers, make decisions for the traffic signal control of intersections inside the subnetwork and send messages to surrounding intersections.

5.2. Network Decomposition. Network decomposition is highlighted in recent DSC research [2] as an important factor for optimizing DSC system. In this work, a grid network is decomposed into grid subnetworks. Besides

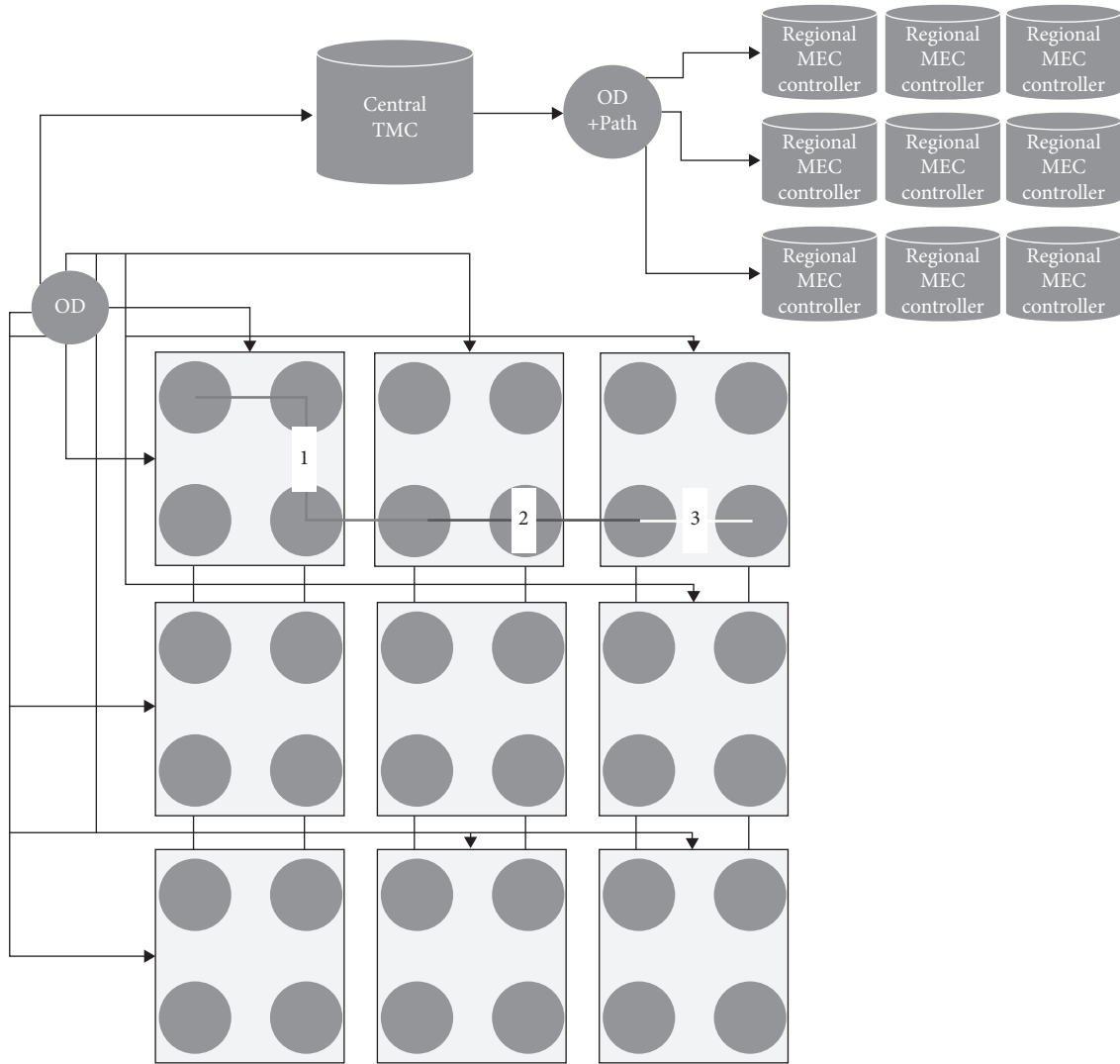


FIGURE 7: An example of network decomposition.

spatial decomposition, we also decompose demand for paths of the whole network into demand for paths of each subnetwork. Thus, each subnetwork has an individual short-term demand profile to optimize network signals inside it as a small CSC system.

Figure 7 shows an example of a 6×6 network decomposed into nine 2×2 subnetworks.

After short-term OD demand is collected at each end of the network, the Central TMC will distribute OD and path information to regional MECs in subnetworks. As assumed in Section 3, travelers will not reroute on their way. The path from the node in the top left corner to the second-to-top right node, illustrated in the figure, is divided into three subpaths through three subnetworks marked as 1, 2, and 3.

Path decomposition is achieved by the Path Decomposition Index (PDI) $\Gamma_s(p, p_s)$, where s is the index for the subnetwork, p is path for the whole network, and p_s is path for subnetwork s . $\Gamma_s(p, p_s) = m$ if the m^{th} node of path p is the first node of path p_s for subnetwork s . For example, suppose the index for the top middle subnetwork in Figure 7

is 2. For path p shown, since the 4^{th} node for the whole path p is the first node of the subpath p_2 inside the subnetwork, we have $\Gamma_2(p, p_2) = 4$.

Say $d_p^s(k, t)$ is defined as short-term demand at node $p(k)$ for path $p \in P_s$ in subnetwork s at timestamp t . Suppose $\Gamma_s(p, p_s) = m$. Then,

$$d_{p_s}^s(1, t) = d_p(m, t). \quad (42)$$

5.3. Two-Layer Process. With the introduction of a CV environment with MEC and network decomposition with PDI, Figure 8 illustrates the two-layer process for a DSC system with short-term OD demand used for this work.

In the first layer, the input is the short-term OD matrix. The central-TMC will gather this information from the regional MEC controllers. The output from this first layer—the short-term demand for each path—is input to the second layer. The second layer has two functions: traffic dynamics (traffic state prediction) on regional MEC

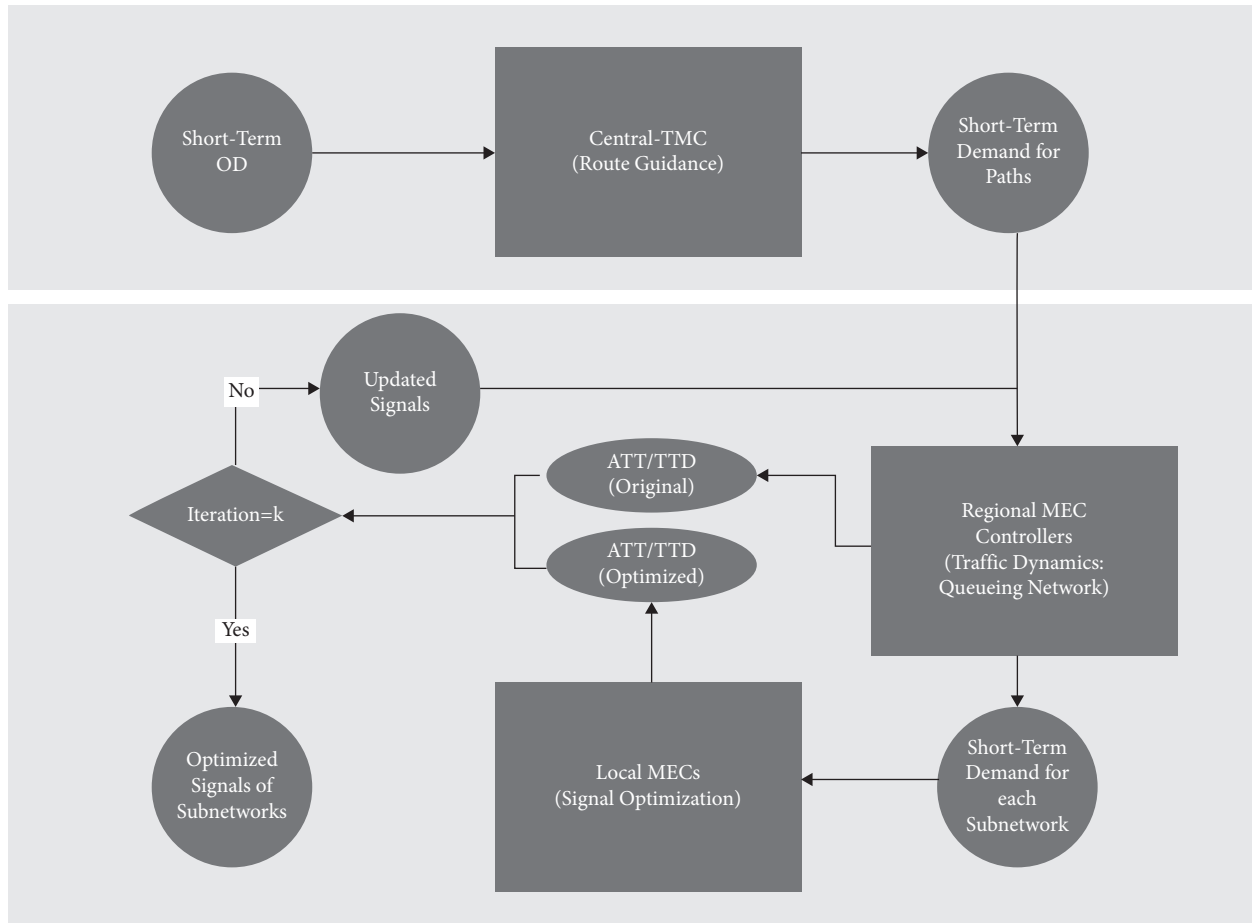


FIGURE 8: Two-layer process for decentralized signal control system.

controllers and signal optimization on local MECs inside each subnetwork.

Each subnetwork can be considered a small CSC system since the signal optimization is achieved via the three-step naïve method, the same as for CSC systems. In addition, traffic dynamics are achieved using the same queueing process. Since the network traffic reacts to changes in signal timing plans, we repredict traffic dynamics after updating signals. This loop will run until the convergent or iteration number is up to a preset constant integer k .

In the CSC system, all three functions are run inside the cloud server of central TMC. However, in the DSC system, the sole function of the Central-TMC is to calculate short-term path demand to provide route guidance. The remaining optimization tasks are allocated to the regional MEC controller inside each subnetwork. In addition, local MECs in the subnetwork are involved in the computation of the signal optimization function. All the computation resources of the DSC system are fully used.

5.4. Summary. This section introduced the environment and decomposition method of the DSC system. The structure of the DSC system was described as a two-layer process.

The DSC system has three main functions: route guidance in central TMC, traffic dynamics prediction in regional

MEC controller, and signal optimizations with local MECs in each subnetwork. PDI plays a vital role in allocating sub-jobs to each regional MEC controller. Considering that each subnetwork functions as a small CSC system, the DSC system, as outlined here, is an extension of the CSC system that enables network expansion.

6. Numerical Simulation

We use MATLAB for our numerical simulation, with computation completed in the Compute Canada server. The maximum number of cores is 40, and the speed for each core is 2.4 GHz. Both the CSC and DSC systems are demonstrated below in five scenarios. We test 1×1 , 2×2 , 3×3 , 4×4 , 6×6 grid networks as illustrated in Figure 9.

The basic settings for all cases are as follows:

- (1) Time horizon: $T = 300$ time units, demand is randomly generated from a uniform distribution (variance equals mean value) for OD pairs within the time horizon with demand interval as 1 time unit.
- (2) Free-flow travel time for all general links: 10 time units.
- (3) Left turn, right turn, and through for signalized links cost 3, 2, and 1 time units, respectively. All-red time for signal control is 2 time units. The discharge rate

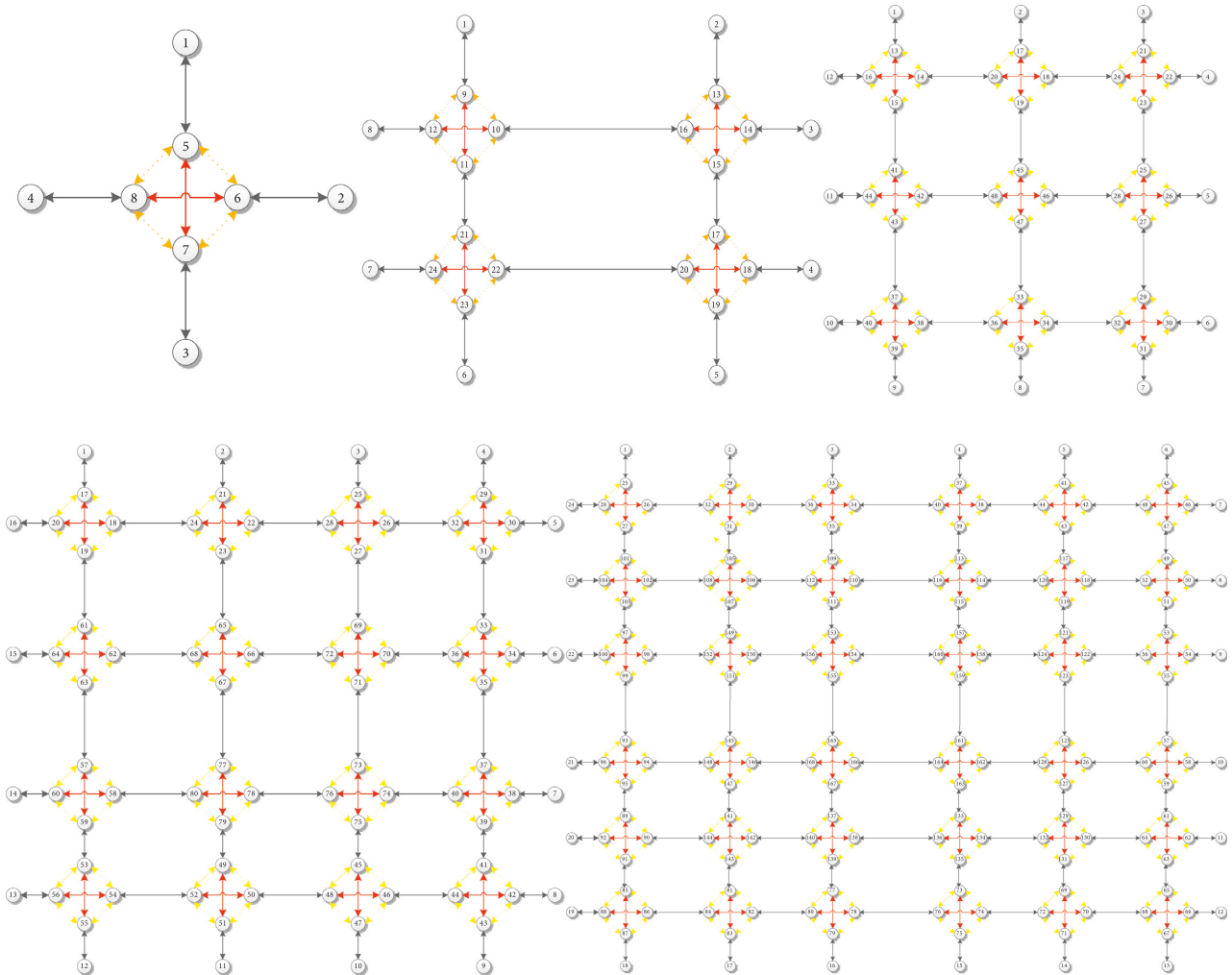


FIGURE 9: Grid networks.

for each signalized link is $w = 2$ vehicles per time unit.

- (4) A standard ring-and-barrier diagram (Figure 2) is used for all intersection signal timings.
- (5) Cycle length is the same for all intersections. Phase splits (i.e., green times) and offsets may vary among different intersections. The unit of control variables and two objectives are in time units.

For DSC-1 cases, the subnetwork size is 1×1 with one intersection. For DSC-2 cases, the subnetwork size is 2×2 with four intersections. If we consider the CSC case as decomposing the whole network into one subnetwork, then CSC, DSC-1, and DSC-2 are regarded as different network decompositions for the same network.

6.1. Scenario 1: CSC, 2×2 Network. Scenario 1 consists of CSC for a 2×2 network. We show the results of the Three-Step Naïve Method step by step.

6.1.1. Cycle Length. The first step is to optimize the cycle. We consider five different demand levels (levels 1–5): 560, 1120, 1680, 2240, 2800 total vehicles over the time horizon with interval as 1 time unit. Results are shown in Figure 10.

Both ATT and TTD are convex, such that we can use the Bisection Method to search for the local optimal point of the cycle length. As demand increases, the optimal cycle length increases as well.

6.1.2. Phase Split. Demand level 3 has an optimal cycle length $cl^* = 48$. The offset is fixed as the initial value 0, and we solve the problem via the global search method.

Set the feasible set of phase splits for problem (32) as

$$\Omega_{\text{phase}} = \{(12\ 12\ 12\ 12), (16\ 16\ 8\ 8), (16\ 8\ 16\ 8), (16\ 8\ 8\ 16), (8\ 16\ 16\ 8), (8\ 16\ 8\ 16), (8\ 8\ 16\ 16)\}. \tag{43}$$

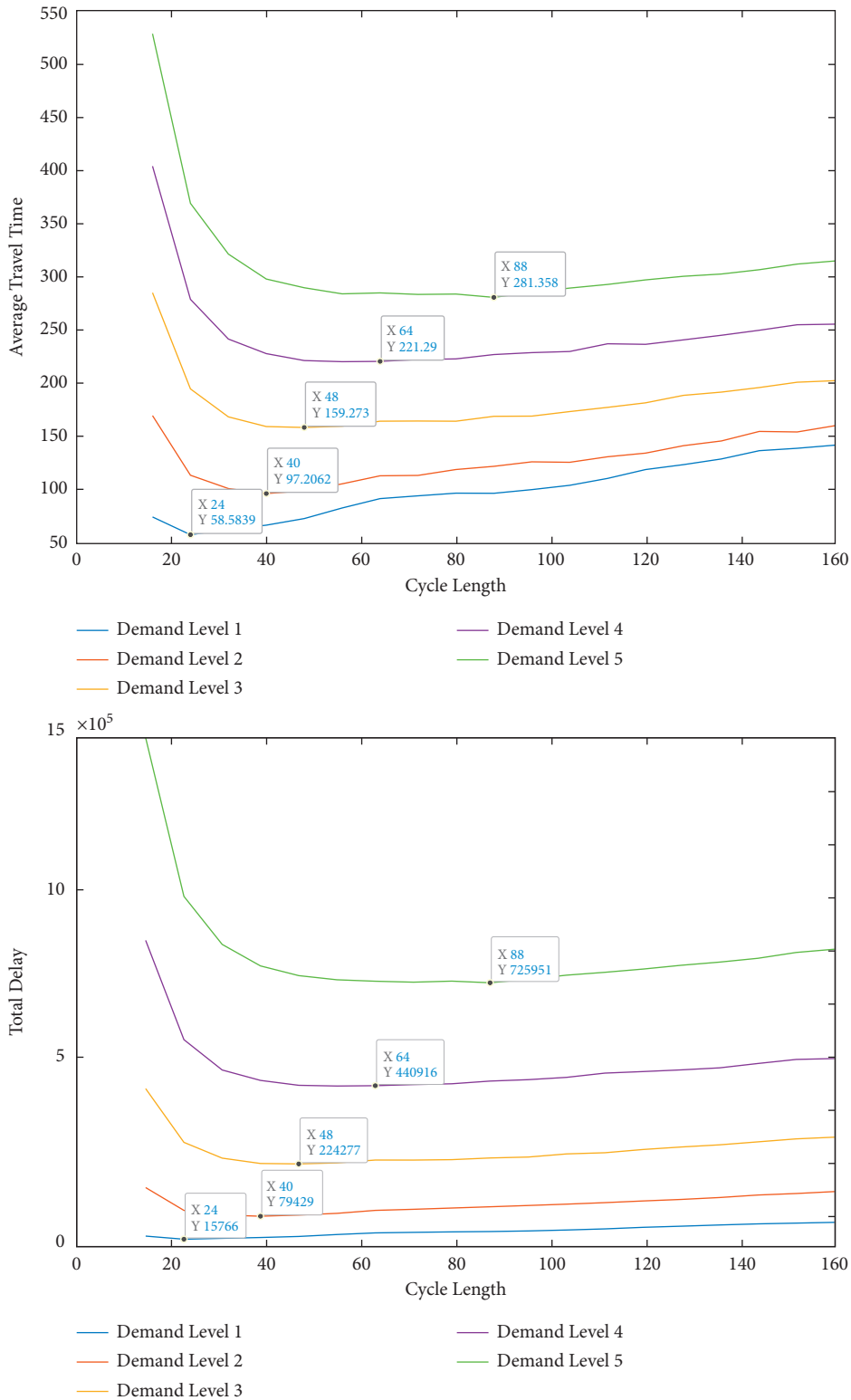


FIGURE 10: ATT and TTD under different demand levels.

The optimal phase is $g_m^* = (12\ 12\ 12\ 12)$, $m = 1, 2, 3, 4$, which results from the uniformly distributed demand. The minimal $ATT = 159.27$ and $TTD = 2.24 \times 10^5$.

6.1.3. *Offset.* Continue with the values above for cl^* , g_m^* , and m . Set $\gamma = 6$, then the step size for the gradient search approach $\alpha = cl^*/\gamma = 8$. The resulting optimal offsets are

TABLE 3: Iterations of scenario 2.

		1			2			3			4			5			
# of iteration		180.64	144.67	133.38	133.35	141.16	144.67	133.38	133.35	144.67	133.38	133.35	141.16	144.67	133.38	133.35	
ATT		4.31E+04	3.04E+04	2.65E+04	2.65E+04	2.96E+04	3.04E+04	2.65E+04	2.65E+04	3.04E+04	2.65E+04	2.65E+04	2.96E+04	3.04E+04	2.65E+04	2.65E+04	
TTD		60	60	60	60	60	60	60	60	60	60	60	60	60	60	60	
Cycle length																	
	Network Size	DSC-1 (subnetwork size is 1 x 1)			DSC-2 (subnetwork size is 2 x 2)			CSC (no subnetwork)									
	Demand Level	Lv. 1	Lv. 2	Lv. 3	Lv. 1	Lv. 2	Lv. 3	Lv. 1	Lv. 2	Lv. 3	Lv. 1	Lv. 2	Lv. 3	Lv. 1	Lv. 2	Lv. 3	Lv. 3
	ATT													26.76	24.42	19.68	
	TTD													2.04E+03	3.54E+03	4.46E+03	
	CT (sec)													0.55	0.64	0.65	
	ATT	71.98	74.69	71.73										64.89	68.39	64.84	
	TTD	7.44E+03	1.63E+04	2.46E+04										7.32E+03	1.43E+04	2.29E+04	
	CT (sec)	6.28	5.51	5.52										14.36	14.76	14.51	
	ATT	105.38	110.81	117.91										100.55	108.08	112.86	
	TTD	1.70E+04	3.39E+04	6.03E+04										1.17E+04	2.18E+04	3.80E+04	
	CT (sec)	16.44	16.01	16.13										6023.88	5954.00	6607.22	
	ATT	135.25	176.37	170.77	129.46	164.24	169.21										
	TTD	2.74E+04	8.98E+04	1.26E+05	2.51E+04	8.13E+04	1.26E+05										
	CT (sec)	45.25	44.17	44.81	299.63	303.61	299.01										
	ATT	234.17	299.44	285.00	210.02	281.35	245.14										
	TTD	8.86E+04	2.34E+05	2.50E+05	7.47E+04	2.00E+05	2.15E+05										
	CT (sec)	287.56	286.19	283.41	772.50	769.49	521.23										

Phase split and offset in each iteration (Each row is green times of 4 phases, and offset for each intersection)

$of_1 = 16$, $of_2 = 24$, $of_3 = 32$, $of_4 = 0$, final $ATT = 154.74$, and $TTD = 2.19 \times 10^5$. ATT is reduced by 2.84% and TTD by 2.33% from the previous step. $CT = 32.45$ seconds.

The results show that the Three-Step Naïve method can solve the CSC problem with OD demand as inputs for a 2×2 network. In addition, the bisection, global search, and gradient search methods are applicable for Step 1, Step 2, and Step 3, respectively. CT is acceptable for the time horizon $T = 300$ time units.

6.2. Scenario 2: DSC-2, 4×4 Network. Scenario 2 shows the two-layer process of the DSC system. The example used is a DSC-2 case for a 4×4 network. The stop iteration number is 5. The cycle length is fixed to 60. There are 351 vehicles over the time horizon $T = 300$ with demand time interval as 1 time unit. The number of CPU cores is 16. $\gamma = 6$ for the Three-step Naïve Method. Results for each iteration are shown in Table 3.

The optimal solution results in $ATT = 133.35$ and $TTD = 2.65 \times 10^4$. In this scenario, $CT = 151.81$ secs. From the first iteration, ATT is reduced by 22.87%, while TTD is reduced by 38.52%.

Table 3 also shows each iteration in the DSC case in the last row. The remaining scenarios measure the impact of CPU core numbers, stop iteration numbers, and different network decompositions based on the performance measures used (ATT , TTD , and CT).

6.3. CPU Cores: CSC, 2×2 . We tested the number of cores to observe the impact on CT . We used a CSC case with a 2×2 network, with 192 vehicles over the time horizon with demand time interval as 1 time unit.

Figure 11 shows that when the number of CPUs increases, CT will decrease from over 250 seconds to around 10 seconds. However, the curve is convex, which means the marginal benefit decreases with respect to the number of CPU cores. This decrease demonstrates the computational limitations of a centralized system even with a powerful computing cloud server.

6.4. Iteration Number: DSC-1, 2×2 . Here we demonstrate how the number of iterations impacts DSC system performance, using a 2×2 network decomposed into four 1×1 subnetworks. There are 192 vehicles over the time horizon with demand time interval as 1 time unit, the same demand as in Scenario 3.

The objective function converges as the number of iterations increase (Figure 12), evidence of the convergence for the two-layer process. In other words, there exists an equilibrium between demand and traffic signal optimization. The two objectives, ATT and TTD , correlate since they have similar trends when the iteration number increases.

6.5. Network Decompositions. Scenario 5 is designed to compare different decompositions under different short-term OD demand profiles. The demand input is shown in Table 4. Demand is increased from level 1 to level 3.

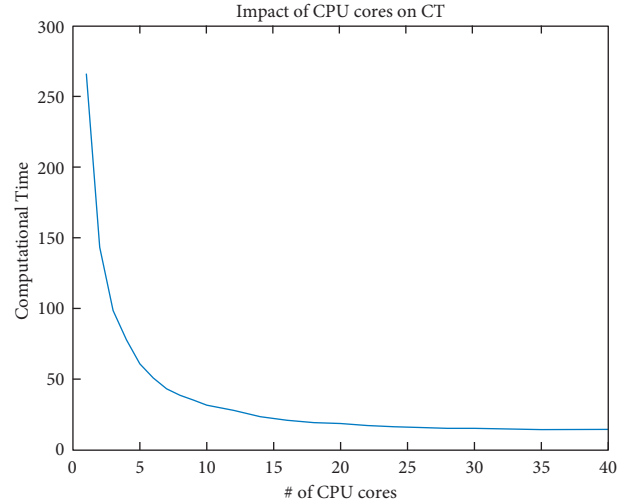


FIGURE 11: Impact of CPU cores on CT .

Short-term OD Demand is generated randomly following uniform distribution over OD pairs and the time horizon with demand interval as 1 time unit. The mean value and standard variance are $\tau, 2\tau, 3\tau$ /od for the three levels, respectively. τ is the adjust coefficient for different network sizes to balance saturation rate under the same demand level for all the networks. The number of control variables for one intersection is 6, so the maximum number of control variables in the simulation is 216 for the 6×6 network.

Since the primary calculation of the system is in the local layer of the two-layer process, and the local MEC has a computing unit located at each intersection, the total number of cores used for the DSC system is equal to the number of intersections inside the network. Limited to the test environment, we will use clusters with the same number of CPU cores to intersections instead. The numbers of cores used for DSC cases are 1, 4, 9, 16, 36 for 1×1 , 2×2 , 3×3 , 4×4 , 6×6 networks, respectively. Since CSC system computation is supposed to be completed in the cloud server, 40 cores are used.

Table 5 outlines the results for Scenario 5. Each case is the result of 10 iterations.

Table 5 outlines the case for CSC only since the network 1×1 only has one intersection. In addition, networks 2×2 and 3×3 cannot be divided into more 2×2 subnetworks. The DSC-2 case is not possible for either network. Since CT increases exponentially as the CSC case network size increases, we do not have CSC cases for 4×4 and 6×6 networks. CT for the CSC case of the 3×3 network is ~ 1 h 30 min, which is high for any real-time application, thereby making it problematic for a larger-sized network in these conditions.

In Table 5, the best value among different demand levels for cases of the same network and decomposition (DSC-1, DSC-2, or CSC) is shaded in grey. For TTD , the best case is always the one with the lowest demand. Since demand reflects the total number of vehicles in the network, the resulting TTD is correlated. However, for ATT , the best case is not consistent with demand level. For example, the best

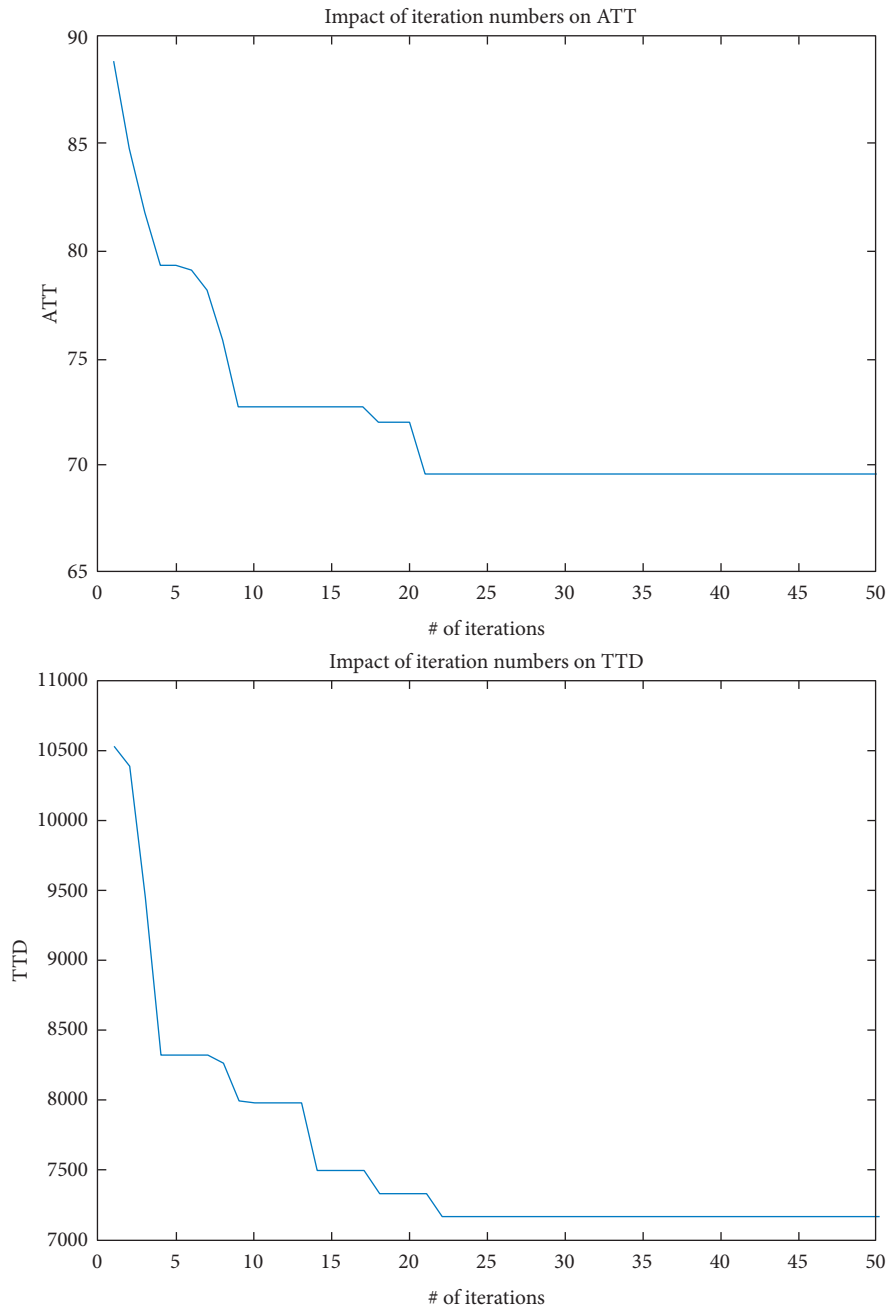


FIGURE 12: Of iterations on the performance of DSC system.

TABLE 4: Demand profile of scenario 5.

Network size	Level 1 (τ/od)	Level 2 ($2\tau/\text{od}$)	Level 3 ($3\tau/\text{od}$)	τ	# of intersections	# of control variables
1×1	96	168	228	12	1	6
2×2	192	360	552	6	4	24
3×3	284	520	816	4	9	54
4×4	351	753	1089	3	16	96
6×6	574	1082	1205	2	36	216

case is demand level 3 for network 1×1 in CSC cases and network 2×2 in both the DSC-1 and CSC cases due to the varying distribution of demand profiles. For larger networks, cases with lowest demand have the best ATT and TTD

because larger networks seem to be oversaturated in the high demand level as compared to smaller networks. For example, ATTs in both DSC-1 and DSC-2 cases of network 6×6 increase by 30% approximately from demand levels 1 to 2. In

TABLE 5: Test results of scenario 5 (ATT and TTD are measured in time units.).

Network size	Demand level	DSC-1 (subnetwork size is 1×1)			DSC-2 (subnetwork size is 2×2)			CSC (no subnetwork)		
		Lv. 1	Lv. 2	Lv. 3	Lv. 1	Lv. 2	Lv. 3	Lv. 1	Lv. 2	Lv. 3
1×1	ATT	—	—	—	—	—	—	26.76	24.42	19.68
	TTD	—	—	—	—	—	—	$2.04E+03$	$3.54E+03$	$4.46E+03$
	CT (sec)	—	—	—	—	—	—	0.55	0.64	0.65
2×2	ATT	71.98	74.69	71.73	—	—	—	64.89	68.39	64.84
	TTD	$7.44E+03$	$1.63E+04$	$2.46E+04$	—	—	—	$7.32E+03$	$1.43E+04$	$2.29E+04$
	CT (sec)	6.28	5.51	5.52	—	—	—	14.36	14.76	14.51
3×3	ATT	105.38	110.81	117.91	—	—	—	100.55	108.08	112.86
	TTD	$1.70E+04$	$3.39E+04$	$6.03E+04$	—	—	—	$1.17E+04$	$2.18E+04$	$3.80E+04$
	CT (sec)	16.44	16.01	16.13	—	—	—	6023.88	5954.00	6607.22
4×4	ATT	135.25	176.37	170.77	129.46	164.24	169.21	—	—	—
	TTD	$2.74E+04$	$8.98E+04$	$1.26E+05$	$2.51E+04$	$8.13E+04$	$1.26E+05$	—	—	—
	CT (sec)	45.25	44.17	44.81	299.63	303.61	299.01	—	—	—
6×6	ATT	234.17	299.44	285.00	210.02	281.35	245.14	—	—	—
	TTD	$8.86E+04$	$2.34E+05$	$2.50E+05$	$7.47E+04$	$2.00E+05$	$2.15E+05$	—	—	—
	CT (sec)	287.56	286.19	283.41	772.50	769.49	521.23	—	—	—

summary, TTD is positively correlated to demand while ATT will depend on demand distribution and signal time, especially in cases of small networks.

Figure 13 illustrates the results of comparing CT of cases in the same network decomposition.

We investigate the CTs of CSC and DSC-1 cases by number of signalized intersections. Here, the goodness of fit for CSC cases is measured by power function. Since the power value is over 7, this indicates that the CT of CSC increases exponentially when number of intersections increase. Given that the relationship is convex, the goodness of fit for DSC-1 is measured using 3-degree polynomial function. However, the coefficients for x^2 and x^3 are relatively small, indicating that the relationship between CT and number of intersections in the DSC-1 cases is nearly linear. R -square values for both cases are over 99%. Fits for the CSC cases using polynomial functions are low on goodness of fit and thus, not shown here. Therefore, based on CT, we conclude that the DSC-1 system is more suitable for a real-time application of larger size networks than the CSC system.

For cases of same network size, we further compare the performance of different decompositions, as per Table 6. Since the intersection of CSC cases and DSC-2 cases are empty, we use the performance ratios over DSC-1 cases for other cases with same demand level and network size.

From the ratio of CSC versus DSC-1 cases in the upper half of Table 6, we find that DSC-1 cases emerge as the most computationally efficient. Ratios for CT in all cases are greater than one. In addition, CT is reduced by around 55% and 99.7% as compared to CSC cases for 2×2 and 3×3 networks, respectively. However, ATT and TTD for DSC-1 cases are inferior. Ratios for ATT and TTD for all cases are smaller than one. In general, DSC-1 cases result in much better CT than CSC cases, but demonstrate poorer performance in ATT and TTD.

When comparing CSC with DSC-1 cases, the improvement of ATT is around 10% for 2×2 network, while the improvement is less than 5% for 3×3 network for the

CSC cases. However, there is significant improvement of TTD for CSC cases in 3×3 network compared to 2×2 , for which ratios are less than 0.7 when compared to those of 2×2 networks (over 0.88). We can see the same trend when comparing DSC-2 with DSC-1 cases. Cases of 6×6 network are better in terms of TTD as compared to 4×4 network cases. We should consider suitable objective functions for optimizing different cases.

In addition to comparing DSC-2 and DSC-1 cases, CT ratios of 6×6 networks are approximately 2.7, smaller than the ratios of the 4×4 networks. These ratios demonstrate that expanding DSC-2 to a more extensive network does not sacrifice much computational efficiency. However, the ratio of CT jumps from less than 3 to over 300 with increases in network size when we look at CSC cases. This change highlights the limitation of CSC cases when expanding network size.

As a final note, we cannot directly compare the DSC-2 and CSC case results in Table 6. However, feasible solutions from the DSC system are a theoretical subset of feasible solutions from CSC systems. Thus, the CSC system could achieve better ATT and TTD than all DSC systems for the same network. The CT ratio for DSC-2 over DSC-1 in a larger network is superior; the CT ratio for CSC over DSC-1 in a larger network is inferior. As a result, we can expect DSC-2 cases are more computationally efficient than CSC cases in the same-sized network.

6.6. Summary. We have presented five scenarios to test and compare the CSC and DSC systems. Scenarios 1 and 2 demonstrate the Three-step Naïve Method in a CSC system of a 2×2 network and two-layer process to optimize signals in a 4×4 network by decomposing it into four subnetworks, respectively. Scenario 3 shows that an increasing number of cores for the computation can increase computational efficiency while marginal revenue decreases. Scenario 4 shows a convergence of the DSC system when iteration numbers increase. Scenario 5 compares cases with different network

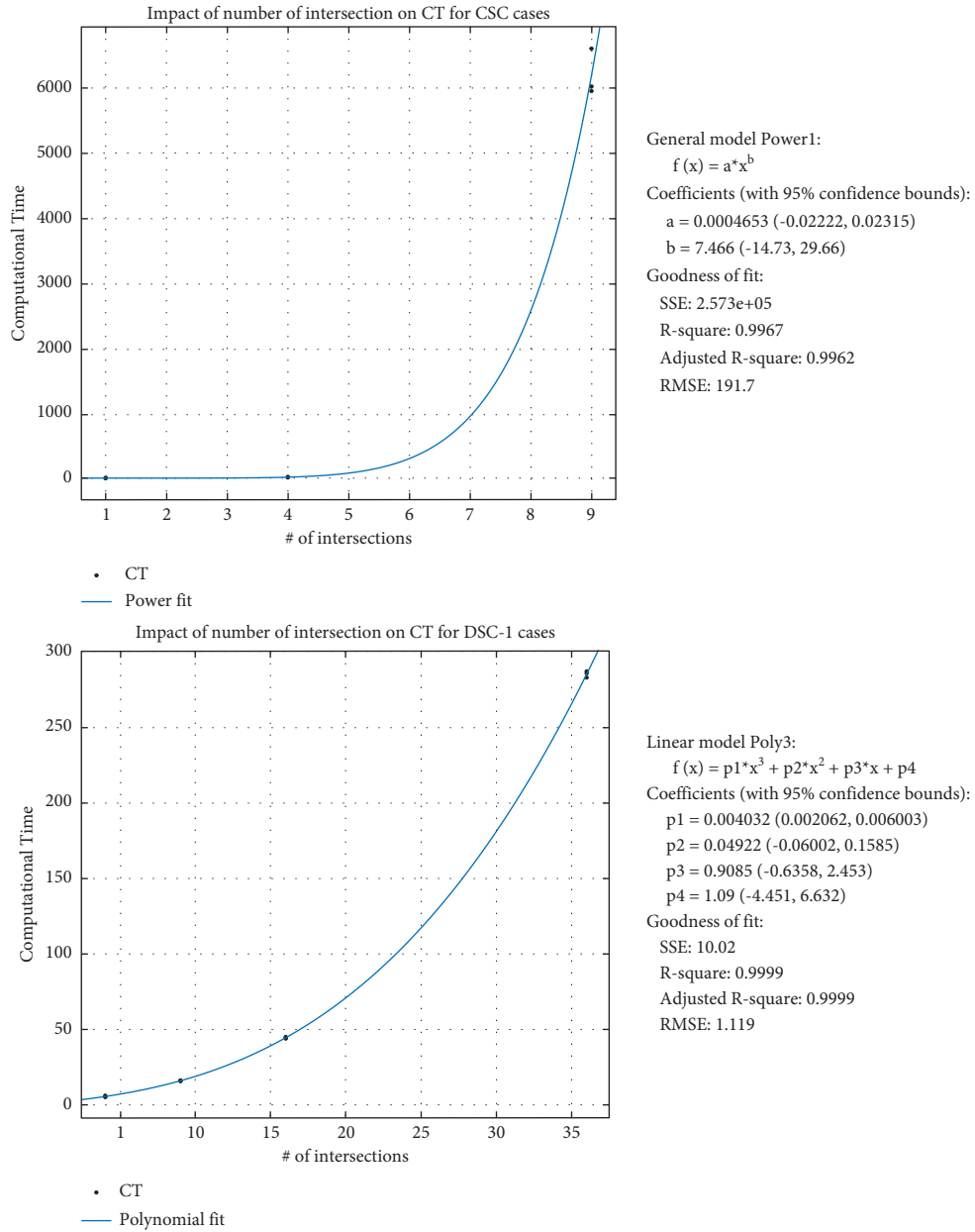


FIGURE 13: Impact of number of intersections on CT, CSC and DSC-1 cases.

TABLE 6: Comparison between different decompositions.

Network		Demand level 1	Demand level 2	Demand level 3
<i>CSC/DSC-1</i>				
2 × 2	ATT	0.90	0.92	0.90
	TTD	0.98	0.88	0.93
	CT	2.29	2.68	2.63
3 × 3	ATT	0.95	0.98	0.96
	TTD	0.69	0.65	0.63
	CT	366.42	371.80	409.72
<i>DSC-2/DSC-1</i>				
4 × 4	ATT	0.96	0.93	0.99
	TTD	0.92	0.91	1.01
	CT	6.62	6.87	6.67
6 × 6	ATT	0.90	0.94	0.86
	TTD	0.84	0.92	0.80
	CT	2.69	2.69	2.76

decompositions and demand levels. Results demonstrate that DSC cases have significantly better performance on CT and weaker performance on ATT and TTD. Network decomposition also has an impact on three scales. Increases in the subnetwork size may see reductions in ATT and TTD and loss of computational efficiency for the cases tested.

7. Conclusions

This research develops and compares CSC and DSC systems using short-term OD demand as inputs in an MEC-enabled CV environment and investigates the performances of different network decompositions for both systems. Control variables considered are cycle length, phase split (green times for all phases), and offset for each intersection inside the network. The two control objectives used for optimization are minimal Average Travel Time (ATT) and Total Travel Delay (TTD). Network dynamics are based on a G/G/n/FIFO open queuing network model, solved by simulation in MATLAB. Signal timing is proposed using a standard ring-and-barrier diagram with four phases and all-red intervals. Considering each phase as a virtual link, we assume that travelers will choose the paths with minimal instantaneous travel time. The optimal control problem for network signals is formulated with OD demand and free-flow travel time as inputs. However, the original problem is NP-complete due to nonconvex variables and nonlinear constraints. Therefore, we use a Three-step Naïve Method to decompose it into three subproblems and develop a DSC system within a CV environment with MEC. Thus, the DSC system decomposes the network into subnetworks, with each subnetwork controlled by an individual agent (regional MEC controller). Agents can exchange information in real-time. Finally, a two-layer process is proposed to solve the DSC system.

Numerical simulations were performed via five scenarios where demand was randomly generated following a uniform distribution. Scenarios 1 and 2 helped identify the basic settings and results of the CSC and DSC systems. Scenario 3 demonstrated the limitations of solving CSC cases against increasing the number of CPU cores. Scenario 4 gave an example for the convergence of the loop in the two-layer process for DSC cases. Scenario 5 investigated both systems in cases with different network decompositions and demand levels. The results show that network decomposition with smaller subnetworks results in better computational efficiency but reduced performance on ATT and TTD. For example, the CSC case for a 3×3 network has a CT of 90 minutes, while the DSC-1 case with the same demand takes 16 seconds—a reduction of over 99%. In addition, the DSC system can be regarded as the physical combination of several small CSC systems if considering each end of the traffic network as connected to the central TMC with fiber. The Three-step Naïve Method used in CSC systems to solve signal optimization is the same method used to solve optimal signal timing plans in each subnetwork of the DSC systems. In this case, the method—a combination of the bisection method, global search, and gradient search approaches—is highly inefficient. Thus, the main difference between the

systems is structure. The results show notable improvement in computational efficiency with some, but not significant, loss of ATT and TTD compared to the CSC system, thus demonstrating the value of the DSC system structure.

The results also revealed the ability of the DSC structure, with the scalable decomposition method, to apply to more extensive networks, which suffer acceptable losses of computational efficiency compared to the CSC system. Although the performances of ATT and TD are positively correlated, some cases revealed that we should choose suitable objectives for the different cases.

This work used a simulation-based method to gather more realism to the framework of network signal optimization. We also highlight the limitations of this work as future research to pursue. First, we must test our system in real-world traffic networks in a CV environment with MEC (which we only describe generally). The simulation in a cloud server with parallel computing is not equal to a DSC system within a real-world CV environment with MEC, constituting the need for field tests. Second, algorithms for signal optimization are simple and time-consuming in the CSC system, the three-step naïve method is inefficient in this regard. When network size increases to 3×3 , CT increases to more than 1.5 hours. This leads the further work to develop innovative algorithms to optimize traffic signal controls in CSC system. Third, queueing assumptions like Point Queue and First In First Out is not realistic, more updated models should be considered to capture traffic flow dynamics in the future. Finally, the route guidance function/traffic assignment model in Central TMC is under-investigated our present study, requiring further research into its limitations.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

The contents of this paper reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the City of Edmonton or Transport Canada. This paper does not constitute a standard, specification, or regulation.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research work was jointly supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, City of Edmonton, Transport Canada, and TELUS. The simulation was finished with the support by Compute Canada.

References

- [1] L. Adacher, A. Gemma, and G. Oliva, "Decentralized spatial decomposition for traffic signal synchronization," *Transportation Research Procedia*, vol. 3, pp. 992–1001, 2014.
- [2] L. Adacher and M. Tiriolo, "Performance analysis of decentralized vs centralized control for the traffic signal synchronization problem," *Journal of Advanced Transportation*, vol. 2020, Article ID 8873962, 1–19 pages, 2020.
- [3] A. H. F. Chow, R. Sha, and S. Li, "Centralised and decentralised signal timing optimisation approaches for network traffic control," *Transportation Research Part C: Emerging Technologies*, vol. 113, pp. 108–123, 2020.
- [4] A. G. Sims and K. W. Dobinson, "The sydney coordinated adaptive traffic (SCAT) system philosophy and benefits," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 2, pp. 130–137, 1980.
- [5] Y.-T. Wu and C.-H. Ho, "The development of Taiwan arterial traffic-adaptive signal control system and its field test: a Taiwan experience," *Journal of Advanced Transportation*, vol. 43, no. 4, pp. 455–480, 2009.
- [6] N. H. Gartner, "OPAC: a demand-responsive strategy for traffic signal control," *Transportation Research Record*, vol. 906, pp. 75–81, 1983.
- [7] P. Mirchandani and L. Head, "A real-time traffic signal control system: architecture, algorithms, and analysis," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001.
- [8] M. Ben-Akiva, D. Cuneo, M. Hasan, M. Jha, and Q. Yang, "Evaluation of freeway control using a microscopic simulation laboratory," *Transportation Research Part C: Emerging Technologies*, vol. 11, 2003.
- [9] F. Luyanda, D. Gettman, L. Head, S. Shelby, D. Bullock, and P. Mirchandani, "ACS-lite algorithmic architecture: applying adaptive control system technology to closed-loop traffic signal control systems," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1856, no. 1, pp. 175–184, 2003.
- [10] D. Manolis, I. Papamichail, E. B. Kosmatopoulos, and M. Papageorgiou, "Automated tuning of ITS management and control systems: results from real-life experiments," *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 119–135, 2016.
- [11] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, "Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network," 2017, <https://arxiv.org/abs/1705.02755>.
- [12] L. B. de Oliveira and E. Camponogara, "Multi-agent model predictive control of signaling split in urban traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 1, pp. 120–139, 2010.
- [13] F. A. De Souza, E. Camponogara, W. Kraus, and V. B. Peccin, "Distributed MPC for urban traffic networks: a simulation-based performance analysis," *Optimal Control Applications and Methods*, vol. 36, no. 3, pp. 353–368, 2015.
- [14] A. H. F. Chow, R. Sha, and Y. Li, "Adaptive control strategies for urban network traffic via a decentralized approach with user-optimal routing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1697–1704, 2020.
- [15] F. Perry Sr, "UFTI DSRC and Other Communication Options for Transportation Connectivity Workshop Overview of DSRC Messages and Performance Requirements," 2017, <https://www.pdfFiller.com/204136636-Forms-114pdf-FORM-JVAT-114>.
- [16] A. Ferdowsi, U. Challita, and W. Saad, "Deep Learning for Reliable Mobile Edge Analytics in Intelligent Transportation Systems," 2017, <https://arxiv.org/abs/1712.04135>.
- [17] G. Wang and F. Xu, "Regional intelligent resource allocation in mobile edge computing based vehicular network," *IEEE Access*, vol. 8, pp. 7173–7182, 2020.
- [18] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Edge Computing: a key technology towards 5G," *ETSI White Pap. No. 11 Mob*, vol. 11, pp. 1–16, 2015.
- [19] A. Hajbabaie, *Intelligent dynamic signal timing optimization program*, vol. 216, ProQuest Diss, E. Eisenhower Parkway, MI, USA, 2012, Theses.
- [20] R. Putha, L. Quadrioglio, and E. Zechman, "Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions," *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, no. 1, pp. 14–28, 2012.
- [21] C. Beard and A. Ziliaskopoulos, "System optimal signal optimization formulation," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1978, pp. 102–112, 2007.
- [22] J. Gregoire, X. Qian, E. Frazzoli, A. De La Fortelle, and T. Wongpiromsarn, "Capacity-aware backpressure traffic signal control," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 164–173, 2015.
- [23] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with average cost for adaptive control of traffic lights at intersections," in *Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, DC, USA, October 2011.
- [24] B. P. Gokulan and D. Srinivasan, "Distributed geometric fuzzy multiagent urban traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 714–727, 2010.
- [25] E. van der Pol and F. A. Oliehoek, "Coordinated Deep Reinforcement Learners for Traffic Light Control," in *Proceedings of the 30th Conf. Neural Inf. Process. Syst. 30th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, December 2016.
- [26] A. Nowé, P. Vrancx, and Y. M. De Hauwere, "Game theory and multi-agent reinforcement learning," *Adaptation, Learning, and Optimization*, vol. 12, 2012.
- [27] T. Nishi, K. Otaki, K. Hayakawa, and T. Yoshimura, "Traffic signal control based on reinforcement learning with graph convolutional neural nets," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, November 2018.
- [28] H. Wei, N. Xu, H. Zhang et al., "CoLight," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Gold Coast, Australia, November 2019.
- [29] C. Chen, H. Wei, N. Xu et al., "Toward a thousand lights: decentralized deep reinforcement learning for large-scale traffic signal control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 3414–3421, 2020.
- [30] H. Wei, G. Zheng, V. Gayah, and Z. Li, "A Survey on Traffic Signal Control Methods," 2019, <https://arxiv.org/abs/1904.08117>.
- [31] M. Mehrabipour and A. Hajbabaie, "A cell-based distributed-coordinated approach for network-level signal timing optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 7, pp. 599–616, 2017.
- [32] J. Withanawasam and A. Karunananda, "Multi-agent based road traffic control optimization," *2017 IEEE 20th*

International Conference on Intelligent Transportation Systems (ITSC), vol. 2017, pp. 977–981, 2018.

- [33] L. Adacher and M. Tiriolo, “Distributed urban traffic signal optimization based on macroscopic model,” *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, vol. 2016, pp. 605–610, 2017.
- [34] M. Lee, H. Chang, C. Carrion, and L. Zhang, “An approach to grouping traffic signals for coordination using clustering methods,” in *Proceedings of the 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 792–797, Naples, Italy, June 2017.
- [35] M. Rinaldi, W. Himpe, and C. M. J. Tampère, “A sensitivity-based approach for adaptive decomposition of anticipatory network traffic control,” *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 150–175, 2016.
- [36] B. Ran, D. E. Boyce, and L. J. LeBlanc, “A new class of instantaneous dynamic user-optimal traffic assignment models,” *Operations Research*, vol. 41, no. 1, pp. 192–202, 1993.
- [37] T. Urbanik, *Signal Timing Manual*, The National Academies Press, Washington, DC, Second Edition, 2015.