Title
Dynamic Traffic Routing and Adaptive Signal Control in a Connected Vehicles Environment

Author
Chai, Huajun

Publication Date
2019

Dynamic Traffic Routing and Adaptive Signal Control in a Connected Vehicles Environment

By

Huajun Chai

Dissertation

Submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Civil and Environmental Engineering

in the

Office of Graduate Studies

of the

University of California

Davis

Approved:

Professor H. Michael Zhang, Chair

Professor Yueyue Fan

Professor Chen-Nee Chuah

Professor Dipak Ghosal

Committee in Charge

2019

To my family.

Contents

List of Figures

List of Tables

Abstract

This dissertation aims to study effective and efficient ways for both travelers and transportation authorities to consider the actions of the other side when they make their corresponding travel or management decisions, such that certain common goals, such as mitigating congestion, reducing cost in travel expenses and improving the overall reliability of the transportation system can be achieved.

A novel dynamic traffic routing (DTR) with an adaptive signal control framework is developed to utilize the fast developing wireless communication technologies that makes V2X (Vehicle To Everything) possible. The hyper-path based dynamic traffic routing method takes stochasticity of link travel time into consideration, which ensures robust and reliable routing decisions. In addition, online travel time updating is incorporated into the DTR model. The online updating presented in this dissertation uses both historical information (a priori knowledge) and new information, thanks to the V2X system, to form a posteriori knowledge about the link travel time. Various distributed traffic signal control methods are proposed and tested concurrently with the DTR model to cope with the different levels of the traffic demand. Simulation models are built to test and compare the models developed in this dissertation against the traffic routing methods and traffic signal control models in the literature. In the extensive simulation tests, we discover that enabling vehicle re-routing in the network can reduce the average travel time as well as reduce the average queue length at the intersections.

The joint dynamic traffic routing and adaptive signal control model developed in this dissertation performs well in terms of average travel time and average travel delay in most cases. However, there are still possibilities that in some extreme cases the proposed routing and control model may fail to produce satisfying results. The underlying logic of DTR does not guarantee to prevent deadlock, a.k.a gridlock, from happening. To address the possibility that deadlock occurs, following the study of dynamic traffic routing and adaptive signal control, I formulate a deadlock avoidance model under dynamic user equilibrium with queue spillback. In the proposed model, travelers' route choice is governed by a simple *"DLA (DeadLock Avoidance) Routing"* rule which is proved to generate deadlock

free routing result. Potential deadlocks during the optimization of the model are detected with an algorithm modified based on *Floyd Warshall Algorithm*. The algorithm then assigns a deadlock potential value to each potential deadlock. The model minimizes this potential, and meanwhile tries to maintain the total travel time in the network at a reasonably low level.

Many transportation applications can potentially take advantage of the research results in this dissertation. We explored one interesting and important application scenario-the parking search problem in Chapter 6 of this dissertation.

Acknowledgments

I would like to give my greatest gratitude to my Ph.D. advisor, Professor H. Michael Zhang, for the continuous support of my Ph.D study and related research, for his guidance, immense knowledge, patience, and most important, encouragement throughout my Ph.D. study at University of California, Davis. I consider myself to be extremely fortunate to have Dr. Zhang as my mentor during my Ph.D., as he is such an easy-going, kind, supportive and knowledgeable person. He always encourages me to explore my own interests, and meanwhile provides me as much support and help as he can. Throughout the years, I have benefited greatly from his extensive experience in transportation research, valuable advice and guidance on my career path, and generous financial support. Professor Zhang has been, and will continue to be, my role model even after I graduate with a Ph.D. degree and start my new life career somewhere else.

I would also like to express my great appreciation to Professor Yueyue Fan and Professor Miguel Jaller in the Department of Civil and Environmental Engineering, Professor Chen-Nee Chuah in the Department of Electrical Engineering, Professor Dipak Ghosal in the Department of Computer Science. They served in my qualify exam committee, and some of them are also in my dissertation committee. They have always been ready to listen to my questions, give me advice and share their expertise and experience in academia. I am also grateful to other faculties and staffs of the department who offered me numerous help during my years of graduate study.

I want to acknowledge my fellow colleagues at UC Davis, Yiru Chen, Jia Li, Hui Deng, Shikai Tang, Rui Ma, Shenyang Chen, Sarder Rafee Musabbir, Zenghao Hou, Xiao Han and Hang Gao for their help during my days in UC Davis. I have been enjoying myself working with them, discussing problem with them, and sometimes hanging out with them after we reached some project milestones. They are always ready to listen and to help. I would also like to give my special thanks to Dr. Rui Ma, who has been giving me great help in my research study during his years in our research group as a Postdoctoral researcher.

Finally, I would like to thank my beloved wife, Huachao Lin and my parents, to whom this dissertation is dedicated. It is impossible for me to make this far without their endless

love and support, and their company, both physically and emotionally, along my journey to Ph.D..

Chapter 1

Introduction

1.1 Background and motivations

Transportation systems nowadays are like complex organisms, within which different parts on the one hand support each other, and on the other hand are tightly constrained by each other. Different traffic modes (cars, buses, bicycles, etc.), different groups and entities (travelers, transportation authorities, legislation, consultant agencies, etc.), transportation infrastructures (different types of roadways, road signages, traffic lights, etc.) and other components are mixed together forming an immensely complex system. It requires enormous and decades of research endeavor to understand, model and optimize such a system.

As the communication technology advances so rapidly in the past decades, the long existed concept of V2X (connect Vehicle to Everything), and hence the broader concept Vehicular Ad-Hoc Network (a.k.a. VANET), finally becomes possible to be implemented in real world. There are now many companies and agencies that develop and provide V2X products all over the world, Continental (Germany), Qualcomm (US), NXP Semiconductors (Netherlands), Robert Bosch (Germany), and Delphi Automotive (U.K.), to name a few [1]. Vehicles in the near future will be equipped with devices that allow them to talk to central control stations to get the most up-to-date information, and to report/publish

[1] V2X Market: 2018 Global Key Players, Regional Analysis, Size, Share, Trends, and Competitive Landscape Forecast To 2023. MarketWatch. https://www.marketwatch.com/press-release/v2x-market-2018-global-key-players-regional-analysis-size-share-trends-and-competitive-landscape-forecast-to-2023-2018-05-25. Published 2018. Accessed April 1, 2019.

their own information to the public. Vehicles can also directly communicate with nearby vehicles to share information like location, speed, headings, destinations, etc. It is a critical component in the Cooperative Adaptive Cruise Control, a.k.a CACC. Vehicles negotiate with other neighboring vehicles in some established protocols the right of way, merge and split maneuver, lane change, overtaking, etc (Milanés et al., 2014; Amoozadeh et al., 2015; Dey et al., 2016). The benefits from CACC are significant, including but not limited to enhancing driving safety, increasing road network throughput, improving string stability of the system, reducing fuel consumptions, etc (Lu et al., 2002; Ploeg et al., 2011; Suthaputchakun et al., 2012; Öncü et al., 2012; Zohdy and Rakha, 2012; Dey et al., 2015; Li and Kockelman, 2016). Furthermore, some cities and companies are planning to build infrastructures that support V2I to facilitate vehicular traffic. Audi, an automotive company, are bringing their talking traffic light technology to 10 U.S. cities[2]. In those 10 cities, Audi cars that are equipped with V2I devices can obtain intersection traffic light information (red/green/yellow signals) before they actually get to that intersection. This allows drivers to make decisions about their driving speeds and route choices ahead of time, which in turn can either save the driver some time by choosing a detour that avoids red light as much as possible or reduce fuel usage by choosing cruising speed appropriately.

The example of the Audi's talking traffic lights reveals that by knowing the traffic light status in advance can benefit travelers greatly. On the other way around, it is also possible to optimally adjust the traffic signal control to adapt to the on-road traffic when knowing travelers' position, speed and route choice. The easiest and simplest adaptive strategy is to increase green time to the approach when vehicle volume in that approach increases, or to reduce green time otherwise. Most of the traffic routing and signal control methods in the past are developed without consideration of V2X technology, and hence it lacks the ability to take advantage of the benefits from the promising V2X technology aforementioned. An efficient, effective and robust framework that jointly routes and controls traffic in the concurrent network is desirable.

[2]Davies C. AudiâĂŹs talking traffic lights tech expands to 10 US cities. SlashGear. https://www.slashgear.com/audi-v2i-traffic-light-information-10-cities-expansion-11530481/. Published May 11, 2018. Accessed April 1, 2019.

1.2 Research objectives and contributions

This dissertation aims to investigate the relationship between traffic control and travelers' route choice behavior, and to develop a novel framework which systematically coordinate those two parts in a concurrent manner to improve the performance of the transportation system in the context of Vehicular Ad-Hoc Network (VANET). The central idea is to take advantage of on-line information availability boosted by VANET, which can help reduce uncertainties and stochasticity while traveling in the network and, in the same time, help guide traffic to their destinations more quickly. Hyperpath based models are convenient and capable to handle routing problems with stochastic link travel time. By incorporating certain link travel time distribution updating paradigms, a traditional hyperpath routing model can be modified into a dynamic traffic routing model. Taking a step further from the study of joint traffic signal control with dynamic routing under VANET environment, the possible deadlock occurrence is another important research problem. By utilizing the *"DLA routing"* strategy, which guarantees deadlock free in networks, as an additional constraint in the user's route choice, it is possible to achieve deadlock free in traffic assignment. However, such strict routing strategy could have high overheads in terms of increased travel time, increased VMT, extra emission, etc. One solution is to find a balance between deadlock occurrence and overall system performance by optimally selecting the percentage of *"DLA routing"* vehicles in the network. Last but not least, I explore the potential of applying the proposed research to some real world applications, such as parking search and management problem.

The work in this dissertation has several contributions to transportation research and current real world practice. They are summarized as follows:

- Developed a hyper-path based dynamic traffic routing method that works jointly with signalized intersections in a connected environment. The effects of different penetration rates of DTR vehicles are studied.

- Developed a traffic control method based on max-pressure algorithm that takes downstream queue, downstream queue capacity and incoming flow rate into consideration. The performance of the proposed signal control method outperforms most of

the popular signal control strategies in the literature.

- Developed an optimal routing strategy under dynamic user equilibrium with queue spill-back that minimizes the potential of deadlock via *"DLA routing"*. The DLA model not only reduces the deadlock potential in the network, but also maintains a reasonable total travel cost.

- The overall DTR and DLA frameworks are ready to be applied to various real world applications. Specifically, the proposed DTR with adaptive signal control framework is applied to solve a practical parking search and management problem.

1.3 Thesis organization

The structure of this dissertation is organized as follows. In Chapter 2 and Chapter 3 a literature review on traffic routing methods and traffic signal control models, respectively, is presented. In Chapter 4, a joint dynamic traffic routing and adaptive signal control framework is developed in stochastic and time-dependent context. Simulation results are presented in the same chapter testing the proposed coherent framework. Some stability analysis is present with respect to dynamic traffic routing and traffic signal control where online information is available to travelers. Deadlock analysis in traffic routing is provided in Chapter 5, with insights from Dynamic Traffic Assignment (DTA). In this chapter, a dynamic user equilibrium model considering queue spillback is developed to minimize the possibility of deadlock occurrence. Chapter 6 sheds some light upon the possible applications of the research in this dissertation in real world scenarios. A study case that applies the dynamic traffic routing developed in earlier chapters to the problem of parking search is presented. Chapter 7 concludes the dissertation with a summary and some discussions about future works.

Chapter 2

Literature Review PART 1: Traffic Routing And Traffic Assignment

In this chapter, we present a literature review on the history as well as the state-of-art of the traffic routing methods and algorithms. Following that an introduction of traffic assignment models is present. The difference between traffic routing and traffic assignment is discussed in the end of this chapter.

2.1 Shortest path algorithms

Shortest path algorithms are the most essential pieces in traffic routing problem. Every traffic routing problem, eventually, will encounter certain types of generalization of the shortest path problem, such as the most reliable path (Fan et al., 2005; Xing and Zhou, 2011; Pan et al., 2013), the shortest path with resource constraints (Beasley and Christofides, 1989), k^{th} shortest path (Yen, 1971), and many other routing problems. Deo and Pang (1984) conducted a thorough survey and taxonomy on different shortest path algorithms. They did an outstanding work summarizing studies on shortest path from the last century. Here in this chapter, we first give an introduction on some of the most important shortest path algorithms, and then introduce some of the state-of-the-art shortest path algorithms.

Before going into different shortest path problems and algorithms, a brief description on the general setting for the shortest path problem is provided. Let $G = (E, V)$ denote a graph (or "network", another term for network, and it is used interchangeably in this dissertation hereafter) with V as the set of vertices (or nodes) and E as the set of edges

(or links). Links E can be directional, or non-directional. Graph in the former case is known as the directed graph, while in the latter case it is known as the undirected graph. Each link in the network can be associated with a certain weight, w_e, where $e \in E$. In the transportation network, this weight, for instance, can be link travel time or link length. If the weight of every link is homogeneous, the shortest path problem becomes identical to the problem of identifying a path with the least intermediate hops.

2.1.1 One-to-One/One-to-Many/Many-to-One shortest path

The simplest and most fundamental problem in the shortest path problem is the one-to-one shortest path. It seeks to find the shortest path in a network from one origin node (it is also called source node in some literature) to one destination node (it is also called sink node in some literature).

Bellman (1956) first introduced the one-to-one shortest path problem as a problem to determine the path in the network of cities from one given city to another given city which minimizes the travel time. A dynamic programming approach is proposed to solve the aforementioned shortest path problem. In the original *Bellman-Ford* formulation, there are N cities, arbitrarily numbered from 1 to N. Every pairs of cities are connected by a bi-directional road, with a given travel time t_{ij} as the required travel time from city i to city j. The problem is to find the shortest path from city 1 to city N. Let c_i be the minimum travel time from city i to city N, where $i = 1, 2, \ldots, N$. It is obvious that $c_N = 0$.

From the principle of optimality, c_i should satisfy the following equation:

$$c_i = \begin{cases} \min\limits_{(i,j)\in E}(t_{ij} + c_j), & i = 1, 2, \ldots, N-1 \\ 0, & i = N \end{cases}$$

The problem is then solved iteratively by using DP to approximate in policy space. Let c_i^k be the minimum cost from node i to node N in k^{th} iteration, with initial condition:

$$c_i^0 = \begin{cases} t_{iN}, & i = 1, 2, \ldots, N-1 \\ 0, & i = N \end{cases}$$

Successive approximation (it is also a form of relaxation) yields:

$$c_i^{k+1} = \begin{cases} \min\limits_{(i,j)\in E}(t_{ij} + c_j^k), & i = 1, 2, \ldots, N-1 \\ 0, & i = N \end{cases}$$

By iterating the above approximation $N-1$ times, the final result is the minimum travel time from node 1 to node N, which is c_1^{N-1}. The actual shortest path can be obtained by traversing from node 1 all the way to node N. The rule is at any node i, the next node to take will be the node j that holds the above relaxation equation. In the above formulation, it does not only calculate the shortest path from node 1 to node N, it also computes the shortest path from any node to Node N at the same time as a by-product. It is sometimes called a "many-to-one" shortest path. It takes trivial work to modify the original *Bellman-Ford* formulation to a "one-to-many" variant. Let c_i be the minimum travel time from city 1 to city i. The principle of optimality now becomes:

$$c_j = \begin{cases} \min\limits_{(i,j)\in E}(t_{ij} + c_i), & i = 1, 2, \ldots, N-1 \\ 0, & i = 1 \end{cases}$$

The initialization and relaxation processes are now:

$$c_i^0 = \begin{cases} t_{i1}, & i = 1, 2, \ldots, N-1 \\ 0, & i = 1 \end{cases}$$

$$c_j^{k+1} = \begin{cases} \min\limits_{(i,j)\in E}(t_{ij} + c_i^k), & i = 1, 2, \ldots, N-1 \\ 0, & i = 1 \end{cases}$$

The "one-to-many" variant of *Bellman-Ford* algorithm is provided in Algorithm 1. The time complexity of *Bellman-Ford* algorithm is $O(V + VE + E) = O(VE)$.

Algorithm 1 Bellman-Ford Shortest Path Algorithm

```
1: procedure
2:     Input Vertices: V, Edges: E, Source Vertex: s
3:     for each vertex v in V do
4:         dist[v] ← ∞ // minimum distance from s to v
5:         pred[v] ← NULL // preceding node of v on the shortest path from s to v
6:     end for
7:     dist[s] ← 0
8:     for i from 1 to |V|-1 do
9:         for each edge (u,v) in E do
10:            if dist[v] > dist[u] + weight(u,v) then
11:                dist[v] ← dist[u] + weight(u,v)
12:                pred[v] ← u
13:            end if
14:        end for
15:    end for
16:    // optional: negative cycle detection
17:    for each edge (u,v) in E do
18:        if distance[u] + weight(u,v) < dist[v] then
19:            There is a negative cycle in the graph.
20:        end if
21:    end for
22: end procedure
```

Following Bellman (1956), Dijkstra (1959) proposed another method to solve the shortest path problem between a pair of nodes without the need to iteratively relax every edge $|V|-1$ times. A temporary list is used to store the vertices that its label of distance is not yet finalized. Every iteration in the "while" loop, the vertex, u, with minimum label of distance is picked and removed from the list. Then the algorithm tries to visit any neighboring vertices of the selected vertex, u. To speed up the algorithm, any neighboring vertices

that are not in Q, i.e. the vertices have been finalized with a permanent label of distance previously, will be ignored in the visit. During the visit, if the current label of distance of any neighboring vertex, v, is larger than the summation of the label of vertex u and weight of edge (u, v), the label of such vertex v will get updated to $dist[u] + weight(u, v)$. At the same time, vertex u becomes a temporary predecessor vertex of v. The process is repeated until all vertices are visited and have been finalized with a permanent label of distance, i.e. Q becomes an empty list. Algorithm 2 shows the details about how *Dijkstra's Algorithm* works.

Algorithm 2 Dijkstra's Shortest Path Algorithm

```
1:  procedure
2:      Input Vertices: V, Edges: E, Source Vertex: s
3:      Temporary vertices container Q ← []
4:      for each vertex v in V do
5:          dist[v] ← ∞ // minimum distance from s to v
6:          pred[v] ← NULL // preceding node of v on the shortest path from s to v
7:          add v to Q
8:      end for
9:      dist[s] ← 0
10:     while Q is not empty do
11:         u ← arg min_{u∈Q} dist[u]
12:         remove u from Q
13:         for each (u,v) in E and v in Q do
14:             alt ← dist[u] + weight(u,v)
15:             if dist[v] > alt then
16:                 dist[v] ← alt
17:                 pred[v] ← u
18:             end if
19:         end for
20:     end while
21: end procedure
```

In the literature, the Dijkstra's Algorithm is sometimes categorized as a Label Setting Algorithm, while the Bellman-Ford's Algorithm is categorized as a Label Correcting Algorithm. The major difference between these two classes of shortest path algorithms is that in the Label Setting Algorithm once the label of a node is set as permanent or visited it is not changed in the following iterations, while in the Label Correcting Algorithm the label of a certain node is subject to change until the termination of the algorithm. In terms of the performance, Dijkstra's Algorithm is faster than Bellman-Ford Algorithm in some cases. However, Bellman-Ford Algorithm is able to cope with graphs with negative weighted links. It is also capable to detect any negative weight cycles inside the graph. In the context of a transportation network, a negative weight is rare (with exceptions in the cases where subsidy or credit is so high that the actual cost to traverse a link may become negative). In most cases, Dijkstra's Algorithm is a better choice.

2.1.2 k-shortest path

Sometimes a single shortest path may not be enough. For instance, when the uncertainty of road traffic is extremely high, a single shortest path planned ahead of time may fail due to the occurrence of unexpected incidents. In such cases, knowing some alternative paths (though not optimal) can be beneficial. This is one motivation in real life for computing k shortest path. The k shortest path algorithm is a generalized form of the general shortest path algorithm aiming to find out a sequence of k paths from origin to destination in an increasing order of cost [1].

There are two variants of k-shortest path problem. The first variant allows loops to exist in the paths, while the second variant requires the paths to be loopless. In a transportation network, loops in a path are usually not desirable as travelers generally tend to avoid loops when they plan their routes or when they travel en-route. Paths with loops in them are less interesting to travelers in general. For this reason, only the second type of the k shortest path algorithm is discussed hereafter in this subsection.

Yen (1971), on top of many other excellent works on the k shortest path algorithm (Bock et al., 1957; Pollack, 1961; Clarke et al., 1963; Sakarovitch, 1966), proposed a new k shortest

[1] Wikipedia Contributors (2019). k shortest path routing. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/K_shortest_path_routing [Accessed 8 May 2019].

path algorithm that guarantees the upper bound computational complexity increases linearly with the value of k. Therefore, Yen (1971)'s algorithm is extremely efficient comparing to its predecessors, achieving an approximate complexity upper bound of $O(\frac{1}{2}KN^2)$ when there are no negative weights [2]. Yen's k-shortest path algorithm is also capable to deal with graph with negative weights. However, to simplify the illustration, in this introduction we assume all the edges have positive weights. The algorithm is summarized as following.

Table 2.1: Notations: Yen's k-shortest path algorithm.

Notation	Definition
N	Total number of nodes in the network.
(i)	i^{th} node in the network with (1) being the source node and (N) being the sink node.
d_{ij}	The length (cost) of the directed arc from i to j. Note that $d_{ij} > 0$.
A^k	The k^{th} shortest path from (1) to (N), with $k = 1, 2, \ldots, K$. The path is $(1) \rightarrow (2^k) \rightarrow (3^k) \rightarrow \ldots \rightarrow (Q_k^k) \rightarrow (N)$, where i^k is the i^{th} node on the k^{th} shortest path.
A_i^k	A deviation from path A^{k-1} at node (i), where $i = 1, 2, \ldots, Q_k$. A_i^k and A^{k-1} share the same path until node (i^k), i.e. $(1) \rightarrow (2^k) \rightarrow \ldots \rightarrow (i^k)$. After (i^k), arc from i^k to $(i+1)^k$ on path A_i^k is different from any path A^j, where $j = 1, 2, \ldots, k-1$.
R_i^k	The root path of A_i^k. It is the sub-path of A_i^k that overlaps with path A^{k-1}, i.e. $(1) \rightarrow (2^k) \rightarrow \ldots \rightarrow (i^k)$.
S_i^k	The spur path of A_i^k. It starts at node i^k and ends at node (N), i.e. $(i^k) \rightarrow \ldots \rightarrow (N)$, i.e. A_i^k.

In Yen's k-shortest path algorithm, there are two lists A and B used to store shortest paths. List A contains the k-shortest paths, i.e. A^k, whereas list B holds the potential

[2]The time complexity is highly depending on the shortest path algorithm used within Yen's k-shortest path algorithm. The computational complexity that the author claims is achieved under the assumption that one uses the shortest path specified in Yen (1972).

k-shortest paths.

Step 1: The first step is to find the shortest path A^1. Any efficient shortest path algorithm can be used to compute A^1. After obtaining A^1, store it in list A.

Step 2: The next step is to compute the rest A^k, where $k = 2, \ldots, K$. It is done in a iterative way. In order to find A^k, the shortest paths $A^1, A^2, \ldots, A^{k-1}$ are required to be known in advance. The computation of A^k is further divided into two parts:

For each $i = 1, 2, \ldots, Q_{k-1}$, repeat the following step 2.1:

Step 2.1: Compute all the deviation paths A_i^k. There are three sub-procedures to compute all the deviation paths:

(a) Identify the root path, R_i^k, by searching the sub-path in A^{k-1} that overlaps for the first i nodes with A^j, where $j = 1, 2, \ldots, k-1$. If R_i^k is found, set $d_i q = \infty$, where (q) is the $(i+1)^{th}$ node of A^j; otherwise, do nothing. Then go to (b).

(b) Compute the spur path S_i^k from node (i) to node (N) using any efficient shortest path algorithm. Since the arc between (i) and (q) is set to ∞, the spur path, S_i^k, found is guaranteed to be different from those of A^j.

(c) If both R_i^k and S_i^k are found, combine the two together to form $A_i^k = R_i^k + S_i^k$. And put this potential k-shortest path into list B. After that, reset the cost of arc from (i) to (q) back to its initial value.

Step 2.2: Choose the path with minimum cost from list B to be put into list A as A^k. Now move on to the next iteration to compute A^{k+1}.

As aforementioned, the Yen's k-shortest path algorithm can achieve a performance of $O(\frac{1}{2}KN^2)$ in terms of the upper bound of the complexity when all edges have positive weights. The space complexity is $N^2 + KN$ with N^2 to store the graph and KN to store list A and list B. Since Yen's k-shortest path algorithm scales up linearly with the value of K for both the time complexity and space complexity, it has a significant advantage over other algorithms that scale exponentially with the value of K, e.g. Pollack (1961). Following Yen (1971). There are numerous research works trying to improve the performance of k-shortest path algorithm. However, a thorough review of all the k-shortest path algorithms is beyond the scope and purpose of this dissertation work. Readers are referred

to the related literature (Brander and Sinclair, 1996; Martins et al., 1998; Eppstein, 1998; Borgwardt and Kriegel, 2005; Hershberger et al., 2007) if interested.

2.2 Shortest hyper-path algorithms

In k shortest path routing, the possible choices of paths are selected within the set of k shortest paths which are computed ahead of time. Though the k shortest path routing gives traveler some extra freedom and reliability in making trip plans, it still lacks the ability to adaptively adjust routing decisions en-route. Once a traveler picks one path and travels on the road, the choice is limited to that path unless a new shortest path calculation is performed. If recalculation is not possible and there are some unexpected incidents on the chosen path, the traveler could experience longer than expected total travel time. In a broader context, link travel time in reality has randomness and uncertainty. The k shortest path by its nature is not sufficient to handle this type of uncertainty and randomness.

To address the uncertainty discussed above, another type of routing method is developed called hyper-path routing. The central idea of hyper-path routing is to design a routing policy or strategy at each node in the network so that travelers can re-course during their trips according to the policy. This routing policy is one instantiation of a more general concept known as hyper-path or hyper-graph (Nguyen and Pallottino, 1989; Gallo et al., 1993; Nielsen et al., 2005). At each node on the hyper-path, an outgoing hyper-link is used to represent the diversion options of travelers' route choices. Each branch of a hyper-link has a certain probability to be chosen depending on the re-routing policy used. In the remaining part of this section, a formal definition of the hyper-path is presented and its application in vehicles routing in a transportation network is discussed.

A hyper-graph (Gallo et al., 1993) is a pair of $G_H = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n nodes and $E = \{e_1, e_2, \ldots, e_m\}$ is the set of m hyper-links, with $e_i \subseteq V$ for $i = 1, 2, \ldots, m$. Hyper-graph is a generalization of a standard graph representation of a network where links (called hyper-links) can have multiple tail nodes or head nodes. When $|e_i| = 2, i = 1, 2, \ldots, m$, a hyper-graph G_H becomes a traditional graph. Similar to a directed graph, a directed hyper-graph is connected by directed hyper-links. A directed hyper-link $e \in E$ is an ordered pair $e = (T(e), H(e))$ of disjoint subsets of nodes, where

$T(e) \subset V$ is the tail nodes connected to hyper-link e and $H(e) \subset V$ is the head nodes of hyper-link e. There are two different types of basic hyper-links known as forward-hyperlinks and backward-hyperlinks (in Gallo et al. (1993), they are called F-arc and B-arc). A backward hyper-link is a hyper-link with $|H(e)|= 1$, while a forward hyper-link is a hyper-link with $|T(e)|= 1$. Figure 2.1 shows the structure of a backward hyper-link and a forward hyper-link.

(a) Backward hyper-links. (b) Forward hyper-links.

Figure 2.1: Two types of hyper-links.

For traffic routing in a transportation network, the forward hyper-link representation is of particular interest as it is very well representing traveler's route choice division behavior (re-course) at a node (Kanturska et al., 2013). In the sequel, we are mostly discussing hyper-graph with forward hyper-links, unless specified otherwise. Denote

$$FS(i) = \{e \in E | i \in T(e)\}, \quad BS(i) = \{e \in E | i \in H(e)\}$$

It is clear that for forward hyper-graph the following holds:

$$|T(e)|= 1$$

A hyper-path (Gallo et al., 1993; Nielsen et al., 2005) Π_{st} of origin s and destination t is an acyclic minimal hyper-graph $G_\Pi = (V_\Pi, E_\Pi)$ that satisfies:

(a) $E_\Pi \subseteq E$

(b) $s, t \in V_\Pi = \bigcup_{e \in E_\Pi} (T(e) \cup \{H(e)\})$

(c) $v \in V_\Pi \setminus \{s\} \Rightarrow v$ is connected to s in G_Π.

Given a hyper-graph G_Π, for each arc $(i,j) \in E_\Pi$ there is an associated conditional probability, π_{ij}^Π, defined as:

$$\pi_{ij}^\Pi = Prob\{traversing(i,j) \in E_\Pi | i \in V_\Pi\}$$

The above term can be interpreted as the conditional probability to traverse arc (i,j) from node i. As shown in the text above, in forward hyper-graph a hyper-link has only one tail node. Assume for hyper-link e, the tail node is i, i.e. $T(e) = \{i\}$. Then we shall have the following as the conditional probability of route division at a node should sum up to 1:

$$\sum_{j \in T(e)} \pi_{ij}^\Pi = 1$$

Let P_Π denote the set of all paths connecting origin s and destination t in hyper-graph G_Π. The probability of a certain elementary path $p \in P_\Pi$, π_p^Π, being chosen is then equal to:

$$\pi_p^\Pi = \Pi_{(i,j) \in E_\Pi} (\pi_{ij}^\Pi)^{\delta_{ij}^p}$$

where $\delta_{ij}^p = 1$ is path p traverses arc (i,j), and $\delta_{ij}^p = 0$, otherwise. π_p^Π can also be interpreted as the proportion of a unit flow going though path p from s to t. Apparently, the following equation holds as the total flow through all paths should sum up to 1:

$$\sum_{p \in P_\Pi} \pi_p^\Pi = 1$$

Consider a scenario where the cost to traverse a link (i,j) consists of two parts: 1) the ordinary link cost c_{ij}, and 2) a node traversing cost w_i^Π. Note that different hyper-path could have different node traversing cost w_i^Π. Introduce another boolean variable δ_i^p denoting if path p traverses node i: $\delta_i^p = 1$ if path p goes through node i, and $\delta_i^p = 0$, otherwise. With all the ingredients needed for calculating the total cost of a hyper-path, we shall have:

$$C_\Pi = \sum_{p \in P_\Pi} c_p \cdot \pi_p^\Pi \tag{2.1}$$

where

$$c_p = \sum_{(i,j) \in E_\Pi} \delta_{ij}^p c_{ij} + \sum_{i \in V_\Pi} \delta_i^p w_i^\Pi \tag{2.2}$$

The hyper-path cost defined in Equation (2.1-2.2) requires the exhaust enumeration of all the paths connecting source node s and target node t. This is impractical and intractable

when the size of the network becomes large. A dynamic programming approach is hence proposed by Nguyen and Pallottino (1989). Let $C_\Pi(i,t)$ denote the cost from node i on hyper-path Π_{st} to target node t. We should have:

$$\begin{aligned} C_\Pi = C_\Pi(s,t) &= \sum_{(s,j)\in FS(s)} \pi_{sj}^\Pi \cdot (w_s^\Pi + c_{sj} + C_\Pi(j,t)) \\ &= w_s^\Pi + \sum_{(s,j)\in FS(s)} \pi_{sj}^\Pi \cdot (c_{sj} + C_\Pi(j,t)) \end{aligned}$$

Similar relation should hold for all node i heading to target node t on hyper-path Π_{st}:

$$C_\Pi(i,t) = \begin{cases} 0, & i = t \\ w_i^\Pi + \sum_{(i,j)\in FS(i)} \pi_{ij}^\Pi \cdot (c_{ij} + C_\Pi(j,t)), & i \in V_\Pi \setminus \{t\} \end{cases}$$

With the above dynamic programming formulation, it is possible to calculate the hyper-path cost for Π_{st} in a backward manner.

Let $C^*(s,t)$ denote the minimum cost for any hyper-path connecting s and t. By Bellman's Principle of Optimality, the above dynamic programming relation then becomes the following:

$$C^*(i,t) = \begin{cases} 0, & i = t \\ \min_{\Gamma_i \in FS(i)} \left\{ w_i^{\Gamma_i} + \sum_{(i,j)\in\Gamma_i} \pi_{ij}^{\Gamma_i} \cdot (c_{ij} + C^*(j,t)) \right\}, & i \in V_\Pi \setminus \{t\} \end{cases}$$

where $\Gamma_i \subseteq FS(i)$ is any subset of the links that have node i as tail node. w^{Γ_i} and π^{Γ_i} are the node traversing cost and route division probability at node i, respectively.

Following the work in Nguyen and Pallottino (1989) and Nguyen and Pallottino (1988), there are many research works on applying hyper-path to different applications in transportation, especially in traffic assignment (Marcotte and Nguyen, 1998; Lozano and Storchi, 2002; Unnikrishnan and Waller, 2009), transit operations (Wu et al., 1994; Nguyen et al., 1998; Kurauchi et al., 2003; Li et al., 2015), traffic routing (Miller-Hooks and Mahmassani, 1998, 2000; Miller-Hooks, 2001; Yang and Miller-Hooks, 2004; Gao and Chabini, 2006; Kanturska et al., 2013; Chai et al., 2017) and so on. With its flexibility in network representation that could be easily modified for different application purposes, hyper-path and hyper-path based routing shall have great potential in the future.

2.3 Backpressure based routing

In packet routing and network communication area, there is a popular approach called backpressure routing for scheduling packets, network resources, etc. It is a routing method designed to direct traffic in a network in the way to maximize the total throughput. The key idea in backpressure routing is to dynamically route traffic based on backpressure gradient between different hops. The term "back pressure" is a concept adapted from the field of fluid dynamics describing the resistance force when fluid flows through pipe systems [3].

Tassiulas and Ephremides (1990) first brought the idea of using backpressure to route traffic in communication networks and developed the original backpressure routing algorithm. In their study they considered a queuing network with multiple-hops and arbitrary topology for a communication network. They assumed that the global knowledge of the queuing lengths at each hop in the network is known. The arrival rates are assumed to be random. At each time step, the backpressure routing algorithm picks the eligible sets of servers(links) to activate, and corresponding routing decisions are made in a centralized fashion, in contrast to fixed scheduling sequences that are set in advance in other static routing schemes. The algorithm is made of two stages: 1) in the first stage select links with maximum weight for certain type of packet. 2) in the second stage route packet based on backlog differential. The algorithm is summarized hereafter.

First step is to identify the set of links for activation for each type of flow f:

$$W_{(a,b)}(t) \triangleq max\{\max_{f}[Q_a^f(t) - Q_b^f(t)], 0\} \tag{2.3}$$

where Q_a^f is the queue backlog of flow f at node a at time t. $\max_f[Q_a^f(t) - Q_b^f(t)]$ finds the type of flow f^{opt} on link (a, b) that has the maximum queue backlog differential between head and tail nodes. Link (a, b) is then included in the set of activated links for flow f^{opt}, and it is closed for all other types of flow on link (a, b). The weight of link (a, b), i.e. $W_{(a,b)}(t)$, is then assigned as that backlog differential of flow f^{opt}. If the maximum differential is negative, $W_{(a,b)}(t) = 0$.

[3] Wikipedia Contributors (2019). Back pressure. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Back_pressure [Accessed 12 May 2019].

The next step is to choose the rate to send flow f^{opt} via link (a, b), i.e. $\mu_{(a,b)}(t)$. This is done by solving the following optimization problem:

$$Maximize: \sum_{(a,b),\mu_{(a,b)}(t)\in\Gamma(t)} \mu_{(a,b)}(t) \cdot W_{(a,b)}(t) \tag{2.4}$$

where $\Gamma(t)$ is the set of all feasible transmission rates that are supported by the network conditions at time t. The backpressure algorithm allocates the full link capacity $\mu_{(a,b)}(t)$ to flow f^{opt} at time t, and route flow f^{opt} via link (a, b). The rest types of flows can not transmit via link (a, b) at time t. Note, the same type of f^{opt} may have multiple activated transmission links. For example, beside link (a, b) it may have another link (a, c) where flow f^{opt} also has the maximum backlog differential. Flow f^{opt} then also transmit via link (a, c) at a rate of $\mu_{(a,c)}(t)$.

The dynamic link activation method by selecting the links with maximum backpressure was proved to yield a maximized throughput for the constrained queuing network. It is also proved that when following the maximum throughput routing policy they proposed, the network is stabilized for any arrival rate for which it is stabilizable. Proofs can be found in Tassiulas and Ephremides (1990). Readers are referred to Tassiulas and Ephremides (1990) for more details. Following Tassiulas and Ephremides (1990), there have been many research works on backpressure routing from many different perspectives, Neely and Urgaonkar (2009); Georgiadis et al. (2006); Dvir and Vasilakos (2011); Moeller et al. (2010); Ying et al. (2011) to name a few. As per Neely and Urgaonkar (2009) and Georgiadis et al. (2006), backpressure routing algorithm has several unique attractive features: 1) by following the algorithm, it leads to maximum network throughput. 2) it is possible to implement the backpressure algorithm without information of traffic arrival rates and link state probabilities. 3) it is proved to be robust to time-dependent network conditions. These three features are ideal for the purpose of routing traffic in a transportation network where overall throughput and system robustness are critical.

2.4 Other routing methods

2.4.1 Learning based routing

Besides the traditional routing algorithms and techniques discussed previously, there is another whole new method that is drawing more and more attention nowadays. It is using different kinds of learning techniques like machine learning, deep learning, particle swarm (swarm intelligent), genetic algorithm, evolutionary algorithm and so on to assist in the task of traffic routing, e.g. Boyan and Littman (1994), Bonabeau et al. (1998), Peshkin and Savova (2002), Fadlullah et al. (2017), to name a few. Some methods have explicit models for optimization like the ant colony optimization algorithm, while others may not have well-defined models for optimization in route planing like deep learning methods that use a deep neural network structure to make predictions.

One of the advantages of the learning based routing method is that it does not necessarily require full and accurate knowledge of the road conditions in the network, and still be able to produce satisfactory routing results. Take reinforcement learning as an example. Reinforcement learning works just like what a normal traveler does in everyday traveling. Every time a traveler makes a trip, it learns the realization of network information and updates its knowledge about the network. This knowledge includes relations and interference between different roads, the historical distribution of network information and so on. The knowledge is stored in the form of a reward function. Next time when the traveler plans his route, he does not need to know exactly the condition of every roads. Based on the knowledge learned and a limited new knowledge obtained for the current network, the traveler can infer an "optimal" route. It is even possible to do adaptive routing during trips as long as a proper reinforcement learning framework is presented.

With all these different existing learning techniques and more new ones to come up, it is almost impossible to thoroughly review and compare each technique. And it is also beyond the scope and targets of this dissertation. This section only serves an elementary introduction to learning based traffic routing techniques. Readers are advised to search in the literature for more in depth understanding of those techniques if one finds them interesting.

2.4.2 Network-On-Chip routing

In Network-On-Chip (NoC) design, efficient routing algorithms also play a very crucial role. As the number of processing elements (PE) integrated on a System-On-Chip (SoC) is increasing at a dramatic rate over the past two decades, the need to communicate between different PEs on SoC is emerging. Though the scale and routing paradigm are quite different from that of a transportation system, there are some similarities in terms of the routing protocol design philosophy and guidelines between the two that can be shared.

A NoC network topology can have different forms depending on the application. According to Palesi and Daneshtalab (2014), there are six commonly seen and commercially used network typologies for NoC design: Shared-Bus, Ring, Crossbar, Mesh, Torus and Butterfly Flat Tree. Different designs of typologies have their own unique strengths and weaknesses. The most similar structure to a road network is the mesh design. See Figure 2.2. It resembles a commonly seen grid network structure in many metropolitan downtown areas.

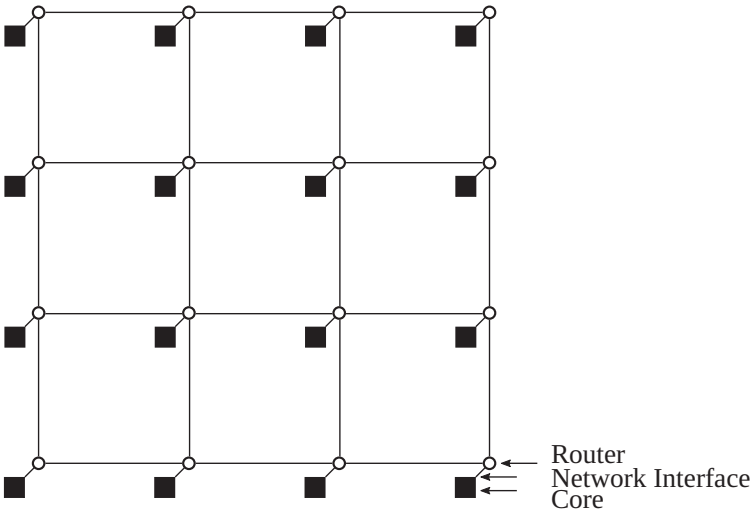


Figure 2.2: A mesh network in NoC.

Various routing mechanisms have been proposed and implemented for NoC applications. For switching mechanisms, two common types are circuit switching and packet switching. The former provides low latency and guaranteed bandwidth, but may have low channel utilization, low throughput and long connection initialization time (Palesi and Daneshtalab, 2014). Packet switching, on the other hand, improves channel utiliza-

tion and network throughput, but may suffer from packet loss, high latency and channel contention. Packet switching relies on buffered flow control to deal with channel allocation and packets buffering (Palesi and Daneshtalab, 2014). Store-and-forward, virtual cut-through and wormhole routing are three commonly used buffer control mechanisms. The first variant, store-and-forward, is the most similar to how individual vehicles are routed through a physical transportation network. In this mechanism, packets buffering works like vehicles being withheld in an intersection. Forwarding a packet to a nearby router resembles the process of routing a vehicle to a neighboring intersection node.

Chiu (2000) developed a deadlock free NoC routing algorithm without the use of virtual channels. The routing rules are simple, as stated below:

Rule 1 Packets are not allowed to take EN turns in an even column, and they are not allowed to make a NW turn in an odd column.

Rule 2 Packets are not allowed to take ES turns in an even column, and they are not allowed to make a SW turn in an odd column.

Rule 3 Packets are not allowed to make 180° turns.

The routing rules specified above is proved to be deadlock-free. In Chiu (2000), a minimal Odd-Even routing algorithm is also presented. Following work in Chiu (2000), Hu and Marculescu (2004) proposed a smart routing algorithm named DyAD-OE for NoC routing that combines Odd-Even routing and XY routing (Intel, 1991) to achieve a deadlock free, dynamic, adaptive and deterministic routing scheme. Short after Hu and Marculescu (2004), Li et al. (2006) in their work re-designed the XY routing algorithm to a congestion-aware deadlock free dynamic routing method called DyXY. They claimed the DyXY algorithm can adaptively route traffic based on congestion condition in the proximity to improve system performance under load, and still guarantee deadlock-free and live-lock free. Lotfi-Kamran et al. (2010) published an enhanced version of DyXY that further reduces latency and strengthens link failure tolerance.

2.5 Traffic assignment models

Traffic assignment models are models built to estimate how traffic flows distribute in a network. These models take traffic demand, network topology and configuration as inputs, and analytically or numerically calculate the way that the traffic flows are loaded onto and distributed in the network. This process can be static, i.e. all demands and/or network configuration are not changing over time, or dynamic, i.e. demands and network configuration are time-dependent. The former is categorized as static traffic assignment, while the latter is dynamic traffic assignment.

Traffic assignment is becoming more and more important to planners, traffic engineers and some government agencies. It is widely used to predict traffic performances in the future for urban traffic management purposes. The estimation results can also serve as an important input for land use planning, as well as for traffic demand analysis including mode choices, trip distribution, destination choices and so on. With its wide range of applications and significant impacts on transportation engineering, it is unwise to omit traffic assignment models in the review of different routing methods. Though there are some clear differences between the two, they are inevitably mingling with each other in many cases. The remaining part of this section gives a brief review on the history of different traffic assignment models, their unique characteristics and their potential usage in a V2X world.

2.5.1 Static traffic assignment

As discussed previously, static traffic assignment is a model that deals with time-invariant traffic demand, and network configuration is static too. Beckmann et al. (1956), first proposed a mathematical model to compute traffic flow distribution in the network. Within their work, they solved the distribution problem both for the User Equilibrium (UE) case and the System Optimal (SO) case. Their model relies on an important while intuitive assumption that each road in the network has a cost function. The cost function is a monotonically increasing function of the quantity of traffic on the road. Following Beckmann et al. (1956), there are numerous studies on solving the traffic assignment problem from different perspectives, Sheffi (1985); De Palma et al. (1998); Koutsoupias and Papadim-

itriou (1999); Correa et al. (2004); Boyce et al. (2005); Nagurney (2013), to name a few.

Under the UE case, travelers are following Wardrop's First Principle (Wardrop, 1952) when they make their route decisions. When more and more people are choosing the same road, the road becomes more and more congested which causes the cost to increase. The cost will keep increasing to a point that new travelers will choose some alternative road as those roads are now with lower cost to travel. In the end, traffic equilibrium is reached as no one has the incentive to change choice.

The SO case, on the other hand, is quite different in terms of travelers' rationale. Travelers are no longer always choosing their actions "selfishly" as opposite to the UE case where everyone seeks to find the best option of their own. In SO, travelers will "cooperate" with each other to make the overall performance better. By operation, some individuals may experience worse than UE cost, while some other individuals may improve their cost compared to UE case. SO is important as it usually has good performance improvement over UE. It is especially interesting to organizations like transportation management authorities. However, SO is not easy to implement as it is not what a "rational" traveler does. To manually force travelers to follow SO rules, certain administrative interventions are needed. Approaches like congestion taxes and tolls are useful tools that nowadays transportation administrative authorities are using widely over the world (Small, 1992; Verhoef et al., 1996; Small and Gómez-Ibáñez, 1997; Zhang and Yang, 2004; de Palma and Lindsey, 2011).

2.5.2 Dynamic traffic assignment

Static traffic assignment discussed in the last subsection, however, is not capable to cope with cases with time-dependent traffic demand, such as scenarios with fluctuated traffic flows intraday. To characterize the time varying characteristics, a more sophisticated and complex traffic assignment model is needed. Merchant and Nemhauser (1978a) and Merchant and Nemhauser (1978b) was the first to formulate a discretized mathematical programming model to solve the dynamic traffic assignment problem. In their model, the demand is fixed and deterministic. There is only one destination with single type of agent. The problem is also limited to the SO case (Peeta and Ziliaskopoulos, 2001). Their model is simple and can be criticized in some aspects, but as a pioneer model in DTA it does illus-

trate the concept of DTA well. Following the track of Merchant and Nemhauser (1978a), different DTA models have been proposed and developed using different approaches and from different research perspectives covering both SO and UE, single destination and multiple destinations, deterministic demand and stochastic demand, and many other interesting scopes. Some of the works are Smith (1984); Friesz et al. (1989); Papageorgiou (1990); Carey (1992); Janson (1991); Ran et al. (1993); Ziliaskopoulos (2000); Mahmassani (2001); Szeto and Lo (2006); Ben-Akiva et al. (2012); Ban et al. (2012a); Qian et al. (2012); Shen and Zhang (2014); Ma et al. (2014); Zhu and Ukkusuri (2015); Patriksson (2015), to name a few.

There are different ways to formulate a DTA problem. Peeta and Ziliaskopoulos (2001) categorizes different DTA models into four groups based on the approaches used: mathematical programming formulation, optimal control formulation, variational inequality formulation and simulation based models. A typical DTA model consists the following components:

1. An objective function for minimization: $Z(\cdot)$
2. A network loading model. For example, the *M-N model* (Merchant and Nemhauser, 1978a):
$$\begin{cases} x_{t+1} = x_t + u_t - v_t \\ v_t = g(x_t) \end{cases}$$
3. Flow conservation.
$$\sum u_t = \sum v_t + F_t$$
4. Other constraints, including traveler's route choice model.

One of the most essential pieces in the DTA model is the network loading model. The network loading model controls how a given inflow is loaded and distributed into a network (Xu et al., 1999). According to Nie and Zhang (2005), there are four types of mainstream dynamic network loading models present in the literature: 1) the *M-N model*; 2) the delay function link model; 3) the point queue model; and 4) the cell transmission model. Each dynamic network loading (DNL) model has its own strengths and weaknesses. A proper DNL model should be selected depending on the scope of the dynamic traffic assignment that one tries to solve.

DTA models, due to its formidable complexity from all the time varying ingredients in the formulation, can be extremely difficult to solve analytically or numerically. Special techniques have been developed to solve different variants of DTA models, including model discretization, approximation, using heuristics and many other innovative approaches. A comprehensive review of all computational methods for solving DTA models is not a focus of this dissertation. Readers are encouraged to delve into the literature for more details.

Chapter 3

Literature Review PART 2: Traffic Signal Control

The following chapter gives a literature review on traffic signal control. We look through the development of the most simple, fundamental signal control designs, as well as the most state-of-art, intelligent and popular signal control strategies. Brief introductions on how each method is designed and implemented are outlined in each section. In addition to that, advantages and limitations of each presented methods are discussed alongside each section.

3.1 Basics of traffic signal control

Traffic is more chaotic and complex at an intersection than on normal road segments due to conflicts of different movements within the intersection. These conflicting movements can come from various of traffic modes, like cars, motorcycles, bicycles, pedestrians and even trains. The conflicts between different movements can greatly increase the risk of traffic collision, which makes traffic intersections more prone to causing traffic accidents especially for unsignalized intersections. According to U.S.DOT Federal Highway Administration, there are over 40% of total U.S. traffic fatalities that are related to intersections (See Table 3.1 [1]). One intuitive and effective solution is to signalize intersections to avoid or limit the possibility of traffic conflicts. This section aims to explain the basics of traffic

[1] Intersection Safety - Safety | Federal Highway Administration. Dot.gov. https://safety.fhwa.dot.gov/intersection/conventional/unsignalized/. Published 2010. Accessed April 1, 2019.

signal control in practice.

Table 3.1: Numbers of U.S. traffic fatalities from year 2010 to 2016.

Year	Total U.S. Traffic Fatalities	Intersection Fatalities	Unsignalized Intersection Fatalities	Unsignalized Intersection Fatalities involving a Pedestrian	Unsignalized Intersection Fatalities involving a Bicyclist
2010	32,999	8,633	6,156	752	171
2011	32,479	8,317	5,875	778	156
2012	33,782	8,851	6,233	860	154
2013	32,894	8,676	6,054	814	179
2014	32,675	8,661	5,986	872	195
2015	35,485	9,665	6,811	864	214
2016	37,461	10,267	7,122	985	200

Cycle Length Total time duration for an entire sequence of signal phases.

Green Split The percentage of effective green time of a certain phase in total cycle length.

Minimum Green/Minimum Red/Yellow Duration The minimum green/red/yellow time required for any signal plan.

Phase A group of movements that are served at the same time.

Movement Turning movement taken by vehicle or pedestrian at an intersection.

Ring Barrier Graph A graph of phases within a set of rings and phases within a set of barriers.

3.2 Fixed time control

Fixed time control, or *pretimed control* in some literature, is the most commonly seen and simplest traffic signal control method. In such control schemes, the entire traffic signal control plan is pre-defined and fixed over specified time frame, and hence the name *fixed time control* or *pretimed control*. Though the concept of fixed time control is simple and straightforward, some use quite sophisticated optimization method to obtain fixed time

plans. The cycle length, green/red time duration and phase sequence are allocated based on the traffic patterns observed historically. As the name indicates, these values are fixed per each plan. There are no detectors necessary in this control mode.

Since the fixed time control lacks the ability to detect sudden changes in traffic demand pattern and the ability to adapt signal control plans to the change of traffic, this kind of control mode is mostly suitable for intersections with little fluctuation in traffic demand. For intersections with high fluctuation in traffic demand, fixed time control will result either in starvation (for approaches with sudden decease in traffic demand) or in serve queuing (for approaches with sudden increase in traffic demand).

3.3 Actuated control

Actuated control, however, does not have a fixed cycle length or fixed phase sequence. Compared to *fixed time control* systems, *actuated control* system is quicker and more flexible in responding to dynamically changing traffic demand.

In *actuated control* mode, there are different types of sensors to detect the presence of vehicles near intersections to actuate signal control. The amount of green time allocated to each phase depends on the traffic demand detected by the sensors. Different from *fixed time control*, *actuated control* works best at isolated intersections with low traffic demand on average. Once the sensor detects vehicles upstream, a signal is sent to the controller to actuate the green time for that approach. Green time is extended if vehicles are detected continuously until maximum green is reached.

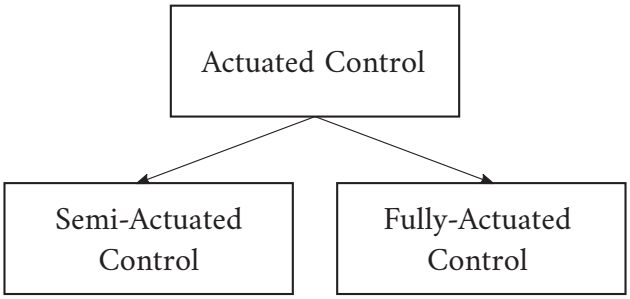
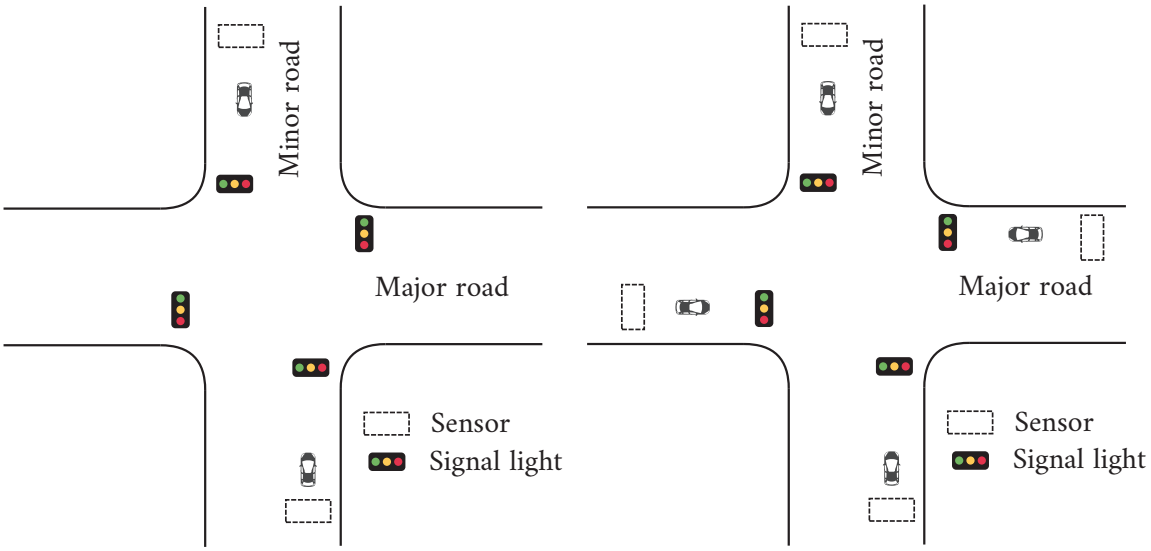


Figure 3.1: Actuated control types.

Based on different applications, *actuated control* can be further divided into two groups: *semi-actuated control* and *fully-actuated control*. See Figure 3.1. In a scenario where there

are a major road and a minor road connected to the same intersection, *semi-actuated control* works best. The major road approach is not installed with sensors and operates in "non-actuated" manner while the minor road is under *actuated control*. Normally, the controller is programmed to give green time to the major road until the minor road receives an "actuated" signal. *Semi-actuated control* performs best for intersections with low speed major road and low traffic demand on minor road (Koonce and Rodegerdts, 2008). In *fully-actuated control*, both major road and minor road are installed with sensors to detect incoming traffic. Whichever approach detects vehicles will actuate the corresponding traffic signal control. This type of control is most ideal for intersections where daily traffic demands and patterns vary significantly (Koonce and Rodegerdts, 2008). See Figure 3.2 for a simple illustration of the two different types actuated signal controlled intersection.



(a) Semi-actuation control. (b) Fully-actuated control.

Figure 3.2: Two different types of actuated signal control.

3.4 Adaptive control

One step further, *Adaptive signal control* is another type of signal control technology that utilizes traffic detection sensors. In contrast to *fixed time control* discussed in the last section, *adaptive control* is capable to addresses the problem of detecting and adapting to the sudden traffic changes. Upstream and/or downstream traffic flows are detected by sensors. Based on the time and location of the detection, a prediction of when and where

those traffic will be is made (Koonce and Rodegerdts, 2008). Some sophisticated algorithms then utilize the prediction to calculate optimal signal timing settings and make appropriate adjustment to the current signal plan.

According to Koonce and Rodegerdts (2008), *adaptive control* can improve average performance metrics by 10%. In some special cases, this improvement can be as high as 50% or more. *Adaptive control* is particularly suitable for scenarios where there are fluctuated traffic demands. For instance, accidents, disruptive events and natural disasters that result in sudden changes to traffic demands. In such cases, *fixed time control* will fail to detect the change and fail to make appropriate adjustment. *Actuated control*, on the other hand, is also not capable to deal with this kind of random and unexpected scenarios. It may perform slightly better than *fixed time control* in these cases. But with random and unexpected traffic flow coming from all directions, the simple actuated plan is very unlikely to be an optimized control plan.

There have been many traffic signal control applications that uses adaptive control methods in the U.S. as well as over the world. Different manufacturers and vendors design different products. The most popular ones that got deployed in the U.S. in the past 20 years (Koonce and Rodegerdts, 2008) include Split Cycle Offset Optimization Technique (SCOOT), Sydney Co-ordinated Adaptive Traffic System (SCATS), Real Time Hierarchical Optimized Distributed Effective System (RHODES) and Adaptive Control Software Lite (ACS-Lite). Though the percentage of adaptive control signals are still low (less than 1% according to Koonce and Rodegerdts (2008)), the future of *adaptive control* is bright with nowadays rapidly developing communication and detection technologies. When technologies like V2X become mature, *adaptive control* will have many more profound applications.

3.5 Coordinated control

In the context of the multiple intersections application, *coordinated control* mode is used to control traffic movements in a cooperative manner. By carefully selecting signal control plans and offsets between different signalized intersections, it synchronizes multiple intersections to improve the operation of the traffic movements in the system.

A very simple but effective way to determine offset between two consecutive intersections is:

$$\Delta = \frac{L}{\overline{v}}$$

where Δ is the offset, L is the distance between the two intersections and $\overline{v}$ is the average travel speed of the link between the two intersections. If vehicle platoons move at a speed close to $\overline{v}$, it takes $\frac{L}{\overline{v}}$ for the platoon to traverse that link. Since this travel time matches the offset between upstream and downstream intersections, i.e. Δ, the signal is in green when the platoon arrives downstream. In such way, the system forms a "green wave" of platoon traveling through the corridor without any stops. The corresponding amount of green time available to a group of vehicles in a progressive signal system is called "green band" in traffic signal control operation. This naive approach might suffers many potential drawbacks, such as low throughput. However, it suffices for illustration purpose.

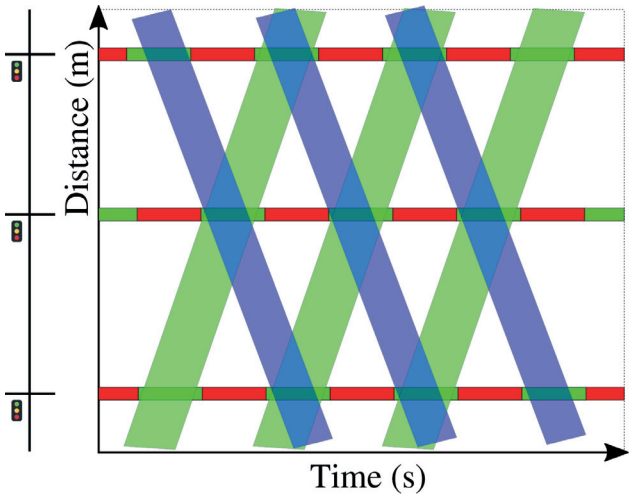


Figure 3.3: A green band (time-space diagram) on a coordinated control corridor.

There are many different methods depending on one's control objective to do *coordinated control* for intersections, including but not limited to cycle/split/offset methods to coordinate traffic. Sydney Co-ordinated Adaptive Traffic System (SCATS), for example, allows traffic engineers to implement maximum throughput, minimum stops and minimum delays based on the vehicle data detected from sensors. It is a cycle-to-cycle system which means the system tries to optimize cycle length, green split and offsets in each cycle [2].

[2]Scats.com.au. (2019). Adaptive control | SCATS. [online] Available at: https://www.scats.com.au/how-scats-works-adaptive.html [Accessed 25 Apr. 2019].

After getting the coordinated signal plan from various software and manual tools, traffic engineers should check and verify the selected plans yield expected outcomes. One approach recommended by Federal Highway Administration (FHWA) (Koonce and Rodegerdts, 2008) is to use a time-space diagram. Figure 3.3 shows a time-space diagram for a coordinated corridor with signal timing plans illustrated for each intersection on the major road. Traffic is allowed to progress within the green/blue (different colors for different directions) regions without the need to stop for red lights. Outside those regions, vehicles are required to stop at least once assuming traffic does not change speed. From this time-space diagram, one can quantify the maximum bandwidth of a certain coordination plan. A common practice is to maximize the bandwidth which means it allows the maximum number of vehicles to travel through the corridor without stops.

Intersection coordination control can reduce travel time, delays and queues in the system for the coordinated movements, it might, however, create negative impacts on the other uncoordinated movements. That being said, *coordinated control* suits best for applications where there is a major corridor (or a network with main arterials and side roads) with multiple signalized intersections. The corresponding crossing roads are minor roads with relatively low traffic demand, while the vehicular traffic on major road is relatively high. Vehicular traffic on major road benefits from shorter travel time, delays, queues and fewer stops, while the impact to traffic on minor roads is kept at a low level due to low traffic demand on those roads.

3.6 Other types of signal control

3.6.1 Max backpressure control

Backpressure is a term originally used in fluid dynamics describing the opposite resistance to the desired flowing direction that causes the pressure drop between high pressure end and low pressure end. The concept is then borrowed and used widely in packet routing, network communication, network control and software development areas (Tassiulas and Ephremides, 1990; Georgiadis et al., 2006; Moeller et al., 2010; Ying et al., 2011; Ji et al., 2013). Backpressure routing algorithm, for instance, seeks to route packets in the direction that has the maximum differential backlog between neighboring hops. This behavior

mimics the natural behavior of liquid fluid flowing through a pipe network via pressure differential [3].

Tassiulas and Ephremides (1990) was the first to propose and develop a *backpressure algorithm* for scheduling policies in a multi-hop radio network. In their *backpressure algorithm*, a weight is assigned to each possible link based on the queue difference between neighboring nodes and the service rate of the corresponding link. See Equation (3.1). After computing all the weights, the set of links (servers) that maximize the weighted backpressure are selected for activation. See Equation (3.2). This gives the final activation vector $\hat{\mathbf{c}}$.

$$D_{ij}(t) = \begin{cases} (X_{q(i)j}(t-1) - X_{h(i)j}(t-1))m_i, & if \quad h(i) \notin V_j \\ X_{q(i)j}(t-1)m_i, & if \quad h(i) \in V_j \end{cases} \tag{3.1}$$

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c} \in S} \{\mathbf{D}^T(t)\mathbf{c}\} \tag{3.2}$$

$X_{q(i)j}(t-1)$: Number of queued customers of class j in queue $q(i)$ at server i by the end of time $t-1$.

m_i: Service rate at server i.

$q(i), h(i)$: Queues at tail and head of link i, respectively.

$\mathbf{D}^T(t)$: Weight vector at time t.

$\mathbf{c}$: One activation vector.

$\hat{\mathbf{c}}$: The activation vector that maximizes backpressure.

V_j: Final destination set of customers of class j.

The type of queue that activates each link i correspondingly are then released via link i at a rate of m_i. Tassiulas and Ephremides (1990) showed that by following the *backpressure algorithm*, network can obtain maximum throughput. They also proved that with any arrival rate $\mathbf{a} \in C'$, the optimal policy is stabilizing the system. C' is defined as:

$$C' = \{\mathbf{a} : \text{there exists } \mathbf{f} \in F_a, \mathbf{c} \in co(S) \text{ such that for the corresponding } \hat{\mathbf{f}}$$
$$\text{we have } m_i^{-1}\hat{f}_i < c_i \text{ if } \hat{f}_i > 0 \text{ and } \hat{f}_i = 0 \text{ if } c_i = 0\}$$

[3] Wikipedia Contributors (2019). Backpressure routing. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Backpressure_routing [Accessed 25 Apr. 2019].

where $co(S)$ is the convex hull of the constraint set S.

Backpressure algorithm looks promising in applying to other fields besides network communication area where throughput of the system is one of the system performance goals. One of such applications is traffic light control in transportation networks. Transportation networks and packets transmission/communication networks share many similarities. Packets in communication networks are like vehicles in transportation networks, where queues can be observed in both types of networks. Servers in communication networks are like intersection controllers in transportation networks controlling which movements are allowed to pass the intersection.

However, there are also a few differences between the two that need to be addressed. First of all, the queues in Tassiulas and Ephremides (1990) can be distinguished by their destinations into different groups. This is not feasible in transportation networks context as the queues with different destinations are physically mixed together. Therefore, it is impossible to calculate the activation link set in the maximum backpressure algorithm based on queue types.

Secondly, weight calculation (Equation 3.1) are based on the backlog differential between tail node and head node of a link. Different queues share the same transmission link. However, in transportation networks queues at intersections can have different movements, i.e. after they pass through the intersection they can be at different locations. See Figure 3.4. Links W, E, N, S are links connected to node C. An example of flows within this intersection is a traffic flow moves from link W to link N by taking a left turn. To account for this difference between a transportation network and a communication network, one solution is to convert the intersection into a hyper-graph. See Figure 3.4. Links in the original graph now become nodes, and nodes in the original graph becomes hyper-links. Links W, E, N, S in the original network (on the left) becomes nodes W, E, N, S in the hyper-graph (on the right). Node C in the original network expands to a hyper-link C that connects all surrounding four nodes. By doing this conversion, the network setup is similar to that in the original backpressure algorithm: 1) traffic flow travels from one end of a hyper-link to the other end of a hyper-link via the hyper-link. 2) Weights and the corresponding backpressure are calculated using the backlog differential between two

neighboring nodes, e.g. the weight of arc (M, N) equals to the queuing difference between node W and node N, or $(Q_W - Q_N)$, the backpressure then equals to $c_{(M,N)} \cdot (Q_W - Q_N)$. 3) The pair of nodes with the highest backpressure get the right of way to flow through the hyper-link C.

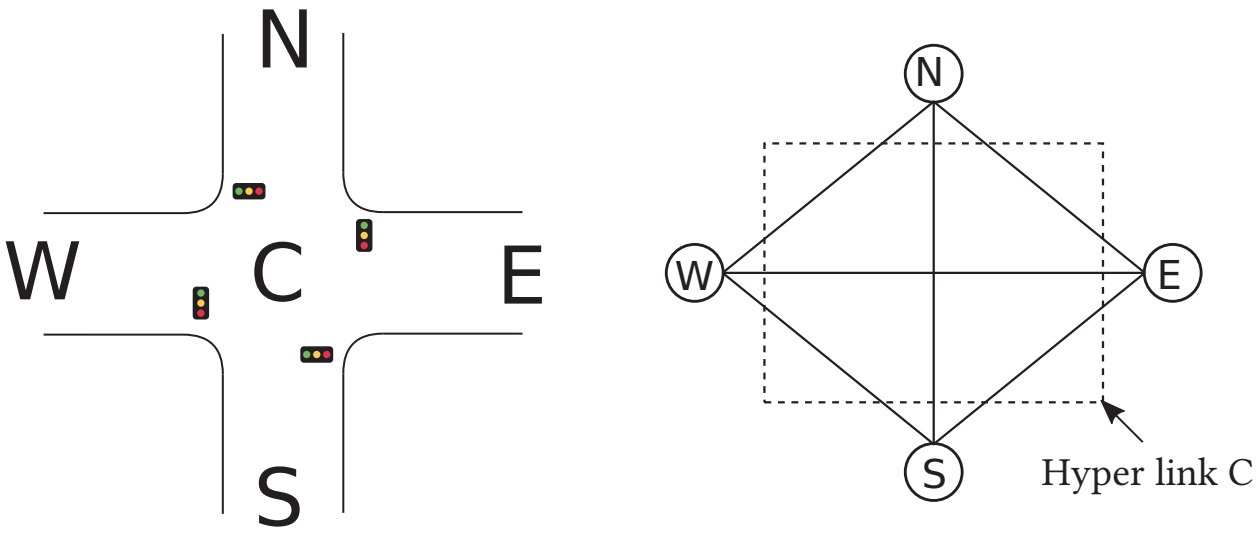


Figure 3.4: An intersection in hyper-graph representation.

Thirdly, the server in communication network has full control of the transmission rate via the link. However, a traffic signal light controller only controls which movement has the right of way. It has no control of how fast the queue in a certain movement can discharge. The discharging flow rate is a value bounded between 0 and f_{max}, i.e. the saturation flow rate. This depends on the traffic conditions for that movement at the intersection.

Finally, the max backpressure algorithm assumes that the server has infinite buffer storage for queues. This is particularly not true for transportation networks as roads usually have a small queue capacities.

With the differences discussed above, the original max backpressure algorithm needs a few modifications before it can be applied to control traffic signal lights. In the remaining of this section, we introduce some works that follow Tassiulas and Ephremides (1990)'s work, and are successfully adapted to transportation traffic signal control.

3.6.1.1 Queue-based max backpressure control

Based on the work by Tassiulas and Ephremides (1990) and Georgiadis et al. (2006), Wongpiromsarn et al. (2012) proposed a distributed signal control method using max backpres-

sure approach that maximizes throughput of the network. Their max backpressure control framework is a specialization of Tassiulas and Ephremides (1990)'s original backpressure algorithm with single commodity. To address the lack of flow rate control, Wongpiromsarn et al. (2012) slightly modified the definition of weights and backpressure as Equation (3.3-3.4).

$$W_{ab}(t) \triangleq Q_a(t) - Q_b(t) \tag{3.3}$$

$$S_p(t) \triangleq \sum_{(L_a, L_b) \in p} W_{ab}(t) \xi_i(p, L_a, L_b, z_i(t)) \tag{3.4}$$

Weights are still calculated by the backlog differential. But the differential is now taken between the "from" link (L_a) and the "to" link (L_b) of a movement. Flow rate used to calculate backpressure is no longer the maximum throughput of the intersection. Instead, it is a function ($\xi_i(\cdot)$) of current phase (p), incoming link (L_a), outgoing link (L_b) and current traffic state ($z_i(t)$). The rest procedures are similar to Tassiulas and Ephremides (1990). Wongpiromsarn et al. (2012) proved in their work that using the modifications specified the distributed max backpressure control method for signal control inherits the properties of the original backpressure algorithm, including maximum throughput, stability and so on.

Almost at the same time as Wongpiromsarn et al. (2012), Varaiya (2013) proposed a signal control method based on max backpressure algorithm. It adopts the same idea of backpressure algorithm for routing and scheduling of packet transmission in a wireless network developed by Tassiulas and Ephremides (1990). However, as the author claims the network model and the notion of stability in Varaiya (2013) is significantly different from those in Wongpiromsarn et al. (2012). We present here the max pressure signal control model from Varaiya (2013).

$$w(l,m)(X) = \chi(l,m) - \sum_{p \in Out_m} r(m,p)\chi(m,p) \tag{3.5}$$

$$\gamma(S)(X) = \sum_{l,m} c(l,m)w(l,m)(X)S(l,m) = \sum_{l,m:S(i,m)=1} c(l,m)w(l,m)(X) \tag{3.6}$$

$$u^*(X) = \arg\max\{\gamma(S)(X)|S \in \mathcal{S}\} \tag{3.7}$$

(l, m): A traffic movement with in an intersection: input link is l, output link is m.

$\chi(l, m)$: Queue length of movement (l, m).

$r(l, m)$: Turning ratio for movement (l, m).

$w(l, m)(X)$: Weight of movement (l, m) under state X.

$c(l, m)$: Saturation flow rate of movement (l, m).

$S(l, m)$: Entry of intersection control matrix. $S(l, m) = 1$ indicates that phase (l, m) is actuated, $S(l, m) = 0$ otherwise.

$\gamma(S)(X)$: The pressure of the intersection under state X and with intersection control matrix S.

X: Intersection state, indicating queuing lengths at the intersection.

At every state X, their *max pressure* selects the intersection control matrix that has the maximum pressure for network signal control, i.e. selecting the phase with maximum pressure as the phase to be actuated. The author in their paper also proved properties of stability and maximum throughput in their max pressure signal control.

3.6.1.2 Delay-based max backpressure control

Motivated also by the backpressure routing from Tassiulas and Ephremides (1990), Wu et al. (2018) proposed a *delay-based backpressure* signal control method to address the fairness concerns in traffic signal control while applying the maximum backpressure algorithm. The control methods in both Varaiya (2013) and Wongpiromsarn et al. (2012) use the number of queuing vehicles as the backlog to calculate weights and the corresponding backpressure. However, this queue-based approach may result in the situations where vehicles in some movements experience excessively long waiting time (or delay). Consider the following scenario. In an intersection with a major road and a minor road, the major road has large traffic demand. Consequently, queues on major roads are long. The minor road has small traffic demand and queue length is short. In this case, the queue-based max backpressure control will always allocate the right-of-way at the intersection to the major road, leaving the vehicles on the minor road being delayed forever. The overall fairness of

this intersection under the scenario described above can be poor.

The so-called delay-based max backpressure control in Wu et al. (2018) modified the weight calculation method. In their method, they measured the sojune time of the Head-Of-Line (HOL) vehicle of $Q_{i,j}$ in the network at time slot t, $W_{i,j}(t)$, and use that as the weight.

$$W_{i,j}(t) \triangleq T_{i,j,1}(t)$$

where $T_{i,j,1}(t)$ is the sojune time of the k^{th} vehicle of $Q_{i,j}$ in the network at time slot t. Time is measured from the point when the vehicle first arrives in the network. Optimal phase selection is based on the modified backpressure calculation method:

$$\overrightarrow{p}^*(t) \in \arg\max_{\overrightarrow{p} \in S_P} \sum_{p_{i,j}=1} \gamma_{i,j} \cdot W_{i,j}(t) \cdot \mu_{i,j}(\overrightarrow{p})$$

They proved that by using the delay-based max backpressure control, the system is stable as long as the arrival rate is strictly within the region of Λ, with the definition of Λ listed below. The property of maximum throughput is also maintained. Simulation tests confirms that the fairness performance regarding delay is better than that of the queue-based approaches, especially in busty and heterogeneous traffic conditions.

$$\Lambda = \{ \overrightarrow{\lambda} | \exists \overrightarrow{\phi} \in Co(S_P) \quad s.t. \lambda_i \leq \phi_{i,j}, \forall (i,j) \in M \}$$

Furthermore, to achieve better flexibility, Wu et al. (2018) also proposed a maximum backpressure control with a combination of queuing length and HOL delay. The optimal selection is based on the following:

$$\overrightarrow{p}^*(t) \in \arg\max_{\overrightarrow{p} \in S_P} \sum_{p_{i,j}=1} \gamma_{i,j} \cdot [\eta_{i,j}^{(W)} W_{i,j}(t) + \eta_{i,j}^{(Q)} Q_{i,j}(t)] \cdot \mu_{i,j}(\overrightarrow{p})$$

The performance of the "hybrid" method is similar to delay-based method, but has better flexibility to adjust strategies in response to different traffic conditions intraday by assigning different weights to queuing length and delays, respectively.

The max backpressure control shines with a lot of great features and benefits when applying in transportation applications. However, there are still some drawbacks. One of the drawbacks of the above backpressure control method in applying in transportation

network is that it does not consider the maximum/minimum green/red time constraints. However, this can be easily fixed by limiting the minimum/maximum number of time slot that a selected phase can be actuated by max backpressure control. Note, this modification could potentially affect the stability and maximum throughout properties. Also, it does not explicitly address green waves in coordination. So it may not perform as well as coordinated control in lightly loaded networks where queuing can be eliminated.

Chapter 4

On Dynamic Traffic Routing And Adaptive Signal Control

4.1 Introduction

Regular vehicles like the ones most of us are driving today don't have real-time road traffic information. The routing strategies used are either based on past experience or based on limited local information. This leads to the fact that regular vehicles can't respond to road incidents in a timely manner to avoid long delay. However, with the emergence of connected vehicles technology it is possible to access both local and global real-time traffic information via the V2X infrastructure. It is an important and challenging research problem to study how to take advantage of the extraordinarily rich information we can get from the connect vehicle system. The main objective of this chapter is to study the interaction between two major components: dynamic vehicle routing and adaptive traffic signal control in a connected vehicle environment. We consider different combinations of route choice strategies and various traffic signal control methods to obtain an effective joint vehicle routing and signal control scheme which will reduce the average travel time within the network.

Vehicle routing problems have been very well studied over the years. A great part of the existing papers studies how to route vehicles in a network efficiently to meet some constraints like location routing (Perl and Daskin, 1985; Min et al., 1998; Nagy and Salhi, 2007) or time constrained routing (Desrosiers et al., 1984; Solomon and Desrosiers, 1988;

Desrosiers et al., 1995; Nie and Wu, 2009). The routing problem in this study is focused particularly on how to route vehicles in the network so that the total time for the vehicles to get to their destinations can be minimized. The underlying problem for that is the well-known shortest path problem(SPP).

In the study of shortest path problem, the existing work can be categorized in two main groups: deterministic shortest path problem(DSPP) and stochastic shortest path problem(SSPP). Bellman (1956) proposed a dynamic programming method to solve the optimal route from one point to another with all link travel time to be deterministic and known ahead of time. SSPP is more interesting than DSPP as in real word link cost is usually not known deterministically but has many uncertainties. In stochastic scenarios, the shortest path problem can be further categorized into two groups according to Gao and Chabini (2006): path problem and optimal routing policy problem. Path problem aims to find a specific path(a deterministic link set) to destination that will attain a certain objective, such as least expected travel time (Miller-Hooks and Mahmassani, 2000) or maximum on-time arrival probability (Fan et al., 2005). Optimal routing policy problem, however, is more complicated than the path problem. According to Gao and Chabini (2006), a routing policy is defined as a decision rule that specifies which node to take next at each decision node based on realized link travel times and the current time. Compared with path problem, optimal routing policy problem in most cases can give a routing solution that is more efficient and reliable. The reason for that is because as the traveller travels in the network, he gains knowledge of the network (in this example, the travel time experienced after the traveller traverses a link). With this knowledge and the dependency of links being known, his anticipation of the future can be changed. The traditional path finding method does not take advantage of the newly learned information and the dependency between links. In order to take these advantages to improve the routing decision, the aforementioned optimal routing policy is needed. The optimal routing policy will not give one fixed path but a decision tree that will guide the traveller to the next node based on the current state(in most of the cases it is the arrival time at the current node) at the decision node. The optimal routing policy is particularly useful in stochastic and time-dependent networks. This routing policy is sometimes known as hyper-path in some literature. Miller-Hooks and

Mahmassani (2000) studied the least expected travel time problem using hyper-path algorithm in a stochastic and time-varying network. Fu (2001) also studied an adaptive routing approach with real-time information. Chen and Nie (2015) studied the stochastic optimal routing problem for vehicles with a limited travelling limit. The problem is formulated as a two-stage stochastic shortest path problem: both stages are a stochastic shortest path problem respectively. A label correcting based algorithm is used to solve the problem. However, their model does not consider time-variant link cost. Wu (2015) studied the travel reliability as an extension to the traditional shortest path problem in stochastic and time-dependent networks by adding the standard deviation to the mean travel time to represent the reliability of a certain route. However, the work does not explicitly consider the time-dependent problem in their formulation.

For signal control, most of the existing papers do not consider the interaction with traffic routing. The traditional control method, no matter adaptive or fixed-time, isolated or coordinated, only aims to reduce the delay or maximize the throughput of the intersection with known and perhaps time-dependent traffic demand (Rosdolsky, 1973; Hunt et al., 1982; Lo, 1999; Mirchandani and Head, 2001; Choy et al., 2003; Tatomir and Rothkrantz, 2004; Cheng et al., 2006; Haddad et al., 2013). This in reality might cause inefficient traffic routing, as traffic may oscillate between different routes due to the impact on travel delay caused by the signal control. Vehicles may make unnecessary reroute to avoid a red light at a certain intersection: for example when they see a red signal for the through movement, they may change their route and take a right turn in order not to wait at that intersection. This myopic behavior not necessarily guarantees to reduce the total travel time as the traveler may experience more red light stops (more delay) at the downstream intersections. It also causes fluctuations of road traffic as travelers are switching their routes too frequently. The fluctuation of road traffic has many negative effects on traffic control (Horowitz, 1984; Friesz, 1985; Zhang and Nagurney, 1996). So a good signal control strategy should take the interaction with vehicle routing behavior into consideration to achieve a better overall performance: not only reducing average travel time, but also maintaining a relatively stable on-road traffic. It requires the work to integrate traffic routing together with signal control. The problem is no longer a simple shortest path problem but a more complicated

time-constrained shortest path problem(TCSPP).

In early literature, there have been many papers trying to solve the combined traffic assignment and traffic control problems, known as CTAC. Smith and Ghali (1990) studied the dynamics of traffic assignment and traffic control. When demand was constant, they were able to get a steady (equilibrium) state, but for dynamic demand they were unable to obtain such results. Later on, Yang and Yagar (1995) formulated the CTAC problem in a bi-level programming formulation. The upper level tried to minimize the system cost by varying the signal settings while the lower level modeled travellers' routing behavior which will give an equilibrium flow state given the signal settings. They also proposed an efficient method to solve the bi-level optimization problem. All the aforementioned papers were focusing on static networks. In reality traffic flows are varying from time-to-time and from day-to-day. A more sophisticated model should be used in order to solve the real world problem. Xiao and Lo (2014) formulated a joint dynamical traffic system that considered both travellers' route choices and traffic light control. The dynamical system permitted the signal controller to interact with and adapt to route choices of travellers, and vice versa in a day-to-day setting. Zaidi et al. (2015) applied the back-pressure algorithms from communication networks to the traffic network with adaptively re-routing traffic.

The aforementioned papers that jointly consider traffic assignment and signal control are all path based, and most of them consider static user equilibrium rather than adaptive control, which means they are unable to provide real-time routing guidance to individual travellers. In real world applications, it is more useful and has more significant impact to travellers if a system can offer good routing guidance at an individual level. This brings attention to combined traffic routing and signal control. Chen and Yang (2000) studied the shortest path problem in traffic-light networks. The constraints in their work were called time-windows which were actually a mathematical representation of phases in traffic light cycles. Their shortest path algorithm could achieve a time complexity of $O(r \times n^3)$, where n denotes the number of nodes in the network and r is the number of different time-windows in a node. However, this work only considered deterministic and static link travel times. Yang and Miller-Hooks (2004) extended the work of Miller-Hooks and Mahmassani (2000) to incorporate the traffic signal operations. Their network

flow was time-varying and stochastic. They studied two different sub-problems: one with actual signal timing known a priori, and the other with probabilistically known signal timings. They proposed algorithms for both problems to solve the adaptive routing in a signalized network. The underlying assumption in their paper was that the distribution of time-varying stochastic link travel times is known a priori. This assumption is too strong when applying to real world scenarios. In reality, travellers might know the distribution of link travel times in the near future (Zhang and Rice, 2003; Chien and Kuchipudi, 2003; Lin et al., 2004). But in a longer scale of time, it is impossible for travellers to know the future link travel time accurately.

The work in this chapter addresses the issues of existing methods with the following contributions:

1. We have proposed a dynamic routing strategy that can constantly update travellers' knowledge of link travel time with consideration of adaptive signal control for efficient routing in real world transportation networks. The joint model of dynamic traffic routing and adaptive signal control developed in this chapter is shown to reduce the average queue length and average travel time, as well as increase the average speed in the network. It shows significant advantage over the other traditional ways of routing and signal control.

2. We have proposed, tested and compared different signal control methods under different scenarios.

3. We have used the OmNet++ and SUMO simulation platform to study the benefit of VANET on the joint routing and signal control strategy proposed. The effect of using real-time information is studied and evaluated.

The rest of the chapter is organized as follows: Section 4.2 exposes some difficulties and existing problems in routing and signal control that motivate us to study the topic. The challenges to address these issues are also stated later in the same section. The joint routing and control problem is stated and algorithms are proposed in Section 4.3. Some numerical experiments are designed and carried out in Section 4.4. Results and analysis are given

in Section 4.5. Section 4.6 summarizes the main results of this work, and discusses future extensions to the current work.

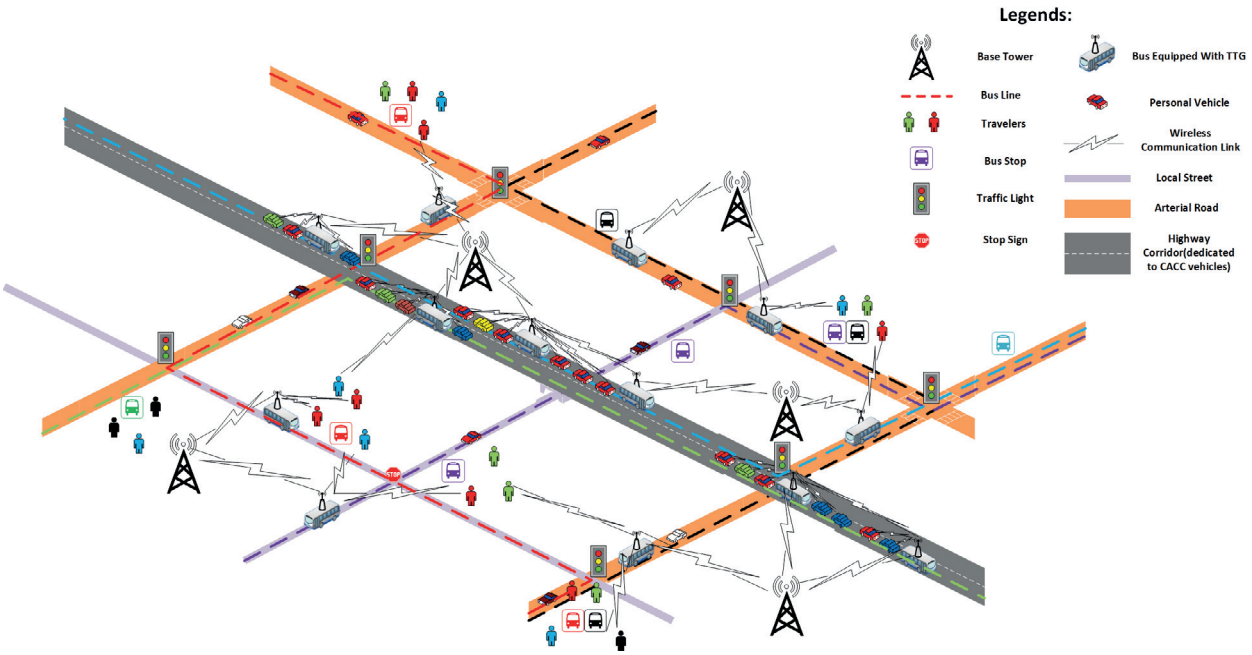


Figure 4.1: Connected vehicles in VANET.

4.2 Context and motivation

We have briefly talked about the problems associated with the traditional routing and signal control methods in Section 4.1. In this section, more details on the existing problems that motivate us to study the topic are discussed and explained. Figure 4.1 shows a typical suburban road network in the future that are facilitated by Vehicular Ad hoc NETwork (VANET) technology. It consists of a freeway (in the middle), some main arterial roads and some minor local roads. The vehicles travelling in the network are all equipped with wireless devices that enable them to communicate with each other (V2V technology). Base towers are served as communication hubs to collect and distribute information to individual vehicles (V2I technology). Technically, the VANET technology provides an opportunity to adopt more sophisticated routing and control methods. Here we list the three most critical aspects of the problems we are facing in joint routing and traffic signal control in real life. These are also the difficulties and challenges that drive us to study the problem.

- **Time-dependency:** The traffic on the road is evolving at all times. A road segment that is not congested during the last hour can become congested in the current hour. Moreover, for the same time-of-day, the same location can be congested today but not congested tomorrow. Such within day and between day fluctuations in traffic conditions and travel times make traffic routing and signal control a difficult task. A dynamic framework that can update the routing decision and signal control timing dynamically should be studied in order to fully consider the time-dependency.

- **Uncertainty:** There are many uncertainties on the road: accidents may happen unexpectedly; disasters like hurricane and flood may destroy and block road segments without early notice. These uncertainties can cause significant delays to travelers if there are no appropriate solutions to deal with them. Uncertainties are hard if not impossible to predict. Instead of predicting the unexpected incidents, we can come up with a good real-time updating framework that can respond to any unexpected incidents in a timely manner.

- **Stochasticity:** Even with the most up-to-date technologies we have today, it is impossible to capture the absolutely precise traffic state. For example, we can use floating cars to estimate the travel time for a certain road segment. But, the penetration rate of such floating cars limits the accuracy of the estimation. Uncertainties discussed previously can also impose difficulties to traffic state estimation. Thus, there is stochasticity residing in the joint routing and signal control problem: link travel time estimation and signal timing estimation. A comprehensive approach which should take careful consideration of such stochasticity is desired to make the routing and signal control strategy reliable, robust and effective.

We show two examples of the problems travelers might encounter if the aforementioned three aspects are not addressed appropriately. As a precursor to this study, we carried out simulations that use traditional routing and signal control methods to observe any possible problems. By traditional routing and signal control methods, we mean that the default traffic routing algorithm (deterministic shortest path algorithm) and the default signal control methods (fixed timing or simple adaptive control) implemented by SUMO

are used in the simulation. Other inputs and parameters are the same as the simulation setups described in Section 4.4.

- **Longer travel time:** When we run simulations using the traditional routing algorithm, we observe that many travellers are not taking the "optimal" routes in terms of expected total travel time to destinations. The reason is that travellers have no access to the most up-to-date travel time information when they plan their routes, and the routes deemed optimal earlier may no longer be optimal later as traffic conditions change. The accuracy of the knowledge of the current and future traffic conditions becomes even more important when traffic signals are considered as one might encounter unexpected delays at intersections. A well designed joint routing and signal control model that can select routes with shorter travel times and avoid waiting at intersections is desired.

- **"Looping" en-route:** Another interesting observation is that a great number of vehicles are travelling back and forth on some links even when new link travel time is available to travellers. They loop in the network and spend much longer time before reaching their destinations. When travellers receive new information and replan their route, sometimes they will find that going back to the previous link will be a "best" choice at the moment. Thus, they would make a "U-turn" and travel backwards. This is due to uncertainties in the network and also the inaccuracy in estimating the future network states, especially the traffic signal timing plans and link travel times. The current link travel time alone is usually not a good resource to calculate the shortest path in a stochastic and time-dependent network. In order to better describe the link travel time characteristics, a well designed travel time updating model that considers both current information and historical information is needed.

With the help of rapidly developing VANET technology, it becomes possible to gain on-line information of on-road traffic like queues, average speed, delay at intersections and so on. In the future, most of the on-road vehicles will be equipped with wireless devices which will enable the vehicles to communicate with each other and with the con-

trol centers. These vehicles are known as Connected Vehicles. They use V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure) technologies to share information in the network. There will be two networks: one is the traditional physical transportation network, and the other is the so-called VANET that uses connected vehicles as mobile nodes. With real time traffic information being available, dynamic routing and adaptive signal control become possible. In our work, we also consider a stochastic and time-varying network.We propose a strategy that will constantly update the knowledge of network, then use this new information to update travellers' route choice. A hyper-path based solution is developed in this work. We also propose five different adaptive signal control strategies (including one method from other's work and one modified based on it) that are interacting with the dynamic routing to achieve a better system performance, which most of the existing papers do not consider.

4.3 Model formulation

We have a general transportation network, which is denoted using graph representation $G = (V, A)$. V is the set of nodes in the network, $|V| = v$; A is the set of edges in the network, $|A| = m$. $V_s \subset V$, is a subset of nodes that are controlled by the traffic lights. $\tau_{ij}(t)$ is the link travel time for link (i, j) at time t. It is a time variant variable. In reality, link time cost is determined by many factors, including background traffic flow, link condition, driver's preference and so on. Table 4.1 shows all the symbols used in the following sections and their corresponding definitions.

There are two components interacting with each other in our problem: dynamic traffic routing and adaptive signal control. Each of these two components has specific inputs, and yields outputs which can be inputs to the other component. Figure 4.2 illustrates the whole system framework for joint traffic routing and signal control. The simulations discussed in the later sections are all designed based on this system framework.

4.3.1 Dynamic traffic routing

There are many different adaptive routing strategies having been studied in the literature. To keep the problem as simple as possible first, this study uses the strategy stated as follows: when a vehicle arrives at a node, it will get updated link travel time, queuing length

Table 4.1: Symbols used in the chapter.

Symbol	Definition
A	Set of edges
V	Set of nodes
v	Number of nodes
m	Number of edges
V_s	Set of nodes that are controlled by traffic lights
$\tau_{ij}(t)$	Travel time on link (i, j) at time t
λ_i	Minimum cost from node i to destination node
λ_i^h	Minimum cost from node i to destination node with upstream node to be node h
π_i	Previous node of node i in the shortest path
ϕ	Intersection delay
$\rho_{ij}(t)$	Probability of link travel time $\tau_{ij}(t)$
ω_i^l	The l^{th} phase at intersection i
Ω_i	The set of phases at intersection i
$\Gamma(i)$	The set of adjacent nodes of node i
T	The time horizon for stochastic and time-varying network, after which the network becomes static

and traffic signal plans at the time it arrives. Based on that new information, the traveller will perform a new shortest path calculation (the algorithm used is given in the following parts of this section), and follow the new calculated shortest path thereafter until he reaches the next node. However, this procedure doesn't have to be performed every node for the purpose of calculation efficiency. Since the routing strategy here is changing dynamically over time, we refer it as *Dynamic Traffic Routing (DTR)* in the following sections in this chapter. In contrast, *Adaptive Routing* without dynamic information updating is simply referred as *AR*.

In a general transportation network setting, traffic conditions are changing constantly. In the old days when real-time traffic information is not available, travellers plan their

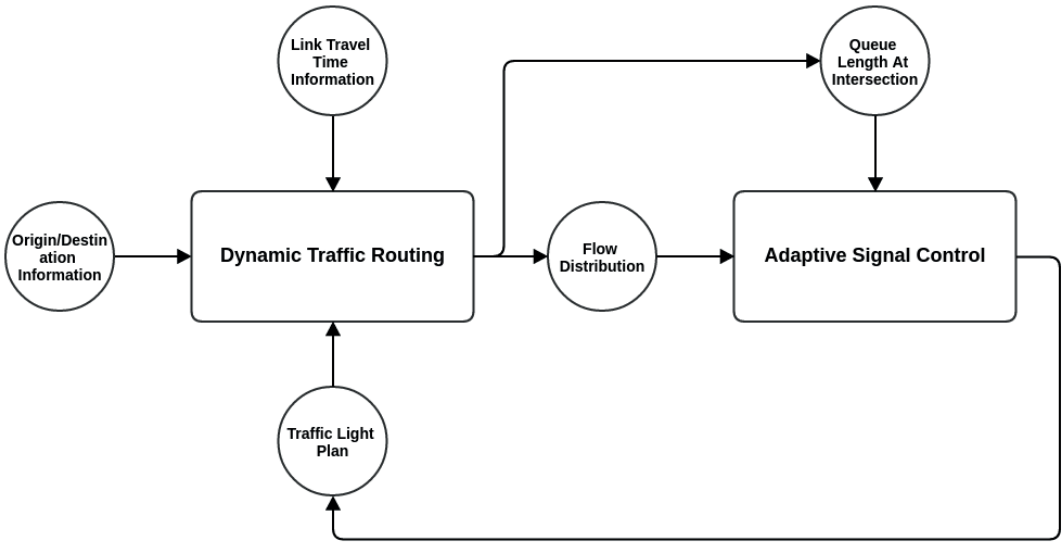


Figure 4.2: Interactions between dynamic traffic routing and adaptive signal control.

routes before their trips using the information that is available at that time. Once a route is selected, the choice is fixed. When some unexpected incidents happen on the links in the route, the travellers will experience longer than expected travel time. However, with the real-time traffic information being available, travellers are able to change their routes en-route. Their knowledge of traffic information will keep refreshing. Once the knowledge is updated, routes can also be updated.

Traditional routing usually uses deterministic shortest path algorithm to obtain new route. A well-known and widely used shortest path algorithm is Dijkstra's Algorithm (Dijkstra, 1959). A simple Dijkstra's Algorithm can be formulated as below:

$$\lambda_i = \min_{j \in \Gamma(i)} \{\lambda_j + \tau_{ij}\}$$

where λ_i is the minimum travel time from node i to destination; τ_{ij} is the link cost for link $(i, j$. This formulation assumes that link travel time is constant during the entire trip. However, this assumption is not realistic in most real world scenarios. Link cost varies from time-to-time and from day-to-day. It is intuitive to formulate link travel time τ_{ij} as a function of time t. So the link cost for link (i, j) at time t will be $\tau_{ij}(t)$. A modified Dijkstra's Algorithm is then

$$\lambda_i(t) = \min_{j \in \Gamma(i)} \{\lambda_j(t + \tau_{ij}(t)) + \tau_{ij}(t)\}$$

4.3.1.1 Stochastic link travel time

In the formulation above, we can see that calculation of the minimal travel time cost to destination depends on the travel cost to destination, i.e. $\lambda_j(t + \tau_{ij}(t))$, of the nodes that have not yet been traveled. This requires an estimation of the future link cost. Due to the randomness and uncertainties mentioned in the previous sections, this estimated link cost subjects to some stochastic fluctuations. It is straightforward to formulate the problem in a stochastic manner.

A link (i, j) can have K possible costs:

$$\tau_{ij}^k(t),\ k = 1, 2, \cdots, K.$$

Each of these link costs $\tau_{ij}^k(t)$ will have an associated probability:

$$\rho_{ij}^k(t),\ k = 1, 2, \cdots, K.$$

4.3.1.2 Hyper-path based stochastic shortest path

Taking the stochastic link cost into consideration, the formulation for the hyper-path based shortest path formulation becomes:

$$\lambda_i(t) = \min_{j \in \Gamma(i)} \left\{ \sum_{k=1}^{K} \left\{ \left[\lambda_j(t + \tau_{ij}^k(t)) + \tau_{ij}(t) \right] \times \rho_{ij}(t) \right\} \right\} \tag{4.1}$$

The solution of the problem is a hyper-path (Miller-Hooks, 2001). Figure 4.3 shows an example of a hyper-path tree. For every node, we maintain two different vectors: 1) the vector of next node to take at every time step; 2) the vector of the cost from current node to destination at every time step. By tracking down the tree by the labels, one can retrieve the shortest path to destination for any time step.

4.3.1.3 Link travel time updating

The fact that a dynamic shortest path algorithm outperforms a static shortest path algorithm is because the dynamic algorithm can use the most up-to-date traffic information to make the shortest path calculation more accurate. In order to achieve this, a dynamic link travel time updating scheme is needed. In the setting of this study, every link maintains a set of possible link travel time realizations as well as their corresponding probabilities. After a certain period of time, every link will have a set of new link travel time realizations

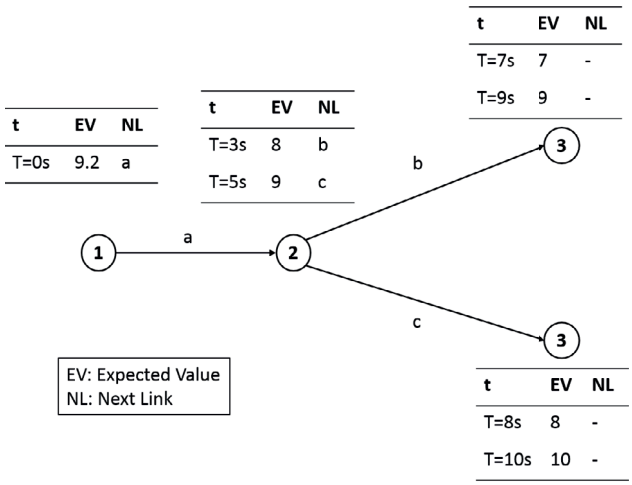


Figure 4.3: An example for hyper-path tree.

(they can be new values or values that are overlapped with the initial set). An updating scheme will merge the old set and the new set.

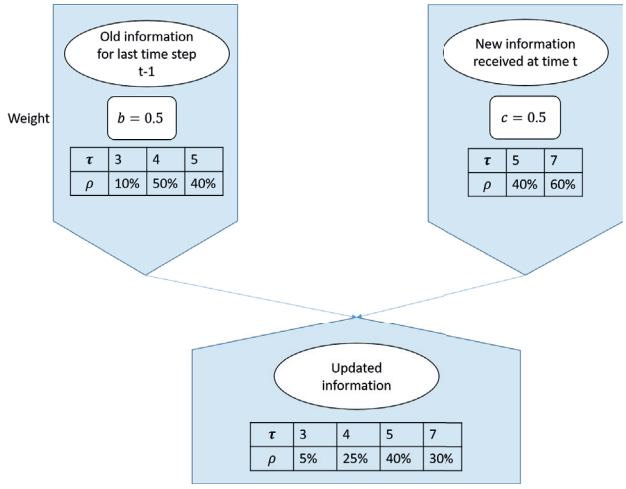


Figure 4.4: An example for travel time updating.

Figure 4.4 shows the updating method used in this study. Travellers will have different preferences towards old information and newly received information. It is characterized by the weights, i.e. b and c respectively, shown in the figure. This preference can be affected by travelers' experience, traffic conditions, time of day and so on. In the example in Figure 4.4, weight for old information is $b = 0.5$, and weight for new information is $c = 0.5$. Updated travel time distribution can be calculated using the following equation:

$$\rho^{updated}(t=\tau) = \begin{cases} b \cdot \rho^{old}(t=\tau), & if\ \tau \in T^{old} \\ c \cdot \rho^{new}(t=\tau), & if\ \tau \in T^{new} \\ b \cdot \rho^{old}(t=\tau) + c \cdot \rho^{new}(t=\tau), & if\ \tau \in T^{old} \cap T^{new} \end{cases} \tag{4.2}$$

$$T^{updated} = T^{old} \cup T^{new} \tag{4.3}$$

Where T^{old}, T^{new} are the sets of old and new link travel times.

4.3.2 Adaptive signal control

Our research work considers a VANET environment which means the traffic information can be very detailed. The most important feature is that this information is real-time. In the old days, accurate real-time traffic information is not always available. Most traffic research papers use the so-called "forecast" or "predicted" data to design traffic control scheme. Due to the nature of the inaccuracy of the data source, the control algorithm is sometimes not working effectively especially under congestion scenarios in which traffic usually has larger unpredictability (Noland and Polak, 2002; Zheng and Van Zuylen, 2010).

With real-time traffic information being available, it now becomes possible for real-time adaptive signal control. Different adaptive control methods are proposed and tested in this study. The first one is a low-density control algorithm which is designed for low traffic volume situations. The second is a high-density algorithm which is for more congested situations. There is also a third control method called *Phase Selection Control* algorithm which chooses a phase candidate to switch to and determines its green duration to give the maximum throughput of the intersection. Varaiya (2013) proposed a traffic control algorithm called *Max Pressure* control. Their control algorithm selects the "optimal" phase based on the calculated pressure for each stage. A stage that has the maximum pressure is then selected as the next stage for the intersection. In their paper, the pressure term is determined by the queues on the in-coming links, queues on the out-going links and the saturation flow rates for those links. Details on the model can be found in Appendix A and in Varaiya (2013). They claimed that their control method can achieve a stable network state in terms of average queue length in the network. However, there are several shortcomings in their work that need remedy. The way they determine green time

split is too simple. By assigning most of the green time to the selected "optimal" phase will increase queue length on the approaches that are not permitted to pass in that phase, and hence increase travel time. This becomes quite significant especially in a homogeneously congested scenario in which each approach has almost the same demand. Another aspect is that their algorithm fails to consider upstream demand which is a very important component for designing an adaptive signal control algorithm, especially if coordination is considered. To address those problems, we further propose a control method called *Modified Max Pressure* control based on the aforementioned *Phase Selection Control* and *Max Pressure Control*, which considers pressures derived from both the queues at the current intersection and the upstream traffic demands. All the algorithms mentioned above are described in details in the following sections.

4.3.2.1 Low density control

When the traffic demand in the network is low, a low density control algorithm is used to control traffic. The algorithm is a fully vehicle-actuated control method. Traffic light will turn green for a certain approach if there are vehicles detected on the lanes of that approach (detectors are located at both ends of a certain link. See Figure 4.8). Green time extension is also based on traffic actuation, which means if there are vehicles coming continuously the green time will extend correspondingly. Green time is subjected to minimum and maximum green times.

Our low density control algorithm works as Figure 4.5:

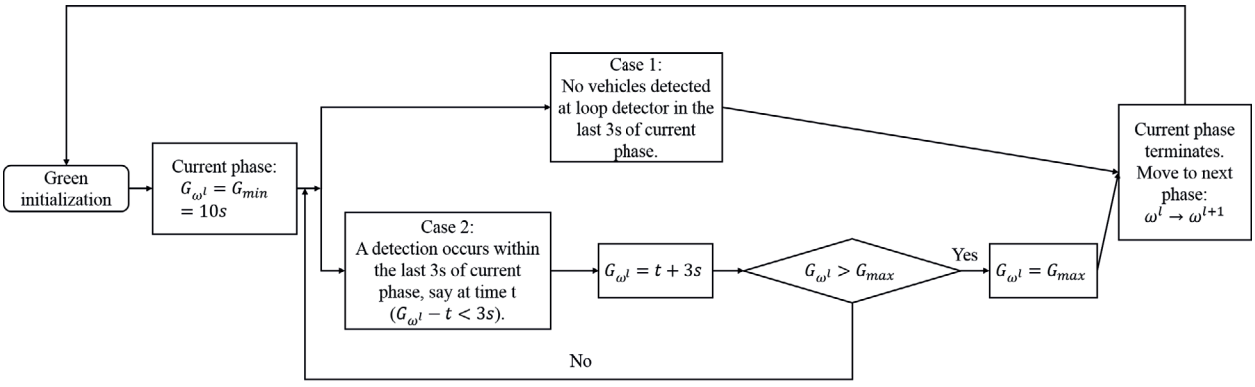


Figure 4.5: Low-density signal control algorithm.

4.3.2.2 High density control

When the traffic demand in the network is high, a high density control algorithm is used to control traffic. In this algorithm, phase split is set to be proportional to the incoming flow for each phase (More rigorously, flow ratio, i.e. flow/saturation flow, should be used. However, in our scenario saturation flow rates are the same for each phase.). Before every updating period for the traffic signal timing, the incoming flows for every phase are collected by the detectors located on the lanes. Minimum and maximum green times are still respected in this case.

Our high density control algorithm works as Figure 4.6:

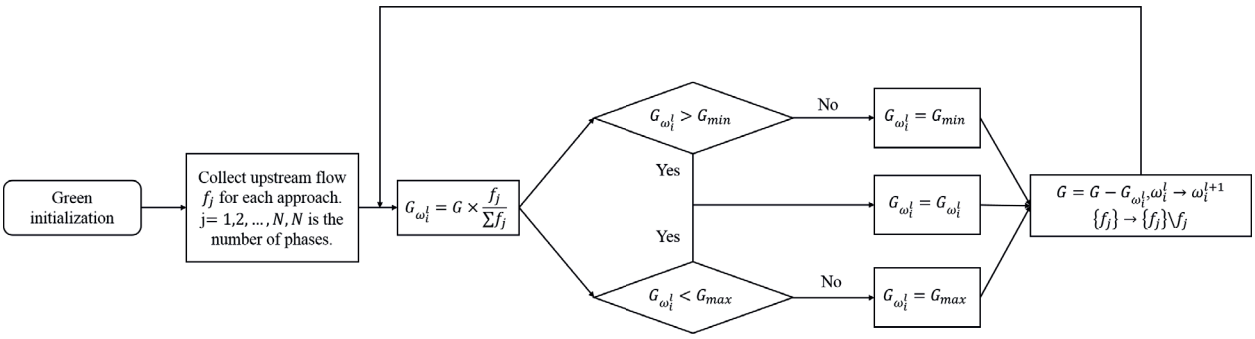


Figure 4.6: High-density signal control algorithm.

4.3.2.3 Phase selection control

The aforementioned control algorithms do not change the phase sequence. In some cases, the choice of the next permitted phase can have a great effect on the traffic. Therefore, it is important to choose a good phase sequence for the time-being other than simply setting it to be fixed. The phase selection control proposed in this study simultaneously determine the next permitted phase and the duration of that phase as following formulation states:

$$(\omega_i^l, G_{\omega_i^l}(t)) = \arg \max_{(\omega_i^l \in \Omega_i, t_g \in [G_{min}, G_{max}])} \{\frac{N_{t_g}^{\omega_i^l}}{t_g}\} \tag{4.4}$$

Note:

$N_{t_g}^{\omega_i^l}$: is the estimated number of vehicles that will arrive at intersection i within time t_g that are also permitted in phase ω_i^l.

4.3.2.4 Modified max pressure control

As mention previously, a *Modified Max Pressure Control* is proposed as follows:

$$(\omega_i^l, G_{\omega_i^l}(t)) = \arg \max_{(\omega_i^l \in \Omega_i, t_g \in [G_{min}, G_{max}])} \{\frac{\alpha \times N_{t_g}^{\omega_i^l} + \beta \times \gamma(\omega_i^l)}{t_g}\} \tag{4.5}$$

$$\begin{aligned} \gamma(\omega_i^l) &= \sum_{n,m} c(n,m) w(n,m) S(n,m) \\ &= \sum_{n,m;S(n,m)=1} c(n,m) w(n,m) \end{aligned} \tag{4.6}$$

$$w(n,m) = x(n,m) + \sum_{p \in Out_m} \gamma(m,p)[d(m,p) - x(m,p)] \tag{4.7}$$

Note:

$N_{t_g}^{\omega_i^l}$: is the estimated number of vehicles will arrive at intersection i within time t_g that are also permitted in phase ω_i^l.

α, β: are weights on upstream flow and queue pressure, respectively.

m, p: link m and link p.

$c(n,m)$: saturation flow of movement (n,m), vehicles per period.

$\gamma(\omega_i^l)$: the portion of vehicles passing intersection i in phase ω_i^l. The $\sum$ in Equation 4.6 sums up the number of passing vehicles in all the permitted movements (n,m) that belong to phase ω_i^l.

$d(m,p)$: the length of the corresponding lane on link m for the connection between link m and link p.

$x(m,p)$: the queue length on the corresponding lane on link m for the connection between link m and link p.

$w(n,m)$: the queue pressure for movement (n,m).

For comparison purpose, the original MP Control is included in the Appendix A. From the formulations for Phase Selection Control and Modified MP Control, one can see that the main difference between these two is the queue length term. Phase Selection Control

only considers upstream demand to determine the phase sequence and duration, while Modified MP Control considers both upstream demand and queue length.

The *Modified Max Pressure Control* algorithm is presented as Algorithm 4 in Appendix A.

4.3.3 A combined adaptive signal control and DTR

Now we have a proper dynamic routing algorithm as well as several signal control schemes (low-density control, high-density control, phase selection control, max pressure control or modified max pressure control). While planning their routes, travellers now not only have to consider travel time on each link but also the delay caused by signal control at intersections. With that being said, we adopted the non-adaptive hyper-path approach from Yang and Miller-Hooks (2004) to incorporate the signal control methods and the Bayesian link travel time updating scheme proposed in the previous context to explicitly take into account signal delays and link travel time dynamics:

$$\mu_i^h(t) = \min_{j \in \Gamma(i)} \left\{ \sum_{k=1}^{K} \left\{ \left[\phi_{ij}^{\omega_i^l}(t) + \tau_{ij}^k \left(t + \phi_{ij}^{\omega_i^l}(t) \right) + \lambda_j^i \left(t + \phi_{ij}^{\omega_i^l}(t) + \tau_{ij}^k \left(t + \phi_{ij}^{\omega_i^l}(t) \right) \right) \right] \cdot \rho_{ij}^k \left(t + \phi_{ij}^{\omega_i^l}(t) \right) \right\} \right\} \tag{4.8}$$

$\mu_i^h(t)$: is the temporary label (travel cost to the destination) of node i at time t, the upstream node is h.

$\lambda_j^i(t)$: is the current label (travel cost to the destination) of node j at time t, the upstream node is i.

$\tau_{ij}(t)$: is the link travel time of link (i, j) at time t.

$\phi_{ij}^{\omega_i^l}$: is the delay caused by the signal light at intersection i at time t, ω_i^l is the l^{th} phase of intersection i, the downstream node is j.

$\Gamma(i)$: is the set of all all adjacent nodes of node i.

With the adaptive signal control methods and periodically updated link travel times, the traffic routing algorithm become adaptive and dynamic. The adaptive algorithm this study uses is described in Algorithm 3 in Appendix A:

4.3.4 Computational issues

Consider the DTR algorithm alone, the computational complexity is $O(V^2 \cdot T)$ per vehicle per re-routing if implemented using the priority-queue data structure. If we adopt vehicle-based DTR implementation, i.e. every vehicle will perform its own re-routing calculation, the complexity will be $O(V^2 \cdot T \cdot N)$, where N is the number of vehicles running in the network. When a network gets very large, the number of nodes (V) and number of vehicles (N) could also get very large which can make the computation so slow that the results it provides are not suitable for real-time applications.

One possible approach to conquer this computational barrier is to use O-D based DTR algorithm instead of vehicle-based. O-D based DTR means we compute a stochastic shortest path using the algorithm mentioned in Algorithm 3 for each possible O-D pair every updating interval t_{update}. This updating interval t_{update} can be wisely chosen that it can reduce computation time, while can still yield an acceptable result. Here we make some approximations. We assume the shortest path solution is the same during t_{update} as long as it's the same O-D pair. We store the stochastic shortest path for every O-D pair in a table, and set the table to be active. After t_{update}, old tables will be discarded by simply setting them to be in-active. Whenever a vehicle has a need for re-routing, it can look up in the current active table using its origin and destination information so that it can obtain a route (which is a hyper-path) to guide itself to the destination. In this way, the computational complexity becomes $O(V^2 \cdot T \cdot V)$. For networks with very large number of vehicles (N) and small number of nodes (V), it can reduce the computation time substantially.

4.4 Simulation

In this section, the DTR algorithm and adaptive signal control methods are tested on a synthetic signalized street network. We run simulations with different settings of parameters to explicitly compare the performance of our DTR algorithm and control methods with the traditional routing and control methods. The metrics we used include average travel time, average speed, average queue length and so on. The goal is to find an "optimal" solution for various traffic conditions. More details on the metrics are discussed later in Section 4.5.

4.4.1 Simulation tools

The VANET provides not only real-time traffic information that can be used for traffic routing and signal control purposes, but also the possibility to launch CACC programs so that freeway capacity and safety can be enhanced (Van Arem et al., 2006; Shladover et al., 2012). The work in this chapter is based on the VANET setting. We use a simulation platform to test the algorithms and control methods proposed: a microscopic traffic simulation tool called SUMO is used to simulate the physical road network and the vehicles travelling in the network; a communication simulator called OmNet++ is used to simulate the communication network. These two simulators have interfaces that allow them to interact with each other. More information about the simulation platforms can be found here (Arellano and Mahgoub, 2013; Amoozadeh et al., 2015).

4.4.2 Simulation network

The network used is a 10×3 grid network which replicates a portion of a typical city road network. The middle East-West (EW) road and all the North-South (NS) cross streets are bi-directional. The other two EW roads are one-way roads. The EW roads have a speed limit of 40 mph or 17.78 m/s, while the cross roads have a speed limit of 25 mph or 11.11 m/s. The block size is 500 meters long. There are a total of 30 nodes and 47 links within the network. All the intersections are controlled by traffic lights. The geometry and layout of the network are shown in Figure 4.7. Each link in the network has double loop detectors located at both entry point and exit point of that link. Figure 4.8 shows the location of loop detectors. The detectors can capture the flow and speed of the road traffic.

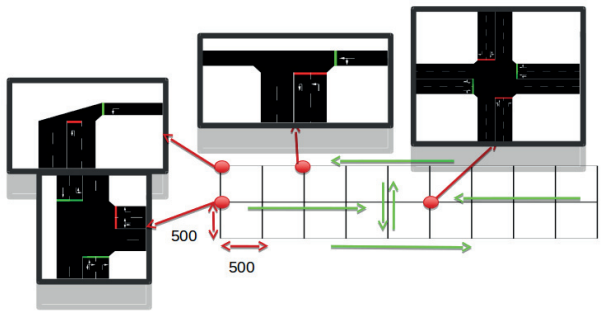


Figure 4.7: A synthetic 10×3 grid network used for simulation.

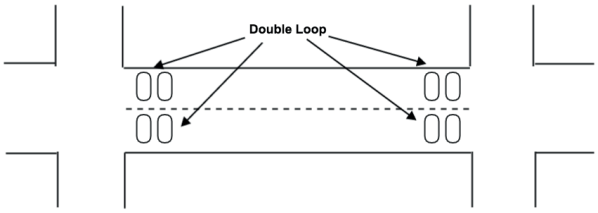


Figure 4.8: Each link has double loop detectors at both ends.

The default signal timings for the intersections are all fixed control but with different cycle lengths: the traffic lights on the middle main street have a default cycle length of 94 s; the traffic lights at the four corners of the network have a default cycle length of 34 s; the rest of the traffic lights have a default cycle length of 68 s. There is no signal coordination in the network. Through VANET, it assumes that each traveler has precise knowledge of current signal status and timing plans throughout the network. The detailed implementations of the information acquisition process for each traveler, however, is not the focus of this research study and therefore is not described here.

4.4.3 Input parameters

Vehicles in the network are generated randomly with a uniform distribution in a certain period of time (different traffic demands will have different loading time, which is given in Table 4.3). The ODs are also randomly generated and assigned to vehicles. All nodes can serve as an origin as well as a destination of a vehicle. Vehicles are generated at nodes. In order to see the performance of the proposed algorithms under different levels of traffic loads, three different levels of traffic demand are used to replicate the free flow, the mildly congested and the heavily congested scenarios. Vehicles that reach their destinations will not re-enter into the network. The simulation terminates when the last vehicle exits the network.

Before the actual run of simulation, a set of small scale pre-run simulations are needed in order to generate the historical set of link travel times which serves as initial inputs for the DTR algorithm. For different traffic demands, the pre-run simulations should use different demand correspondingly. With trial-and-error, we find that usually a number of 10 or more pre-run simulations are needed in order to generate such a historical set. An example of link travel time set is given in Table 4.2. Between time $t = 0$ and time

$t = T$, all links have stochastic link travel times. Each link may have a different number of possible link travel time realizations (in Table 4.2 all links have 3 different possible link travel times.).

Table 4.2: An example of historical set of link travel time.

		link 1			link 2			⋯	link N		
t=0	$\tau(s)$	10	6	8	2	3	4	⋯	4	5	6
	ρ	0.1	0.2	0.7	0.4	0.2	0.4	⋯	0.1	0.2	0.7
t=1	$\tau(s)$	5	6	7	2	3	5	⋯	5	6	7
	ρ	0.3	0.2	0.5	0.1	0.3	0.6	⋯	0.2	0.4	0.4
...	$\tau(s)$	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
	ρ	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
t=T	$\tau(s)$	10	15	20	5	7	9	⋯	8	9	10
	ρ	0.8	0.1	0.1	0.2	0.5	0.3	⋯	0.1	0.5	0.4

In real world applications, this table is maintained by an information center. The center takes charge of maintaining and updating this table from time to time. In the setting of this study, the updating frequency of the table is 1 second. Every time step, the center will collect the travel time realizations in the last second on all the links to obtain a new dataset, denoted as I_{new}. The dataset in the old table, denoted as I_{old}, is then updated to a new dataset ($I_{updated}$) using the method explained in Section 4.3.

$$b \cdot I_{old} + c \cdot I_{new} \rightarrow I_{updated} \tag{4.9}$$

Every time a vehicle needs to do a re-route, it sends request to the center to request the most up-to-date traffic information. Any new shortest path will be calculated based on that information. The simulation configurations are given in Table 4.3.

Vehicles are loaded onto the network randomly following a uniform distribution during time T_L. T_L is the vehicle loading time, which is given in Table 4.3. There are three levels of traffic demand respectively, which are 500 vehicles, 3000 vehicles and 6000 vehicles.

Table 4.3: Simulation parameters.

Link Travel Time Table Updating Interval	1s
Vehicle Re-routing Interval	Per intersection
Vehicle Loading Time (T_L)	3 min/5 min/10 min
Traffic Signal Updating Interval	120s

4.4.4 Simulation scenarios

We design different simulation scenarios to test our algorithms: different combinations of routing strategies and traffic signal control methods under different traffic demand levels.

Table 4.4: Different combinations of simulation scenarios.

Number of Vehicles in The Simulation		Traffic Routing Method		Traffic Signal Control
500	⊗	Dijkstra's Algorithm	⊗	Fix Timing Control
				Low Density Control
3000		Adaptive Routing		High Density Control
				Phase Selection Control
6000		Dynamic Traffic Routing		Modified Max Pressure Control
				Max Pressure Control

For the purpose of comparison, the fixed timing control is also implemented. There are three different types of intersections(see network in Section 4.4). Each has a different timing plan.

- Cycle length=94s:

Table 4.5: Duration of phases for type 1 intersections.

	Phase # 1	Phase # 2	Phase # 3	Phase # 4	Phase # 5	Phase # 6	Phase # 7	Phase # 8
Phase Duration (s)	31	5	6	5	31	5	6	5

- Cycle length=83s:

Table 4.6: Duration of phases for type 2 intersections.

	Phase # 1	Phase # 2	Phase # 3	Phase # 4	Phase # 5	Phase # 6
Phase Duration (s)	31	5	6	5	31	5

- Cycle length=68s:

Table 4.7: Duration of phases for type 3 intersections.

	Phase # 1	Phase # 2	Phase # 3	Phase # 4
Phase Duration (s)	31	3	31	3

For each simulation run, there is a maximum simulation time limit which is preset to 1 hour (3600 seconds). In most cases, all vehicles can reach their destinations before this time limit unless there is gridlock in which vehicles are stalled in the network. In rare cases where there are still vehicles in the network after the time limit, the simulation is forced to terminate. We will show in the following section an example of gridlock under a less effective signal control method.

4.5 Results and discussion

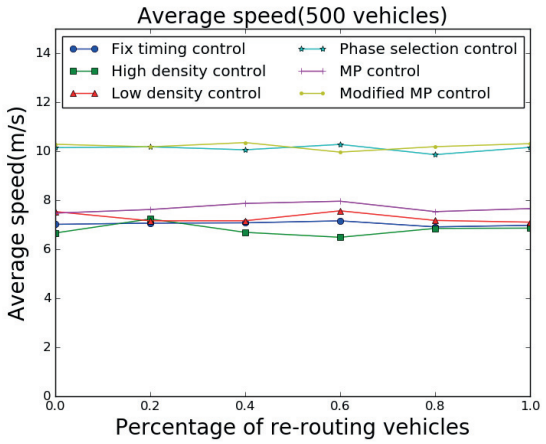
Results of different simulation scenarios (see Table 4.4) are analyzed here. To eliminate the effect of stochasticity in our problem due to stochastic link travel time, we run 10 times of simulations for each scenario and analyze the results based on the average of those ten simulation runs from different aspects to study the effect of that particular aspect. It consists of four sub-parts: effects of different traffic signal control methods, effects of different number of re-routing vehicles, effects of link travel time updating and effects on the macroscopic fundamental diagram. The main metrics are average travel time, average travel speed and average queue length in the network. These metrics are good indicators to evaluate the performances among different strategies. Strategies that have smaller travel time, higher travel speed and less queue are favored over the ones that have larger travel time, lower travel speed and more queue. The MFD analysis is to study how the joint

traffic routing and signal control algorithm proposed can affect the network throughput in an aggregated way.

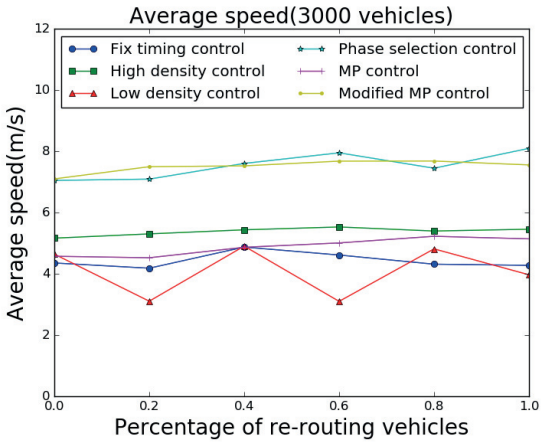
In order to see the performance of the proposed methods under some extreme conditions, an accident scenario is also tested and analyzed.

4.5.1 Effects of different traffic signal control methods

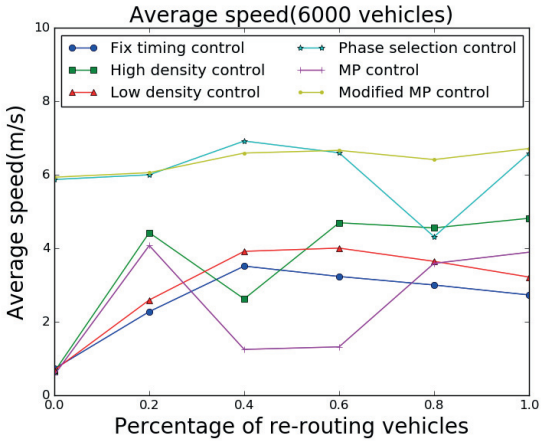
First, we study the performance of the four different control methods with DTR: Low Density Control, High Density Control, Phase Selection Control, and Modified Max Pressure Control. Comparisons are made among the four together with Fixed Timing Control and the original Max Pressure Control under the three different traffic demand levels. Various metrics are studied in the following sections.



(a) Case 1: 500 vehicles

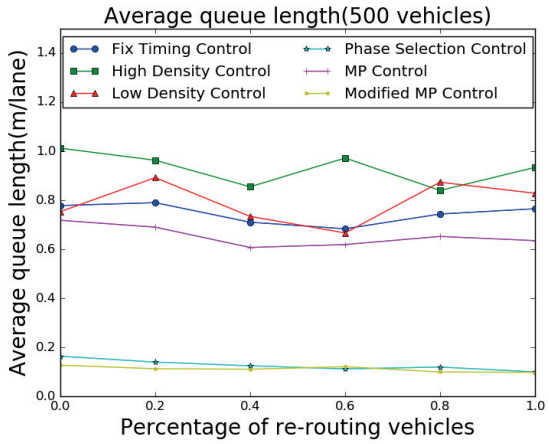


(b) Case 2: 3000 vehicles

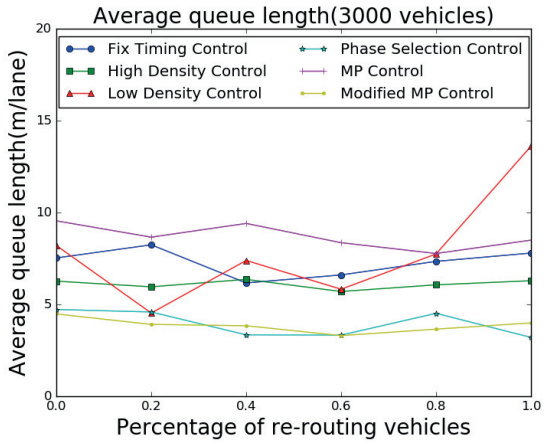


(c) Case 3: 6000 vehicles

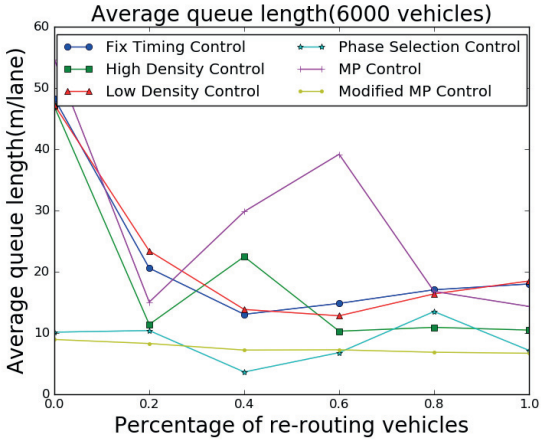
Figure 4.9: Average speed in the network with different number of vehicles.



(a) Case 1: 500 vehicles



(b) Case 2: 3000 vehicles



(c) Case 3: 6000 vehicles

Figure 4.10: Average queue length in the network with different number of vehicles.

Figure 4.9 and Figure 4.10 show the results for different control methods under different percentage of travelers who seek re-routing. We can see that Phase Selection Control and Modified MP Control always have the smallest average queue length and largest average speed in all three scenarios with different traffic demands. The performances of Phase Selection Control and Modified MP Control are very similar to each other in all cases. This indicates that in this network, pressure from upstream demand has a dominant impact on designing the "optimal" signal timing, while the queue at an intersection is a less important factor. It is also interesting to note that the performance of Phase Selection and Modified Max Pressure controls are in most cases not sensitive to the number of travelers who engage in re-routing, while this is not true for other controls when traffic demand

in the network is heavy. In the latter cases, more re-routing travelers does not necessarily lead to better network performance, an observation consistent with earlier literature on the adoption of Advanced Traveler Information Systems.

In the low demand scenario (500 vehicles), the rest four control methods perform very similarly, with the original MP Control working slightly better than the other three. This is because in such low density, there is very little probability to form a long queue at intersections. Incoming flows can be easily accommodated by even a fixed timing plan in every cycle. The benefit of allowing changing of phase sequence and duration, which is a key point of Phase Selection Control and Modified MP Control, becomes important in this case. That's why these two methods can outperform the rest.

In the mid-range demand scenario, the High Demand Control tends to work better than the Fix Timing Control, the Low Density Control and the original MP Control. In this range of demand, queues are formed from time to time. It is beneficial to give more preference to the movements that have greater demands to dissipate any potential queues by assigning longer green times to the corresponding phases. With the fact that High Density Control works better than the original MP Control, it confirms that upstream demand has a greater impact on designing traffic signal timing than the queue length in this demand level. Again, the Phase Selection and Modified MP control still outperform the other four control methods.

When the network is highly congested, like the case with 6000 vehicles, the performances of the rest four control methods (not including Phase Selection Control and the Modified MP Control) become complex. In this scenario, queues will be formed frequently at intersections, and normally can't be dissipated completely in one cycle when the cycle length is fixed, which is the case of all these four control methods. Performance will be random, and highly depend on the initial input which is stochastic in our simulation.

By examining the three different traffic demand scenarios, one important conclusion is that upstream demand plays a very important role in designing a good signal control method. Per signalized intersection point of view, the phase sequence and phase duration can be equally important when demand gets high. Failing to consider either one will result in degraded performance.

4.5.2 Effects of different number of re-routing vehicles

The x-axis in Figure 4.9 and Figure 4.10 is the percentage of re-routing vehicles in the network. Different number of re-routing vehicles can have quite a different effect on the overall performance even when the signal control remains the same. This is intuitive to understand when the percentages are 0 and 100%, representing all vehicles to be normal vehicles and all vehicles are re-routing vehicles, respectively.

In the low density case (The case of 500 vehicles), the average speed in the network is almost the same when the number of re-routing vehicles changes from 0 to 100%. However, the average queue length is slightly decreasing when the number of re-routing vehicles in the network becomes larger. When densities get higher (the cases of 3000 and 6000 vehicles), the relation is no longer monotonic. Average speed is highest, and average queue length is smallest when the percentage of re-routing vehicles is somewhere between 0 and 100%. This shows that in congested scenarios, it is not always true that the more re-routing vehicles the better. In those cases, the network is already packed with vehicles. It is hard for vehicles to move around. Any attempt to change route will incur further congestion burden to the network, which will possibly in turn exacerbate the current congestion level.

Furthermore, if all travellers have access to the same piece of information, after re-routing the result will be that the least congested links will be chosen as their preferred links. Hence traffic will flood to those links making them become congested. If another re-routing is performed, the newly allocated traffic will further shift to some other links that are not congested. So the traffic will periodically oscillate between different links, which will always leave some links congested and some links not congested. The consequence of this is underusing the capacity of the network. This is very similar to the concept of Herd Behavior in behavioral finance (Scharfstein and Stein, 1990; Banerjee, 1992). A sweet spot for those cases in terms of average speed and average queue length lies between 0 and 100% which depends on the demand, the signal control methods and many other latent factors yet to be identified. To overcome the negative effect of all-people re-routing, we can limit the accessibility to information so that only a portion of travellers can have the most up-to-date traffic information and be able to do re-routing. The proportion of re-routing travellers is an optimizable variable that can be closely related to traffic conditions and

network geometry.

4.5.3 Effects of link travel time updates

As mentioned already in Section 4.1, the major difference between our work and the existing work is that the link travel time is constantly updated with the most current link travel time. Here, we study the performance difference between our method and the traditional one which does not have the functionality to update the link travel time. In Figure 4.11, we have two different routing methods: Adaptive routing (AR) and Dynamic traffic routing (DTR). The former is a traditional stochastic routing policy mentioned in Section 4.1. It does not have the ability to update link travel time dynamically. The latter is the routing method we proposed. It allows dynamic updating of link travel time.

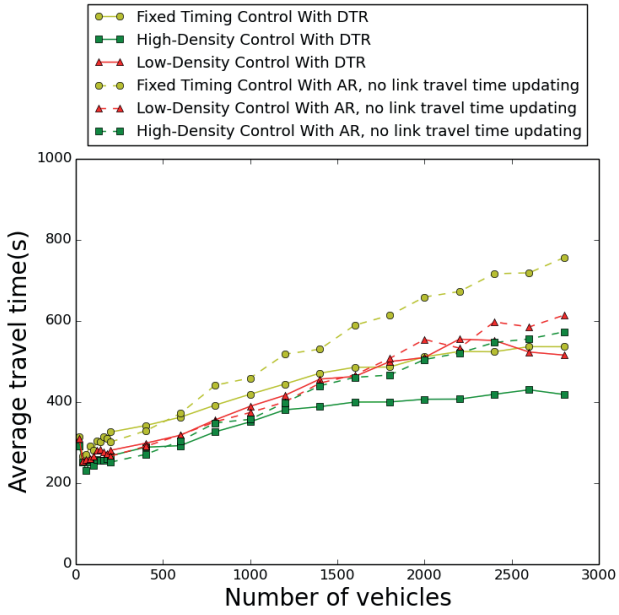


Figure 4.11: Average travel time with and without link travel time updating.

The dash lines are for AR, while the solid lines are for DTR. We compare these two under three different signal control methods: fixed timing control, low-density control and high-density control (No Phase Selection Control or Modified Max Pressure Control Method was tested in this case). Network traffic loads from 200 up to 2800 vehicles with an increment of 200 are tested. From the figure, we can see that all dash lines are above the corresponding solid lines (hence longer average travel time) which means the DTR algorithm consistently works better than traditional AR algorithm. As traffic load becomes

heavier, this benefit becomes larger.

4.5.4 Effects on the macroscopic fundamental diagram

Another metric to evaluate the overall performance of a network is the throughput. In this subsection, we measure the throughput (in veh/h) of the network under different signal control methods. The results are shown in Figure 4.12. The x-axis is the number of running vehicles in the network, meanwhile the y-axis is the corresponding throughput (a.k.a. network exit flow) at that moment. we use different colors to indicate the temporal evolution of the simulation process: starting from the blue end and finishing in the red end. We combine simulation results from different demand levels into one figure: 500, 3000 and 6000 vehicles. In this way, we can construct an MFD as complete as possible to include both the free flow part and congested part. The 1-second data are quite noisy and moving average (with 50 steps) is applied to filter the noise. From the figure we can see that the relationship between throughput and number of running vehicles can be varying as the signal control method changes. In this experiment, 50% vehicles are re-routing vehicles and traffic signal control methods are: *Phase Selection Control*, *Max Pressure Control* and *Modified Max Pressure Control*. The small circles are for 500-vehicle case; the "+" signs are for the 3000-vehicle case and the small squares are for 6000-vehicle case. The colors are in a time sequence that corresponds to the simulation time. It starts from the bluish color and ends with the reddish color.

All three sub-figures have a portion of data points starting from the lower part of the figure (blue end), and are deviating from the "MFD curve". This portion belongs to the initial vehicle loading part while subjects to a very large uncertainty as the demand is generated randomly. This variation can be eliminated if we leave out the loading part (see Figure 4.13). From the MFD plot, we can see that with *Phase Selection Control* and *Modified Max Pressure Control*, the MFD is closer to a trapezoidal shape than the *Max Pressure Control* with a fairly large density region that sustains high exit flow. The *Max Pressure Control* case is more noisy. Under this control method, the network is gridlocked so the simulation terminates after the maximum simulation time is reached. The flow rate in the network is close to 0 while the density of the network is some positive value. This is represented as the rising part starting at a density of around 40 veh/km and a flow of about 0 veh/h

(normal MFD should originate from (0,0)). The former two control methods also have a higher maximum flow which is the highest point in the MFD, which indicates that the network has a higher throughput under *Phase Selection Control* and *Modified Max Pressure Control*. Between these two control methods, there is no significant difference from the MFD point of view.

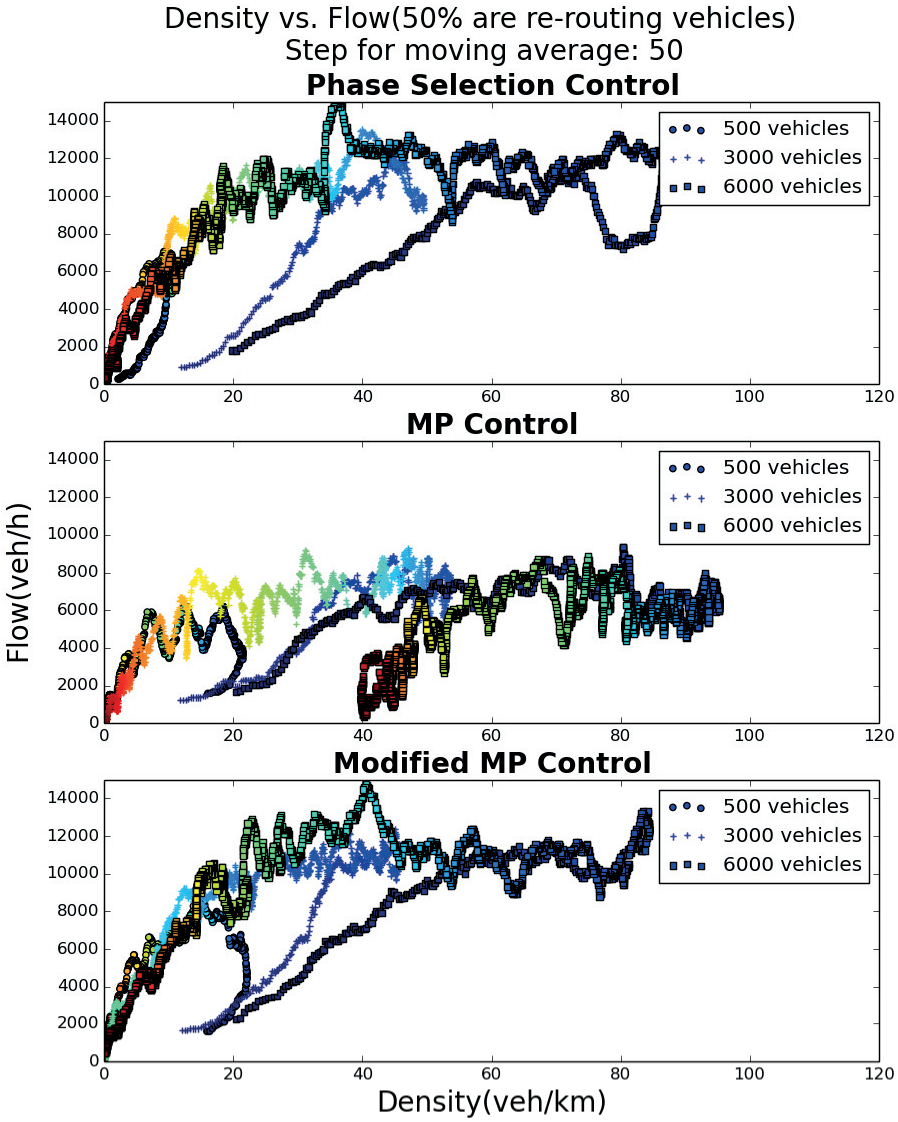


Figure 4.12: Throughput vs. number of vehicles in the network.

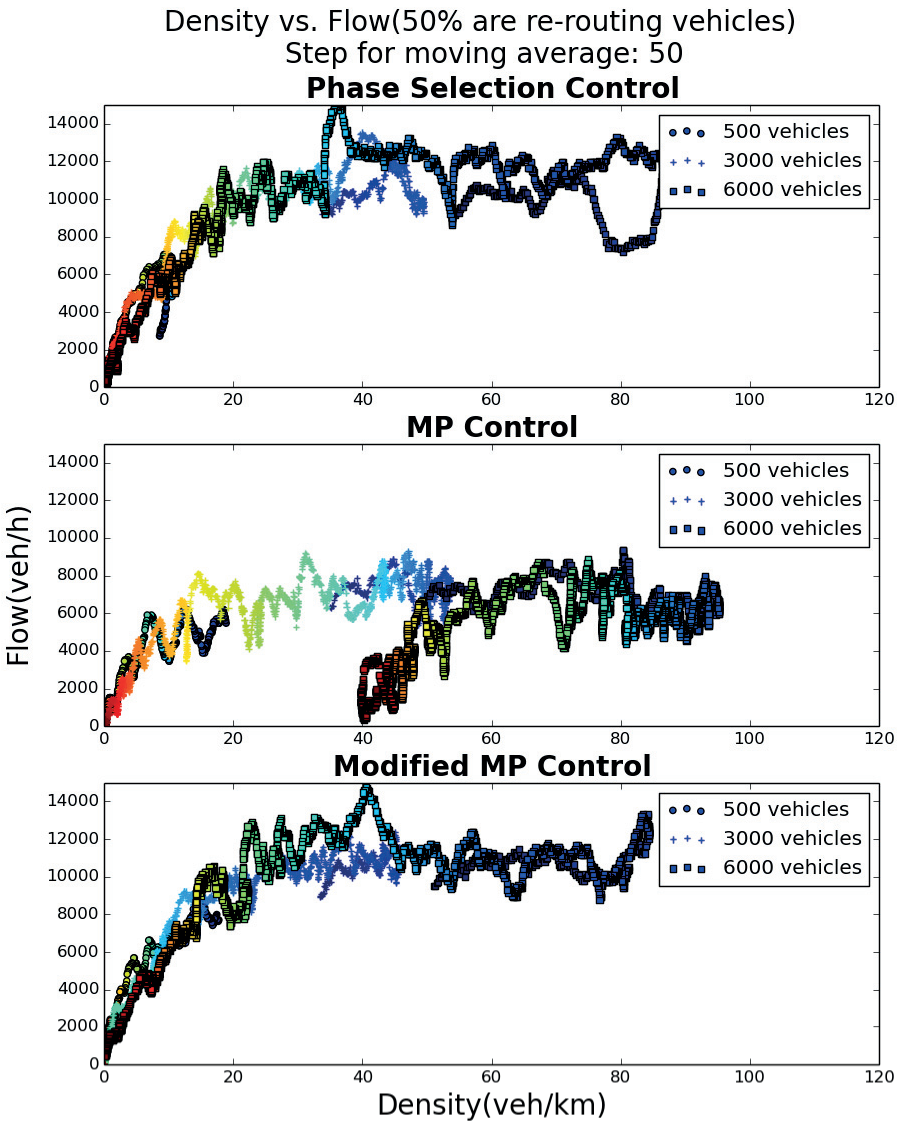


Figure 4.13: Throughput vs. number of vehicles in the network (No initial loading).

4.5.5 Traffic accident scenario

The previous simulations are all under normal traffic conditions, which showed that the routing algorithm and traffic signal light control algorithm we proposed work satisfactorily. We still want to see how the proposed algorithms will perform under unusual scenarios. Here, a traffic accident scenario is designed in order to test that. We manually pick three links on the main arterial to be the locations where the accidents occur (see Figure 4.14). This is to mimic a typical traffic accident scenario in down-town road networks.

The links with accidents are assumed to be blocked during the time of accidents. Vehicles on those links are forced to stop behind the accident location. They are allowed to resume moving only after the accidents clear. The detailed information of duration and locations of accidents are given in Table 4.8.

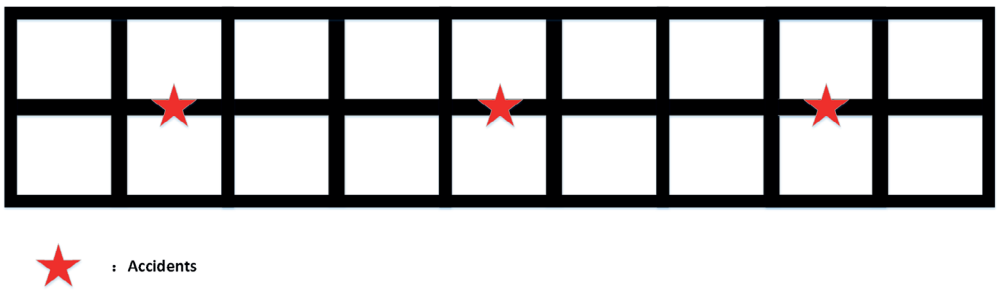


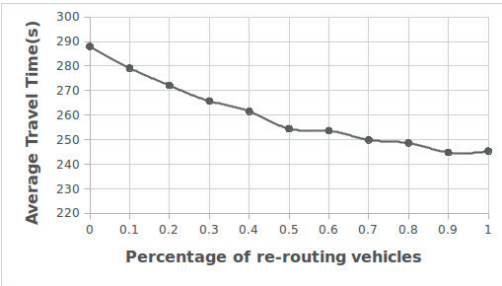
Figure 4.14: A traffic accident scenario: three links with accidents.

Table 4.8: Links with accidents.

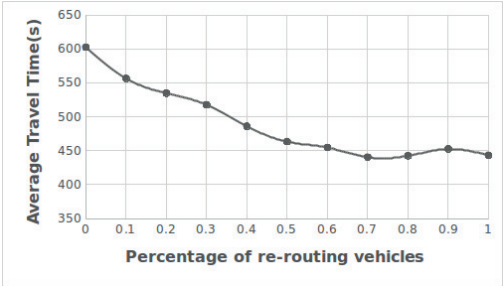
Link	Begin Time (s)	End Time (s)	Position	Lane Index
12to15	40	200	250	1
19to18	60	250	150	0
15to16	200	450	300	1

We tested three different scenarios: 500 vehicles, 3000 vehicles and 6000 vehicles. The traffic light control method is *Phase Selection Control*. The results are shown in Figure 4.15. From the figure, we can see that in the 500-vehicle and 3000-vehicle cases, the average travel time decreases as the percentage of re-routing vehicles increases; however, in the 6000-vehicle case the average travel time shows a slight "U" shape pattern where the minimum is reached when re-routing vehicle percentage is between 0.6$\sim$0.8. The observation of these simulations also confirms the explanation in the previous sections. When traffic demand is relatively low (500 and 3000 vehicles), most of the links are not congested yet. When some of the links are blocked or partially blocked by accidents, vehicles can reduce their travel time by taking some detours prior their entering into the blocked links. The network is not highly congested so these re-routing vehicles won't cause new congestions when they re-route and the network performs better when more vehicles re-route to avoid the accident locations. However, when the network is already heavily congested, if all ve-

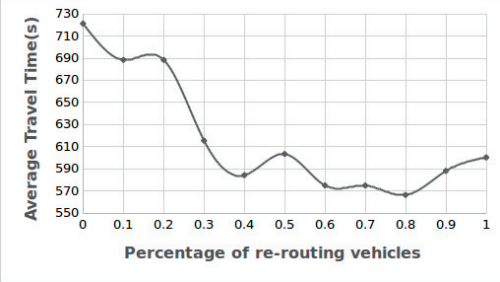
hicles are able to re-route, the result will not necessarily be the best since their routing decisions are not aimed at reducing network-wide travel time, and in some cases even becomes worse. The simulation suggests that in the 6000-vehicle case, a good choice will be allowing $60\% \sim 80\%$ vehicles to re-route. This percentage range, however, is likely to vary by network, traffic demand level, and driver population.



(a) Case 1: 500 vehicles



(b) Case 2: 3000 vehicles



(c) Case 3: 6000 vehicles

Figure 4.15: Average travel time under accident scenario.

4.6 Conclusions and future work

In this chapter, we proposed a joint adaptive routing and traffic signal control algorithm to improve traffic operations in a VANET enabled traffic environment. Our dynamic routing algorithm (DTR) is an extension of the LET algorithm of Miller-Hooks and Mahmassani (2000) with periodic updates of link travel times. The proposed algorithm also takes into account the delay caused by real-time traffic signal operations. Besides several traditional traffic signal control strategies, namely fixed-timing, vehicle actuated control (known as low density control in this chapter) and adaptive Webster's (known as high density control in this chapter), we also proposed two new traffic signal control strategies, the Phase Selection Control and the Modified Max Pressure Control, to take into account the effects of both incoming demand and current queues on traffic signal operations.

The most difficult part of tackling the joint routing and signal control problem is modeling the interaction between these two components. The interaction is implicit, and thus hard to model analytically. Also, the routing problem is user-based and time-dependent. Hence, the computation cost for running simulation is not negligible as route needs to be re-calculated per traveller and per time interval which can be as small as a second depends on the resolution desired. This puts a significant burden on computation. In order to test the effects of different parameters on the performance of the proposed algorithms, a sizable number of simulations with various parameters need to be run. This is another computationally demanding task. So it needs a computationally efficient algorithm to speed up the simulations. Furthermore, some approximation techniques are needed to shorten the simulation time even more. These tasks are not trivial both from algorithmic point of view and implementation point of view.

Our simulation results shows that the DTR algorithm works well under higher demand scenarios together with the adaptive traffic signal control methods proposed in this study. Enabling vehicle re-routing in the network can reduce the average travel time as well as reduce the average queue length at the intersection. The optimal re-routing ratio lies between 0-1.0, which our simulation tells us for 6000 vehicles this number is around 0.5. This number will likely vary over networks, traffic demand, and driver population. With the dynamic travel time updating model proposed in this chapter, the re-routing algorithm can further reduce the average travel time in the network by taking advantage of the most current link travel time information.

The different signal control methods proposed and tested in this study under different scenarios tells us that the Phase Selection Control and the Modified Max Pressure Control work better than the rest four control methods (including the original Max Pressure Control and fixed timing control). They tend to respond to traffic and accommodate traffic better than the rest. Average speed is higher and average queue length is shorter when these two control methods are applied. Among all the six control methods (including the original Max Pressure Control), the original Max Pressure Control performs the worst as its logic to find the optimal phase and its corresponding duration is not well designed. With the Phase Selection Control and the Modified Max Pressure Control, the MFD of the

network is closer to a trapezoidal shape compared to that of the Original Max Pressure Control. For the latter, its MFD is more chaotic and fluctuating. The maximum flow rate is also lowered compared to the former two.

The joint dynamic traffic routing and adaptive signal control approach is tested against a traffic accident scenario. With links blocked or partially blocked by accidents, the dynamic traffic routing (DTA) can efficiently re-route traffic to other uncongested links to avoid long delay. According to the simulation results, the average travel time in the network can be reduced by $17\% \sim 27\%$ when the percentage of re-routing vehicles is chosen properly. The optimal percentage of re-routing vehicles in this case relies heavily on the traffic demand level. In uncongested or mildly congested scenarios, the more re-routing vehicles in the network, the less the average travel time. In a highly congested network, however, the optimal number of re-routing vehicles lies somewhere between $0\% \sim 100\%$, which again depends on the traffic demand, driver population, network geometry and so on.

A possible extension of this work would be to consider global signal optimization. The signal control methods proposed in this work are all distributed ones. The phase selection method, for example, seeks local optimality but not global optimality. Coordination could be another extension to the current work. These new research directions will be even more challenging than the current joint routing and distributed signal control problem that we have dealt with in this research study and are worthy of serious investigation.

Chapter 5

Deadlock Avoidance In Traffic Routing and Assignment

5.1 Introduction

In traffic routing, most of the existing literatures focus primarily on reducing experienced or expected total travel time (Bellman, 1958; Fleischmann et al., 2004; Kim et al., 2005; Gueziec, 2008; Gendreau et al., 2015; Chai et al., 2017). However, travel time alone sometimes does not suffice as a measurement to evaluate the performance of a transportation system. In some cases, the objective to minimize the travel time for travelers will result in the other way around: deadlock in the network (in some literature, it is interchangeably called gridlock). When a deadlock occurs, no traffic can move forward until a manual intervention is taken to break the deadlock. Travelers within a deadlock will experience extremely high travel time. Beyond that, urban deadlock has other undesirable consequences: increase driving fatigue, increase accidents under stressful conditions and increase pollution from vehicles (Semiz, 2016). Since deadlock has a great negative impact on the performance of a transportation system and the travelers in the system, a proper mechanism should be designed and studied to reduce the possibility of deadlock occurrence.

Deadlock avoidance is a long-established research topic in network communication community. There are many research work in the past studying the deadlock free packets routing strategies. The various strategies implement different methods or use different

technologies to impose certain routing restriction on traffic to ensure a deadlock free routing result. In network communication and packets routing, deadlock can postpone packet delivery indefinitely. A set of packets could been blocked for ever in a deadlock. One approach to resolve the deadlock situation is to use preemption of packets in such cases by either discarding or re-routing the preempted packets (Ni and McKinley, 1993). However, a more realistic and commonly used technique is to avoid deadlock from happening in the first place by carefully designing the routing rules. Such routing algorithms order the network resources and let the packets use those resources in strictly monotonic orders so that the circular waiting can be avoided (Ni and McKinley, 1993). There are two main categories of deadlock free routing techniques: one utilizes virtual channels to facilitate deadlock free routing, the other one imposes certain restriction on certain turning movements to ensure deadlock free routing, a.k.a the turn model (Glass and Ni, 1992). Virtual channels are the abstraction that share the same physical channel (Chiu, 2000). Dally and Seitz (1988) used virtual channels in their work to design a deadlock free non-adaptive routing algorithm. Following that, virtual channel techniques have been used by many other researchers to design partially and fully adaptive routing algorithms that ensure deadlock free routing (Duato, 1993; Su and Shin, 1993; Dally and Aoki, 1993). Though the virtual channel technique is powerful and effective, adding virtual channels to the network can be costly and could bring significant overhead. It needs to add buffer space and design complex control scheme for routers which could affect the communication performance of the network and the reliability of the routers (Glass and Ni, 1992; Chien, 1993; Chiu, 2000). Furthermore, there are applications where virtual channels are not physically possible to implement. For instance, routing in transportation network can not rely on virtual channel based routing algorithms as links in the transportation network do not permit adding any "virtual" lanes. To deal with the aforementioned problems, Glass and Ni (1992) proposed a deadlock-free routing algorithm that did not use virtual channels. Their model prohibits the minimum number of turns that break all of the cycles to avoid deadlock and achieve partially adaptive routing. Following their work, there have been many more improved works done by other researchers (Chiu, 2000; Hu and Marculescu, 2004; Li et al., 2006; Lotfi-Kamran et al., 2010). It has been applied to dealing with both

partial and full adaptive routing. One of the merits of turn models is that it only requires control at specific router nodes. Hence, the design and implementation of such a model is easy and causes less overhead to the network. It also fits the needs of routing in transportation networks perfectly as specific routing strategies can be implemented by controlling traffic via traffic lights to either permit or forbid certain turning movements.

In the context of transportation research, there are also numerous studies tackling the problem of deadlock in traffic network. Mendes et al. (2012) studied how a traffic jam spreads in complex networks when demands increase significantly between certain origin and destination nodes using two different traffic models. Their models can be used to reveal the links most vulnerable to deadlock and hence provide solutions to network design. It is generally recognized that queue spillback is one of the major cause of deadlock in the transportation network (Daganzo, 1995, 1998). Under high traffic demand, the influx at upstream of a road becomes larger than the exit flow at downstream. Queue builds up quickly if the inflow keeps entering into the congested road until the point that the built-up queue reaches back to the upstream point. At this point, congestion propagates to the upstream road and starts to spread out in the network if traffic demand keeps at a sufficiently high level, and eventually deadlock could occur. Besides the microscopic explanation of the occurrence of deadlock, Daganzo (2007) proposed a macroscopic model to describe urban gridlock from network level. The author in the paper described an adaptive control approach that monitors and controls aggregated vehicular accumulations to mitigate and alleviate gridlock in urban areas. However, the macroscopic model fails to characterize the dynamics of the deadlock system. The existence of exit flows and its recovery period in the dynamic traffic assignment context are not well answered. Mahmassani et al. (2013) explored some limiting properties of network-wide traffic flow conditions in a large-scale urban street network in the context of dynamic traffic assignment. Their study provides some insights on the characteristics and dynamics of gridlock in urban network with regards to different traffic conditions and patterns. In the study of traffic assignment, both static and dynamic, there is no guarantee that the traffic assigned to the network is free of deadlock. According to Geroliminis et al. (2007), after the occurrence of a deadlock, it is extremely difficult if not impossible for the transportation system to re-

cover back to better states even if the traffic demand decreases significantly. This indicates that it is more preferable to avoid deadlock from happening by preventing road network from entering the states of high traffic accumulation than to mitigate or resolve deadlock after deadlock occurred. Osorio et al. (2015) proposed a simulation-based signal control method to reduce the probability of gridlock. There are other similar approaches to address the problem of deadlock (Zhang et al., 2010; Claes and Holvoet, 2011; Lämmer and Treiber, 2012). However, there are two major things that most of the existing works fail to consider: 1) deadlock avoidance/prevention is not studied under equilibrium traffic flow; 2) the efficiency of the deadlock avoidance/prevention strategies. In this chapter, we propose a novel DLA model that explicitly addresses these two aspects of deadlock avoidance problem in traffic assignment.

The rest of this chapter is organized as follows: Section 5.2 states the problem and the notations used in the chapter. An brief introduction to *"Odd-Even routing"* method in packets routing and its similarity to traffic routing problem are presented in Section 5.3. Section 5.4 defines the deadlock potential studied in this chapter. Following that, the deadlock avoidance (DLA) model with DUE constraints is formulated in Section 5.5. Section 5.6 talks about approximation of link travel time, route travel time and user's route choice complementarity. Section 5.7 describes the scheme used to discretization for the DLA model. Numerical studies with different control strategies are presented in Section 5.8. Section 5.9 concludes the chapter with some discussions on future research.

5.2 Problem statement

A network with multiple $O-D$ pairs is denoted as $G(N,E)$. The time horizon for the problem setting is T. It should be noted that the time horizon T in this problem is large enough so that all vehicles can exit the network at the end of the time horizon. Dummy origins $o' \in O$ and dummy destinations $s' \in S$ are added to the original network to form an extended network. Between each pair of (o,o'), there are two dummy links added, i.e. (o,o') and (o',o). The same applies to destination node pairs (s,s'). The purpose of adding two dummy links are twofold: 1) in the DLA model specified in Section 5.5, there is a penalty term defined at each node $(\phi^r_{(i,j;h)})$ that requires the information of incoming

node (h) from upstream. Without dummy link (o, o') or (s', s), the model at origin nodes and destination nodes will become ill-defined; 2) as mentioned in Ma et al. (2014) and Yu et al. (2018), the user's route choice model requires that travel time from each node to destination node to be well-defined. Without dummy link (s', s), the travel time from a certain dummy destination node s' to some other dummy destination nodes will become infinity as there is no outgoing link from s' back to s. This leads to the user's route choice model to be ill-defined at dummy destination nodes. The notations used in this chapter are listed in Table 5.1.

Table 5.1: Notations used in the chapter.

Type	Notation	Definition
Sets	N	set of nodes in the network
	L	set of links in the network
	L_o	set of dummy origin links
	L_s	set of dummy destination links
	L_e	set of links in the expanded network, i.e. $L_e = L \cup L_o \cup L_s$
	S	set of dummy destination nodes
	Ξ_t	set of transitive closures in the network at time t.
	ξ_t^i	i^{th} transitive closure in the network at time t.
	$L_{\xi_t^i}$	set of links in i^{th} transitive closure at time t.
Constants	$\tau^0_{(i,j)}$	free flow travel time of link (i, j)
	$\tau^\omega_{(i,j)}$	shockwave travel time of link (i, j)
	T	total time horizon
	$\overline{Q}_{(i,j)}$	queue capacity of link (i, j)
	$\overline{C}_{(i,j)}$	homogeneous flow capacity of link (i, j)
	$H^{(o',s')}$	traffic demand between origin-destination pair (o', s')
	$\phi^r_{(i,j;h)}$	penalty of movement $h \to i \to j$ at node i for group r travelers
	α^r	percentage of group r travelers in the entire population.
	TTT	total travel time.

	$A(t)$	adjacency matrix of the network(considering queue) at time t. Only links with queue that are higher than threshold are included in the adjacency matrix.
Variables	$\overline{A}(t)$	adjacency matrix considering queues.
	$A^*(t)$	transitive closure computation matrix
	$P_{ij}(t)$	possibility of a link (i,j) get congested
	$p^{s',r}_{(i,j;h)}(t)$	inflow rate for group r of link (i,j) from node h towards destination s' at time t.
	$p_{(i,j)}(t)$	inflow rate of link (i,j) at time t.
	$v^{s',r}_{(i,j)}(t)$	exit flow rate for group r of link (i,j) towards destination s' at time t.
	$v_{(i,j)}(t)$	exit flow rate of link (i,j) at time t.
	$q^{u,s',r}_{(i,j)}(t)$	upstream queue for group r of link (i,j) towards destination s' at time t.
	$q^{u}_{(i,j)}(t)$	upstream queue of link (i,j) at time t.
	$q^{d,s',r}_{(i,j)}(t)$	downstream queue for group r of link (i,j) towards destination s' at time t.
	$q^{d}_{(i,j)}(t)$	downstream queue of link (i,j) at time t.
	$\pi^{s',r}_{(i,j)}(t)$	minimum travel time at time t from node j, with incoming node as i, to destination s' for group traveler r.
	$\tau_{(i,j)}(t)$	actual link travel time of link (i,j) at time t.
	$\theta^{s',r}_{(i,j)}(t)$	approximation coefficient of actual travel time from node i to destination s' via link (i,j) at time t for group traveler r.

When traffic demand increases or there are accidents happening on some links, queues may build up within the network. If the discharging flow rate of a queue is always less than the incoming flow rate, the queue will propagate towards upstream and eventually spill back to upstream links. In cases where spill-back happens, traffic on upstream links can no longer travel to the downstream links that are fully occupied by queues. Note

in this study, only vehicular traffic is considered. Therefore, "vehicle" and "traffic" are used interchangeably in the remaining of this chapter. If the queue spill-back propagation spreads further upstream in the network, there is a probability that a cycle, i.e. deadlock, will form (See Figure 5.1). When a deadlock is formed, the vehicles within the deadlock can not proceed any further. Without proper manual intervention, the deadlock can not resolve by itself. As mentioned in Section 5.1, deadlock can significantly deteriorate the

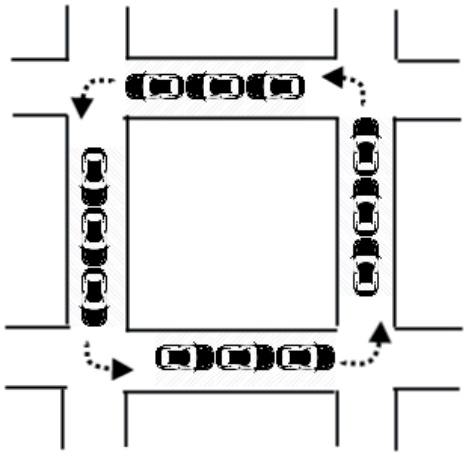


Figure 5.1: Deadlock in the network.

performance of a transportation system. To avoid deadlock from happening, in this study, we want to "guide" some vehicles in the network to some special routes so that the network can have a minimal possibility of getting deadlocked. To accomplish this, two different groups of vehicles are introduced in the network: one is normal vehicle group, the other is DeadLock-Avoidance-routing vehicle group (it will be referred as DLA-routing vehicle hereafter in this chapter). The DLA-routing vehicles are vehicles that comply to some certain routing restrictions, which are explained in details in Section 5.3. For instance, DLA-routing vehicles may not allow to turn left at some certain intersections; meanwhile, normal vehicles do not have such restrictions. In the model we use r to indicate different group of vehicles: $r = 0$ for normal vehicles and $r = 1$ for DLA-routing vehicles. In the extreme case where all vehicles in the network are DLA-routing vehicles, then the network is guaranteed to be deadlock free. Details of proof are provided in Theorem 1 in Section 5.3. However, eliminating deadlock does not come free. There are certain overhead

introduced by the "Odd-Even" routing rule specified in Section 5.3. By strictly following the "Odd-Even" routing rule, most of the vehicles are not traveling via the shortest route to their destinations. Therefore, travel time is inevitably increased for some travelers. On the other hand, the extra detours made by the DLA-routing vehicles also contribute to the increased VMT in the network, and hence more GHG and other pollutants into the environment. One of the objectives in the research is to find the “sweet point” between deadlock free and less travel time by carefully selecting the ratio between normal vehicles and DLA-routing vehicles in the network.

In this study all travelers are assumed to be rational. This assumption implies that travelers always tend to choose the best possible options. Under this assumption, Dynamic User Equilibrium (DUE) holds within the problem studied in this chapter. The work in this chapter tries to solve the following problems in order to obtain the "optimal" outcome for the network.

1. The ratio between normal vehicles and DLA-routing vehicles in the network.
2. The departure time of vehicles from their origin nodes.
3. The optimal route choice for each type of vehicles in the network to their destination.

5.3 Odd-Even routing

Before we delve into the optimal deadlock free strategy under DUE, we first look at a routing technique studied in network communication that guarantees deadlock free in the network. Chiu (2000) proposed an adaptive routing method called “odd-even” routing (OE routing for short) designing adaptive wormhole routing algorithms for mesh networks without virtual channel. In their model the locations where some turns can be taken are restricted to explicitly avoid deadlock from happening in the network.

In a mesh network, columns are numbered with integer numbers starting from 0. A column is called odd column if it is numbered with an odd index. The same for even column. In Chiu (2000), their adaptive OE routing model is governed by two rules which basically specify how packets can route at nodes in the network. There is a comprehensive

proof in Chiu (2000) to show that deadlock is guaranteed not to exist in the mesh network if every packet follows the routing rules specified in the paper.

The transportation network is similar to the mesh network studied in the network communication, especially for road networks in most downtown areas where most of the roads are in grid-net patterns. The *Odd-Even Routing* method proposed in Chiu (2000) for 2-D mesh network can also be applied to the transportation network with some minor adaptations. Vehicles in a transportation network are like packets in a communication mesh network. Traffic signal lights at intersections that control the movement of traffic are analogous to the routers at each node in communication network that control the movements of packets. From any perspective, the two networks work in a resembling way. The most significant difference between a 2-D mesh network in communication field and a transportation road network is that there can be multiple queues at each nodes going to different directions instead of just one queue in the communication network. However, this difference dose not limit the applicability of the *Odd-Even Routing* method in a transportation network. Hereafter in this section, a brief proof in the context of the transportation network will be given to prove the fact that with every vehicle following *Odd-Even Routing* rules specified as follows, the network is guaranteed to be deadlock-free.

Odd-Even turn model:

Rule 1: Any vehicle is not allowed to take an EN turn at any intersection located in an even column, and it is not allowed to take an NW turn at any intersection located in an odd column.

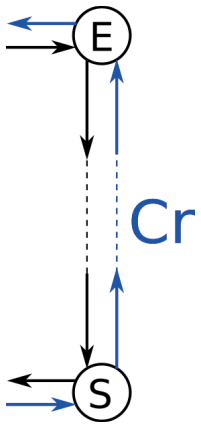
Rule 2: Any vehicle is not allowed to take an ES turn at any intersection located in an even column, and it is not allowed to take an SW turn at any intersection located in an odd column.

Rule 3: No U-turns are allowed in the network.

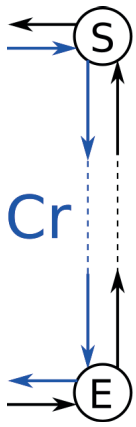
Note: The directions in the space are denoted as SOUTH, NORTH, EAST and WEST. A turn is called a ES turn if the direction of the movement is from EAST to SOUTH. The same is for EN, SW and NW turns.

Theorem 1. *A grid transportation network is free of deadlock as long as all vehicles in the network follow the turning rules specified in the Odd-Even turn model.*

Proof. We prove the theorem by contradiction. Assume there is a set of links $l_1, l_2, \ldots, l_m$ that are in deadlocked state, which indicates the links are connected in a cyclic path, denoted as $\mathscr{C}$. Since *U-turns* are not allowed in the routing rule specified above, it is safe to conclude that this cyclic path must contain both horizontal and vertical links. Otherwise, it is impossible to form a close cyclic path in the grid network. In the cyclic path $\mathscr{C}$, let's denote set of the rightmost links as C_r. Links in C_r are connected in a sequence and heading in the same direction. Assume the starting node and ending node of these connected links are node S and node E. There are two different ways to connect the links in C_r illustrated in Figure 5.2: one is clockwise and the other is counter-clockwise.



(a) Case 1: Counter Clockwise Deadlock Loop



(b) Case 2: Clockwise Deadlock Loop

Figure 5.2: The rightmost links on the cyclic path.

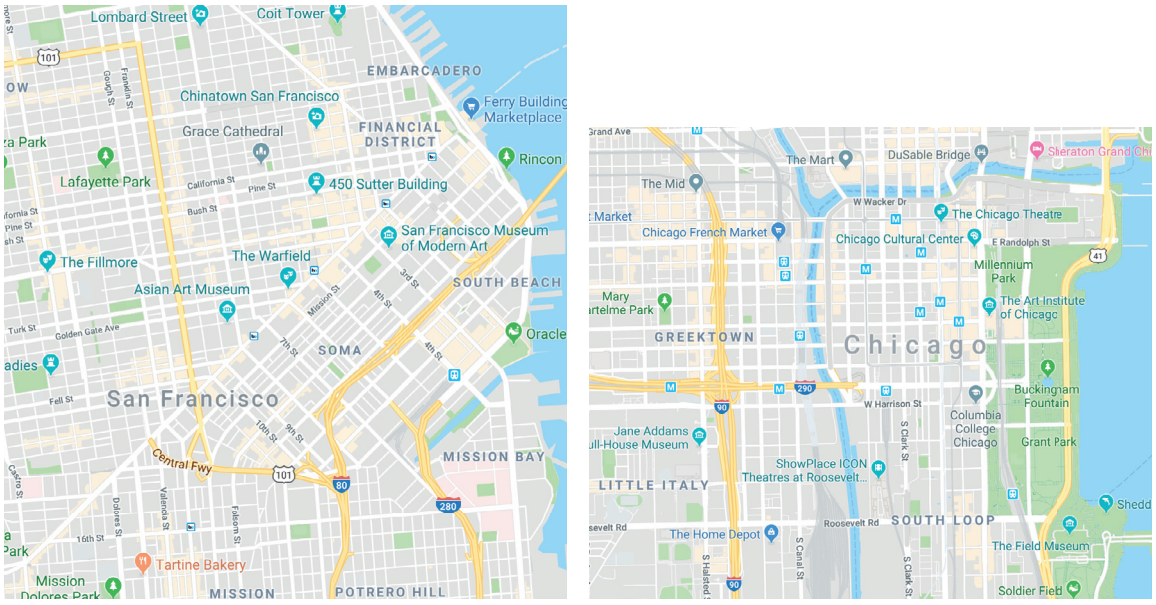
In the first case, the links orient from SOUTH to NORTH. In this case, vehicles at node S are taking a EN turn and vehicles at node E are taking a NW turn. According to Rule 1 in *Odd-Even turn model*, EN and NW turns are not allowed to co-exist in the same column. This contradicts our assumption that node S and node E are in the same right-most column of the deadlock link loop. For the second case, the links orient from NORTH to SOUTH. In this case, vehicles at node S are taking a ES turn and vehicles at node E are taking a SW turn. According to Rule 2 in *Odd-Even turn model*, ES and SW turns are not allowed to co-exist in the same column. This also contradicts out assumption that node S

and node E are in the same right-most column of the deadlock link loop. With the above contradictions, we prove the theorem.

□

5.3.1 Similarity and difference between transportation network and mesh network

The adaptive routing model in Chiu (2000) only works in regular shaped grid communication networks. However, the idea of regulating turning maneuvers at certain locations to avoid deadlocks can be adapted to different applications, and used for different network configurations. It is especially applicable to grid networks commonly seen in downtown areas. Figure 5.3 are maps of the City of San Francisco and the City of Chicago showing that the majority of the road network are in regular grid shape. It is relatively easy to implement a similar regulatory routing rule as the one in Chiu (2000) in physical transportation network. There are a few differences that needs to be considered before implementing such rules in real world.



(a) Map of Downtown San Francisco. (b) Map of Downtown Chicago.

Figure 5.3: Maps of Downtown San Francisco and Downtown Chicago (from *Google Maps*).

First of all, the packets in a communication network occupy no physical space which means the queues in such network have no physical length. However, in transportation

network, vehicles have physical lengths and thus queues have a spatial length.

Secondly, queues in a communication network do not distinguish different directions which means they all share the same queue. However, queues in a transportation road network spread out different queues into different lanes based on the direction to go. For instance, left turn traffic usually has a dedicated left turn lane, while right turn traffic and through traffic might share the same lane.

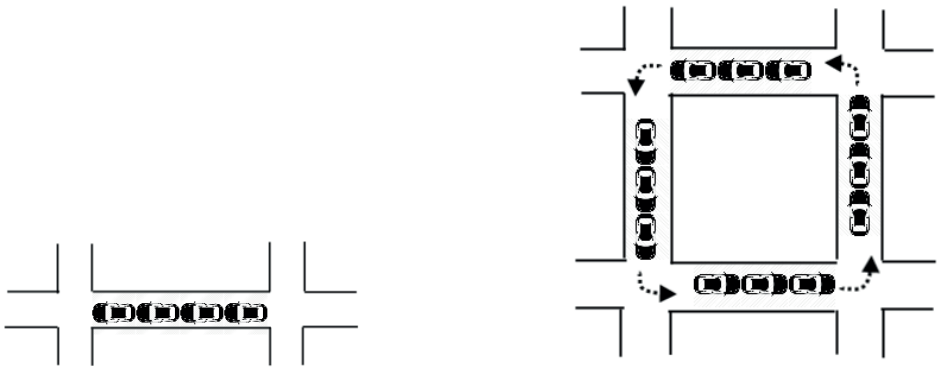
Except for the differences aforementioned, the transportation network studied in this chapter is very similar to the communication network studied in Chiu (2000).

5.4 Deadlock potential and transitive closure

In order to avoid deadlock from happening, it is intuitive to define a probability that a deadlock occurs and then try to minimize that probability. Here we introduce a deadlock probability measurement called deadlock potential. It defines the probability of a cycle of links to be in a deadlock state.

5.4.1 Deadlock potential

A link a has a queuing capacity of Q_a. When there is traffic going through this link, a queue could build up on the link. Let's denote the downstream queue as q_a^d, where $0 \leq q_a^d \leq Q_a$. If $q_a^d = Q_a$, it means queue on this link completely occupies the link, and thus queue spill-back occurs.



(a) a link with queue (b) a cycle with queue

Figure 5.4: Links with queues in the network.

Definition: A link with queue has a potential to spill-back. Such spill-back potential

is defined as: $P_a = q_a^d / Q_a$.

In a cycle of links $L = \{a_1, a_2, ...a_k\}$, if every link has spill-back, then there is a deadlock in this cycle L.

Definition: Deadlock potential of a cycle L is defined as: $DLP_L = \sum_{a \in L} P_a$.

At time t, there could be n_t number of deadlock cycles in the network.

5.4.2 Transitive closure

To find deadlocks in a network is equivalent to identify the cycles in the corresponding graph. Transitive closure algorithm is an algorithm used to compute reachability in a graph. With trivial modifications, it can be applied to identify cycles in a graph (Aho et al., 1972; Koskinen and Herlihy, 2008; Mak et al., 2011). Various methods and techniques have been proposed in the past to find the transitive closure of a graph. The most well-known algorithm was proposed by Floyd (1962) and Warshall (1962), known as Floyd-Warshall algorithm. The original Floyd-Warshall algorithm is used to compute all possible paths with costs through the graph between each pair of vertices . The computational complexity is $O(V^3)$. A detailed description of Floyd-Warshall algorithm is provided in Appendix B.1.

With some trivial modifications, the Floyd-Warshall algorithm can be applied to calculate the connectivity between any pair of vertices in the graph, i.e. compute the transitive closure of the graph. A slightly modified algorithm based on Floyd-Warshall algorithm is present in Appendix B.2 to calculate transitive closures in a graph.

5.5 Optimal deadlock avoidance strategy under DUE with queue spillbacks

As discussed in Section 5.2, one of the objective is to balance between avoiding deadlock in the network and reducing total travel cost by optimally choosing the number of *DLA-routing vehicles* in the network and the departure time of each vehicle. The total travel time (TTT) can be written as Equation (5.1). It consists of two parts: travel time on links which corresponds to the first part on the right-hand-side, and queue waiting time at origins which corresponds to the second part on the right-hand-side.

$$TTT = \sum_{s' \in S} \int_0^T [\sum_{(o',o) \in L_o} \int_0^t v^{s'}_{(o',o)}(\xi) d\xi - \sum_{(s,s') \in L_s} \int_0^t p^{s'}_{(s,s')}(\xi) d\xi] dt + \sum_{s' \in S} \int_0^T q^{d,s'}_{(o',o)}(t) dt \quad (5.1)$$

According to Ma et al. (2014), it is proved that the total travel time of a network with a single destination can be re-written as Equation (5.2). Details of the proof can be found in Ma et al. (2014).

$$TTT = \sum_{(s,s') \in L_s} \int_o^T t \cdot p_{(s,s')}(t) dt \quad (5.2)$$

Yu et al. (2018) extends the single destination case to multiple destinations case, as shown in Equation (5.3). In this study we adopt Yu et al. (2018)'s multiple destinations formulation as in Equation(5.3).

$$TTT = \sum_{s' \in S} \sum_{(s,s') \in L_s} \int_0^T t \cdot p^{s'}_{(s,s')}(t) dt \quad (5.3)$$

In this section, an optimal control framework aimed to find the optimal percentage of DLA-routing vehicle is proposed with DUE, queue spill-backs and other constraints. The optimization objective is to minimize the potential of deadlock through optimally choosing the number of DLA-routing vehicle in the network, and meanwhile maintain the total travel time within a reasonable range. The objective function is shown in Equation (5.4). The first term on the right-hand-side is the summation of downstream queues in all deadlocks over all time steps, while the second term is the total travel time. α and β in the formulation are weight scalars for deadlock potential and total travel time.

$$\min_{\gamma^r} z \triangleq \alpha \cdot \sum_{t \in [0,T]} \sum_{\xi^\nu_t \in \Xi_t} \sum_{(i,j) \in L_{\xi^\nu_t}} \sum_{s'} \sum_{r} q^{d,s',r}_{(i,j)}(t) + \beta \cdot TTT \quad (5.4)$$

Assume that at any time t, there are $N_{TC}(t)$ deadlock cycles, i.e. $|\Xi_t| = N_{TC}(t)$. The transitive closure algorithm in Appendix B.2 can be used to compute $N_{TC}(t)$ given $P_{ij}(t)$.

The equivalent algebra formulation is (Fischer and Meyer, 1971):

$$A^*(t) = I \oplus \overline{A}^1(t) \oplus \overline{A}^2(t) \oplus \cdots \oplus \overline{A}^{n-1}(t) \quad (5.5)$$

$$\overline{A}(t) = A(t) \circ P(t) \tag{5.6}$$

, where $\oplus$ is an operator in Max-Plus algebra and $\circ$ is the element-wise production operator. $A(t)$ is the adjacency matrix. Link deadlock potential matrix, **P(t)**, is calculated by Equation (5.7). More details about max-plus algebra can be found in Heidergott et al. (2014).

$$P_{ij}(t) = \begin{cases} \dfrac{\sum_{s'} \sum_{r} q_{(i,j)}^{d,s',r}(t)}{\overline{Q}_{(i,j)}} & if(i,j) \in L_e \\ \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

In Equation (5.7), each link is calculated at every time step t a spill-back potential $P_{ij}(t)$. The term $q_{(i,j)}^{d,s',r}(t)$ in Equation (5.7) is the downstream queue at link (i,j). The vehicles in this queue are from group r and heading toward destination s'. Though each downstream queue is divided into different groups based on their destinations and vehicle groups, in Double Queue (DQ) model there is no segregation between those queues. Hence, as the definition given in Section 5.4, all queues at downstream of link (i,j) are summed up before it is divided by the queue capacity of that link to get the link deadlock potential.

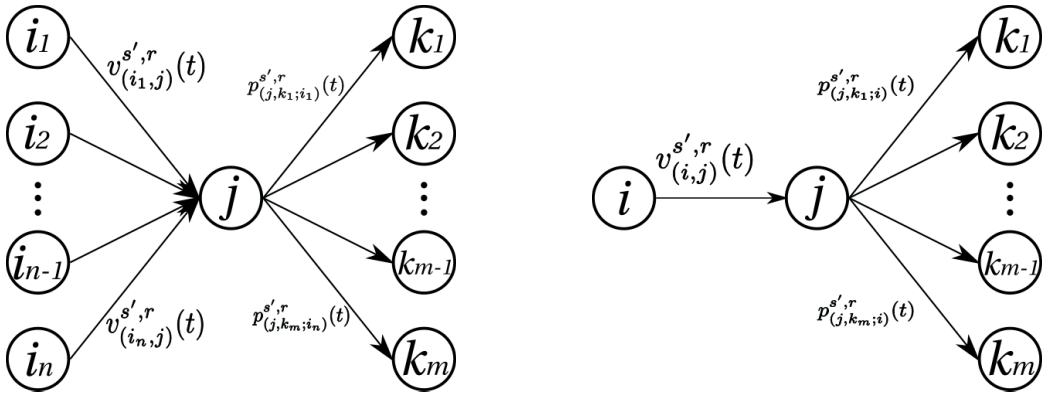
5.5.1 Flow dynamics and constraints

In this study, we assume inflow carries the information of where it comes from. Once it enters a link, it is mixed with other flows (from different incoming links) on that link. It loses the information of what node it comes from when it is exiting this link. With that explained, inflow to a link is denoted as $p_{(i,j;h)}^{s',r}(t)$, and exit flow is denoted as $v_{(i,j)}^{s',r}(t)$. The notation $(i,j;h)$ is designed to keep track of the information of incoming node. Note that exit flow, $v_{(i,j)}^{s',r}(t)$, does not have h in its subscript. This notation scheme makes it possible to distinguish inflows with regards to their corresponding turning movements. Later in the traveler's route choice model, different turning movements will be given different penalties to influence travelers' behavior.

Flow conservation:

In the nodal DUE model, flow conservation is usually applied at each individual node. The total inflows to a node should be equal to the total exit flows out of a node. In Figure 5.5a, total inflow to node j equals: $\sum_{i:(i,j)\in L_e} v^{s',r}_{(i,j)}(t)$ and total exit flow out from node j equals $\sum_{i:(i,j)\in L_e} \sum_{k:(j,k)\in L_e} p^{s',r}_{(j,k;i)}(t)$. A nodal flow conservation is shown as Equation (5.8).

$$\sum_{i:(i,j)\in L_e} \sum_{k:(j,k)\in L_e} p^{s',r}_{(j,k;i)}(t) = \sum_{i:(i,j)\in L_e} v^{s',r}_{(i,j)}(t) \quad \forall i \in N \setminus S, s' \in S, t \in [0,T] \tag{5.8}$$



(a) Flow conservation at a node. (b) Flow conservation of a link.

Figure 5.5: Flow conservation at node and link level.

As discussed earlier in this section, flows are associated with different attributes: destination (s'), groups (r) and incoming direction (using $(i,j;h)$ to denote a movement of $h \to i \to j$). Different from a generic nodal DUE model, flow conservation has to be held at each individual link level as well. In Figure 5.5b, the exit flow of link (i,j) is equal to $v^{s',r}_{(i,j)}(t)$. This exit flow of link (i,j) is then distributed among all the outgoing links from node j, i.e. $(j,k_1),(j,k_2),\ldots,(j,k_m)$. Hence, there is a link level flow conservation as shown in Equation (5.9).

$$v^{s',r}_{(i,j)}(t) = \sum_{k:(j,k)\in L_e} p^{s',r}_{(j,k;i)}(t) \quad \forall i \in N \setminus S, s' \in S, t \in [0,T] \tag{5.9}$$

UE route choice:

We assume travelers are all rational. They always choose the minimum cost paths to their destinations. From a link-node point of view, at each node, flows are assigned to the next link that is on the minimal cost path from current node to destination. In static traffic assignment case, this route choice model is referred as Wardrop's principle (Wardrop,

1952). The same principle holds in dynamic traffic assignment (Mahmassani and Herman, 1984; Friesz et al., 1993; Ban et al., 2012b; Ma et al., 2014). In this particular study, the principle is slightly adapted to accommodated the movement-based incoming flow $p^{s',r}_{(i,j;h)}(t)$. At every time step t, the minimal travel time from node i, with incoming node as h, to destination s' is denoted as $\pi^{s'}_{(h,i)}(t)$. If any downstream node j of node i is on a minimal cost path, i.e. $\phi^r_{(i,j;h)}(t) + \tau_{(i,j)}(t) + \pi^{s',r}_{(i,j)}(t + \tau_{(i,j)}(t)) - \pi^{s',r}_{(h,i)}(t) = 0$, there shall be some non-negative flow $p^{s',r}_{(i,j;h)}(t)$ flowing from node h through node i to node j bounding for destination s'; otherwise, $p^{s',r}_{(i,j;h)}(t) = 0$. This principle translates into the complementarity constraint as shown in Equation (5.10). $\phi^r_{(i,j;h)}(t)$ is a penalty term associated with turn $(i, j; h)$ for group r travelers at time t. For *DLA-routing vehicles*, according to the routing rule specified in Section 5.3, they are not permitted to make certain turns at certain intersections. A penalty $\phi^r_{(i,j;h)}(t)$ works perfectly to serve the purpose of regulating turning maneuvers of *Odd-Even routing vehicles* at certain intersections in Equation (5.10). For *Odd-Even routing vehicles*, a turn $(i, j; h)$ that is not permitted by the *OE Routing Rules* has a large penalty (for illustration purpose, this penalty can be set to ∞). In this case, the right-hand-side of Equation (5.10) is positive, which implies that $p^{s',r}_{(i,j;h)}(t) = 0$. Therefore, no *OE Routing vehicle* flow is moving through turn $(i, j; h)$. For *OE Routing vehicles* at permitted turns or normal vehicles at any turns, the penalty is zero reflecting the fact that no regulation is imposed for these cases.

$$0 \leq p^{s',r}_{(i,j;h)}(t) \perp [\phi^r_{(i,j;h)}(t) + \tau_{(i,j)}(t) + \pi^{s',r}_{(i,j)}(t + \tau_{(i,j)}(t)) - \pi^{s',r}_{(h,i)}(t)] \geq 0 \tag{5.10}$$

Equations (5.11-5.12) are the queue dynamics of the Double Queue Model for upstream queue and downstream queue, respectively.

$$q^{u,s',r}_{(i,j)}(t) = \int_0^t [\sum_h p^{s',r}_{(i,j;h)}(\xi) - v^{s',r}_{(i,j)}(\xi - \tau^{\omega}_{(i,j)})]d\xi + q^{u,s',r}_{(i,j)}(0) \tag{5.11}$$

$$q^{d,s',r}_{(i,j)}(t) = \int_0^t [\sum_h p^{s',r}_{(i,j;h)}(\xi - \tau^{0}_{(i,j)}) - v^{s',r}_{(i,j)}(\xi)]d\xi + q^{d,s',r}_{(i,j)}(0) \tag{5.12}$$

Equations (5.13-5.14) are summations of the upstream queues and downstream queues, respectively, on link (i, j) at time t over different destinations, different traveler groups and different incoming nodes.

$$q^{u}_{(i,j)}(t) = \sum_{s'} \sum_{r} q^{u,s',r}_{(i,j)}(t) \tag{5.13}$$

$$q^{d}_{(i,j)}(t) = \sum_{s'} \sum_{r} q^{d,s',r}_{(i,j)}(t) \tag{5.14}$$

Equations (5.15-5.16) are summations of the incoming flow and exit flow, respectively, on link (i, j) at time t over different destinations, different traveler groups and different incoming nodes.

$$p_{(i,j)}(t) = \sum_{s'} \sum_{h} \sum_{r} p^{s',r}_{(i,j;h)}(t) \tag{5.15}$$

$$v_{(i,j)}(t) = \sum_{s'} \sum_{r} v^{s',r}_{(i,j)}(t) \tag{5.16}$$

Upstream queues are capped by the total queue capacity $\overline{Q}_{(i,j)}$, see Equation (5.17). Total inflow $p_{(i,j)}(t)$ to link (i, j) is capped by the total link flow capacity $\overline{C}_{(i,j)}$, see Equation (5.18). Total exit flow $v_{(i,j)}(t)$ out of link (i, j) is capped by the effective link flow capacity $\delta_{(i,j)}(t)\overline{C}_{(i,j)}$, see Equation (5.19).

In this study, the primary focus is exploring how *OE Routing* could benefit the overall system in regard to deadlock potential and total travel cost. Therefore, signal control is not explicitly studied. To keep problem concise and solvable, all intersection nodes are controlled by fixed-timing signal lights in contrast to optimal control studied in Yu et al. (2018). For fixed timing control, green split remains the same between cycles. Each incoming link (j, i) at node i gets a fixed (pre-given) green time allocation $\delta_{(j,i)}(t)$, which reduces the effective exist flow capacity as in Equation (5.19).

$$q^{u}_{(i,j)}(t) \leq \overline{Q}_{(i,j)} \tag{5.17}$$

$$p_{(i,j)}(t) \leq \overline{C}_{(i,j)} \tag{5.18}$$

$$v_{(i,j)}(t) \leq \delta_{(i,j)}(t)\overline{C}_{(i,j)} \tag{5.19}$$

$$q^{d,s',r}_{(i,j)}(t), q^{u,s',r}_{(i,j)}(t), p^{s',r}_{(i,j;h)}(t), v^{s',r}_{(i,j)}(t) \geq 0 \quad \forall (i,j) \in L_e, s' \in S, t \in [0,T] \tag{5.20}$$

Ma et al. (2014) derived the relation between the link queue capacity $\overline{Q}_{(i,j)}$ and the link flow capacity $\overline{C}_{(i,j)}$ based on triangular fundamental diagram. The relation between those two is adopted in this study, and is shown in Equation (5.21).

$$\overline{Q}_{(i,j)} = (\tau^0_{(i,j)} + \tau^{\omega}_{(i,j)}) \cdot \overline{C}_{(i,j)} \tag{5.21}$$

At time $t = 0$, vehicles are not entering the network yet. All downstream and upstream queues, as Equations (5.22-5.23) present, equal 0 except for the dummy links (o, o'). On dummy links (o, o'), initially, all vehicles are queued up waiting to enter the network. Upstream and downstream queues are equal to total demand at that dummy link (o, o'), as Equation (5.22) shows. Note that $\gamma^r, r \in \{0, 1\}$ is homogeneous in the network among different origins. Equation (5.24) indicates there is no inflow to the dummy links (o, o') at origins.

$$q^{u,s',r}_{(o',o)}(0) = q^{d,s',r}_{(o',o)}(0) = \gamma^r H^{(o',s')}, \quad \text{where } \sum_r \gamma^r = 1, r \in \{0,1\} \tag{5.22}$$

$$q^{u,s',r}_{(i,j)}(0) = q^{d,s',r}_{(i,j)}(0) = 0 \tag{5.23}$$

$$p^{s',r}_{(o',o)}(t) = 0 \tag{5.24}$$

In Section 5.2, we assume a large enough time horizon T is picked to guarantee there are no vehicles left in the network at the end of time T. This explicitly means that: 1) for every link there is no upstream queue nor downstream queue, and 2) for every link there is no inflow coming into link (i, j) after time $T - \tau^0_{(i,j)}$. The first is true otherwise there will be queues at time T, which contradicts the assumption of simulation horizon T. The second is also true due to the fact that if there are incoming flow $p^{s',r}_{(i,j;h)}(t)$ during $[T - \tau^0_{(i,j)}, T]$, these flows can not exit their link (i, j) in less than free flow travel time $\tau^0_{(i,j)}$. Therefore, there will be flows left on link (i, j) at time T, which also contradicts the assumption of simulation horizon T. These two constraints are described in Equation (5.25) and Equation (5.26).

$$q_{(i,j)}^{u,s',r}(T) = q_{(i,j)}^{d,s',r}(T) = 0, \forall (i,j) \in L_e, r = \{0,1\} \tag{5.25}$$

$$p_{(i,j;h)}^{s',r}(t) = 0, \forall t \in [T - \tau_{(i,j)}^{0}, T], r = \{0,1\} \tag{5.26}$$

Extra Constraints:

$$p_{(i,j;h)}^{s',r}(t) = 0 \quad s' \in S \tag{5.27}$$

$$q_{(i,j)}^{u,s',r}(t_L) = q_{(i,j)}^{d,s',r}(t_L) = 0 \quad s' \in S, (i,j) \in L_o \tag{5.28}$$

Equation (5.27) states that no u-turns are allowed in the network. This is not only a requirement of *OE Routing Rule*, but also an implicit requirement for the double queue model with dummy links to behave close to real scenario. We discover that without such a constraint, sometimes in the network, flows will enter dummy links and circulate in the dummy links for certain amount of time before re-entering the regular links (see Figure 5.6) even if destination node s' is not its final destination. This behavior is due to the fact that dummy links have infinity queue capacities and trivial (0, in this study) link travel time. By sending flows into dummy links and circulating flows in dummy links, the system avoids those flows being queued up on regular links. Since fewer flows are queued on regular links, links become less congested. However, such flow circulation is not well characterizing flow propagation and traffic queuing in real world. In real world scenario, traffic is only allowed to queue on regular links except for dummy origin links where flows initially enter the network. In order to model such regulations, we impose the constraint to limit circulation flows, see Equation (5.27).

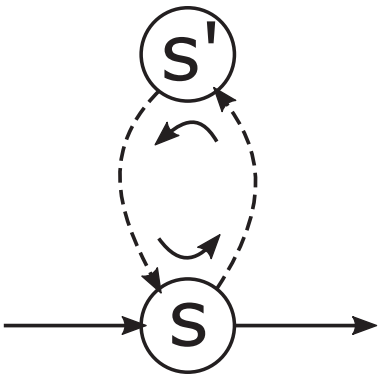


Figure 5.6: Flow circulates in dummy links.

Equation (5.28) states the latest departure time requirement: at time $t = t_L$, both downstream and upstream queues at dummy origin links are cleared. This requirement resembles the case where travelers are required to depart within a certain time frame. This constraint has another purpose in the DLA model. Since the objective function consists of two parts: queue length on regular links and total travel time in the network, traffic tends to stay in dummy links as much as possible to prevent queues being built up on regular links when optimizing the objective function. However, this behavior does not well reflect how travelers in real word behave. In real world scenario, travelers have no incentive to stay in dummy links as long as their total travel cost is minimized. Equation (5.28) is thus designed to cope with this problem. This constraint forces all traffic to load onto regular links before $t = t_L$ so that queues can be observed on regular links as expected.

5.5.2 Feasibility of the proposed problem

In the subsection, a brief proof is presented to show the feasibility of the proposed problem by manually constructing a feasible solution to the DLA model described in Section 5.5.

Proof. Assume there are K total $O-D$ pairs in the network. Randomly number the $O-D$ pairs with subscripts from 1 to K. All travelers in the network are from group $r = 0$, i.e. all travelers are driving normal vehicles.

For $k^{th}O-D$ pair $(o', s')_k$, find the shortest path between this pair of origin and destination pair. Identify the link on the shortest path with lowest link flow capacity. Denote this link flow capacity as $C(o', s')_k$. Load and discharge flow between origin and destination pair $(o'.s')_k$ at a flow rate equal to $C(o', s')_k$. With such flow discharging scheme, the total

discharging time $T_{(o',s')_k}$ for traffic demand between origin and destination pair $(o',s')_k$, i.e. $H_{(o',s')_k}$, is

$$\lfloor \frac{H_{(o',s')_k}}{C_{(o',s')_k}} \rfloor \leq T_{(o',s')_k} \leq \lceil \frac{H_{(o',s')_k}}{C_{(o',s')_k}} \rceil \tag{5.29}$$

After the traffic flow between origin and destination pair $(o',s')_k$ is completely discharged, move on the $(k+1)^{th}$ origin and destination pair $(o',s')_{k+1}$ and repeat the discharging step described above. Repeat this process until all K origin destination pairs are served.

Let $\overline{T_{(o',s')_k}} \triangleq \lceil \frac{H_{(o',s')_k}}{C_{(o',s')_k}} \rceil$, and $\overline{T} \triangleq \sum_{k=1}^{K} \overline{T_{(o',s')_k}} = \sum_{k=1}^{K} \lceil \frac{H_{(o',s')_k}}{C_{(o',s')_k}} \rceil$. By setting the time horizon T to be significantly larger than $\overline{T}$, it is guaranteed that all demand flows can be served and no flow remains in the network at the end of T.

By following the above steps, the constructed solution is proved to be a feasible solution. Hence, we proved the DUE problem has at least one feasible solution, i.e. is feasible.

□

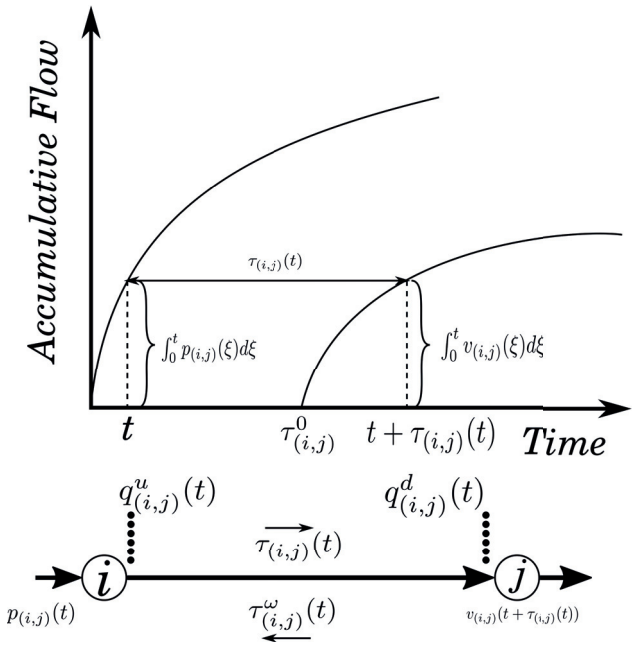


Figure 5.7: Link travel time in double queue model.

5.6 Approximation

5.6.1 Link travel time approximation

In DUE constraint (5.10), a delay term $\pi^{s',r}_{(i,j)}(t+\tau_{(i,j)}(t))$ that depends on future time (t) and state $(\tau_{(i,j)}(t))$ is used to describe travelers' route choice in a dynamic setting. It depends on knowing link travel time $\tau_{(i,j)}(t)$ to evaluate the DUE constraints. Since this link travel time $\tau_{(i,j)}(t)$ is implicitly defined in the DUE model (see Figure 5.7 for an illustration), it is difficult to explicitly keep track of travel time in the network. In this study, we adopt a similar link travel time approximation scheme with Ma et al. (2017) and Yu et al. (2018). An approximation coefficient $\alpha_{(i,j)}(t)$ is used to approximate link travel time at time t based on free flow travel time $\tau^0_{(i,j)}$, downstream queue $q^d_{(i,j)}(t+\tau^0_{(i,j)})$ and link capacity $\overline{C}_{(i,j)}$. See Equation (5.30).

$$\tau_{(i,j)}(t) = \tau^0_{(i,j)} + \alpha_{(i,j)}(t) \cdot \frac{q^d_{(i,j)}(t+\tau^0_{(i,j)})}{\overline{C}_{(i,j)}} \tag{5.30}$$

5.6.2 Time-dependent delay terms approximation

Similar to Ma et al. (2015) and Yu et al. (2018), a "pseudo derivative" (PD) term $\sigma^{s',r}_{(i,j)}(t)$ is used to approximate the time-dependent and state-dependent delay term $\pi^{s',r}_{(i,j)}(t+\tau_{(i,j)}(t))$ with a time-dependent delay term $\pi^{s',r}_{(i,j)}(t+\tau^0_{(i,j)})$. The latter does not depend on any future states. See Equation (5.31).

$$\pi^{s',r}_{(i,j)}(t+\tau_{(i,j)}(t)) = \pi^{s',r}_{(i,j)}(t+\tau^0_{(i,j)}) + \sigma^{s',r}_{(i,j)}(t) \cdot \left(\tau_{(i,j)}(t) - \tau^0_{(i,j)}\right) \tag{5.31}$$

Substitute Equation (5.30) into Equation (5.31), we get:

$$\begin{aligned}
\pi^{s',r}_{(i,j)}(t+\tau_{(i,j)}(t)) &= \pi^{s',r}_{(i,j)}(t+\tau^0_{(i,j)}) + \sigma^{s',r}_{(i,j)}(t) \cdot \left(\tau^0_{(i,j)} + \alpha_{(i,j)}(t) \cdot \tfrac{q^d_{(i,j)}(t+\tau^0_{(i,j)})}{\overline{C}_{(i,j)}} - \tau^0_{(i,j)}\right) \\
&= \pi^{s',r}_{(i,j)}(t+\tau^0_{(i,j)}) + \left(\sigma^{s',r}_{(i,j)}(t) \cdot \alpha_{(i,j)}(t)\right) \cdot \tfrac{q^d_{(i,j)}(t+\tau^0_{(i,j)})}{\overline{C}_{(i,j)}}
\end{aligned} \tag{5.32}$$

5.6.3 Travels' route choice complementarity approximation

With the approximation of link travel time in Equation (5.30) and route travel time approximation in Equation (5.31), the DUE route choice complementarity in Equation (5.10)

is therefore becoming Equation (5.33).

$$0 \leq p^{s',r}_{(i,j;h)}(t) \perp [\phi^{r}_{(i,j;h)}(t)+\tau^{0}_{(i,j)}+\pi^{s',r}_{(i,j)}(t+\tau^{0}_{(i,j)})+\frac{q^{d}_{(i,j)}(t+\tau^{0}_{(i,j)})}{\overline{C}_{(i,j)}}\cdot\alpha_{(i,j)}(t)\cdot(\sigma^{s',r}_{(i,j)}(t)+1)-\pi^{s',r}_{(h,i)}(t)] \geq 0 \tag{5.33}$$

Similar to Yu et al. (2018), a combined approximation coefficient $\theta^{s',r}_{(i,j)}(t)$ is introduced:

$$\theta^{s',r}_{(i,j)}(t) = \alpha_{(i,j)}(t) \cdot \left(\sigma^{s',r}_{(i,j)}(t) + 1\right) \tag{5.34}$$

5.7 Discretization of the DLA model

As stated in Ma et al. (2014), it is very difficult if not impossible to obtain analytical solutions to state-constrained optimal control problems with time delay in continuous time domain. To find numerical solutions, the continuous-time model in Section 5.5 has to been properly discretized in time to convert into a finite-dimensional optimization problem. After such discretization, numerical solutions can be solved using various methods. There are different choices in time discretization. In order to avoid possible negative queues in the network (Ban et al., 2012b; Ma et al., 2014; Yu et al., 2018), in this study a similar implicit time step discretization scheme to the one used in Ma et al. (2017) is adopted. Specifically, the time step λ should be small enough so that all the time-dependent variables in the model should be multiples of the time step length selected. Let integer N_λ denote the total number of time steps, hence we have the following relations:

$$0 \triangleq t_{\lambda,1} < \cdots < t_{\lambda,N_\lambda} < t_{h,N_\lambda+1} \triangleq T \tag{5.35}$$

$$t_{\lambda,n} = \lambda \cdot (n-1), \quad n = 1, 2, \ldots, N_\lambda + 1 \tag{5.36}$$

We also define $n^{0;\lambda}_{(i,j)} \triangleq \tau^{0}_{(i,j)}/\lambda$ and $n^{\omega;\lambda}_{(i,j)} \triangleq \tau^{\omega}_{(i,j)}/\lambda$ for any $(i,j) \in L$. With the discretization scheme described above, the deadlock avoidance model routing under DUE and queue spillback in discrete-time form is then present in the remaining of this section. In Subsection 5.7.5, an iterative solving algorithm is used to solve the proposed problem. The superscript l denotes the l^{th} iteration in the solving process.

5.7.1 Objective function

$$\min_{\{\gamma^r\}} z^l = \alpha \cdot \sum_{n}^{N_\lambda} \sum_{\xi_n^\nu \in \Xi_n} \sum_{(i,j)\in L_{\xi_n^\nu}} \sum_{s'\in S} \sum_{r} q_{(i,j)}^{d,s',r,l}(n) + \beta \cdot [\sum_{s'\in S} \sum_{(s,s')\in L_s} \lambda \cdot \sum_{n=1}^{N_\lambda} n \cdot p_{(s,s')}^{s',l}(n)] \quad (5.37)$$

5.7.2 Flow dynamics and constraints

$$q_{(i,j)}^{u,s',r,l}(n) = \lambda \cdot \sum_{t=1}^{n} \sum_{h} p_{(i,j;h)}^{s',r,l}(t) + q_{(i,j)}^{u,s',r,l}(1) \quad s' \in S, (i,j) \in L_e, (h,i) \in L_e, n = 2, \ldots, n_{(i,j)}^{\omega,\lambda} \quad (5.38)$$

$$q_{(i,j)}^{u,s',r,l}(n) = \lambda \cdot \sum_{t=1}^{n} \sum_{h} p_{(i,j;h)}^{s',r,l}(t) - \lambda \cdot \sum_{t=1}^{n-n_{(i,j)}^{\omega,\lambda}} v_{(i,j)}^{s',r,l}(t) + q_{(i,j)}^{u,s',r,l}(1) \quad s' \in S, (i,j) \in L_e, (h,i) \in L_e, n = n_{(i,j)}^{\omega,\lambda} + 1, \ldots, N_\lambda + 1 \quad (5.39)$$

$$q_{(i,j)}^{d,s',r,l}(n) = -\lambda \cdot \sum_{t=1}^{n} v_{(i,j)}^{s',r,l}(t) + q_{(i,j)}^{d,s',r,l}(1) \quad s' \in S, (i,j) \in L_e, n = 2, \ldots, n_{(i,j)}^{0,\lambda} \quad (5.40)$$

$$q_{(i,j)}^{d,s',r,l}(n) = \lambda \cdot \sum_{t=1}^{n-n_{(i,j)}^{0,\lambda}} \sum_{h} p_{(i,j;h)}^{s',r,l}(t) - \lambda \cdot \sum_{t=1}^{n} v_{(i,j)}^{s',r,l}(t) + q_{(i,j)}^{d,s',r,l}(1) \quad s' \in S, (i,j) \in L_e, (h,i) \in L_e, n = n_{(i,j)}^{0,\lambda} + 1, \ldots, N_\lambda + 1 \quad (5.41)$$

Flow conservation:

$$v_{(i,j)}^{s',r}(n) = \sum_{k:(j,k)\in L_e} p_{(j,k;i)}^{s',r}(n) \quad s' \in S, (i,j) \in L_e, n = 1, \ldots, N_\lambda + 1 \quad (5.42)$$

Flow and queue summation:

$$q_{(i,j)}^{u,l}(n) = \sum_{s'} \sum_{r} q_{(i,j)}^{u,s',r,l}(n) \quad (i,j) \in L_e, n = 1, \ldots, N_\lambda + 1 \quad (5.43)$$

$$q_{(i,j)}^{d,l}(n) = \sum_{s'} \sum_{r} q_{(i,j)}^{d,s',r,l}(n) \quad (i,j) \in L_e, n = 1, \ldots, N_\lambda + 1 \quad (5.44)$$

$$p^{l}_{(i,j)}(n) = \sum_{s'} \sum_{h} \sum_{r} p^{s',r,l}_{(i,j;h)}(n) \quad (i,j) \in L_e, n = 1, \ldots, N_\lambda + 1 \tag{5.45}$$

$$v^{l}_{(i,j)}(n) = \sum_{s'} \sum_{r} v^{s',r,l}_{(i,j)}(n) \quad (i,j) \in L_e, n = 1, \ldots, N_\lambda + 1 \tag{5.46}$$

$$q^{u,l}_{(i,j)}(n) \leq (n^{0}_{(i,j)} + n^{\omega}_{(i,j)}) \cdot \overline{C}_{(i,j)} \tag{5.47}$$

$$p^{l}_{(i,j)}(n) \leq \overline{C}_{(i,j)} \tag{5.48}$$

$$v^{l}_{(i,j)}(n) \leq \delta^{l}_{(i,j)}(n) \cdot \overline{C}_{(i,j)} \tag{5.49}$$

5.7.3 Dynamic UE route choice

$$p^{s',r,l}_{(i,j;h)}(n) \cdot \left[\phi^{r}_{(i,j;h)}(n) + n^{0,\lambda}_{(i,j)} + \pi^{s',r,l}_{(i,j)}(n + n^{0,\lambda}_{(i,j)}) + \frac{q^{d,l}_{(i,j)}(n + n^{0,\lambda}_{(i,j)})}{\overline{C}_{(i,j)}} \cdot \theta^{s',r,l}_{(i,j)}(n) - \pi^{s',r,l}_{(h,i)}(n) \right] = 0 \tag{5.50}$$

$$s' \in S, (i,j) \in L_e, n = 1, \ldots, N_\lambda - n^{0,\lambda}_{(i,j)} + 1$$

$$n^{0,\lambda}_{(i,j)} + \pi^{s',r,l}_{(i,j)}(n + n^{0,\lambda}_{(i,j)}) + \frac{q^{d,l}_{(i,j)}(n + n^{0,\lambda}_{(i,j)})}{\overline{C}_{(i,j)}} \cdot \theta^{s',r,l}_{(i,j)}(n) - \pi^{s',r,l}_{(h,i)}(n) \geq 0 \tag{5.51}$$

$$s' \in S, (i,j) \in L_e, n = 1, \ldots, N_\lambda - n^{0,\lambda}_{(i,j)}$$

5.7.4 Initial conditions and other constraints

$$\sum_{r} q^{u,s',r,l}_{(o',o)}(1) = \sum_{r} q^{d,s',r,l}_{(o',o)}(1) = H^{(o',s'),\lambda} \quad (o',s') \in W, s' \in S, (o',o) \in L_o \tag{5.52}$$

$$\sum_{r} q^{u,s',r,l}_{(i,j)}(1) = \sum_{r} q^{d,s',r,l}_{(i,j)}(1) = 0 \quad s' \in S, (i,j) \in L \cup L_s \tag{5.53}$$

$$p^{s',r,l}_{(o',o)}(n) = 0 \quad s' \in S, (i,j) \in L_o, n = 1, \ldots, N_\lambda + 1 \tag{5.54}$$

$$p^{s',r,l}_{(i,j;h)}(n) = 0 \quad s' \in S, (i,j) \in L, n = N_h - n^{0,\lambda}_{(i,j)} + 2, \ldots, N_h \tag{5.55}$$

$$q^{u,s',r,l}_{(i,j)}(N_h + 1) = q^{d,s',r,l}_{(i,j)}(N_h + 1) = 0 \quad s' \in S, (i,j) \in L_e \tag{5.56}$$

$$q^{u,l}_{(i,j)}(n), q^{d,l}_{(i,j)}(n), q^{u,s',l}_{(i,j)}(n), q^{d,s',l}_{(i,j)}(n), p^{s',r,l}_{(i,j;h)}(n), v^{s',r,l}_{(i,j)}(n) \geq 0 \tag{5.57}$$

$$s' \in S, (i,j) \in L_e, j \in \Pi, n = 1, \ldots, N_h + 1$$

$$p^{s',r,l}_{(i,j;h)}(n) = 0 \quad s' \in S, (i,j) \in L, h = j, n = 1, \ldots, N_h \tag{5.58}$$

$$q^{u,s',r,l}_{(i,j)}(N_L + 1) = q^{d,s',r,l}_{(i,j)}(N_L + 1) = 0 \quad s' \in S, (i,j) \in L_o \tag{5.59}$$

Equation (5.60) describes the penalty imposed at every turn. The first row in the equation specifies that for *DLA-routing vehicles* the turning penalty is ∞ if turn $(h \to i \to j)$ is not permitted per OE routing rule. For other cases, there is no penalty.

$$\phi^r_{(i,j;h)} = \begin{cases} \infty, & r = 1 \text{ and turn } (h \to i \to j) \text{ is not permitted} \\ 0, & otherwise \end{cases} \tag{5.60}$$

5.7.5 An iterative solution procedure

Similar to Ma et al. (2015), in the study an iterative algorithm illustrated in Figure 5.8 is developed to solve the DLA problem.

The detailed steps are summarized as follows.

Step 1: Set iteration count $l = 1$. Randomly generate the set of theta $\{\theta^{s',r,l}_{(i,j)}(t)\}$. Since there are no previous link queue information prior iteration $l = 1$, we can not calculate transitive closures using Algorithm 5 and Algorithm 6 in Appendix B. To serve as the initialization purpose, we set $\Xi^l_t \equiv \emptyset, \forall t \in [0, T]$, which means there is no transitive closure in the 1^{st} iteration.

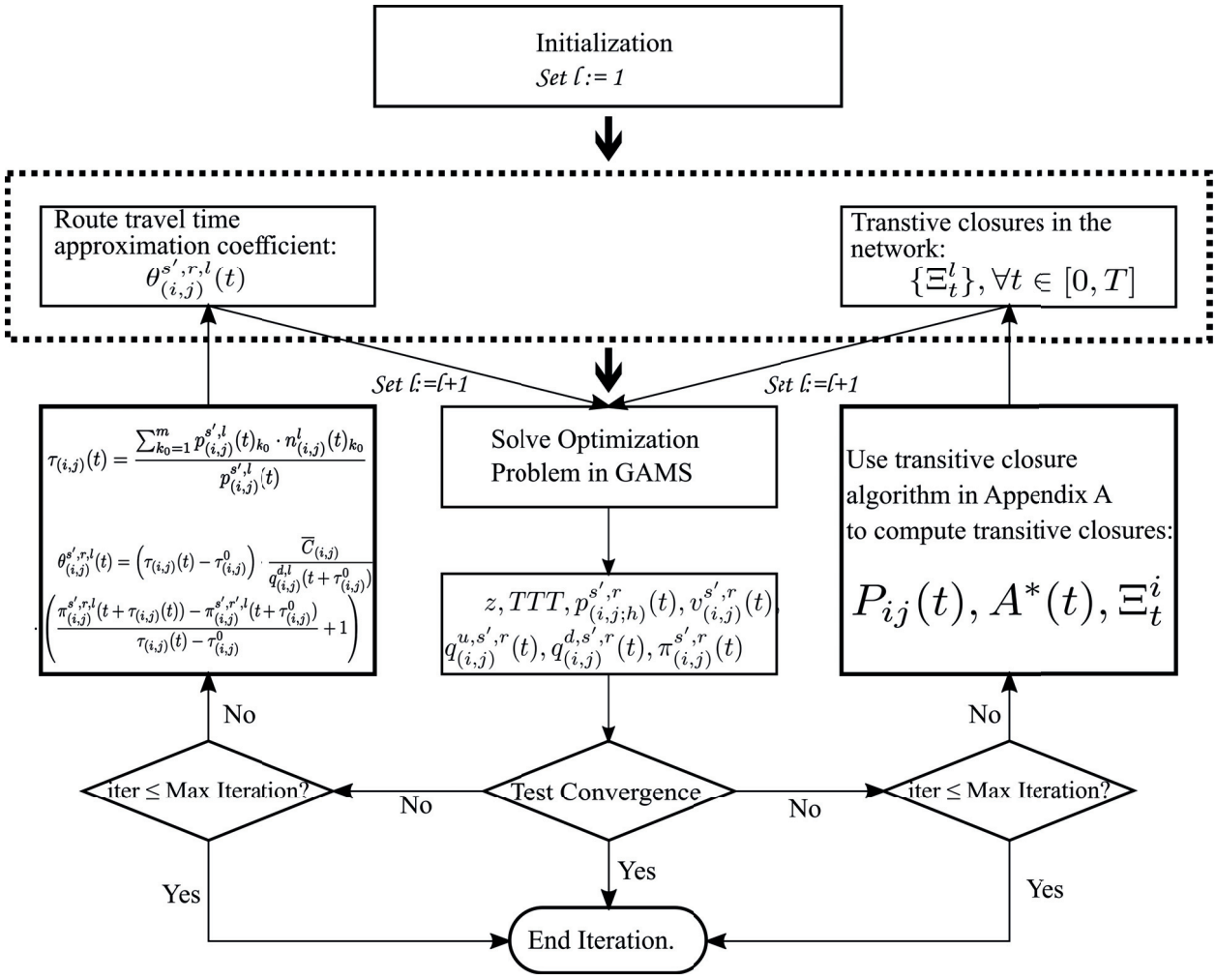


Figure 5.8: A heuristic-based iterative solving algorithm.

Step 2: Feed $\{\theta^{s',r,l}_{(i,j)}(t)\}$ and $\{\Xi^l_t\}, \forall t \in [0,T]$ as inputs to GAMS/NLPEC solver. Solve the optimization problem in GAMS to obtain:

$$z^l, TTT^l, p^{s',r,l}_{(i,j;h)}(t), v^{s',r,l}_{(i,j)}(t), q^{u,s',r,l}_{(i,j)}(t), q^{d,s',r,l}_{(i,j)}(t), \pi^{s',r,l}_{(i,j)}(t)$$

Step 3: Check for convergence. A convergence measuring term δ^l based on mean absolute percentage deviation (MAPD) is used to check model convergence.

$$\delta^l \triangleq \frac{100\%}{\bar{l}} \sum_{l^*=1}^{\bar{l}} \frac{|z^l - z^{l-l^*}|}{z^{l-l^*}}$$

$\bar{l}$ is a predefined fixed number. It controls how many previous iterations the MAPD takes into consideration. If $l < \bar{l}$, skip Step 3 and go to Step 4. If MAPD $\delta^l \leq \overline{\delta}$,

where $\overline{\delta}$ is the convergence threshold, stop the iteration algorithm. Otherwise, check if iteration number $l \geq l_{max}$, Stop iterative algorithm.

Step 4.1: Update transitive closure set. Use $q_{(i,j)}^{u,s',r,l}(t)$ and $\overline{Q}_{(i,j)}$ as input to Algorithm 6 in Appendix B.2 to calculate transitive closures: $\{\Xi_t^l\}, \forall t \in [0, T]$.

Step 4.2: Update route travel time approximation coefficient. In Step 2, inflow and exit flow to each link (i, j) have been calculated. As Figure 5.7 shows, link travel time $\tau_{(i,j)}(t)$ can be derived using cumulative curves. A detailed explanation and illustration can be found in Yu et al. (2018).

$$\tau_{(i,j)}(t) = \frac{\sum_{k_0=1}^{m} p_{(i,j)}^{s',l}(t)_{k_0} \cdot n_{(i,j)}^{l}(t)_{k_0}}{p_{(i,j)}^{s',l}(t)}$$

Then use the following equation to calculate new $\theta_{(i,j)}^{s',r,l}(t)$:

$$\theta_{(i,j)}^{s',r,l}(t) = \left(\tau_{(i,j)}(t) - \tau_{(i,j)}^{0}\right) \cdot \frac{\overline{C}_{(i,j)}}{q_{(i,j)}^{d,l}(t + \tau_{(i,j)}^{0})} \cdot \left(\frac{\pi_{(i,j)}^{s',r,l}(t + \tau_{(i,j)}(t)) - \pi_{(i,j)}^{s',r,l}(t + \tau_{(i,j)}^{0})}{\tau_{(i,j)}(t) - \tau_{(i,j)}^{0}} + 1 \right)$$

Step 5 : Set $l = l + 1$. Go to Step 2.

5.8 Numerical results

5.8.1 A two-by-two grid network

Figure 5.9 is a 2x2 grid network. Nodes are connected by bi-directional links. There are 4 sets of ODs (see Table 5.2). The node numbers in parenthesis are dummy ODs. For instance, node 5 is a dummy node connected to node 1 via dummy links $(5, 1)$ and $(1, 5)$. Link flow capacity and free flow link travel time are displayed along with the links shown on Figure 5.9.

		Destinations			
		1(5)	2(6)	3(7)	4(8)
Origins	1(5)	-	-	-	30
	2(6)	-	-	15	-
	3(7)	-	15	-	-
	4(8)	30	-	-	-

Table 5.2: OD for the 2x2 grid network: high demand case.

		Destinations			
		1(5)	2(6)	3(7)	4(8)
Origins	1(5)	-	-	-	15
	2(6)	-	-	15	-
	3(7)	-	15	-	-
	4(8)	15	-	-	-

Table 5.3: OD for the 2x2 grid network: high demand case: low demand case.

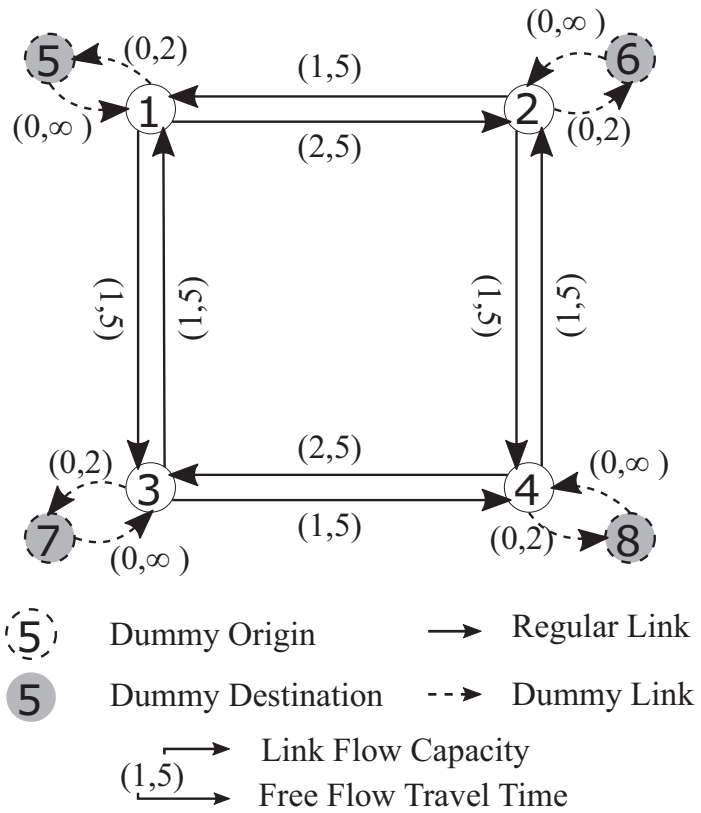


Figure 5.9: A 2x2 grid network.

Travelers in the network will seek their own shortest paths to destinations. In this particular network with all link travel time shown in Figure 5.9, the "possible" flow pattern is shown in Figure 5.10. Different colors represent traffic flow between different OD pairs. If everyone in the network follows their desired paths, this will lead to a deadlock cycle, $1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1$, in the network.

On the contrary, if everyone in the network follows the *OE-routing rules* specified in

Section 5.3, traffic flows will be re-distributed in the network as Figure 5.11 shows. As *OE-routing rules* specifies, EN turn and ES turn are not permitted. Hence, travelers are not allowed to make turn "$5 \rightarrow 2 \rightarrow 4$" or turn "$3 \rightarrow 4 \rightarrow 2$" when traveling in the network. Therefore, travelers from node 3 to node 2 can only travel via node 1. Traffic demand between dummy origin node 7 and dummy destination 6 now only travels via path "$7 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 6$". Traffic flows between the rest OD pairs are not affected. Note in Figure 5.11 that the deadlock cycle in the middle is now resolved.

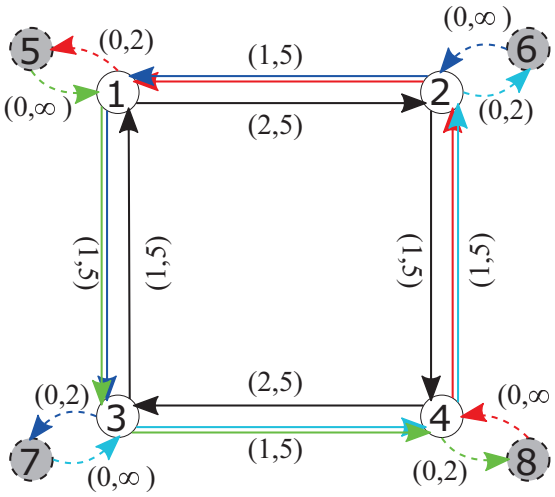


Figure 5.10: Gridlock in a 2x2 grid network.

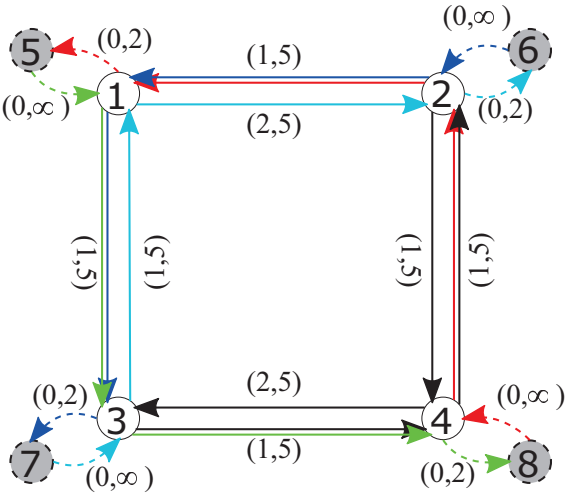


Figure 5.11: OE routing in a 2x2 grid network.

To assess how the performance of the proposed DLA model varies under different levels of traffic demand, in this numerical study section, two scenarios with different traffic demand levels are tested against the DLA model proposed: 1) a high traffic demand case representing congestion scenario where heavy queues are built across the network (Table 5.2); 2) a low traffic demand case where the majority of the links are free of queues (Table 5.3). The two scenarios share the same network configuration. With in each scenario, various sub-scenarios consist of different percentages of *OE Routing vehicles* are tested.

5.8.2 Performance under high-demand scenario

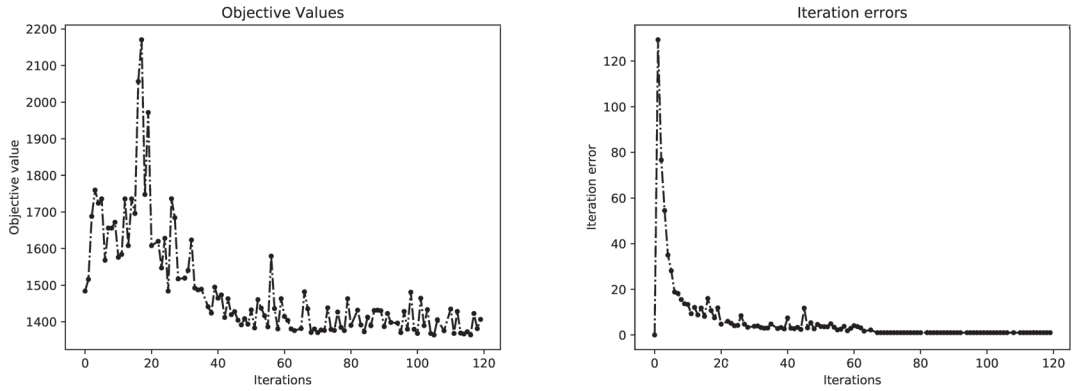
In this study, an optimal *OE Routing* ratio is solved using the procedures described in Section 5.7.5 with traffic demand specified in Table 5.2. This testing scenario is designed for a relatively high demand case, i.e. high level of congestion or long queues are expected to occur within the network. In order to create such a congested scenario, several ways

besides setting higher OD demands in 5.2, are carefully designed to accomplish that. They will be explained in details below.

First of all, dummy destination links in Figure 5.9 has a flow capacity of 2, meanwhile the regular links have 5 or 10 link flow capacities. Each dummy link, in this case, becomes a bottleneck in the system. This is different with the setups in Ma et al. (2014) and Yu et al. (2018) where dummy links have infinity flow capacities. There are two reasons to have dummy links with small flow capacities: 1) this setup is closer to real world scenario. Dummy destinations are like parking structures in real world with dummy links as entrance links to the parking structure. In reality, the entrances have finite flow capacity. And this capacity is often smaller than that of a regular road. 2) In Yu et al. (2018) they have traffic signal light to control flows at intersections. This reduces the effective flow capacity of a link controlled by signal light based on the corresponding green split ratio. Thus, inflow capacity sometime is greater than exit flow capacity. This makes it possible to build up downstream queues on links. In this study, however, there is no traffic light to control flows. Exit flow capacity is equal to inflow capacity of a link. If downstream link (dummy link) has infinity flow capacity, any flow flowing from a regular link that connects directly to dummy link to that dummy link will get "absorbed" by the dummy link. Thus, it is unlikely to have downstream queue built up on that regular link. In addition to the low flow capacities on dummy destination links, t_L is set to be 5 so that all traffic from origins should depart before $t = 5$. This forces traffic to load onto the regular links in a relatively short period of time to create queues or congestion.

5.8.2.1 Convergence test

In this part, we evaluate how the DLA model converges using the iterative procedures described in 5.7.5. The optimal DLA model is solved iteratively until convergence. Figure 5.12a and Figure 5.12b show the convergence curve of the objective value and iteration errors over iterations, respectively. The model converges quickly in the first 60 iterations. After iteration 60, the objective value does not change too much with a fluctuation range between 1360 and 1450. At iteration 117, the optimal objective value is 1360.



(a) Objective Value (b) Iteration Error

Figure 5.12: Objective value and iteration error over iterations.

5.8.2.2 Queue statistic vs time

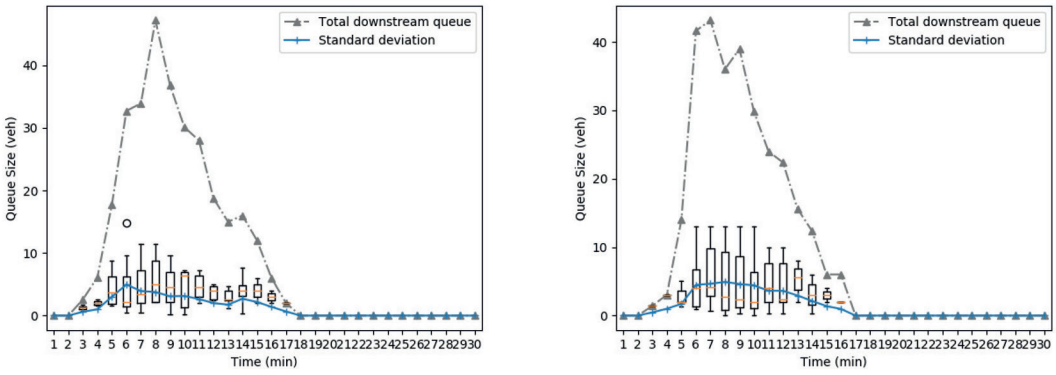
In this part, we test the proposed DLA model against two different scenarios: Scenario (I) optimal DLA under DUE constraints, and Scenario (II) DUE with 0% *OE Routing vehicles*. Figure 5.13a and Figure 5.13b show the total number of downstream queues in the network at each time step for Scenario (I) and Scenario (II), respectively. In Scenario (I), optimal solution of the DLA model is reached when the percentage of *OE Routing vehicles* (γ) equals 33.33%. Table 5.4 shows the objective values, total number of queues and total travel time for the two scenarios. Objective values and total number of queues are smaller in Scenario (I) than Scenario (II), while total travel time is almost the same in the two scenarios. Overall, the objective value is reduced by approximately 12% in Scenario (I) compared to Scenario (II). Total number of queues is reduced by approximately 27%. The result indicates that by deploying an appropriate percentage of *DLA-routing vehicles* in the entire vehicle population (in this numerical testing case, this optimal percentage is 33.33%), the network can benefit from fewer queues and less travel time.

Table 5.4: Objective value, total number of queues and total travel time.

	Objective value	Total queue	Total travel time
Scenario I	1052	330	722
Scenario II	1190	450	740

In addition to the smaller total number of queues, smaller queue fluctuations across

different links are observed in Scenario (I). As shown in Figure 5.13, the variation of average number of queues across all links at each time step in Scenario (I) is smaller than that in Scenario (II). This can be observed from both the boxplot and the standard deviation curve. This means that with an optimal percentage of *DLA-routing vehicles* equal to $\gamma = 33.33\%$, the traffic is more evenly distributed across different links comparing to the scenario with no *DLA-routing vehicles*. With more balanced queues across the network, link queue capacities can be better utilize as links with shorter queues "accommodate" queues from other links with longer queues. This reduces the deadlock potential of links in the network. The opposite case, however, where queues are more scattered and varying significantly across different links, fails to spread queues to links with shorter queue, leaving the extra queue capacity of those links underutilized. With longer queues, links have greater deadlock potential. Thus, the network in this scenario is more vulnerable to deadlock.



(a) Scenario I: optimal $\gamma = 0.3$. (b) Scenario II: $\gamma = 0$.

Figure 5.13: Total downstream queue at each time step.

Table 5.5 shows that with certain percentage of *DLA-routing vehicles* the average number of queues in the network can be reduced. Here, the number of downstream queues are averaged over all links and the entire time horizon. With fewer queues in the network, it is less likely to have deadlocks. Similar to the queue balancing across links discussed in the last paragraph, queues are also balanced temporally in Scenario (I) as the standard deviation of the downstream queues is smaller in Scenario (I) than that in Scenario (II). High queuing demands are pushed to other time steps with lower queuing demands. In

this way, queue capacities at different time steps are better utilized. By smoothing queues out along the time horizon, it prevents possible queue spikes. Hence, deadlock possibility can be reduced.

Table 5.5: Mean and standard deviation of downstream queue.

	Mean	Standard deviation
Scenario I	4.3	3.2
Scenario II	5.4	4.2

5.8.2.3 Performance with different market penetration rate

In addition to the 0% *OE Routing vehicles* case and optimal case studied in the previous sections, this part applied the proposed DLA model under various percentages of *OE Routing vehicles* ranging from 0% to 100%, with 10% increment between each two cases (except for the optimal case which is 33.33%). We evaluate how the *DLA-routing rule* performs under different penetration market rate of *DLA-routing vehicles*.

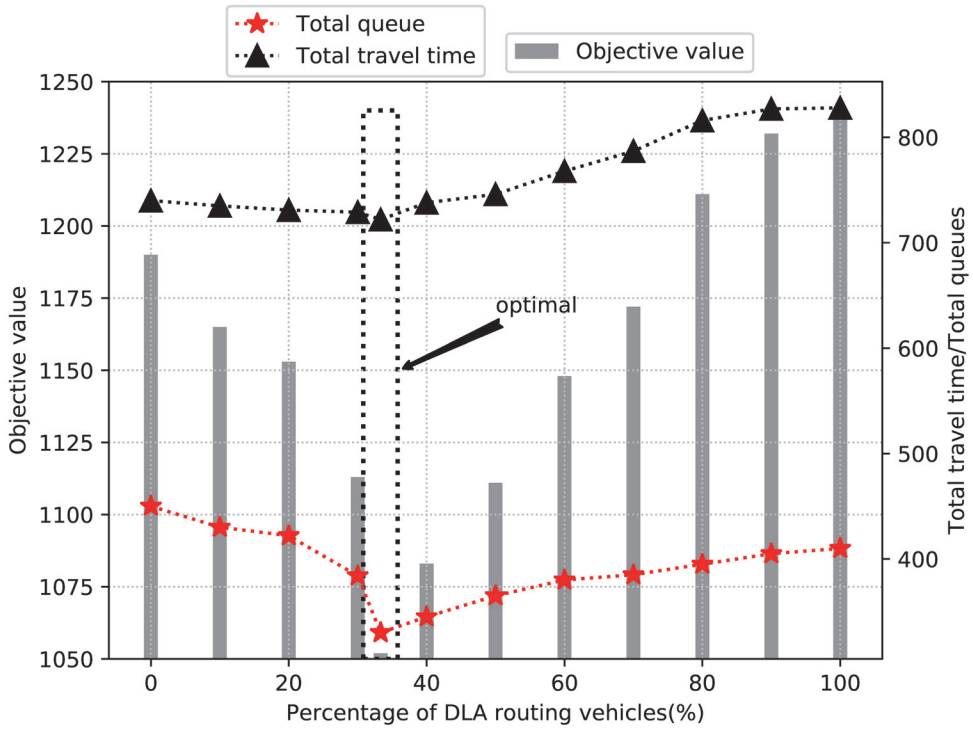


Figure 5.14: Objective values of different cases with different percentage of DLA-routing vehicles.

The results in Figure 5.14 show that the objective value first decreases when γ increases from 0% to 33.33%, and then increases when γ goes from 33.33% up to 100%. The dashed box is for the optimal case. The total number of queues follows the same pattern as that of the objective value. The total travel time, however, does not have the "V" shaped pattern when percentage of *DLA-routing vehicles* increases from 0% to 100%. Total travel time decreases slightly from 740 to 722 when γ is between 0% and 33.33%, then increases monotonically, from 722 up to 828, as percentage of *DLA-routing vehicles* increases from 33.33% to 100%. When γ is between 0% and 33.33%, total travel time rises quickly when γ increases; after 33.33%, the increase of total travel time is less dramatic. The objective value under 100% *DLA-routing vehicles* is greater than 0% *DLA-routing vehicles* case, largely because of the extra detours made by the *DLA-routing vehicles* which oftentimes have longer travel time compared to regular routes. Total number of queues is smaller when there are 100% *DLA-routing vehicles*, thanks to the *Odd-Even Routing rules*. However, the reduction of total number of queues, in the case, is not large enough to offset the increase of total travel time, resulting in a higher objective value. This confirms our initial analysis in Section 5.2.

To sum up, in high traffic demand scenario, by introducing an optimal of 33.33% *DLA-routing vehicles*, the objective value can be reduced by approximately 5%, with an approximate 27% of queue reduction which significantly reduces the probability of deadlocks.

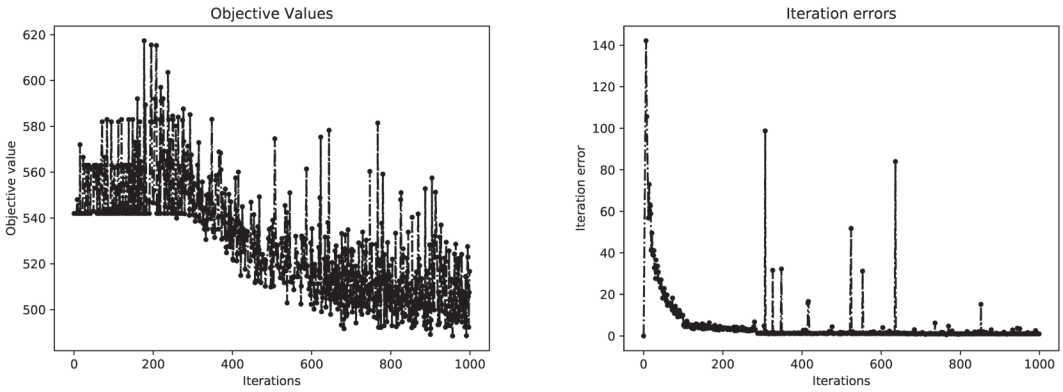
5.8.3 Performance under low demand scenario

In this part, the DLA model is tested with a lower demand to evaluate how well the model performs in responses to low traffic demand. The OD demand profile is in 5.3 (compared to high demand case, OD demand between dummy node 5 and dummy node 8 is reduced from 30 to 15). The optimal solution of DLA model under the low demand is reached when $\gamma = 0$, which means the *OE Routing* strategy could potentially deteriorate system performance in low demand scenarios. When traffic demand is low, traffic on road is not congested, i.e. queues are small or non-existent. It is unnecessary to re-route traffic to other routes to reduce congestion as the DLA model intends to accomplish. On the contrary, by re-routing traffic using *OE-routing* strategy in low demand scenarios, it would induce extra travel time for those re-routed traffic due to the detours from *OE Routing*. Therefore, *OE-routing* could increase the total travel cost in this case while provide almost

no improvement to the deadlock potential. The rest of this subsection will assess other metrics of the DLA model under low demand.

5.8.3.1 Convergence

As shown in Figure 5.15, the DLA model converges nicely after iteration 600 with a few random spikes. The optimal solution is reached at iteration 950 with an optimal objective value of 465.



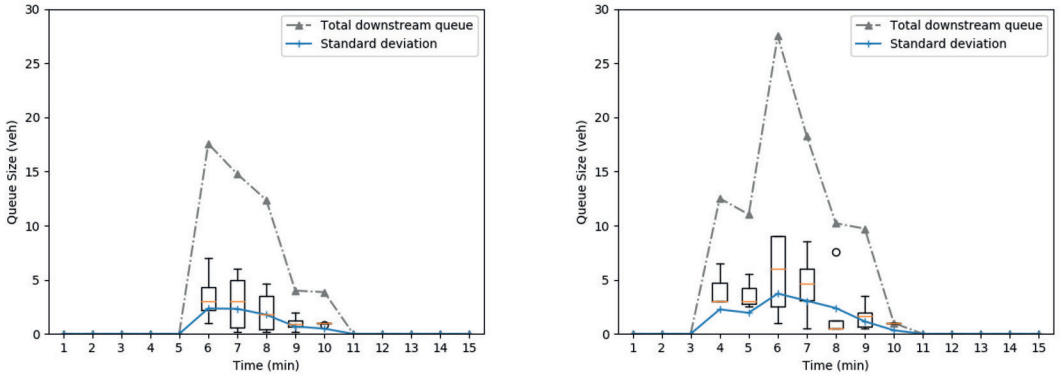
(a) Objective Value (b) Iteration Error

Figure 5.15: Objective value and iteration error over iterations.

Comparing to high demand case (Figure 5.12), convergence speed becomes slower in low demand. One possible explanation is that in low demand scenario traffic has more freedom to choose its route to destination, i.e. the feasible solution region to the DLA model in each iteration is larger. Therefore, the time and effort to reach the optimal solution would increase, which translates to more iterations to convergence. Coincidentally, the iteration process to convergence shows more fluctuations than that in high demand case (Figure 5.12).

5.8.3.2 Compared to Cases with OE Routing Vehicles

As mentioned in the precious section, the optimal solution is reached in low demand case when $\gamma = 0$. For comparison purposes, two scenarios are tested and compared: Scenario (III) optimal DLA under DUE constraints, and Scenario (IV) DUE with 33.3% *OE Routing vehicles*. Figure 5.16a and Figure 5.16b shows the downstream queue statistics for these two scenarios, respectively.



(a) Scenario III: optimal $\gamma = 0$.

(b) Scenario IV: $\gamma = 0.3$.

Figure 5.16: Total downstream queue at each time step.

In contrast to high demand scenarios as discussed in Section 5.8.2, queues are smaller in Scenario (III) where there are no *DLA-routing vehicles* than Scenario (IV) where there are 33.33% *DLA-routing vehicles*. Furthermore, the standard deviation of downstream queues is smaller in Scenario (III) compared to that in Scenario (IV). The reason that makes it different with high demand scenarios is that in low demand scenarios queue levels are much lower than high demand scenarios. It is less necessary to do deadlock prevention via *DLA-routing*. On the contrary, introducing *DLA-routing vehicles* will have very limited improvement in terms of the total number of queues, or even will have negative impacts (see the results of performance under different market penetration rate in following section). Meanwhile, the *DLA-routing vehicles* traveling in the network will promote longer travel time compared to regular vehicles due to the extra detours they make. As a result, *DLA-routing vehicles* yields no or negative benefits to the system performance in low demand scenarios.

The effect of the market penetration rate of *DLA-routing vehicles* on the system performance is also evaluated with the value γ ranging from 0% to 100%. See Figure 5.17. The optimal case is annotated with a dashed box.

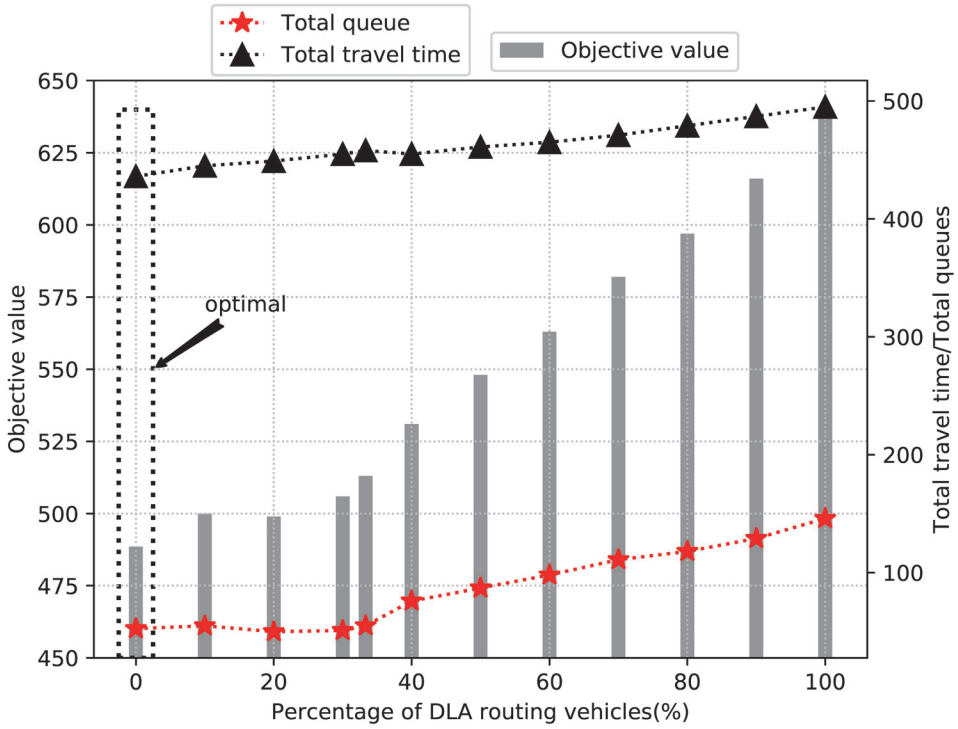


Figure 5.17: Objective values of different cases with different percentage of DLA-routing vehicles.

The objective value and the total travel time increase monotonically as the percentage of *DLA-routing vehicles* varies from 0% to 100%. Between $\gamma = 0\%$ and $\gamma = 30\%$, the increment speed of objective value is relatively moderate. The increment speed of objective value rapidly jumps up when γ is larger than 30%. The increment speed of total travel time is relatively steady compared to that of objective value. The total number of queues, however, does not monotonically increase as γ increases. Between $\gamma = 0\%$ and $\gamma = 30\%$, the number of queues are approximately unchanged with very limited fluctuations. After $\gamma = 30\%$, the number of queues gradually picks up.

To sum up, in low demand scenarios, *DLA-routing vehicles* provide zero or even negative benefits to the overall transportation system. The higher percentage of *DLA-routing vehicles* results in a worse overall performance. Table 5.6 is the summary of improvements of objective value, total queues and total travel time between $\gamma = 0\%$ and $\gamma = 100\%$.

Table 5.6: The performance improvement of scenario with $\gamma = 0\%$ over $\gamma = 100\%$.

Objective value	Total queues	Total travel time
24%	65%	12%

5.9 Conclusions and future work

In this chapter, we formulated a link-based Deadlock Avoidance model (or DLA model) to minimize the probability of having deadlocks in the network. City road networks are sometimes vulnerable to deadlocks, especially for those with heavy traffic demands frequently. A strategy that prevents or reduces the possibility of deadlock occurrence can be extremely helpful to improve the city core traffic. The DLA model in this chapter incorporates the idea of deadlock free routing algorithm by restricting certain movements at certain locations in the network for a special type of vehicles, i.e. *DLA-routing vehicles*. By optimally selecting the number of *DLA-routing vehicles* in the network, the DLA model is able to reduce the possibility of having deadlocks while still maintaining reasonable average total travel cost.

Numerical studies for both high and low demand cases have been designed, tested and compared. The results show that the proposed DLA model performs well in congested, high demand scenarios, reducing the total number of queues in the network by 27%. Therefore, the deadlock potential is lowered correspondingly. The optimal percentage of *DLA routing vehicles* in high demand case is equal to 33.33% which implies that by introducing only 33.33% *DLA routing vehicles* in the whole vehicle population, it is possible to significantly reduce the possibility of having deadlocks in the network. In addition to reduction in total number of queues, there are two other unique characteristics that the DLA model exhibits. Fisrt, the DLA model tends to spread queues evenly among different road links to utilize queuing capacity in the network as fully as possible. Furthermore, the DLA model also spreads out queues over temporally to make queues much more evenly distributed over time compared to cases without *DLA routing vehicles*. With these two queue smoothing, the DLA model is able to further reduce the deadlock potential. In low demand scenario, however, having *DLA routing vehicles* does not provide much benefits. Instead, the extra detours made by *DLA routing vehicles* contribute to total travel cost

increasing. Numerical test confirms that within the low demand scenario, the optimal solution is with 0% *DLA routing vehicles*. The more *DLA routing vehicles* in this situation, the worse the overall performance will be.

To sum up, the DLA model proposed in this chapter provides an efficient and effective approach to reduce the deadlock probability. Networks with high congestion benefit the most from the DLA model with respect to deadlock potential reduction. While the model of DLA is self-explanatory and working, to implement the proposed *DLA routing* scheme in real world may not be as straightforward. Traditional human-driven cars lack the capability to talk to traffic controllers. As a consequence, it is difficult, if not impossible, to completely prevent traditional human-driven cars from making certain turns that are prohibited by the *OE Routing rules* through imposing a turning penalty as what the model does. However, with V2I technology and autonomous driving technology making great break-through in the near future, monitoring and controlling vehicles will become easier and easier. Thus, imposing a special turning penalty to some special vehicles, i.e. *DLA routing vehiles*, will become a feasible task.

There are a few possible extensions to the current work. Currently, there is no traffic signal control modeled in the framework. In the future, signal control can be added into the optimization model to make it more realistic. Another possible extension could be dynamically changing the market penetration rate of *DLA routing vehicles* in response to the traffic demand and overall network conditions, instead of one fixed percentage in the current model.

Chapter 6

An Application of DTR in Parking Search Problem

6.1 Introduction

The difficulty of finding parking spots for travelers, as well as the congestion and other externalities produced by the parking process, have become a major issue in many metropolitan areas. Travelers usually have to cruise on streets to search available parking facilities due to the lack of information on the parking availability ahead of time. Travelers search for parking suffer additional delay, and the search process itself causes more congestion and environmental impacts on the streets (Caicedo, 2010; Waterson et al., 2001; Höglund, 2004; Guo et al., 2013).

Parking availability used to be visible only at the facility so that travelers have to be physically present at the parking facilities to check parking availability, which was either shown on board signs as "full or available", or digital signs with the number of available spaces. Once travelers find a garage is full, they have to keep driving to the next parking garage, which introduces extra trips in the traffic networks. Even a parking garage is not full, drivers may encounter difficulties of finding a space and experience longer cruising times within the parking facilities due to higher occupancy. Without the knowledge of parking availability before the arrivals at a parking facility, drivers cannot make good parking decisions and usually, suffer from extra trips and more time for finding available parking spaces.

In this chapter, we propose a parking guidance system to assist travelers to make better parking and routing choices to reduce the travel cost for each traveler, by utilizing both online parking information and real-time and historical link travel time information. In the last decade, enabled by smart infrastructure and smart phones, travelers are able to receive the information of the availability of parking spaces via websites or apps without being at the parking facilities. In this way, travelers can choose from the available parking garages, and avoid the extra trips for parking searches. Along with parking space availability, the online information system can also release parking charges to the drivers, so that travelers can calculate their expected costs for choosing certain parking facilities. Besides the parking cruising time and monetary costs, the cost of travel time from the origin to the parking facility also influence drivers' decisions on parking selection. Rational travelers tend to minimize their own costs by choosing proper routes and parking garages. Chai et al. (2017) proposed a joint dynamic traffic routing and adaptive signal control model, which provided dynamic routing guidance to travelers and help them reach their destinations quicker. Simulation results show that the joint routing and signal control model reduces the average travel time by 15-20%. The dynamic routing component of that model is chosen in this study to guide drivers to travel from their origins to their chosen parking garages.

The parking search process is a well-studied topic in the literature (Thompson and Richardson, 1998; Bellés et al., 2007; Benenson et al., 2008; Boyles et al., 2015). Network models with parking searches are generally categorized as analytical or simulation-based. Analytical models usually apply equilibrium traffic assignment as the foundation with parking as an integral component. Parking facilities are treated as additional links and nodes in the studied networks, and the traffic is usually modeled as system variables. One advantage of analytical models is that optimality conditions and equilibrium analysis can be studied, and analytical relations among system variables can be derived. For instance, Boyles et al. (2015) proposed an equilibrium formulation to incorporate parking searches into traffic network assignment models. They considered a stochastic network with uncertainty of parking availability on certain network links, where on-street parking is associated with availability probabilities. Travelers that seek parking are parked propor-

tional to the corresponding probability. However, it is usually quite difficult to characterize dynamical behavior of driver's route and/or parking destination choices in analytical approaches. The simulation models, on the other hand, can explicitly apply route and parking choice behavioral rules to travelers/agents (Caliskan et al., 2007; Dieussaert et al., 2009). In other words, the simulation models are amenable to modeling various types of parking choice behaviors. For instance, a driver may have a fixed choice of parking garage if it is available. When it is fully occupied and becomes unavailable, the driver cruises to other parking garages (Leurent and Boujnah, 2012), and this process can be easily modeled in simulations. A driver in the simulation can also cruise without a pre-set parking destination, and stop searching once an available parking garage is reached. Drivers with more information on parking availability may behave in a more complex manner, which is modeled in this chapter. Simulation approaches can explicitly show how the drivers would act with exogenous information, and it is relatively easy to test the performance of different parking policies. In this study, we adopt the simulation approach to evaluate how our proposed parking guidance system can affect both individual and network level performance in terms of travel costs and utilization of parking facilities.

In this study, we propose a parking guidance system with dynamic parking destination choice and routing model (see Figure 6.1). The routing and parking searches are modeled as a time-dependent process, where the parking and travel time information is updated periodically, and new recommendations for a parking location and the route to reach it are provided to drivers based on the updated information. It is noted that link travel times are treated as stochastic so that variations of travel times due to demand fluctuations and measurement errors can be modeled in traffic routing. The dynamics of choice switching have been well studied for the day-to-day traffic dynamics. Studies on dynamic choice switching usually focus on the path choices, see a comprehensive review in Yang and Zhang (2009), with some recent exceptions on link choices, such as He et al. (2010) and Guo et al. (2015). Among various dynamic choice switching processes, the proportional-switch adjustment process proposed by Smith (1984) is considered as the most intuitive formulation, and many studies are developed based on this pioneering work in Smith (1984), such as Smith and Wisten (1995), Huang and Lam (2002), Peeta and Yang (2003),

Smith and Mounce (2011), Mounce and Carey (2011), Zhang et al. (2015), to name a few. The basic concept of the proportional-switch adjustment process is that the percentage of population switching from one choice to another is proportional to the cost difference between the choices before and after the switch. Apparently, for an individual driver, the parking choice switching is not deterministic, and the switching probability can be derived from the cost difference of the corresponding choices.

The contributions of this work are: (1) we propose a dynamical parking destination choice switching model that can balance the parking demands on different parking garages and reduce the average cost of the travelers; (2) we develop a parking guidance system that combines the dynamic traffic routing with the parking destination choice switching model, and evaluate its performance under different levels of traffic demand and market penetration rates. The simulation results show that the proposed parking guidance system can significantly reduce the system total costs and balance the parking demand over multiple parking garages under various scenarios, especially for scenarios with high traffic demand. The rest of the chapter is organized as follows: Section 6.2 states the problem, describes the routing algorithm and gives the general dynamic parking garage choice switching model. Simulation setup and numerical results are presented and discussed in Section 6.3. Section 6.4 summarizes the results of this work and discusses its future extensions.

6.2 Modeling of the system

In this study, each driver equipped with the parking guidance system is able to make real-time decisions of their parking destination, as well as the routes from the current location to the chosen parking destination during their trips. Through the parking guidance system, drivers have access to the dynamic online updated traffic information, including expected link travel times, parking garage occupancy, expected incremental usage of the garage in the near future and so on. Based on the information received, travelers evaluate the cost of each available choice: travel time, parking fee, inside parking garage searching cost and walking cost respectively (See Figure 6.1). After obtaining cost information, drivers can choose their parking destination by simply choosing the parking garage with

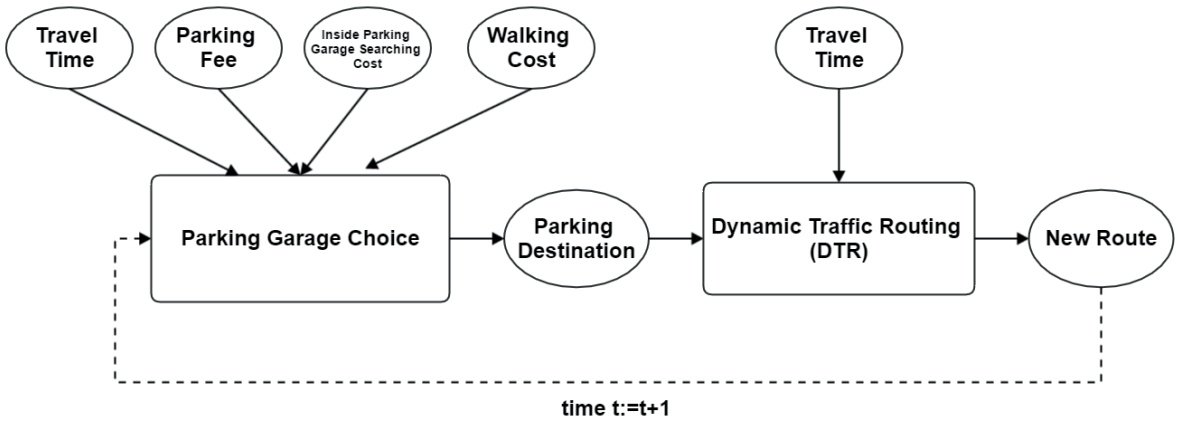


Figure 6.1: System framework.

the lowest cost among all. Then for each driver with a destination chosen, dynamic traffic routing is performed to identify the least expected cost route for the driver. Drivers can receive online information during the trip so that the above-mentioned decision-making process is conducted every time step iteratively. Figure 6.1 shows the framework of proposed decision-making process.

We here list the notation used throughout this chapter in Table 6.1. The network is composed of all nodes in N and O and the bi-directional links connecting them so that the network is fully connected.

6.2.1 Costs associated with parking destination choices

In this study, we apply a parking destination choice model as an analogy of the dynamic choice swapping mechanism in Smith (1984). Note that in Smith (1984), the choice swapping was proposed to determine the change of route choices, while in this study, the choice swapping is for the change of parking destination choices. The route choices in this study do not involve the Smith's choice swapping mechanism; rather, the dynamic route choices are determined by the hyper-path based stochastic shortest path algorithm proposed in Chai et al. (2017). From the perspective of destination-specified vehicle groups, i.e., destination-specified traffic flows, a proportion of each group may switch their destinations if the cost of the original destination is higher than that of the other destination. From the perspective of a single driver, such a proportion means the probability of switch-

Table 6.1: Notations

Symbol	Definition
$i \in N$	Regular Node i in the network
$o \in O$	Origin Node o in the network
$t \in \{0, 1, 2, \ldots, T\}$	Time Step within the study time period $[0, T]$
$p, q \in P$	Parking garages
$OC_p(t)$	Expected occupancy of parking garage p at time t
$C_p(t)$	Generalized parking cost for parking in parking garage p at time t
$F_p(OC_p(t))$	Parking fee($); Monetary cost for parking garage p at time t (exogenously given)
$S_p(OC_p(t))$	Cost ($) for searching parking in parking garage p, given the occupancy as $OC_p(t)$
W_p^d	Constant walking cost ($) from parking garage p to final destination d
$x_p^i(t)$	Number of vehicles at node i (including vehicles traveling on links to node i) choosing parking garage p as the parking place at time t
$y_p^i(t)$	Generalized total cost for vehicles at node i choosing parking garage p as the parking place at time t
$\tau_p^i(t)$	The (discrete) expected travel time (converted to monetary cost) from node i to parking garage p at time t (exogenously given)
ϵ_C	Initial parking fee
ϵ_S	Expected inside parking garage searching cost when the parking garage is empty

ing his or her original destination to the new one. Since such proportions are determined by the cost difference of every two destination choices, in the following, we show how to calculate the costs associated with the parking destination choice.

As shown in Figure 1, a driver considers the summation of following expected costs

for selecting a certain parking destination.

- Expected travel time cost $\tau_p^i(t)$ - the time spent on road before reaching the selected parking garage.

- Expected searching cost inside a parking garage $S_p(OC)$ - the corresponding monetary cost (converted using the value of time) of the time spent in the parking garage before an available parking space is found. In some literature, it is also known as the Cruising Cost (Qian and Rajagopal, 2014a), which is a function of the garage occupancy. In Axhausen et al. (1994) they proposed that the searching time is a function of garage occupancy in an off-street parking garage as:

$$S_p(OC) = \frac{\epsilon_C}{1 - \frac{OC}{K}} \tag{6.1}$$

  where $S_p(OC)$ is the searching cost, OC is the occupancy, K is the capacity of the parking garage, ϵ_C is the average time spent on searching an available space in an empty parking garage.

  This is a hyperbolic function. When the parking occupancy approaches 100%, the searching cost becomes extremely large so that it penalizes vehicles from parking in this parking garage. Such a setting fits the reality, as the time needed to find an available parking slot increases dramatically when the parking garage is almost fully occupied.

- Expected parking fees $F_p(OC)$ - the monetary cost charged by the parking service provider. As dynamic parking pricing schemes are emerging to better coordinate the parking demand and supply in some pioneering projects such as SFPark, we define the parking fee as a function of current occupancy of the parking garage. In this study, we apply a hyperbolic function, which is similar to the in-garage searching time function.

$$F_p(OC) = \frac{\epsilon_S}{1 - \frac{OC}{K}} \tag{6.2}$$

Note that the function of parking fee can be in other forms, and the current one is used for illustration purposes. Constant parking fee, as a special case, can be implemented as well when needed. There are many existing studies (He et al., 2015; Qian and Rajagopal, 2014a,b) on the parking pricing strategies. However, it is beyond the scope of this study. Therefore, we simply pick one parking pricing scheme in this chapter and leave the study on parking pricing strategies as a future research topic.

- Walking cost W_p^d - the corresponding monetary cost (converted using the value of time) of the constant time spent in walking from the parking garage p to the final destination d.

Some discussion on the choice of cost functions

Under the current parking fee function discussed above the parking fee may become large when the occupancy approaches its capacity. Such construction of the parking fee is not the same as the contemporary parking fee policies at most parking facilities in the real world. However, the parking fee function is purposely constructed as the current form. The first reason is that mathematically we would like to guarantee that the parking facilities cannot be full by setting such exceedingly large cost when the parking occupancy approaches its capacity. The second reason is that we would like to have similar function types for the searching cost and the parking fee cost, so that their proportionality helps us evaluate the generalized cost function. Since the searching cost and parking fee cost share the same structure and are proportion to each other, neither of them would dominate its influence on the parking choices.

The current settings are used for better illustration. As shown in the analysis and the numerical results in Section 6.3, under such settings, drivers without the searching for parking guidance system would suffer from high costs, while drivers with such a system can reduce their generalized costs. It is straightforward that any monotonically increasing function can be used to replace the current cost functions for searching and parking fee costs, as long as the travelers' choices on highly occupied parking facilities are discouraged.

The generalized parking cost for parking garage p at time t is the sum of the monetary

cost (parking fee), the inside parking garage searching cost and the walking cost to the final destination.

$$C_p(t) = F_p(OC_p(t)) + S_p(OC_p(t)) + W_p^d \tag{6.3}$$

Then the generalized total cost y_p^i is the summation of the generalized overall parking cost C_p and the expected travel time τ_p^i (See Equation (6.4) below). The model in our work is built on the assumption that travelers make their choices to minimize the (expected) total cost. When parking destination is determined, only travel cost τ_p^i is used to dynamically route the travelers to the parking destination.

$$y_p^i(t) = \tau_p^i(t) + C_p(t + \tau_p^i(t)) = \tau_p^i(t) + S_p(OC_P(t + \tau_p^i(t))) + F_p(OC_p(t + \tau_p^i(t))) + W_p^d \tag{6.4}$$

In the above cost function, parking occupancy at "future" time instants $(t + \tau_p^i)$ are involved. An estimation method based on current destination choices of all vehicles in the network is shown in Equation (6.10). Other than instantaneous ones, using estimated parking occupancy may help to avoid the "all-or-nothing" assignment and thus reduce possible oscillations in the system.

At each time step, the expected travel time $\tau_p^i(t)$ from any node to any parking garage can be calculated using the dynamic traffic routing algorithm and historical link travel times. The detailed calculation method can be found in Chai et al. (2017) and thus is omitted in this chapter due to space budget. We only outline the routing process in the following subsection.

6.2.2 Dynamic traffic routing

In the real world, traffic conditions change constantly. It is difficult to know ahead of time link travel time, link flow and other information accurately. Traditionally, travelers can only rely on their experience or very limited information about the current traffic state to make their route choice decisions. When a route is planned, the traveler is not able to re-plan a new route easily. Usually, the pre-planned route is not the best route, especially when unexpected incidents occur on the route. This may increase the total travel times of impacted travelers significantly. With the rapid growth of modern technologies, now it is

possible for the travelers to acquire recent traffic information from various sources before and during traveling. In this way, travelers can re-plan their routes accordingly to avoid potentially congested roads and thus reduce their travel times.

Chai et al. (2017) developed a joint model of dynamic traffic routing and adaptive signal control. In their work, a hyper-path based stochastic shortest path algorithm was used, and different traffic signal control methods were proposed and tested. We use the Dynamic Traffic Routing (DTR) algorithm from Chai et al. (2017) to calculate a hyper-path for each traveler, and use that hyper-path to dynamically route vehicles to the shortest routes to destinations. The inputs of the DTR algorithm are traffic signal plans and link travel time distributions.

Once a hyper-path is calculated, travelers follow the routing suggestion of the hyper-path thereafter until a new hyper-path is calculated (depends on how frequency the travelers update their route decision, the hyper-path can be re-calculated as frequently as every second as long as the computation power allows).

To fully take advantage of the online traffic information, a dynamic link travel time updating scheme is used to integrate the most up-to-date traffic information into the model. Every link maintains a set of possible link travel time realizations as well as their corresponding probabilities. After a certain period of time, every link will have a set of new link travel time realizations (they can be new values or have some overlapped part with the initial set).

For detailed formulations and algorithms for DTR and dynamic link travel time updating scheme, please refer to the work by Chai et al. (2017).

6.2.3 Dynamic parking garage choice switching model

In Smith (1984), they proposed a deterministic flow switching model which characterizes the flow swapping between two different routes to be proportional to the cost difference between the two. In this study, we propose a parking destination choice model as an analogy of the dynamic choice swapping mechanism in Smith (1984). Note that original Smith's swapping mechanism is for day-to-day dynamics of a group of drivers, and they can switch their route choices every day according to the costs of routes. In this study, the choice swapping inheres similar pattern from Smith's swapping, but the concept is

distinguishably different. There are two significant differences. The first one is that the choices are on parking destinations in this study, while the original Smith's swapping was on route choices. The second difference is that the change of flow in this study is made from time step to time step within a day, while the change of flow in Smith's swapping was made every day in day-to-day dynamics.

$$\Delta x_p^i(t) = \alpha \left[-\sum_{q \in P} (y_p^i(t) - y_q^i(t))_+ \cdot x_p^i(t) + \sum_{q \in P} (y_q^i(t) - y_p^i(t))_+ \cdot x_q^i(t) \right] \tag{6.5}$$

Where $(a)_+ = \begin{cases} a, \text{ if a} \geq 0 \\ 0, \text{ otherwise} \end{cases}$, and $\alpha > 0$ is a scalar to translate the unit of cost to a proportion of flow.

Equation (6.5) presents the flow swapping between two different parking destination choices. $\Delta x_p^i(t)$ is the change of the number of drivers at node i selecting destination p at time t, which is calculated as the number of leaving drivers switching to other parking destinations minus the number of incoming drivers switching from other parking destinations. Note that $x_p^i(t)$, of drivers at node i choosing destination p, means the number of all drivers who are on the road segments heading to node i and have chosen destination p at time t. Any driver of such a group can switch her/his destination choice at the sequential time step, while they are still on the road segment heading to node i.

However, Equation (6.5) requires the time step $\tau \to 0$ so that the group of travelers $x_p^i(t)$ remains the same. Otherwise, the flow at node i in the next time step, i.e. $x_p^i(t+1)$, will probably not be the same group of travelers in the current time step, i.e. $x_p^i(t)$. Such a requirement may not be feasible in real-world applications as parking choice switching frequency might not be set to be extremely high due to cost and efficiency considerations. Furthermore, it is observed from the right-hand-side of Equation (6.5) that the change of flow with certain parking destination choices is determined by the number of drivers with all parking destination choices. Such information may not be necessarily needed if we further rewrite the swapping rules in probabilistic terms in Equation (6.6). In Equation (6.6), each driver with a selected parking destination may switch his/her destination with a probability, if the current choice does not have the least expected cost. Such a probability

is taken to be proportional to the cost difference between two parking destination choices.

Simplified from Equation (6.6), from the perspective of each driver chosen destination p, the probability of switching from the current destination choice to the other one is

$$Prob^{i;v}_{p\to q}(t) = \begin{cases} 0 & if\ y^i_p(t) < y^i_q(t) \\ \alpha'(y^i_p(t) - y^i_q(t)) & otherwise \end{cases}, p \neq q \tag{6.6}$$

where $Prob^{i;v}_{p\to q}(t)$ is the probability of the individual vehicle v at node i at time t switching its parking destination from parking garage p to parking garage q. Drivers are willing to switch to a new parking destination only if the new parking destination can lower their total costs. If the total cost towards a certain parking garage q is greater than the total cost of the current choice p, the driver will not consider switching to that parking garage at all, which means $Prob^{i;v}_{p\to q}(t) = 0$. α' is a regularizer to ensure that the summation of $Prob^{i;v}_{p\to q}(t)$ is within a reasonable range. Aggregating all drivers, the probability shown in Equation (6.6) is equivalent to Equation (6.5).

There have been a number of studies on choosing so that the swapping behavior is well-defined between different choices. For instance, Leurent and Boujnah (2012) proposed the following method to calculate the regularizer α':

$$\alpha' = \frac{1}{\sum_{q\neq p}(y^i_p(t) - y^i_q(t)) + H} \tag{6.7}$$

where H in Equation (6.7) represents the willingness to stay with the current choice. In this study, we assume there is only a single group of drivers among which all the drivers share the same "H". We did not study the effects of different preference of "H". However, in reality, "H" could be different for different drivers. This would be an interesting problem to study in the future research, which is beyond the scope of this study. We will treat "H" as a homogeneous variable in the context of this study.

The probability that a driver keeps the current choice in the next time step is

$$1 - \sum_{q \neq p} Prob^{i;v}_{p \to q}(t) = 1 - \frac{\sum_{q \neq p} (y^i_p(t) - y^i_q(t))_+}{\sum_{q \neq p} (y^i_p(t) - y^i_q(t))_+ + H} \tag{6.8}$$

$$= \frac{H}{\sum_{q \neq p} (y^i_p(t) - y^i_q(t))_+ + H}. \tag{6.9}$$

It shows that the larger H is, the more likely the driver would not change the current parking destination choice. In the simulation we conducted in this study, $H = \$1.0$ is used.

6.2.4 Parking garage occupancy dynamics

Between the current time instant and the expected arrival time instant at the parking destination, the parking garage will receive incoming vehicles for parking, which means a proper estimation of the parking occupancy should consider these incoming vehicles during a future time period. In this study, we consider the period of morning commute during which vehicles arrive at the parking garage and remain in the garage thereafter.

Given the parking destination choices of all drivers in the network at the current time instance, the best estimation of the future parking occupancy at garage p is calculated as the sum of the current occupancy and the number of expected arrival vehicles in Equation (6.10).

For $i \in N$,

$$OC_p\left(t + \tau^i_p(t)\right) = OC^{act}_p(t) + \sum_{\substack{j \in N \\ t + \tau^j_p(t) \leq t + \tau^i_p(t)}} x^j_p(t). \tag{6.10}$$

$OC^{act}_p(t)$ is the actual occupancy of parking garage p at time step t, which is provided by the real-time counts at the garage. The second term is the summation of number of vehicles from all nodes $i \in N$ to parking garage p.

The constraint in Equation (6.10) $t + \tau^j_p(t) \leq t + \tau^i_p(t)$ ensures that we only count the vehicles that will arrive at the parking garage p at the same time t without duplication counts, as no vehicles can appear at multiple nodes at the current time t. It excludes the vehicles that are expected to arrive later than time $t + \tau^i_p(t)$ from the counts in the second term in Equation (6.10).

The expected occupancy of a parking garage in the future is of great importance, because the calculation of future parking costs, which underpins the driver's parking destination and route choices, relies on the future occupancy. In the simulation, the future occupancies are built as a vector of occupancy at each time step in the future. The vector is then updated by Equation (6.10) every updating interval $t^p_{update} = 10$ seconds, which is ten times to the simulation time step length 1 second.

6.3 Simulation results and analysis

Simulations are carried out using SUMO and OmNet++. SUMO is a microscopic traffic simulator which can simulate the network and vehicle movements in the network using various car-following models. OmNet++ is a simulation framework used to simulate computer networks and network protocols. We use it to simulate the online information release mechanism in our framework. See Amoozadeh et al. (2015) and Chai et al. (2017) for more details on the simulation tools. The synthetic network used for the simulation is shown in Figure 6.2.

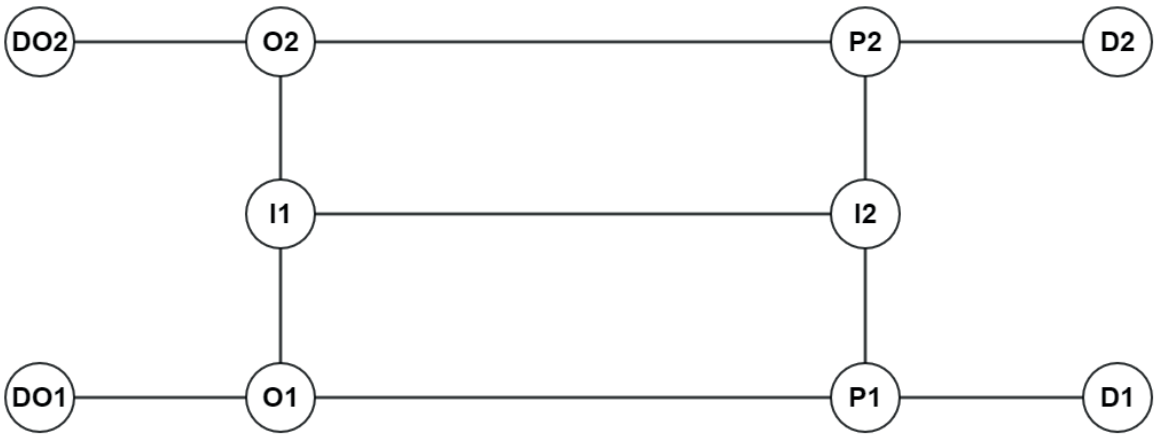


Figure 6.2: A synthetic network.

In this network, there are 2 origins (DO1 and DO2) and 2 destinations (D1 and D2). There are 2 parking garages (P1 and P2) associated with the destinations, which means both destinations have access to both parking garages with different walking cost W^d_p. Each parking garage has its corresponding capacity and parking fee policy. The parking

garage information is shown in Table 6.2. In this study, we assume all time costs can convert into monetary costs in dollar with the value of time 0.5 dollar per minute, i.e., $1/120$ dollar per second. Other nodes in the network are all regular intersections that are controlled by a fixed traffic signal plan (Since signal control is not a focus of this study, we use fixed timing control to simplify the problem. Nevertheless, other more complicated signal control methods are also applicable.).

Table 6.2: Parking garage capacity, parking cost and walking cost for the synthetic network.

Parking Garage	Initial Parking Fee ($)	Capacity	Walking Cost To Node (s)	
			D1	D2
P1	0.3	800	10	60
P2	0.3	1500	60	10

Vehicles are generated at the two origin nodes. Each vehicle is initially assigned to one destination node, while it can park at either parking garage. All nodes except origin and destination nodes are controlled by traffic lights, which are considered in the dynamic traffic routing algorithm. There are two types of vehicles running in the network - traditional vehicles without the access to online information and re-routing vehicles that have online information access. Only re-routing vehicles are able to change their route choices and parking destination choices during the trip when necessary. For those unguided vehicles, they drive directly to the pre-set parking destination. If they find the parking garage is full when they reach the parking garage, they randomly select a nearby parking garage as a second choice.

To guarantee proper configurations of the simulation, we need to have the following assumptions for the simulation:

- Assumption 1: In this research, we only consider the scenarios where the total parking demand is less than the total parking supply. In the final state of simulation, every vehicle should find a parking space. The scenarios where parking demand exceeds parking supply are beyond the scope of this study.

- Assumption 2: All drivers have bounded rationalities when making a decision on

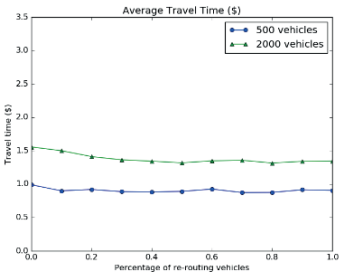
which parking garage to go to. Drivers tend to choose the choices that minimize their total costs as a primary goal. But their choices may also be affected by some other constraints like familiarity to/preference for some specific parking garages, hence the parking garage choice switch model discussed in the last section is not an all-or-nothing model but a proportional switch model.

- Assumption 3: Re-routing vehicles have full access to the online information of link travel times and parking garage information.

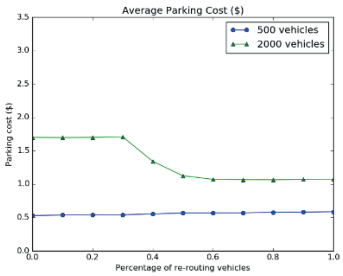
6.3.1 Effects of penetration rates of re-routing vehicles

In this subsection, we present the effects of penetration rates (i.e. the percentage of re-routing vehicles in the entire vehicles entity. These vehicles are equipped with the parking guidance system.) on network performance under different demand levels. Figure 6.3 shows the results of various costs, i.e. travel times, parking cost and total cost, with different market penetration rates ranging from 0 to 1.0. All time costs are converted into monetary value (in dollar) for comparison purpose using the value of time as \$0.5/min. Two different levels of total traffic demand are tested: 500 vehicles and 2000 vehicles. When loaded to the network, the origin and destination nodes of a vehicle are randomly selected from two origin nodes and two destination nodes, respectively. The vehicles depart during the first 1500 seconds simulation time, which is further divided into three time periods, including 600 seconds before peak period, 450 seconds peak period and 450 seconds after peak period. Within each time period, the flow rate is constant. The demand rate at the peak period is 6 times of that in the off-peak periods: during peak hours the demand rate is 0.8 veh/s and 0.15 veh/s for off-peak hours.

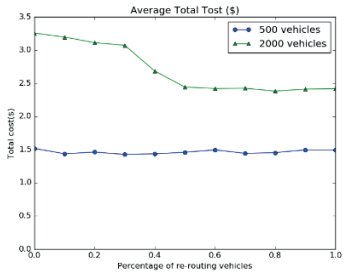
Figure 6.3a shows the average travel time on the roads before reaching the parking garage. Figure 6.3b shows the average parking cost. Figure 6.3c shows the average total costs in the network. The x-axis is the percentage of vehicles with routing and parking guidance, that can switch routes and parking lots en-route. Figure 6.3 shows that the average travel time is at its minimal when market penetration rate is higher than 80%, though the difference is not significant compared to other penetration rates when the demand is light. The effect of enabling en-route route and parking choices in the network is more sig-



(a) Average travel time ($).



(b) Average parking cost ($).



(c) Average total cost for individual travelers ($).

Figure 6.3: Average costs comparison over different penetration rates.

nificant when the network is more congested. This result is consistent with the findings in Chai et al. (2017). In their study, the vehicles cannot change their destinations en-route, while in this study vehicles are able to switch their parking destinations if necessary. The result indicates that by allowing drivers switch their destinations en-route, the Dynamic Traffic Routing (DTR) algorithm (originally not designed for such destination switching) can also reduce average travel time.

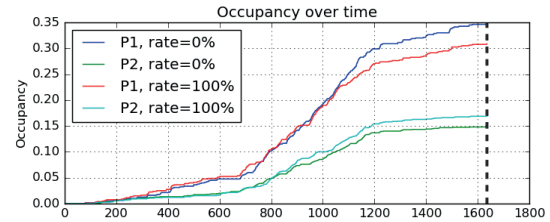
On the parking cost side, since the parking garages are always at low occupancy under low demand level (500 vehicles case), the cost does not change much across market penetration rates. When the demand gets much higher (2000 vehicles), at least one of the parking garages is relatively packed with parked vehicles, so that the parking cost increases dramatically. In Figure 6.3b, when there are 2000 vehicles of total demand and market penetration rate is below 0.3, the average parking cost is almost constant. When market penetration rate is over 0.3, the average parking cost decreases monotonically as the penetration rate increases until it reaches 0.6, after which the cost curve becomes flat. This indicates that the penetration rate plays a great role in reducing the parking cost when the network is congested, and there is a critical range (0.4-0.6) of the penetration rate where parking and routing guidance provides the most benefits. This is consistent with findings from earlier Advanced Traveler Information Studies with respect to market penetration rates, e.g. Al-Deek and Kanafani (1993) stated in their paper that the benefits of guiding traffic in an accident scenario will decrease when the percentage of guided traffic exceeds some critical threshold.

Figure 6.3c combines the average travel time and average parking cost for an individual

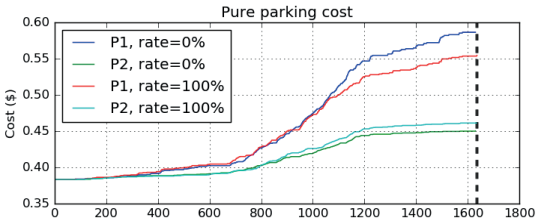
traveler to provide his/her commuting cost. As we can observe, individual's total commuting cost curves have similar patterns as in the average parking cost case. When the demand is low (500 vehicles), the individual's average commuting cost does not change much. When the demand is high (2000 vehicles), the decrease of commuting cost is more dramatic after the penetration rate reaches 0.5, compared with no vehicles are provided routing and parking guidance. Again, a thresh value of market penetration exists for obtaining the most benefits in the high traffic demand case.

6.3.2 Balanced utilization of parking garages

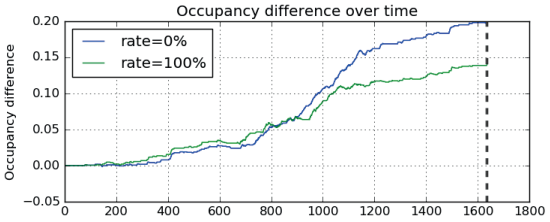
Figure 6.4 and Figure 6.5 show the occupancy and other parameters of the parking garages over the entire simulation horizon with a total number of vehicles equal to 500 and 2000 respectively. We run two sets of simulations with market penetration ranges from 0 and 1.0. The variables we are looking at are occupancy of each parking garage over time and the parking cost without walking cost of each parking garage over time, which is the summation of parking fee and inside parking garage searching cost. For simplicity, we use pure parking cost to denote the parking cost without walking cost. Figure 6.4d and Figure 6.5d describe the demand profiles for the two different cases.



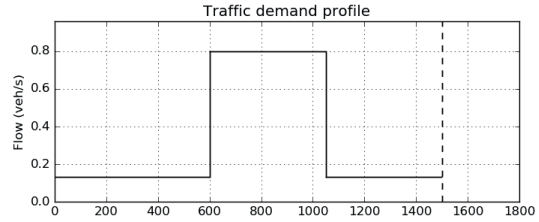
(a) Occupancy over time.



(b) Pure parking cost over time.



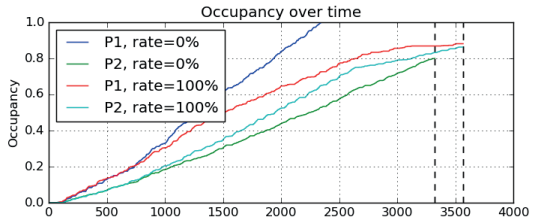
(c) Occupancy difference over time.



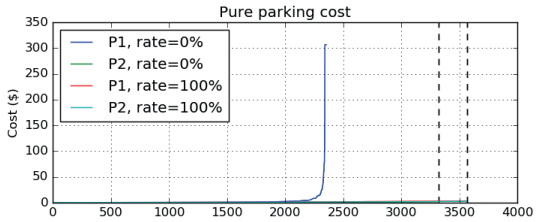
(d) Vehicle demand profile.

Figure 6.4: Case 1 - 500 vehicles.

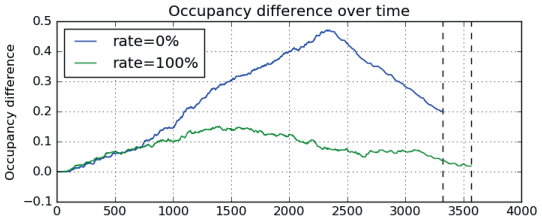
Figure 6.4a and Figure 6.5a show the parking garage occupancies over time with 500 and 2000 vehicles respectively. In the 500 vehicles case, when there are 100% guided ve-



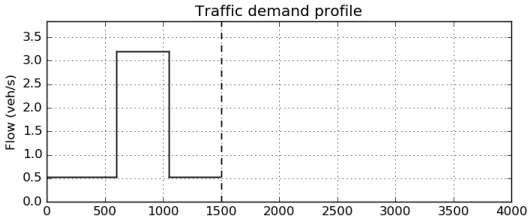
(a) Occupancy over time.



(b) Pure parking cost over time.



(c) Occupancy difference over time.



(d) Vehicle demand profile.

Figure 6.5: Case 2 - 2000 vehicles.

hicles the simulation terminates (the time the last vehicle in the network finds its parking space) about 20s earlier than the case in which all vehicles are not guided. However, in the 2000 vehicles case, the one with 100% penetration terminates about 250s later than the one with 0% penetration. The dynamic routing and parking guidance system tries to reduce the overall cost for the travelers, which includes both travel cost and parking related cost. It does not necessarily guarantee that the travel time is reduced. Hence, it is possible that with re-routing enabled, a certain traveler may experience longer travel time en-route and simulation takes longer time to terminate as what is observed in 2000 vehicles. However, this observation depends highly on the simulation settings like parking demand, parking lot capacities and parking fee schemes. Sometimes the simulation with parking guidance system might terminate earlier than the one without guidance system, which is exactly what happens in the 500 vehicles case.

Figure 6.4b shows that in 500 vehicles case, the pure parking cost difference between different parking garages is reduced when vehicles are 100% assisted by the parking and route guidance system compared to the case that there is no guidance at all. Figure 6.5b shows a more extreme scenario where the pure cost of one parking garage explodes to infinity when there is no parking and route guidance system implemented. This implies that that garage is at full capacity when simulation ends (See Equation (6.2)).

The simulation results also show that the parking switching mechanism enables more efficient utilization of the parking facilities. The occupancy difference between two parking garages are reduced, and parking garages are less likely to get fully occupied as demands can be routed to other parking garages, as shown in Figure 6.5a. Parking garage P1 has a 100% occupancy and P2 has a final occupancy around 80% when there is no parking and routing guidance. When guidance is enabled, the occupancies of parking garage P1 and P2 approach to 90% asymptotically and never reach 100%. The difference between red and light blue lines gets smaller and smaller and approaches to 0 asymptotically after time 1500s, when all vehicles are loaded into the network (see the green line in Figure 6.5c); meanwhile the difference between dark blue and green lines remains over 0.2, which doesn't show the asymptotic pattern (see the dark blue line in Figure 6.4c). In the low demand scenario, the occupancy difference does not diminish to 0, no matter how many guided vehicles are in the network. Nevertheless, when guidance is enabled in the low demand scenario, the occupancy difference between parking garages is still decreasing, as shown in Figure 6.4c.

Since the demands are better allocated to the available parking garages, the pure parking cost is much lower when our guidance system is deployed, as shown in Figure 6.5b. Without guidance, the pure parking cost of parking garage P1 will rise sharply to untenable levels at around t=2300s when that parking garage is fully occupied. When guidance is enabled, the pure parking costs of both parking garages are kept within a reasonable amount (under $20). In the 500 vehicles case, however, the pure parking cost of each parking garage remains at a relatively low level regardless of the penetration rate, largely because both lots have adequate capacity to handle all parking demand.

6.3.3 Sioux Falls network

To evaluate the performance of the proposed framework in a more realistic setting, a medium size network of Sioux Falls city is used. See Figure 6.6. The network is an abstraction of the real Sioux Falls network with 5 parking garages locating at node 9, node 10, node 11, node 15 and node 16. The capacity and parking cost related information are given in Table 6.3.

Two different demand levels are tested with the Sioux Falls network: 3000 vehicles

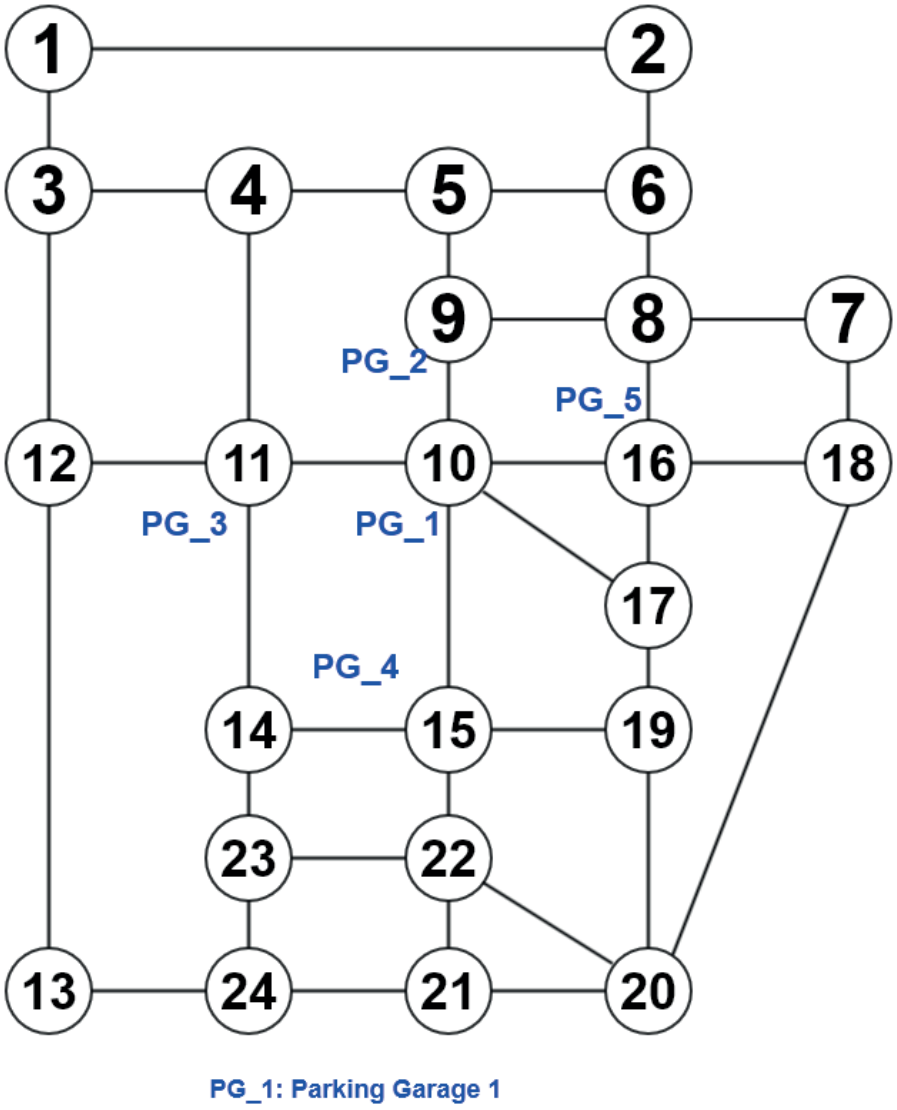


Figure 6.6: Sioux Falls network with 5 parking garages.

and 5000 vehicles separately. Vehicles are generated with the randomly selected origin and destination from the OD list in Table 6.4. Node 10 gets the highest probability to be chosen as destination since it represents the downtown area. There is a peak period from 400s to 700s during which the flow rate is six times of that during off-peak hours. See Figure 6.7c and Figure **??**c for demand profile details.

Two parking garages, P9 and P10, are selected to study the performance of the guidance system. Figure 6.7 and Figure 6.8 show the occupancy and pure parking cost over time

Table 6.3: Parking garage capacity, parking cost and walking cost for the Sioux Falls network.

Parking Garage	Initial Parking Fee ($)	Capacity	Walking Cost To Node (s)				
			9	10	11	15	16
PG_1	0.3	1500	15	1	20	30	20
PG_2	0.3	1000	1	15	35	45	35
PG_3	0.3	1000	35	20	1	50	40
PG_4	0.3	1000	45	30	50	1	50
PG_5	0.3	1000	35	20	40	50	1

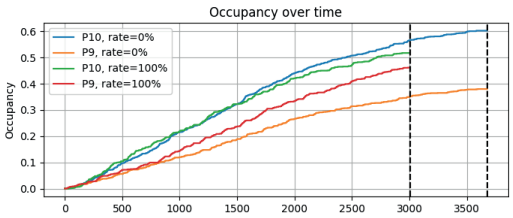
Table 6.4: Origin and destination nodes for the Sioux Falls network.

Origin Nodes						Destination Nodes				
1	2	3	6	7	8	9	10	11	15	16
12	13	18	20	21	24					

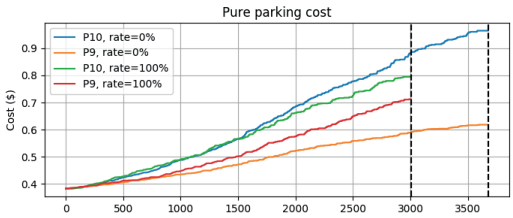
for 3000 vehicles and 5000 vehicles cases respectively. From the figures, we can see that with 100% of re-routing vehicles in the network, the occupancy difference between the two parking garages is reduced compared to 0% re-routing vehicles case for both 3000 vehicles and 5000 vehicles. The pure parking cost follows the same pattern. This is consistent with the results from the simple synthetic network.

Figure 6.9 shows how the costs change when the penetration rate of re-routing vehicles varies from 0% to 100%. Figure 6.9a shows that the average travel time decreases as the penetration rate of re-routing vehicles increases from 0% to 100% for both 3000 vehicles and 5000 vehicles cases. The average travel times for 3000 vehicles and 5000 vehicles are roughly the same for most of the penetration rates. This is because both 3000 vehicles and 5000 vehicles make the network quite congested with these many vehicles. The demand difference between the two does not make much difference in terms of average travel time.

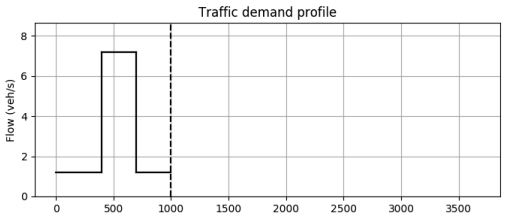
Figure 6.9b shows the average parking-related costs. We show three types of costs in the figure: walking cost, parking fee plus searching cost and total parking cost, to see how the penetration rate of re-routing vehicles affects each of the costs. Parking fee and searching cost remains almost the same as re-routing vehicles percentage changes in the 3000



(a) Occupancy over time.

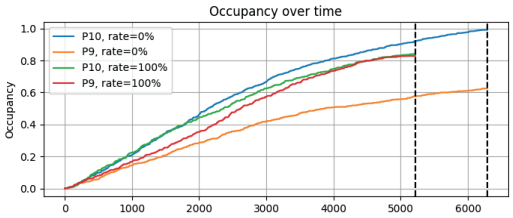


(b) Pure parking cost over time.

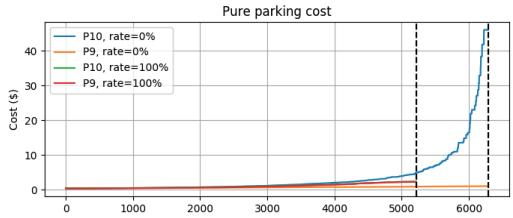


(c) Vehicle demand profile.

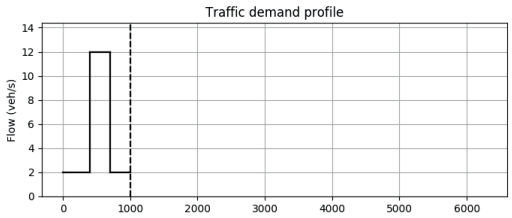
Figure 6.7: Case 1 - 3000 vehicles.



(a) Occupancy over time.



(b) Pure parking cost over time.



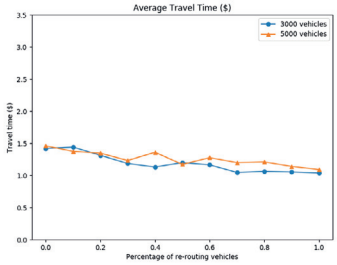
(c) Vehicle demand profile.

Figure 6.8: Case 2 - 5000 vehicles.

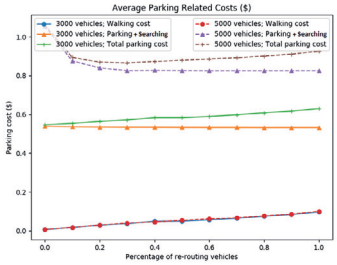
vehicles case. In this case, the total demand (3000 vehicles in total) is low, and the garages have sufficient capacities (5500 slots in total) to meet the parking demand. Flow switching between different parking garages are kept within a moderate range. The switching has small impact on the parking fee and searching cost. In the 5000 vehicles case, the switching between different garages is more unbalanced: more drivers switch their parking choices from Garage 1 to other garages than the reversed switching. If there was no parking and routing guidance system, vehicles heading to Garage 1 would not be able to

switch to other parking garages. Under this circumstance, parking fee and searching cost at parking Garage 1 will be very high. In the meanwhile, the other 4 parking garages may have very low utilization, leaving those unused capacities wasted. With parking and routing guidance system, vehicles can be guided to less occupied parking lots to avoid high parking fee and searching cost, and parking garage occupancies are more balanced.

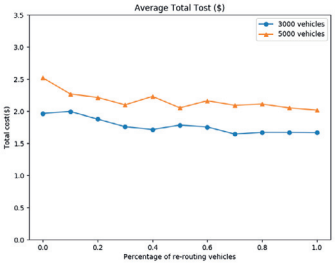
Walking cost goes up as the penetration rate of re-routing vehicles increases. This is because when vehicles are not re-routing, their walking costs are at minimum since they always parked at the nearest parking garage to their final destinations. With re-routing, some of them may park at some other parking garages which increase the walking costs compared to the former case. Even though walking cost may increase after the re-routing, drivers may still prefer to do so. The reason is that by re-routing, drivers can reduce their generalized(overall) costs. Figure 6.9c shows that with higher penetration rate of re-routing vehicles, the average travel cost decreases for both the 3000 vehicles case and the 5000 vehicles case.



(a) Average travel time ($).



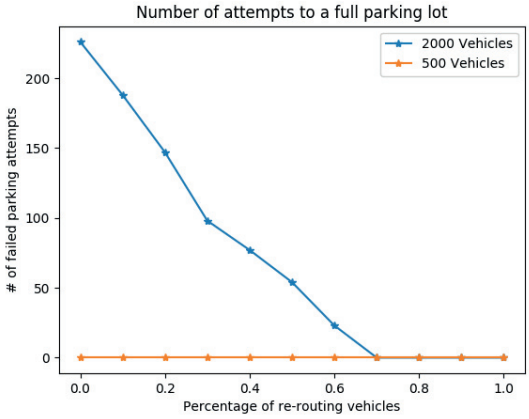
(b) Average parking related cost ($).



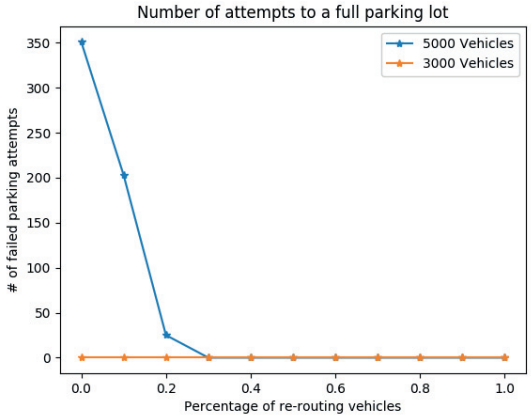
(c) Average total cost for individual travelers ($).

Figure 6.9: Average costs comparison over different penetration rates.

Figure 6.10 shows the total number of failed parking attempts to a fully occupied parking lot. When a traveler arrives at a parking lot in hope to find an available parking space, he/she is forced to search for a new parking lot as the one he/she is currently at is fully occupied. The failed parking attempts not only will increase the VMT/VHT (Vehicle Miles Traveled/Vehicle Hours Traveled) of that vehicle, but also will introduce more congestion to other traffic on road. The higher number of failed parking attempts to some extent indicates the poorer performance of a network. Travelers in a network without dynamic routing and parking guidance system are not able to know the occupancy information of



(a) A synthetic network with 2 origins and 2 destinations.



(b) Sioux Falls network.

Figure 6.10: Number of failed parking attempts to a full parking lot.

a certain parking lot ahead of time. This will inevitably result in some failed parking attempts. In Figure 6.10a when there are 2000 vehicles and 0% of re-routing vehicles, there are a total of 226 failed parking attempts in the network. This number decreases significantly as the percentage of re-routing vehicles increase. When there are greater than or equal to 70% re-routing vehicles in the network, the failed parking attempts become 0. When there are 500 vehicles in the network, there are no failed parking attempts as the parking lots are operating well below their capacities. Same results can be observed in the larger network as Figure 6.10b shows. Failed parking attempts are decreasing monotonically when the percentage of re-routing vehicles are increasing from 0% to 30% with 5000 vehicles presented. Failed parking attempts remain 0 when the percentage of re-routing vehicles exceeds 30%. The above results indicate that with the proposed dynamic parking and routing guidance system, vehicles can choose their parking destinations more efficiently.

6.3.4 Computational efficiency

All the simulation tests are run on a Xeon E5-2963 v3 (18 cores, 2.3 GHz) server, one simulation run per thread. For the smaller synthetic network, it takes 300 seconds on average to run a single simulation with 2000 vehicles. For the larger Sioux Falls network, it takes 1500 seconds on average to run a single simulation with 5000 vehicles. The parking

garage choice model takes trivial computation effort to perform a single calculation. The main computation cost comes from the DTR calculation as we are re-calculating a new hyper-path every couple of seconds. In real implementation, some of the computation tasks like DTR calculation can be distributed to every individual vehicle to speed up the computation.

6.4 Conclusion

In this chapter, we proposed a dynamic parking and route guidance system with Dynamic Traffic Routing (DTR) and parking choice. By iteratively updating the travel and parking costs using online information and estimation, travelers can switch their parking destinations with least expected costs en-route before reaching their destinations.

The proposed system was implemented and tested on a microscopic simulation platform. Simulation results showed that the average travel time and the total commuting cost of individuals can be significantly reduced when the penetration dates exceed a critical threshold value (60% in our case) under heavy traffic demand. From the system perspective, the benefit of real-time guidance becomes more significant when the traffic demand level is high. From the parking resource administrator's point of view, unbalanced parking garages mean inefficient use of parking resources and high parking costs in certain parking garages. The former results in a waste of parking resources, while the latter will increase the out-of-pocket costs of travelers. In the proposed system, vehicles can have access to the online traffic and parking garage information, so that they can switch to alternative parking garages to reduce their total costs. Simulation results showed that the proposed system can better allocate the parking demand so that the usage of parking resources is balanced. The occupancy difference between different parking garages becomes less compared to that without the proposed guidance system. The simulation results showed that neither of the parking garages would be fully occupied when total supply of parking resource is greater than the total demand. The results also showed that when parking destination switching was not enabled, some parking garage would be 100% occupied, which results in extremely high parking costs; when all vehicles can switch their parking destinations, the parking cost is kept at a reasonable level. For now, the system is only tested in simula-

tion, which many complicated relationships may be oversimplified. Later on, field implement in a small scale (maybe on campus) will be carried out to see if the proposed system can work in the real world as expected. Another possible challenge of the proposed system is the implementation cost. Gathering and distributing the real-time information require advanced infrastructures and equipment. Thanks to the rapidly developing technologies nowadays, access to that information is much easier and its cost is much lower than before: 1) most parking facilities have already started to collect real-time occupancies and parking prices; 2) map/traffic providers like HERE Maps, INRIX and Waze have very detailed and accurate link travel time information. Once the proposed system obtains that information from individual parking garages and the online map service providers, broadcasting the information to the travelers through mobile apps should be the major task to complete the proposed system. Most of the information is either public and open-sourced or can be purchased at reasonable costs.

For future research, one possible extension to the current work is to study the effects of different pricing schemes. In the real world, parking garage usually employs different parking charge policy during a different time of day to regulate the incoming flow to the parking garage, which can be further studied with the proposed destination switching algorithm. Also, in the current work, we did not distinguish and examine the effects of dynamic pricing and dynamic re-routing separately. In the next step, those two parts will be studied separately. Analysis and evaluations will be performed to see whether those two can benefit each other or there are some canceling effects, and how significant the effects are. Another interesting extension topic is to study the effects of incomplete information on the performance of the proposed system. The assumption in the current research assumes that perfect information is available to re-routing vehicles. Perfect information means every detailed information is available, which is not realistic in real-world application. It is an interesting but also challenging problem to study how to use partial information to make reliable and efficient parking guidance. One last interesting topic is to consider both short term and long term parking reservation behaviors in dynamic parking and route guidance system. Overall, the implementation and expansion of the proposed system may benefit urban travelers in the long run and thus merit more future

research.

Chapter 7

Conclusion and Future Work

7.1 Summary of contributions

This dissertation aims to investigate the relationship between traffic control and travelers' route choice behavior, and to develop a novel framework to systematically coordinate those two parts in a concurrent manner to improve the performance of the transportation system in the context of Vehicular Ad-Hoc Network (VANET). The central idea is to take advantage of on-line information availability boosted by VANET to reduce uncertainties and stochasticities while traveling in the network and, in the same time, to guide traffic to their destinations more quickly and efficiently. Traffic stability and deadlock concerns are also explicitly considered and studied. Based on the dynamic traffic routing and adaptive traffic signal control framework proposed, possible applications that can utilize the framework are discussed.

Firstly, we proposed a joint adaptive routing and traffic signal control algorithm to improve traffic operations in a VANET enabled traffic environment. Our dynamic routing algorithm (DTR) is an extension of the LET algorithm of Miller-Hooks and Mahmassani (2000) with periodic updates of link travel times. The hyper-path based dynamic traffic routing method takes stochasticity and randomness of link travel time into consideration, which ensures routing decision to be robust and reliable. In addition, the periodic link travel time online updating presented in this dissertation uses both historical information (a priori knowledge) and new online information, thanks to the V2X system, to form a posteriori knowledge about link travel time. Furthermore, the proposed algorithm also

takes into account the delay caused by real-time traffic signal operations. Besides several traditional traffic signal control strategies, namely fixed-timing, vehicle actuated control (known as low density control in our paper) and adaptive Webster's (known as high density control in our paper), I also proposed two new traffic signal control strategies, the Phase Selection Control and the Modified Max Pressure Control, to take into account the effects of both incoming demand and current queues on traffic signal operations.

Simulation models are built to test and compare the models developed in this dissertation against the traffic routing methods and traffic signal control models in the literature. Our simulation results show that the DTR algorithm works well under higher demand scenarios together with the adaptive traffic signal control methods proposed. Enabling vehicle re-routing in the network can reduce the average travel time as well as reduce the average queue length at the intersection. However, the more is not necessary the better. We discover that with a certain percentage between 0% to 100% of DTR vehicles in the network, the average travel time can reach its minimal, which our simulation tells us for 6000 vehicles this number is around 50%. This number will likely vary over networks, traffic demand, and driver population. With the dynamic travel time updating model proposed in the paper, the re-routing algorithm can further reduce the average travel time in the network by taking advantage of the most current link travel time information.

The different signal control methods proposed and tested under different scenarios tells us that the Phase Selection Control and the Modified Max Pressure Control work better than the rest four control methods (including the original Max Pressure Control and fixed timing control). They tend to respond to traffic and accommodate traffic better than the rest. Average speed is higher and average queue length is shorter when these two control methods are applied. Among all the six control methods (including the original Max Pressure Control), the original Max Pressure Control performs the worst as its logic to find the optimal phase and its corresponding duration is not well designed. With the Phase Selection Control and the Modified Max Pressure Control, the MFD of the network is closer to a trapezoidal shape compared to that of the Original Max Pressure Control. For the latter, its MFD is more chaotic and fluctuating. The maximum flow rate is also lowered compared to the former two.

The joint dynamic traffic routing and adaptive signal control approach is also tested against a traffic accident scenario. With links blocked or partially blocked by accidents, the dynamic traffic routing (DTA) can efficiently re-route traffic to other uncongested links to avoid long delay. According to the simulation results, the average travel time in the network can be reduced by $17\% \sim 27\%$ when the percentage of re-routing vehicles is chosen properly. The optimal percentage of re-routing vehicles in this case relies heavily on the traffic demand level. In uncongested or mildly congested scenarios, the more re-routing vehicles in the network are, the less the average travel time is. In a highly congested network, however, the optimal proportion of re-routing vehicles lies somewhere between $0\% \sim 100\%$, which again depends on the traffic demand, driver population, network geometry and so on.

Secondly, we investigate one crucial issue in traffic routing and dynamic traffic assignment that most of the literature, current and past, failed to consider: deadlock avoidance in traffic routing and traffic assignment. The dynamic traffic routing and adaptive signal control interaction model developed earlier in this dissertation behaves well in terms of average travel time and average travel delay in most cases. However, there is possibility that in some extreme cases the proposed routing and control model may fail to produce satisfying results. The underlying logic of DTR does not guarantee to prevent deadlock, a.k.a gridlock, from happening. To address the possibility that deadlock occurs, following the study of dynamic traffic routing and adaptive signal control, I formulate a deadlock avoidance dynamic user equilibrium model with queue spillback. Travelers' route choice is governed by a simple *"Odd-Even Routing"* rule which is proved to generate deadlock free routing result. Potential deadlocks are detected with an algorithm modified based on *Floyd Warshall Algorithm*, and assigned a deadlock potential value to each potential deadlock. The model minimizes this potential, and meanwhile tries to maintain the total travel time in the network at a reasonably low level.

Numerical studies for both high and low demand cases have been designed, tested and compared. The results show that the proposed DLA model performs well in congested, high demand scenarios, reducing the total number of queues in the network by 27%. Therefore, the deadlock potential is lowered correspondingly. The optimal percent-

age of *OE Routing vehicles* in high demand case equals 33.33%, which implies that by introducing only 33.33% *OE Routing vehicles* in the whole vehicle population, it is possible to significantly reduce the possibility of having deadlocks in the network. In addition to reduction in total number of queues, there are two other unique characteristics that the DLA model exhibits. Fisrt, the DLA model tends to spread queues evenly among different road links to utilize queuing capacity in the network as fully as possible. Furthermore, the DLA model also spreads out queues over temporally to make queues much more evenly distributed over time compared to cases without *OE Routing vehicles*. With these two queue smoothing, the DLA model is able to further reduce the deadlock potential. In low demand scenario, however, having *OE Routing vehicles* does not provide much benefits. Instead, the extra detours made by *OE Routing vehicles* contribute to total travel cost increase. Numerical test confirms that with low demand scenario the optimal solution is with 0% *OE Routing vehicles*. The more *OE Routing vehicles* in this situation, the worse the overall performance will be.

The DLA model proposed in this dissertation provides an efficient and effective approach to reduce the deadlock probability. Networks with high congestion benefit the most from the DLA model with respect to deadlock potential reduction. While the model of DLA is self-explanatory and working, it may not be straightforward to implement the proposed *OE Routing* scheme in real world. Traditional human-driven cars lack the capability to talk to traffic controllers. As a consequence, it is difficult, if not impossible, to completely prevent traditional human-driven cars from making certain turns that are prohibited by the *OE Routing rules* through imposing a turning penalty as what the model does. However, with V2I technology and autonomous driving technology making great break-through in the near future, monitoring and controlling vehicles will become easier and easier. Thus, imposing a special turning penalty to some special vehicles, i.e. *OE Routing vehiles*, will become a feasible task.

Finally, we explored the possibility of applying the dynamic traffic routing and adaptive traffic signal control in VANET environment to various applications. One particular application studied in this dissertation is on parking search via dynamic traffic routing. In the last chapter of this dissertation, we proposed a dynamic parking and route guidance

system with Dynamic Traffic Routing (DTR) and parking choice. By iteratively updating the travel and parking costs using online information and estimation, travelers can switch their parking destinations with least expected costs en-route before reaching their destinations.

The proposed system was implemented and tested on a microscopic simulation platform. Simulation results showed that the average travel time and the total commuting cost of individuals can be significantly reduced when the penetration rates exceed a critical threshold value (60% in our case) under heavy traffic demand. From the system perspective, the benefit of real-time guidance becomes more significant when the traffic demand level is high. From the parking resource administrator's point of view, unbalanced parking garages mean inefficient use of parking resources and high parking costs in certain parking garages. The former results in a waste of parking resources, while the latter will increase the out-of-pocket costs of travelers. In the proposed system, vehicles can have access to the online traffic and parking garage information, so that they can switch to alternative parking garages to reduce their total costs. Simulation results showed that the proposed system can better allocate the parking demand so that the usage of parking resources is balanced. The occupancy difference between different parking garages becomes less compared to that without the proposed guidance system. The simulation results showed that neither of the parking garages would be fully occupied when total supply of parking resource is greater than the total demand. The results also showed that when parking destination switching was not enabled, some parking garage would be 100% occupied, which results in extremely high parking costs; when all vehicles can switch their parking destinations, the parking cost is kept at a reasonable level. For now, the system is only tested in simulation, which many complicated relationships may be oversimplified. Later on, field implement in a small scale (maybe on campus) will be carried out to see if the proposed system can work in the real world as expected. Another possible challenge of the proposed system is the implementation cost. Gathering and distributing the real-time information require advanced infrastructures and equipment. Thanks to the rapidly developing technologies nowadays, accessing to that information is much easier and its cost is much lower than before: 1) most parking facilities have already started to collect

real-time occupancies and parking prices; 2) map/traffic providers like HERE Maps, INRIX and Waze have very detailed and accurate link travel time information. Once the proposed system obtains that information from individual parking garages and the online map service providers, broadcasting the information to the travelers through mobile apps should be the major task to complete the proposed system. Most of the information is either public and open-sourced or can be purchased at reasonable costs.

7.2 Outlook of future work

Our future work will extend the current work based on (deadlock free) dynamic traffic routing with adaptive signal control to multiple directions.

A possible extension of this work would be to consider global signal optimization. The signal control methods proposed in this work are all distributed methods. The phase selection method, for example, seeks local optimality but not global optimality. Coordination could be another extension to the current work. These new research directions will be even more challenging than the current joint routing and distributed signal control problem that we have dealt with in this paper and are worthy of serious investigation.

Another possible extension could be dynamically changing the market penetration rate of *DLA-routing vehicles* in response to the traffic demand and overall network conditions, instead of one fixed percentage in the current model.

For parking search application, one possible extension to the current work is to study the effects of different pricing schemes. In the real world, parking garage usually employs different parking charge policy during a different time of day to regulate the incoming flow to the parking garage, which can be further studied with the proposed destination switching algorithm. Also, in the current work, we did not distinguish and examine the effects of dynamic pricing and dynamic re-routing separately. In the next step, those two parts will be studied separately. Analysis and evaluations will be performed to see whether those two can benefit each other or there are some canceling effects, and how significant the effects are. Another interesting extension topic is to study the effects of incomplete information on the performance of the proposed system. The assumption in the current research assumes that perfect information is available to re-routing vehicles. Perfect infor-

mation means every detailed information is available, which is not realistic in real-world application. It is an interesting but also challenging problem to study how to use partial information to make reliable and efficient parking guidance. One last interesting topic is to consider both short term and long term parking reservation behaviors in dynamic parking and route guidance system. Overall, the implementation and expansion of the proposed system may benefit urban travelers in the long run and thus merit more future research.

Appendix A

Chapter 4

A.1 Algorithm 1: DTR algorithm with adaptive signal control

Algorithm 3 DTR algorithm with adaptive signal control

procedure INITIALIZATION
  for each node $i \in V$ **do**
    for each node $h \in \{\Gamma(i), i\}$ **do**
      for each $t \in S$ **do**
        if $i \neq D$ **then**
          $\lambda_i^h(t) = \infty$
          $\pi_i^h(t) = \infty$
        else
          $\lambda_i^h(t) = 0$
          $\pi_i^h(t) = 0$
        end if
      end for
    end for
  end for
  Create an empty vector H
  Insert D into H: $H \leftarrow D$
end procedure

18:

19: **procedure** UPDATE LABELS

20: **for** each $i \in \Gamma(j)$ **do**

21: **for** each $h \in \{\Gamma(i), i\}$ **do**

22: **for** each $t \in T$ **do**

23:

$$\mu_i^h(t) = \min_{j \in \Gamma(i)} \left\{ \sum_{k=1}^{K} \left\{ \left[\phi_{ij}^{\omega_i^l}(t) + \tau_{ij}^k \left(t + \phi_{ij}^{\omega_i^l}(t) \right) + \lambda_j^i \left(t + \phi_{ij}^{\omega_i^l}(t) + \tau_{ij}^k \left(t + \phi_{ij}^{\omega_i^l}(t) \right) \right) \right] \cdot \rho_{ij}^k \left(t + \phi_{ij}^{\omega_i^l}(t) \right) \right\} \right\}$$

24: **if** $\mu_i^h(t) < \lambda_i^h(t)$ **then**

25: $\lambda_i^h(t) = \mu_i^h(t)$

26: $\pi_i^h(t) = j$

27: $H = H \bigcup i$

28: **end if**

29: **end for**

30: **end for**

31: **end for**

32: Go to procedure INITIALIZATION

33: **end procedure**

34:

35: **procedure** CHECK VEHICLE STATE

36: **if** next node is reached and destination node is not reached **then**

37: Go to Procedure Updating Queue Distribution

38: **else if** next node is reached and it is destination node **then**

39: Go to Procedure Stop

40: **else**

41: $t \leftarrow t + 1$

42: Go to Procedure RE-ROUTING

43: **end if**

44: **end procedure**

45: **procedure** UPDATE LINK TRAVEL TIME

46:

$$\rho^{updated}(t=\tau) = \begin{cases} b \cdot \rho^{old}(t=\tau), & if\ \tau \in T^{old} \\ c \cdot \rho^{new}(t=\tau), & if\ \tau \in T^{new} \\ b \cdot \rho^{old}(t=\tau) + c \cdot \rho^{new}(t=\tau), & if\ \tau \in T^{old} \cap T^{new} \end{cases}$$

$$T^{updated} = T^{old} \cup T^{new}$$

47: **end procedure**

48:

49: **procedure** RE-ROUTING

50: Go to Procedure Initialization

51: **end procedure**

52: **procedure** STOP

53: STOP

54: **end procedure**

A.2 Algorithm 2: Modified Max Pressure Algorithm

Algorithm 4 Modified Max Pressure Algorithm

procedure CHOOSE PHASE AND DURATION
  for each $t \in [G_{min}, G_{max}]$ **do**
    for each movement m **do**
      $f_m(t) = \frac{\alpha \times N_t^m + \beta \times Q_t^m}{t}$
      (N_t^m: is the number of vehicles that will arrive within time t that will move in movement m)
      (Q_t^m: is the queue length on the link of movement m)
    end for
  end for
  $f_{high} = -1$
  for each $t \in [G_{min}, G_{max}]$ **do**
    for each phase p **do**
      sum the flow rates of each movement in the phase at that time
      if summed flow rate $\geq f_{high}$ **then**
        f_{high}=summed flow rate
        next phase = p
        next phase duration = t
      end if
    end for
  end for
end procedure

A.3 Original MP Control

The MP Control policy (Varaiya (2013)) $u^* : \chi \to \mathcal{S}$. For $X \in \chi$ assign the weight of each movement (n, m)

$$w(n,m)(X) = x(n,m) - \sum_{p \in Out_m} r(m,p)x(m,p) \tag{A.1}$$

and assign the pressure of each network signal control matrix $S \in \mathcal{S}$

$$\gamma(S)(X) = \sum_{n,m} c(n,m)w(n,m)(X)S(n,m) = \sum_{n,m;S(n,m)=1} c(n,m)w(n,m)(X) \tag{A.2}$$

The MP policy u^* is:

$$u^*(X) = argmax\{\gamma(S)(X)|S \in \mathcal{S}\} \tag{A.3}$$

Appendix B

Chapter 5

B.1 Floyd-Warshall Algorithm

Algorithm 5 Floyd-Warshall Algorithm

```
1:  procedure
2:      let dist be a |V| × |V| array of minimum distances
3:      for u from 1 to |V| do
4:          for v from 1 to |V| do
5:              if w(u,v) is not NULL then // the weight of the edge (u,v)
6:                  dist[u][v] ← w(u,v)
7:              else
8:                  dist[u][v] ← ∞
9:              end if
10:         end for
11:     end for
12:     for each vertex v do
13:         dist[v][v] ← 0
14:     end for
15:     for k from 1 to |V| do
16:         for i from 1 to |V| do
17:             for j from 1 to |V| do
18:                 if dist[i][j] > dist[i][k] + dist[k][j] then
19:                     dist[i][j] ← dist[i][k] + dist[k][j]
```

```
20:                 end if
21:             end for
22:         end for
23:     end for
24: end procedure
```

B.2 Modified Transitive Closure Algorithm

Algorithm 6 Modified Transitive Closure Algorithm

1: **procedure** INITIALIZATION
2: **Input** Vertices: V, Edges: E, Link Deadlock Potential: $P_{ij}(t) = \sum_{s'} \sum_{r} q_{(i,j)}^{d,s',r}(t)$
3: **for** each edge (u, v) in E **do**
4: **if** $P_{ij}(t) > 0$ **then**
5: $w(u, v) = 1$
6: **else**
7: $w(u, v) = 0$
8: **end if**
9: **end for**
10: **end procedure**
11: **procedure** TRANSITIVE CLOSURE
12: **Input** Vertices: V, Edges: E, Link Weights: $w(u, v)$
13: **let** reach be a |V| $\times$ |V| array
14: **for** u from 1 to |V| **do**
15: **for** v from 1 to |V| **do**
16: **if** w(u,v) is not NULL **then** // the weight of the edge (u,v)
17: reach[u][v] $\leftarrow w(u, v)$
18: **else**
19: reach[u][v] $\leftarrow 0$
20: **end if**
21: **end for**
22: **end for**
23: **for** each vertex v **do**
24: reach[v][v] $\leftarrow 1$
25: **end for**

```
    for k from 1 to |V| do
        for i from 1 to |V| do
            for j from 1 to |V| do
                reach[i][j] ← reach[i][j] or (reach[i][k] and reach[k][j])
            end for
        end for
    end for
end procedure
procedure COMPUTE CYCLES
    Input Reachability Matrix: reach
    let Ξ_t be the list of cycles in the network at time t
    let indices be the indices where diagonal(reach) == 1
    for each index i in indices do
        for each index j in indices and i ≠ j do
            if reach[i][j] == 1 then
                if index i ∈ cycle c in Ξ_t then
                    if index j ∉ c then
                        c ← c ∪ {j}
                    end if
                else
                    Add a new cycle (i, j) to the list of cycles Ξ_t
                end if
            end if
        end for
    end for
end procedure
```

References

Vicente Milanés, Steven E Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, and Masahide Nakamura. Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):296–305, 2014.

Mani Amoozadeh, Hui Deng, Chen-Nee Chuah, H Michael Zhang, and Dipak Ghosal. Platoon management with cooperative adaptive cruise control enabled by vanet. *Vehicular communications*, 2(2):110–123, 2015.

Kakan C Dey, Li Yan, Xujie Wang, Yue Wang, Haiying Shen, Mashrur Chowdhury, Lei Yu, Chenxi Qiu, and Vivekgautham Soundararaj. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc). *IEEE Transactions on Intelligent Transportation Systems*, 17(2):491–509, 2016.

Xiao-Yun Lu, J Karl Hedrick, and Michael Drew. Acc/cacc-control design, stability and robust performance. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, volume 6, pages 4327–4332. IEEE, 2002.

Jeroen Ploeg, Alex FA Serrarens, and Geert J Heijenk. Connect & drive: design and evaluation of cooperative adaptive cruise control for congestion reduction. *Journal of Modern Transportation*, 19(3):207–213, 2011.

Chakkaphong Suthaputchakun, Zhili Sun, and Mehrdad Dianati. Applications of vehicular communications for reducing fuel consumption and co 2 emission: The state of the art and research challenges. *IEEE Communications Magazine*, 50(12):108–115, 2012.

Sinan Öncü, Nathan Van de Wouw, WP Maurice H Heemels, and Henk Nijmeijer. String stability of interconnected vehicles under communication constraints. In *2012 IEEE 51st IEEE conference on decision and control (cdc)*, pages 2459–2464. IEEE, 2012.

Ismail H Zohdy and Hesham Rakha. Game theory algorithm for intersection-based cooperative adaptive cruise control (cacc) systems. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1097–1102. IEEE, 2012.

Kakan C Dey, Li Yan, Xujie Wang, Yue Wang, Haiying Shen, Mashrur Chowdhury, Lei Yu, Chenxi Qiu, and Vivekgautham Soundararaj. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc). *IEEE Transactions on Intelligent Transportation Systems*, 17(2):491–509, 2015.

Tianxin Li and Kara M Kockelman. Valuing the safety benefits of connected and automated vehicle technologies. In *Proceedings of the 95th Annual Meeting of the Transportation Research Board, Washington, DC, USA*, pages 10–14, 2016.

YY Fan, RE Kalaba, and JE Moore. Shortest paths in stochastic networks with correlated link costs. *Computers & Mathematics with Applications*, 49(9):1549–1564, 2005.

Tao Xing and Xuesong Zhou. Finding the most reliable path with and without link travel time correlation: A lagrangian substitution based approach. *Transportation Research Part B: Methodological*, 45(10):1660–1679, 2011.

Yiyong Pan, Lu Sun, and Minli Ge. Finding reliable shortest path in stochastic time-dependent network. *Procedia-Social and Behavioral Sciences*, 96:451–460, 2013.

John E Beasley and Nicos Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394, 1989.

Jin Y Yen. Finding the k shortest loopless paths in a network. *management Science*, 17(11): 712–716, 1971.

Narsingh Deo and Chi-Yin Pang. Shortest-path algorithms: Taxonomy and annotation. *Networks*, 14(2):275–323, 1984.

Richard Bellman. On a routing problem. Technical report, DTIC Document, 1956.

Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

Frederick Bock, Harold Kantner, and John Haynes. *An algorithm (the r-th best path algorithm) for finding and ranking paths through a network*. Armour Research Foundation, 1957.

Maurice Pollack. The kth best route through a network. *Operations Research*, 9(4):578–580, 1961. ISSN 0030364X, 15265463. URL `http://www.jstor.org/stable/167129`.

S Clarke, A Krikorian, and J Rausen. Computing the n best loopless paths in a network. *Journal of the Society for Industrial and Applied Mathematics*, 11(4):1096–1102, 1963.

Michael Sakarovitch. *The k Shortest Routes and the k Shortest Chains in a Graph*. 1966.

Jin Y. Yen. Finding the lengths of all shortest paths in n -node nonnegative-distance complete networks using 1/2n3 additions and n3 comparisons. *J. ACM*, 19(3):423–424, July 1972. ISSN 0004-5411. doi: 10.1145/321707.321712. URL `http://doi.acm.org/10.1145/321707.321712`.

Andrew William Brander and Mark C Sinclair. A comparative study of k-shortest path algorithms. In *Performance Engineering of Computer and Telecommunications Systems*, pages 370–379. Springer, 1996.

Ernesto De Queiros Vieira Martins, Marta Margarida Braz Pascoal, and Jose Luis Esteves Dos Santos. The k shortest paths problem. 1998.

David Eppstein. Finding the k shortest paths. *SIAM Journal on computing*, 28(2):652–673, 1998.

Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

John Hershberger, Matthew Maxel, and Subhash Suri. Finding the k shortest simple paths: A new algorithm and its implementation. *ACM Transactions on Algorithms (TALG)*, 3(4): 45, 2007.

Sang Nguyen and Stefano Pallottino. Hyperpaths and shortest hyperpaths. In *Combinatorial Optimization*, pages 258–271. Springer, 1989.

Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete applied mathematics*, 42(2-3):177–201, 1993.

Lars Relund Nielsen, Kim Allan Andersen, and Daniele Pretolani. Finding the k shortest hyperpaths. *Computers & Operations Research*, 32(6):1477–1497, 2005.

Urszula Kanturska, Valentina Trozzi, and Michael GH Bell. Scheduled hyperpath: A strategy for reliable routing and scheduling of deliveries in time-dependent networks with random delays. *Transportation Research Record*, 2378(1):99–109, 2013.

Sang Nguyen and Stefano Pallottino. Equilibrium traffic assignment for large scale transit networks. *European journal of operational research*, 37(2):176–186, 1988.

Patrice Marcotte and Sang Nguyen. Hyperpath formulations of traffic assignment problems. In *Equilibrium and advanced transportation modelling*, pages 175–200. Springer, 1998.

Angelica Lozano and Giovanni Storchi. Shortest viable hyperpath in multimodal networks. *Transportation Research Part B: Methodological*, 36(10):853–874, 2002.

Avinash Unnikrishnan and Steven Travis Waller. User equilibrium with recourse. *Networks and Spatial Economics*, 9(4):575, 2009.

Jia Hao Wu, Michael Florian, and Patrice Marcotte. Transit equilibrium assignment: a model and solution algorithms. *Transportation Science*, 28(3):193–203, 1994.

Sang Nguyen, Stefano Pallottino, and Michel Gendreau. Implicit enumeration of hyperpaths in a logit model for transit networks. *Transportation Science*, 32(1):54–64, 1998.

Fumitaka Kurauchi, Michael GH Bell, and Jan-Dirk Schmöcker. Capacity constrained transit assignment with common lines. *Journal of Mathematical modelling and algorithms*, 2(4): 309–327, 2003.

Qianfei Li, Peng Will Chen, and Yu Marco Nie. Finding optimal hyperpaths in large transit networks with realistic headway distributions. *European Journal of Operational Research*, 240(1):98–108, 2015.

Elise Miller-Hooks and Hani S Mahmassani. Optimal routing of hazardous materials in stochastic, time-varying transportation networks. *Transportation Research Record*, 1645 (1):143–151, 1998.

Elise D Miller-Hooks and Hani S Mahmassani. Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, 34(2):198–215, 2000.

Elise Miller-Hooks. Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks*, 37(1):35–52, 2001.

Baiyu Yang and Elise Miller-Hooks. Adaptive routing considering delays due to signal operations. *Transportation Research Part B: Methodological*, 38(5):385–413, 2004.

Song Gao and Ismail Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological*, 40(2):93–122, 2006.

Huajun Chai, H Michael Zhang, Dipak Ghosal, and Chen-Nee Chuah. Dynamic traffic routing in a network with adaptive signal control. *Transportation Research Part C: Emerging Technologies*, 85:64–85, 2017.

Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In *29th IEEE Conference on Decision and Control*, pages 2130–2132. IEEE, 1990.

Michael J Neely and Rahul Urgaonkar. Optimal backpressure routing for wireless networks with multi-receiver diversity. *Ad Hoc Networks*, 7(5):862–881, 2009.

Leonidas Georgiadis, Michael J Neely, Leandros Tassiulas, et al. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends® in Networking*, 1(1): 1–144, 2006.

Amit Dvir and Athanasios V Vasilakos. Backpressure-based routing protocol for dtns. *ACM SIGCOMM Computer Communication Review*, 41(4):405–406, 2011.

Scott Moeller, Avinash Sridharan, Bhaskar Krishnamachari, and Omprakash Gnawali. Routing without routes: The backpressure collection protocol. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 279–290. ACM, 2010.

Lei Ying, Sanjay Shakkottai, Aneesh Reddy, and Shihuan Liu. On combining shortest-path and back-pressure routing over multihop wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 19(3):841–854, 2011.

Justin A Boyan and Michael L Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Advances in neural information processing systems*, pages 671–678, 1994.

Eric Bonabeau, Florian Henaux, Sylvain Guérin, Dominique Snyers, Pascale Kuntz, and Guy Theraulaz. Routing in telecommunications networks with ant-like agents. In *International Workshop on Intelligent Agents for Telecommunication Applications*, pages 60–71. Springer, 1998.

Leonid Peshkin and Virginia Savova. Reinforcement learning for adaptive routing. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 2, pages 1825–1830. IEEE, 2002.

Zubair Md Fadlullah, Fengxiao Tang, Bomin Mao, Nei Kato, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. State-of-the-art deep learning: Evolving machine intelligence toward tomorrowâĂŹs intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials*, 19(4):2432–2455, 2017.

Maurizio Palesi and Masoud Daneshtalab. *Routing algorithms in networks-on-chip*. Springer, 2014.

Ge-Ming Chiu. The odd-even turn model for adaptive routing. *IEEE Transactions on parallel and distributed systems*, 11(7):729–738, 2000.

Jingcao Hu and Radu Marculescu. Dyad: smart routing for networks-on-chip. In *Proceedings of the 41st annual Design Automation Conference*, pages 260–263. ACM, 2004.

A Intel. Touchstone delta system description. *Intel Corporation*, 1991.

Ming Li, Qing-An Zeng, and Wen-Ben Jone. Dyxy: a proximity congestion-aware deadlock-free dynamic routing method for network on chip. In *Proceedings of the 43rd annual Design Automation Conference*, pages 849–852. ACM, 2006.

Pejman Lotfi-Kamran, Amir-Mohammad Rahmani, Masoud Daneshtalab, Ali Afzali-Kusha, and Zainalabedin Navabi. Edxy–a low cost congestion-aware routing algorithm for network-on-chips. *Journal of Systems Architecture*, 56(7):256–264, 2010.

Martin Beckmann, Charles B McGuire, and Christopher B Winsten. Studies in the economics of transportation. Technical report, 1956.

Yosef Sheffi. Urban transportation networks. 1985.

Andre De Palma, IU E Nesterov, et al. Optimization formulations and static equilibrium in congested transportation networks. 1998.

Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413. Springer, 1999.

José R Correa, Andreas S Schulz, and Nicolás E Stier-Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

David E Boyce, Hani S Mahmassani, and Anna Nagurney. A retrospective on beckmann, mcguire and winsten's studies in the economics of transportation. *Papers in regional science*, 84(1):85–103, 2005.

Anna Nagurney. *Network economics: A variational inequality approach*, volume 10. Springer Science & Business Media, 2013.

John Glen Wardrop. Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers*, 1(3):325–362, 1952.

Kenneth A Small. Using the revenues from congestion pricing. *Transportation*, 19(4):359–381, 1992.

Erik Verhoef, Peter Nijkamp, and Piet Rietveld. Second-best congestion pricing: the case of an untolled alternative. *Journal of Urban Economics*, 40(3):279–302, 1996.

Kenneth A Small and Jose A Gómez-Ibáñez. Road pricing for congestion management: the transition from theory to policy. *Transport Economics*, pages 373–403, 1997.

Xiaoning Zhang and Hai Yang. The optimal cordon-based network congestion pricing problem. *Transportation Research Part B: Methodological*, 38(6):517–537, 2004.

Andre de Palma and Robin Lindsey. Traffic congestion pricing methodologies and technologies. *Transportation Research Part C: Emerging Technologies*, 19(6):1377–1399, 2011.

Deepak K Merchant and George L Nemhauser. A model and an algorithm for the dynamic traffic assignment problems. *Transportation science*, 12(3):183–199, 1978a.

Deepak K Merchant and George L Nemhauser. Optimality conditions for a dynamic traffic assignment model. *Transportation Science*, 12(3):200–207, 1978b.

Srinivas Peeta and Athanasios K Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and spatial economics*, 1(3-4):233–265, 2001.

Michael J Smith. The stability of a dynamic model of traffic assignment-an application of a method of lyapunov. *Transportation Science*, 18(3):245–252, 1984.

Terry L Friesz, Javier Luque, Roger L Tobin, and Byung-Wook Wie. Dynamic network traffic assignment considered as a continuous time optimal control problem. *Operations Research*, 37(6):893–901, 1989.

Markos Papageorgiou. Dynamic modeling, assignment, and route guidance in traffic networks. *Transportation Research Part B: Methodological*, 24(6):471–495, 1990.

Malachy Carey. Nonconvexity of the dynamic traffic assignment problem. *Transportation Research Part B: Methodological*, 26(2):127–133, 1992.

Bruce N Janson. Dynamic traffic assignment for urban road networks. *Transportation Research Part B: Methodological*, 25(2-3):143–161, 1991.

Bin Ran, David E Boyce, and Larry J LeBlanc. A new class of instantaneous dynamic user-optimal traffic assignment models. *Operations Research*, 41(1):192–202, 1993.

Athanasios K Ziliaskopoulos. A linear programming model for the single destination system optimum dynamic traffic assignment problem. *Transportation science*, 34(1):37–49, 2000.

Hani S Mahmassani. Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and spatial economics*, 1(3-4): 267–292, 2001.

WY Szeto and Hong K Lo. Dynamic traffic assignment: properties and extensions. *Transportmetrica*, 2(1):31–52, 2006.

Moshe E Ben-Akiva, Song Gao, Zheng Wei, and Yang Wen. A dynamic traffic assignment model for highly congested urban networks. *Transportation research part C: emerging technologies*, 24:62–82, 2012.

Xuegang Jeff Ban, Jong-Shi Pang, Henry X Liu, and Rui Ma. Continuous-time point-queue models in dynamic network loading. *Transportation Research Part B: Methodological*, 46(3): 360–380, 2012a.

Zhen Sean Qian, Wei Shen, and HM Zhang. System-optimal dynamic traffic assignment with and without queue spillback: Its path-based formulation and solution via approximate path marginal cost. *Transportation research part B: methodological*, 46(7):874–893, 2012.

Wei Shen and HM Zhang. System optimal dynamic traffic assignment: Properties and solution procedures in the case of a many-to-one network. *Transportation Research Part B: Methodological*, 65:1–17, 2014.

Rui Ma, Xuegang Jeff Ban, and Jong-Shi Pang. Continuous-time dynamic system optimum for single-destination traffic networks with queue spillbacks. *Transportation Research Part B: Methodological*, 68:98–122, 2014.

Feng Zhu and Satish V Ukkusuri. A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, 55:363–378, 2015.

Michael Patriksson. *The traffic assignment problem: models and methods*. Courier Dover Publications, 2015.

YW Xu, Jia Hao Wu, Michael Florian, Patrice Marcotte, and DL Zhu. Advances in the continuous dynamic network loading problem. *Transportation Science*, 33(4):341–353, 1999.

Xiaojian Nie and H Michael Zhang. A comparative study of some macroscopic link models used in dynamic traffic assignment. *Networks and Spatial Economics*, 5(1):89–115, 2005.

Peter Koonce and Lee Rodegerdts. Traffic signal timing manual. Technical report, United States. Federal Highway Administration, 2008.

Bo Ji, Changhee Joo, and Ness B Shroff. Delay-based back-pressure scheduling in multihop wireless networks. *IEEE/ACM Transactions on Networking*, 21(5):1539–1552, 2013.

Tichakorn Wongpiromsarn, Tawit Uthaicharoenpong, Yu Wang, Emilio Frazzoli, and Danwei Wang. Distributed traffic signal control for maximum network throughput. In *2012 15th international IEEE conference on intelligent transportation systems*, pages 588–595. IEEE, 2012.

Pravin Varaiya. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195, 2013.

Jian Wu, Dipak Ghosal, Michael Zhang, and Chen-Nee Chuah. Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection. *IEEE Transactions on Vehicular Technology*, 67(2):896–909, 2018.

Jossef Perl and Mark S Daskin. A warehouse location-routing problem. *Transportation Research Part B: Methodological*, 19(5):381–396, 1985.

Hokey Min, Vaidyanathan Jayaraman, and Rajesh Srivastava. Combined location-routing problems: A synthesis and future research directions. *European Journal of Operational Research*, 108(1):1–15, 1998.

Gábor Nagy and Saïd Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672, 2007.

Jacques Desrosiers, François Soumis, and Martin Desrochers. Routing with time windows by column generation. *Networks*, 14(4):545–565, 1984.

Marius M Solomon and Jacques Desrosiers. Survey paper-time window constrained routing and scheduling problems. *Transportation science*, 22(1):1–13, 1988.

Jacques Desrosiers, Yvan Dumas, Marius M Solomon, and François Soumis. Time constrained routing and scheduling. *Handbooks in operations research and management science*, 8:35–139, 1995.

Yu Marco Nie and Xing Wu. Shortest path problem considering on-time arrival probability. *Transportation Research Part B: Methodological*, 43(6):597–613, 2009.

Liping Fu. An adaptive routing algorithm for in-vehicle route guidance systems with real-time information. *Transportation Research Part B: Methodological*, 35(8):749–765, 2001.

Peng Will Chen and Yu Marco Nie. Stochastic optimal path problem with relays. *Transportation Research Part C: Emerging Technologies*, 59:48–65, 2015.

Xing Wu. Study on mean-standard deviation shortest path problem in stochastic and time-dependent networks: A stochastic dominance based approach. *Transportation Research Part B: Methodological*, 80:275–290, 2015.

HG Rosdolsky. A method for adaptive traffic control. *Transportation Research*, 7(1):1–16, 1973.

PB Hunt, DI Robertson, RD Bretherton, and M Cr Royle. The scoot on-line traffic signal optimisation technique. *Traffic Engineering & Control*, 23(4), 1982.

Hong K Lo. A novel traffic signal control formulation. *Transportation Research Part A: Policy and Practice*, 33(6):433–448, 1999.

Pitu Mirchandani and Larry Head. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415–432, 2001.

Min Chee Choy, Dipti Srinivasan, and Ruey Long Cheu. Cooperative, hybrid agent architecture for real-time traffic signal control. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 33(5):597–607, 2003.

Bogdan Tatomir and Leon Rothkrantz. Dynamic traffic routing using ant based control. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3970–3975. IEEE, 2004.

Shih-Fen Cheng, Marina A Epelman, and Robert L Smith. Cosign: A parallel algorithm for coordinated traffic signal control. *Intelligent Transportation Systems, IEEE Transactions on*, 7(4):551–564, 2006.

Jack Haddad, Mohsen Ramezani, and Nikolas Geroliminis. Cooperative traffic control of a mixed network with two urban regions and a freeway. *Transportation Research Part B: Methodological*, 54:17–36, 2013.

Joel L Horowitz. The stability of stochastic equilibrium in a two-link transportation network. *Transportation Research Part B: Methodological*, 18(1):13–28, 1984.

Terry L Friesz. Transportation network equilibrium, design and aggregation: key developments and research opportunities. *Transportation Research Part A: General*, 19(5):413–427, 1985.

Ding Zhang and Anna Nagurney. On the local and global stability of a travel route choice adjustment process. *Transportation Research Part B: Methodological*, 30(4):245–262, 1996.

MJ Smith and M Ghali. The dynamics of traffic assignment and traffic control: A theoretical study. *Transportation Research Part B: Methodological*, 24(6):409–422, 1990.

Hai Yang and Sam Yagar. Traffic assignment and signal control in saturated road networks. *Transportation Research Part A: Policy and Practice*, 29(2):125–139, 1995.

Lin Xiao and Hong K Lo. Combined route choice and adaptive traffic control in a day-to-day dynamical system. *Networks and Spatial Economics*, pages 1–21, 2014.

Ali A Zaidi, Balazs Kulcsar, and Henk Wymeersch. Traffic-adaptive signal control and vehicle routing using a decentralized back-pressure method. In *Control Conference (ECC), 2015 European*, pages 3029–3034. IEEE, 2015.

Yen-Liang Chen and Hsu-Hao Yang. Shortest paths in traffic-light networks. *Transportation Research Part B: Methodological*, 34(4):241–253, 2000.

Xiaoyan Zhang and John A Rice. Short-term travel time prediction. *Transportation Research Part C: Emerging Technologies*, 11(3):187–210, 2003.

Steven I-Jy Chien and Chandra Mouly Kuchipudi. Dynamic travel time prediction with real-time and historic data. *Journal of transportation engineering*, 129(6):608–616, 2003.

Wei-Hua Lin, Amit Kulkarni, and Pitu Mirchandani. Short-term arterial travel time prediction for advanced traveler information systems. In *Intelligent Transportation Systems*, volume 8, pages 143–154. Taylor & Francis, 2004.

Robert B Noland and John W Polak. Travel time variability: a review of theoretical and empirical issues. *Transport Reviews*, 22(1):39–54, 2002.

Fangfang Zheng and Henk Van Zuylen. Uncertainty and predictability of urban link travel time. *Transportation Research Record: Journal of the Transportation Research Board*, 2192(1): 136–146, 2010.

Bart Van Arem, Cornelie JG Van Driel, and Ruben Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *Intelligent Transportation Systems, IEEE Transactions on*, 7(4):429–436, 2006.

Steven E Shladover, Dongyan Su, and Xiao-Yun Lu. Impacts of cooperative adaptive cruise control on freeway traffic flow. *Transportation Research Record: Journal of the Transportation Research Board*, 2324(1):63–70, 2012.

Wilmer Arellano and Imad Mahgoub. Trafficmodeler extensions: A case for rapid vanet simulation using, omnet++, sumo, and veins. In *2013 High Capacity Optical Networks and Emerging/Enabling Technologies*, pages 109–115. IEEE, 2013.

David S Scharfstein and Jeremy C Stein. Herd behavior and investment. *The American Economic Review*, pages 465–479, 1990.

Abhijit V Banerjee. A simple model of herd behavior. *The quarterly journal of economics*, 107 (3):797–817, 1992.

Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

Bernhard Fleischmann, Stefan Gnutzmann, and Elke Sandvoß. Dynamic vehicle routing based on online traffic information. *Transportation science*, 38(4):420–433, 2004.

Seongmoon Kim, Mark E Lewis, and Chelsea C White. Optimal vehicle routing with real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):178–188, 2005.

Andre Gueziec. Traffic routing based on segment travel time, May 20 2008. US Patent 7,375,649.

Michel Gendreau, Gianpaolo Ghiani, and Emanuela Guerriero. Time-dependent routing problems: A review. *Computers & operations research*, 64:189–197, 2015.

Ibrahim Semiz. Icvs: A paradigm to address urban traffic gridlock and associated problems. *IEEE Intelligent Transportation Systems Magazine*, 8(2):43–52, 2016.

Lionel M. Ni and Philip K. McKinley. A survey of wormhole routing techniques in direct networks. *Computer*, 26(2):62–76, 1993.

Christopher J Glass and Lionel M Ni. The turn model for adaptive routing. *ACM SIGARCH Computer Architecture News*, 20(2):278–287, 1992.

William J Dally and Charles L Seitz. Deadlock-free message routing in multiprocessor interconnection networks. 1988.

José Duato. A new theory of deadlock-free adaptive routing in wormhole networks. *IEEE transactions on parallel and distributed systems*, 4(12):1320–1331, 1993.

Chien-Chun Su and Kang G Shin. Adaptive deadlock-free routing in multicomputers using only one extra virtual channel. In *Parallel Processing, 1993. ICPP 1993. International Conference on*, volume 1, pages 227–231. IEEE, 1993.

William J. Dally and Hiromichi Aoki. Deadlock-free adaptive routing in multicomputer networks using virtual channels. *IEEE transactions on Parallel and Distributed Systems*, 4 (4):466–475, 1993.

Andrew A Chien. A cost and speed model for k-ary n-cube wormhole routers. *Urbana*, 51:61801, 1993.

GA Mendes, LR Da Silva, and Hans Jürgen Herrmann. Traffic gridlock on complex networks. *Physica A: Statistical Mechanics and its Applications*, 391(1-2):362–370, 2012.

Carlos Daganzo. The nature of freeway gridlock and how to prevent it. 1995.

Carlos F Daganzo. Queue spillovers in transportation networks with a route choice. *Transportation Science*, 32(1):3–11, 1998.

Carlos F Daganzo. Urban gridlock: Macroscopic modeling and mitigation approaches. *Transportation Research Part B: Methodological*, 41(1):49–62, 2007.

Hani S Mahmassani, Meead Saberi, and Ali Zockaie. Urban network gridlock: Theory, characteristics, and dynamics. *Transportation Research Part C: Emerging Technologies*, 36: 480–497, 2013.

Nikolas Geroliminis, Carlos F Daganzo, et al. Macroscopic modeling of traffic in cities. In *Transportation Research Board 86th Annual Meeting*, number 07-0413. No. 07-0413, 2007.

Carolina Osorio, Xiao Chen, Michael Marsico, Mohamad Talas, Jingqin Gao, and Shitao Zhang. Reducing gridlock probabilities via simulation-based signal control. *Transportation Research Procedia*, 6:101–110, 2015.

Yong Zhang, Yu Bai, and XG Yang. Strategy of traffic gridlock control for urban road network. *China Journal of Highway Transportation*, 23:96–102, 2010.

Rutger Claes and Tom Holvoet. Gridlock: A microscopic traffic simulation platform. In *International Conference on Models and Technologies for Intelligent Transportation Systems*, 2011.

Stefan Lämmer and Martin Treiber. Self-healing networks-gridlock prevention with capacity regulating traffic lights. In *2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pages 61–65. IEEE, 2012.

Hao Yu, Rui Ma, and H Michael Zhang. Optimal traffic signal control under dynamic user equilibrium and link constraints in a general network. *Transportation research part B: methodological*, 110:302–325, 2018.

Alfred V. Aho, Michael R Garey, and Jeffrey D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.

Eric Koskinen and Maurice Herlihy. Dreadlocks: efficient deadlock detection. In *Proceedings of the twentieth annual symposium on Parallelism in algorithms and architectures*, pages 297–303. ACM, 2008.

Terrence Mak, Fei Xia, Alex Yakovlev, Maurizio Palesi, et al. Embedded transitive closure network for runtime deadlock detection in networks-on-chip. *IEEE Transactions on Parallel and Distributed Systems*, 23(7):1205–1215, 2011.

Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

Stephen Warshall. A theorem on boolean matrices. *Journal of the ACM (JACM)*, 9(1):11–12, 1962.

Michael J Fischer and Albert R Meyer. Boolean matrix multiplication and transitive closure. In *12th Annual Symposium on Switching and Automata Theory (swat 1971)*, pages 129–131. IEEE, 1971.

Bernd Heidergott, Geert Jan Olsder, and Jacob Van Der Woude. *Max Plus at work: modeling and analysis of synchronized systems: a course on Max-Plus algebra and its applications*, volume 48. Princeton University Press, 2014.

Hani Mahmassani and Robert Herman. Dynamic user equilibrium departure time and route choice on idealized traffic arterials. *Transportation Science*, 18(4):362–384, 1984.

Terry L Friesz, David Bernstein, Tony E Smith, Roger L Tobin, and Byung-Wook Wie. A variational inequality formulation of the dynamic network user equilibrium problem. *Operations research*, 41(1):179–191, 1993.

Xuegang Jeff Ban, Jong-Shi Pang, Henry X Liu, and Rui Ma. Modeling and solving continuous-time instantaneous dynamic user equilibria: A differential complementarity systems approach. *Transportation Research Part B: Methodological*, 46(3):389–408, 2012b.

Rui Ma, Xuegang Ban, and Jong-Shi Pang. A link-based differential complementarity system formulation for continuous-time dynamic user equilibria with queue spillbacks. *Transportation Science*, 52(3):564–592, 2017.

Rui Ma, Xuegang Jeff Ban, Jong-Shi Pang, and Henry X Liu. Submission to the dta2012 special issue: Approximating time delays in solving continuous-time dynamic user equilibria. *Networks and Spatial Economics*, 15(3):443–463, 2015.

Felix Caicedo. Real-time parking information management to reduce search time, vehicle displacement and emissions. *Transportation Research Part D: Transport and Environment*, 15(4):228–234, 2010.

BJ Waterson, NB Hounsell, and Kiron Chatterjee. Quantifying the potential savings in travel time resulting from parking guidance systemsâĂŤa simulation case study. *Journal of the Operational Research Society*, 52(10):1067–1077, 2001.

Paul G Höglund. Parking, energy consumption and air pollution. *Science of the Total Environment*, 334:39–45, 2004.

Liya Guo, Shan Huang, and Adel W Sadek. A novel agent-based transportation model of a university campus with application to quantifying the environmental cost of parking search. *Transportation Research Part A: Policy and Practice*, 50:86–104, 2013.

Russell G Thompson and Anthony J Richardson. A parking search model. *Transportation Research Part A: Policy and Practice*, 32(3):159–170, 1998.

Borja Beltrán Bellés, Stefano Carrese, and Emanuele Negrenti. Parking search model. In *Proceedings of the sixth triennal symposium on transportation analysis (TRISTAN)*, 2007.

Itzhak Benenson, Karel Martens, and Slava Birfir. Parkagent: An agent-based model of parking in the city. *Computers, Environment and Urban Systems*, 32(6):431–439, 2008.

Stephen D Boyles, Shoupeng Tang, and Avinash Unnikrishnan. Parking search equilibrium on a network. *Transportation Research Part B: Methodological*, 81:390–409, 2015.

Murat Caliskan, Andreas Barthels, Bjorn Scheuermann, and Martin Mauve. Predicting parking lot occupancy in vehicular ad hoc networks. In *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*, pages 277–281. IEEE, 2007.

Karel Dieussaert, Koen Aerts, Thérèse Steenberghen, Sven Maerivoet, and Karel Spitaels. Sustapark: an agent-based model for simulating parking search. In *AGILE International Conference on Geographic Information Science, Hannover*, 2009.

Fabien Leurent and Houda Boujnah. Traffic equilibrium in a network model of parking and route choice, with search circuits and cruising flows. *Procedia-Social and Behavioral Sciences*, 54:808–821, 2012.

Fan Yang and Ding Zhang. Day-to-day stationary link flow pattern. *Transportation Research Part B: Methodological*, 43(1):119–126, 2009.

Xiaozheng He, Xiaolei Guo, and Henry X Liu. A link-based day-to-day traffic assignment model. *Transportation Research Part B: Methodological*, 44(4):597–608, 2010.

Ren-Yong Guo, Hai Yang, Hai-Jun Huang, and Zhijia Tan. Link-based day-to-day network traffic dynamics and equilibria. *Transportation Research Part B: Methodological*, 71:248–260, 2015.

MJ Smith and MB Wisten. A continuous day-to-day traffic assignment model and the existence of a continuous dynamic user equilibrium. *Annals of Operations Research*, 60 (1):59–79, 1995.

Hai-Jun Huang and William HK Lam. Modeling and solving the dynamic user equilibrium route and departure time choice problem in network with queues. *Transportation Research Part B: Methodological*, 36(3):253–273, 2002.

Srinivas Peeta and T-H Yang. Stability issues for dynamic traffic assignment. *Automatica*, 39(1):21–34, 2003.

Mike Smith and Richard Mounce. A splitting rate model of traffic re-routeing and traffic control. *Transportation Research Part B: Methodological*, 45(9):1389–1409, 2011.

Richard Mounce and Malachy Carey. Route swapping in dynamic traffic networks. *Transportation Research Part B: Methodological*, 45(1):102–111, 2011.

Wen-yi Zhang, Wei Guan, Ji-hui Ma, and Jun-fang Tian. A nonlinear pairwise swapping dynamics to model the selfish rerouting evolutionary game. *Networks and Spatial Economics*, 15(4):1075–1092, 2015.

Zhen Sean Qian and Ram Rajagopal. Optimal dynamic parking pricing for morning commute considering expected cruising time. *Transportation Research Part C: Emerging Technologies*, 48:468–490, 2014a.

KW Axhausen, JW Polak, M Boltze, and J Puzicha. Effectiveness of the parking guidance information system in frankfurt am main. *Traffic Engineering+ Control*, 35(5):304–9, 1994.

Fang He, Yafeng Yin, Zhibin Chen, and Jing Zhou. Pricing of parking games with atomic players. *Transportation Research Part B: Methodological*, 73:1–12, 2015.

Zhen Sean Qian and Ram Rajagopal. Optimal occupancy-driven parking pricing under demand uncertainties and traveler heterogeneity: A stochastic control approach. *Transportation Research Part B: Methodological*, 67:144–165, 2014b.

Haitham Al-Deek and Adib Kanafani. Modeling the benefits of advanced traveler information systems in corridors with incidents. *Transportation Research Part C: Emerging Technologies*, 1(4):303–324, 1993.