Revealed Path Choice Behavior and Network Pruning for Efficient Path Finding

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Xi Zhou
Doctor of Civil and Infrastructure Engineering
Volgenau School of Engineering, 2014

Director: Mohan Venigalla, Associate Professor
Department of Civil, Environmental and Infrastructure Engineering

Spring Semester 2014
George Mason University
Fairfax, VA

DEDICATION

To my loving husband Liangbo Deng, my sweet daughter Huining, and my adorable son Ruiming.

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Mohan Venigalla, for the opportunity to participate in this challenging research project and for his valuable guidance and expert advice.

My thanks also go to the other members of my dissertation committee: Professors Mark Houck, Shanjiang Zhu, and Chaowei Yang. Their suggestions and constructive comments guided me throughout the research process.

Last but not least, I would like to thank my husband, my parents, and my parents-in-law for their love, encouragement, and support. Without them, this dissertation would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

REVEALED PATH CHOICE BEHAVIOR AND NETWORK PRUNING FOR EFFICIENT PATH FINDING

Xi Zhou, Ph.D.

George Mason University, 2014

Dissertation Director: Dr. Mohan Venigalla

This dissertation addresses two separate problems related to transportation networks. In the first part, route choice behavior revealed from real world trips is studied. In part two, efficient pruning of large transportation networks for expediting one-to-one path search is studied.

Part I:

Route-choice behavior is influenced by a variety of factors ranging from physical attributes, such as the characteristics of the street network, to abstract variables, such as personal preferences of the driver. Street network composition and network variables, such as roadway type, the mere presence and density of signalized intersections, and path characteristics, such as frequency of turn movements, are expected to have a significant influence on a driver's route-choice. This study explores the impacts of roadway type, signal control, and turning movements on route-choice by comparing observed paths in the real world and computed shortest paths for a set of origin and destination (O-D) pairs.

Robust methodologies are devised and Python scripts are developed to conduct data processing and statistical analysis.

The comparison of real paths and computed paths indicated that drivers do not necessarily choose the theoretical shortest time paths and shortest distance paths in the real world. Drivers are willing to spend more time or travel longer distance on the paths that require fewer turns. Paired sample t-tests indicated that real paths have more signalized turns than computed shortest paths. Moreover, drivers seem more prone to making a turn (left or right) at a signal controlled intersection, while at the same time trying to minimize the number of turns occurring at non-signalized intersections. Statistical evidence also indicated that drivers tend to minimize left turns along their selected path. The number of right turns (normalized per mile), on the other hand, does not have a significant influence on route-choice. The mere presence of signalized intersections along alternative routes does not influence path choice. Statistical results showed that in terms of the number of signals per mile, theoretical paths are not different from real paths. A methodology is developed to quantify the impact of turning movements in the form of turn penalties, and to integrate them into path finding algorithms. However, the optimal path yielded by incorporating turn penalties into these algorithms has not significantly increased the chance of matching theoretical paths to actual paths.

<u>Part II</u>:

Computational efficiency of path finding algorithms is very dependent on the size of the street network. Most of the shortest-path algorithms extend the search into the areas of the network that are not part of the solution to the path finding problem. For this reason, the full network must be "pruned" or narrowed to a more relevant sub-network. Commercially available route guidance systems / solutions have successfully used network pruning methods for faster and real-time solution to shortest path algorithms, but those solutions are proprietary in nature. Hence, the literature available on this subject is very limited. This research is expected to fill the gap in available literature on the methodologies for efficient network pruning.

This study also examines computational accuracy and efficiency of pruning large networks into sub-networks in the search for the shortest path between a given pair of origin and destination nodes in the network (one-to-one path search). A bounding-box approach is introduced to prune the network. Computational experiments are conducted with different buffer sizes for the bounding box. Real-world paths are analyzed for their geographic relationship to the driver's origin and destination and the concept of "proportional buffer" is introduced to define the boundaries of the sub-network. An approach to extracting a sub-network, within which the search work will be limited, is developed. Compared to the most commonly used uniform buffer method, the proportional buffer method can accelerate the computation while maintaining the same level of accuracy.

CHAPTER 1 INTRODUCTION

1.1 Research Need

On a typical street network, travelers are faced with a choice among multiple paths for travel between their origin and destination (O-D pair). This choice is of interest to transportation planners. The step of traffic assignment in the Four-Step transportation planning process requires an understanding of path or route-choice behavior. Wardrop's first and second principles of equilibrium (Wardrop and Whitehead, 1952), popularly known as Wardrop Criteria, provide a theoretical basis for formulating and solving the traffic assignment problem.

Wardrop Criteria make generalized assumptions about route-choice behavior. The criteria assume that a traveler will minimize his or her individual impedance (User Equilibrium or UE) or total system-wide impedance (System Optimal or SO). Both UE and SO problems involve only the minimization of individual or system-wide travel time, travel cost or a combination thereof. However, to a large extent how a driver chooses his or her route is subjective. Personal knowledge of the road network, road and traffic conditions, experience, and even personal preferences make different drivers perceive different "best" routes for the same O-D pair. As a result, the route-choice in the real world is stochastic rather than deterministic in nature. There is no empirical evidence that validates real world route-choice behavior conforming to Wardrop Criteria.

Despite this lack of empirical evidence, Wardrop equilibrium models have been used and are still used today, to predict drivers' choice of route in real-life street networks (Correa and Stier-Moses, 2011). It is necessary therefore, to further explore factors other than travel time or travel distance that have an impact on drivers' decisions. Research has shown that factors influencing route-choice behavior include demographic variables (such as age, gender, profession, or household structure). Furthermore, road and traffic conditions (such as time of day, travel cost, road classification, or congestion), trip characteristics (such as time of day, purpose or mode), and environment conditions (such as weather or accident), also impact drivers' decisions.

In most urban areas, the road transportation network is comprised of various roadway types (such as freeways, major and minor arterials, collectors, and local roads).Compared to suburban or rural areas, urban areas tend to have a much higher density of intersections. Therefore, traffic signals, turning movements, and road types enroute to the destination may significantly affect route-choice. Most of the research to date regarding these factors has been based on stated preference surveys. A few studies have used real-world route-choice observations, but the sample sizes have not been large enough to provide substantial conclusions. This research, on the other hand, uses a large dataset of real-world trip data to study the influence of the aforementioned network infrastructure variables on route-choice.

At the core of implementing route-choice behavior in network models is the process of finding the path, often the shortest path, in a given network. Depending on the purpose of the implementation, the search for the shortest path has three variations: 1) one-to-one search, 2) one-to-many search, and 3) many-to-many search. The objective of the one-to-one search is to find shortest path between two specific locations. The latter two modes, one-to-many search and many-to-many search, are usually confronted in travel demand modeling.

Of late, typical applications of one-to-one shortest path problem include the use of the in-vehicle navigation systems or web search for finding the path between two addresses. This application is also relevant for traveler information systems. With the maturation of Global Positioning System (GPS) and Geographic Information System (GIS) technology, the personal navigation system has been used more and more widely. Typically, the navigation system has limited computing capability, while the recommended path must be presented within a short period.

Extensive research has been conducted to improve the efficiencies of various path-finding algorithms. Most traditional methods have concentrated on improving the algorithm itself, such as the structure to store the data, or sort and search strategies. These efforts can reduce the response time, but they still require cumbersome computations. The methods may not fit the requirements of personal navigation systems in terms of either storage space or computing capacity.

Path finding algorithms are available in commercial hardware such as in-vehicle navigation systems, or Apps in present day “smart-phones” have evolved and do an excellent job in serving the customers. However, by their very nature, the methods used in these tools / applications are proprietary. The network pruning methods used in commercially available navigation tools are not known in the public domain. Available literature on network pruning has been limited. Therefore, the need for new research on network pruning methods is great in the public domain.

Heuristic methods for efficient network pruning can therefore expedite the search. The essential idea of heuristic methods is to utilize the prior information contained in the network structure, like locations of origin and destination nodes, to reduce the searching areas before, or during the course of, running the search algorithms.

The computing time required for path finding algorithms is very dependent on the size of the street network. The complexity of the search for paths increases with network size. More often than not, the one-to-one search includes only a portion of the network. An acceptable alternative route in practice should not be very far away from the proximity determined by given origin and destination locations. In other words, for a large network, only a small portion of it would be relevant to potential routes.

Therefore, if the portion of network (sub-network) can be pre-extracted and the search area can be limited within it, the computing time should be shorter. In this research the term “network pruning” is used to describe this process. Naturally, the smaller the sub-network, the faster the computation. However, if the sub-network that is used does not arrive at the same shortest path solution as found in the search of the full network, network pruning can fail. Also, network pruning would completely fail if the sub-network were to be too small to find even a single path connecting the O-D pair.

The challenge is to analytically and efficiently determine an effective sub-network for an O-D pair. A variety of experiments has been conducted to test the size and orientation of sub-networks (Karami, 2008). It is not realistic to try different sub-networks every time the navigation system plans a trip. For a given O-D pair the total time required to find the shortest path in a sub-network should not be more than the time required to find the path in the entire network.

1.2 Study Objectives

Based on the need for research stated above, the following three objectives have been identified:

1. Develop methodologies to extract trip information from the GPS dataset;
2. Examine the influence of signals and turning movements on drivers' path choices in an urban street network, as well as the distribution of road classes along paths;
3. Develop a heuristic method for pruning large networks and extract a sub-network where the one-to-one shortest path search algorithms are applied.

To meet these objectives, the research in this study relies heavily on real-world trips tracked by GPS equipment in the Twin Cities of Minneapolis-St. Paul, Minnesota.

The first objective has been accomplished by Geographic Information Systems (GIS) technologies. Algorithms were developed to identify the start and end positions of trips, to build paths represented by node and/or link sequences, to aggregate road usage by functional classifications, and to count the number of turns and signals along paths.

The second objective has been accomplished by statistical analyses of the real-world travel and network data for identifying intersection variables that influenced drivers' choices and judgments about optimal paths. Methods to incorporate the revealed route-choice behavior into the traditional path finding algorithm(s) have also been explored.

The third objective has been accomplished by developing a method to determine better size and orientation of the potential sub-network for a given O-D pair. The relationship between real-world paths and their origin/destination locations were investigated to explore the underlying pattern(s) to help build rules for extracting sub-network. Experiments were conducted to test network connectivity, efficiency and accuracy of sub-network.

The research employed data from real trips made in a major urban area. Therefore, these routes are expected to cover the full scope of drivers' reasoning and behaviors in their path finding in the real world.

1.3 Dissertation Outline

Following this introductory chapter is a review in Chapter 2 of the relevant literature on route-choice behavior and path finding methods. Chapter 3 introduces the study approach and research methodologies used in this dissertation. Chapter 4 describes efforts to preprocess on real-world trips. In other word, this chapter explains the preparation of street network data and the process of matching real-world trip trajectories to the digital network. Chapter 5 describes methodologies used to identify possible factors that may influence path choices. Chapter 6 conducts statistical analyses and explores the impacts of identified factors. Chapter 7 describes the methodology developed to determine turn penalties. Chapter 8 presents experiments, comparing and evaluating their accuracy and efficiency of sub-network concept. Finally, Chapter 9 provides a summary of the research findings and gives recommendations for future research.

CHAPTER 2 LITERATURE REVIEW

This chapter reviews past research on route-choice behavior and efficient path finding methods. It provides the necessary backdrop to understanding the data, approach and methodologies, and models adopted by previous researchers who have studies factors influencing route behavior. The chapter also reviews the state of the art in the computational complexities of path finding algorithms.

2.1 Effect of Network Variables on Route-Choice Behavior

Route-choice in the real world is not necessarily based on the path with the shortest impedance. Rather, most drivers choose a route that they perceive to be the best according to their personal knowledge and experience (Liu, 1996).

The problem of route-choice for a traveler might be stated as follows: Given the other characteristics of the trip to be made (purpose, time, origin, destination, and travel mode, for instance), attributes of the alternative routes, and traveler's personal characteristics, choose the "best" route through the transportation network following some criterion (Antonisse, Daly, and Ben-Akiva, 1989).

In practice, route-choice application is more stochastic than deterministic in nature. In a deterministic case drivers are assumed to choose the theoretically best route.

A number of probabilistic route-choice models have been developed and studied on the basis of Wardrop's principles (Wardrop and Whitehead, 1952).

The route-choice models are derived from utility theory: each person tries to maximize utility when faced with a choice among competing routes. The utility is usually defined as a function of variables, each of which represents attributes of the alternative. Coefficients are given to each attribute, in terms of their influence on utilities. A route-choice utility function can be described as:

$$U_k^n = \sum_i \theta_{in} c_{ink} \quad \forall k \in K_{od}$$

Where:

U_k^n: Utility of route *k* that is made by person *n*

n: Individual person

c_{ink}: Cost generated by attribute *i* of route *k*

θ_{in}: Positive coefficient

K_{od} : The set of all routes between a specific origin and destination pair

It is not necessary for the characteristics of each known alternative route to have the same importance in a driver's final decision. As a result, for any of the attributes the coefficient may or may not be the same for different drivers.

On the basis of the relative importance of factors of influence, the route-choice model first identifies the set of sufficiently attractive alternatives for specific travelers. From this set, travelers make their choices, with the chosen route being the one that best satisfies their needs and is consistent with their personal constraints and preferences. In the set, alternative routes may be numerous, and every route is probably not perceived by all users (Cascetta et al., 2002).

A variety of logit / probit route-choice models was developed that vary by the basic structure of the model. The multinominal probit (MNP) model (Daganzo and Sheffi, 1977) and multinominal logit (MNL) model may be considered the earliest models. With modifications to MNL, models such as the C-logit model (Cascetta et al., 1996), the implicit availability/perception (IAP) logit model (Cascetta and Papola, 1998), and the path-size logit model (Ben-Akiva and Bierlaire, 1999) can overcome the overlapping problem while still retaining the MNL structure. Other commonly used logit-based models include the PCL model of Chu (1989), the CNL model of Vovsha (1997), and the Logit Kernel (Mixed Logit) Model (McFadden and Train, 2000; Ben-Akiva and Bolduc, 1996).

The best-known factors that influence route-choice are travel distance and travel time. However, other network factors and drivers' experiences, habits, and other behavioral considerations may produce variations in route-choices. Empirical research has shown that numerous criteria could be used to formulate a route. Therefore, assuming travel time (or distance) as the sole criterion of route-choice may be an overly simplistic abstraction of individual driver behavior and may result in an inaccurate representation of traffic in transportation planning models.

In most urban areas, the transportation network is a mixture of various classes of roads. Freeways and principal arterials crisscrossing the network cater to long trips with mobility as the major function. Minor arterials, collectors, and local streets complete network connectivity necessary for access and mobility. As a result, the availability of a certain class of roads is an important factor affecting drivers' decisions on route. Compared to suburban or rural areas, urban areas have a much higher density of intersections. The number of traffic lights and turning movements along the path may significantly affect drivers' choice of route. Studies have been conducted to identify which factors influence drivers' route-choice behavior and how they influence that behavior. However, there is still a lack of literature on the impact of these three specific factors (road classes, signals, and turning movements) based on a large dataset of real-world observations.

Jackson and Jucker (1981) found that travel time reliability, defined as the difference between the 90th percentile and the median travel times, could be an important influence factor on commute route-choices. Travel time reliability may be positively correlated with criteria like number of traffic signals along a route or the safety of the route. However, the study did not explore this relevance further.

The decision-making process is a learning process, which is central to the driver's cognition (Polydoropoulou, Ben-Akiva, and kaysi, 1994). Therefore, information acquired through the experience of earlier travel choices is considered when a driver is making the next decision. Polydoropoulou, Ben-Akiva, and kaysi (1994), Srinivasan and Mahmassani (2000) have discussed the influence of travel information on route-choice theoretically. Abdel-Aty et al. (1994) conducted a stated preference survey in Los Angeles that indicated that about 60 percent of drivers listen to a travel information report. The study did not indicate how the drivers responded to this information. Two other surveys in Sweden and Israel found that on average two-third of commuters will change their travel behaviors based on real-time information (Stern, Holm and Maarseveen, 1993).

Chen and Jovanis (2003) investigated drivers' responses to incident and congestion information. They studied whether drivers would follow the guidance to change routes if they were advised to turn or switch to a freeway. Although the study found that the influence of advice on turns is significant, it did not reveal whether drivers tried to minimize the number of turns along the whole route, as the guidance suggests only which link drivers should take immediately next without showing guidance for the rest of the route.

An investigation of route-choice behavior on morning commutes (Li, 2004) compared routes that commuters most frequently chose over alternatives. Statistical analysis showed that the primary routes employed a higher freeway percentage and fewer signals than alternative routes. However, primary routes have not been compared to the computed shortest paths.

Jan, Horowitz and peng (2000) have used GPS data to investigate variations in path choice. They found that travelers often took paths that greatly deviated from the shortest paths, but they did not explain why travelers choose these routes.

Zhuang et al. (2012) examined 50 trips between four O-D pairs, and found that the experienced routes (GPS tracking routes used by taxis) have lower frequencies of signalized intersections and turning movements than theoretical shortest path routes.

A number of studies based on stated preference surveys have been carried out to identify factors influencing route-choice other than travel time and distance. Some studies pointed out that drivers tend to minimize signals and turns along the path. However, very few studies presented any empirical evidence supported by field data. Among those studies that considered empirical data, either the sample sizes were too small, or these studies did not investigate influential factors in depth. In terms of the number of traffic signals and turning movements, or the distribution of the trip among different road classes along a path, past studies did not reveal how these factors impacted path choice or qualitatively identify these impacts.

2.2 The Path Finding Problem

The shortest path problem has been widely studied for over 50 years in various fields like transportation, telecommunications, computer science, and operation research. A typical shortest path problem is defined in mathematical notations as follow:

- A directed network (digraph) that consists of a finite set of nodes *N* and a finite set of arcs (or links) $A \subseteq N \times N$ is defined as $G(N, A; c)$.
- A link $a = (i, j) \in A$ is directed from node i to node j and has an associated function c_{ij} which represents the cost that occurs on this link.
- A path *P* from origin *o* to destination *d* is a sequential list of links: *(o, i)… (j, d)*, and the total cost of it $\sum c_a, a \in P$ is the sum of costs on each individual link in this list.

- Among all the possible paths from o to d, the shortest path should be the one with the minimum total cost.

Assumptions:

- The network G should be strictly connected (i.e., for each pair of nodes u and v, there exists a directed path from u to v).
- The arc cost can be either positive or negative, but there is no directed cycle with negative cost (Gallo and Pallottino, 1988). In this study, only positive cost will be discussed.
- For the node pair i and j, if the travel is bidirectional, there exists two distinct directed links (i, j) and (j, i). They may or may not have the same link cost.

In a transportation network, link cost represents the impedance of an individual vehicle traversing the link, which is usually defined by link length or link travel time. Considering differences in travel speeds between various links on the same road, the link travel time is probably a more accurate description than link length in the cases of routing motorized vehicles.

An important concept regarding the shortest path problem is the shortest path tree (SPT). A shortest path tree *SPT(r)* can be defined as a spanning directed tree contained in the network G and rooted at a given node *r*. For each $v \in N$, this tree contains a shortest path from *r* to *v* (Lawler, 1976). For all three modes of shortest path problem (one-to-one, one-to-many, and many-to-many), how to find the SPT of the given origin node is the most critical step to solve the shortest path problem. Only for the one-to-one search, it may not need to find all nodes on the completed shortest path tree. Once the destination node is reached and the shortest path to the destination is determined, the search could stop.

2.2.1 Traditional Path Finding Algorithms

Most traditional algorithms are based on the labeling method that divides all nodes in the network into three sets: unreached, labeled, and scanned. The scanned nodes have been determined to be included in the shortest path tree, and the labeled nodes are those that are waiting to be selected as the next node to scan. Thus, the labeled nodes form the set of candidates.

Gallo and Pallottino (1988) developed a prototype algorithm to implement the labeling method to find a shortest path tree starting from a root node (Figure 2-1). The $FS(u)$ in this prototype denotes the forward star of node u. It is a set of links which start from the node u to its neighbors. $FS(u) = \{(u, j) \in A\}$.

```
Procedure SPT(r):
   begin
      TINIT(r); QINIT(r);
      repeat
         QOUT(u);
         foreach (u, v ) ϵ FS(u) do if D[u] + luv < D[v] then
            begin QIN(v); TUPDATE(u, v) end
         until Q = empty
      end;

Procedure TINIT(r):
   begin
      for i:=  1 to n do begin P[i] := r; D[i] := +∞ end;
      P[r] := 0; D[r] := 0
   end;

Procedure QINIT(r):
   begin
      Q := {r}
   end;

Procedure TUPDATE(u, v):
   begin
      P[v] := u; D[v] := D[u] + luv
   end;

Procedure QOUT(u):
   begin
      select u ϵ  Q; Q :=  Q – {u}; update Q
   end;

Procedure QOUT(u):
   begin
      if not u ϵ  Q then Q :=  Q  + {v};
      update Q
   end;
```

Where:

r – the root node where the tree starts

Q – the set of candidate nodes

u, v – network nodes

luv – the link connecting *u* and *v*

P[u] – the precedent node of *u*

D[u] – the cost from *r* to *u*

TINIT(*r*) – initial the tree starting from root *r*

QINIT(*r*) – initial the candidate set *Q*

TUPDATE(*u, v*) –update the tree with the link *luv*

QOUT(*u*) – remove scanned node *u* out of *Q*

QIN(*v*) – add node *v* into *Q*

Figure 2-1: Prototype Algorithm of the Labeling Method

In this prototype, the most crucial procedure is *QOUT*, which selects the next eligible node u to scan from the candidate set *Q*. There exist various data structures to manage the candidate set *Q*, and various ways to examine and operate the set. Generally, most of shortest path algorithms follow this prototype, whereas the variations in the implementation of data structures resulted in a variety of algorithms in practice.

The Bellman-Ford-Moore algorithm (Moore, 1957; Ford, 1956; Bellman, 1958) is one of the first shortest path algorithms based on the labeling method. It maintains the candidate set in a First-In-First-Out (FIFO) queue. The next node to be scanned is the one at the head of the queue, no matter whether it has the minimum label. The Bellman-Ford-Moore algorithm cannot provide the shortest path between two specific nodes until the complete shortest path tree is identified, so it is more suitable for the problems that need a one-to-many search or many-to-many search. As a result, the Bellman-Ford-Moore algorithm is often used in transportation planning and travel demand modeling where multiple routes need to be identified.

Dijkstra's algorithm (Dijkstra, 1959) is another early proposed and widely known shortest path algorithm. It uses a different strategy to label the nodes. Dijkstra's algorithm maintains a linked-list for candidate set Q. After the full list is examined, the node with the minimum label in current Q is selected as the next node to be scanned. With this strategy, Dijkstra's algorithm can be terminated once the destination node becomes the next node eligible for scanning; there is no need to complete the whole shortest path tree. Thus, it is quite appropriate for a one-to-one search, often used for personal navigation systems.

The different features in the practical application, like the one addressed above, can distinguish these two algorithms as either a label-correcting algorithm (LC), the Bellman-Ford-Moore algorithm; or a label-setting algorithm (LS), Dijkstra's algorithm. Most algorithms developed later can be classified into these two categories (Fu, Sun and Rilett, 2006).

Pape (1974, 1980, and 1983) proposed a LC algorithm with a double-ended queue to manage the labeled nodes. Here, a node is added to the queue from either the head or the tail and removed from the head only. Another similar algorithm (Pallottino, 1979 and 1984) uses two queues, one's tail connected by the other's head. The nodes can be added to both queues at their tails, but removed from the head of the first queue only.

Both the algorithm with a double-ended queue and the algorithm with two-queue separate the candidate nodes into two subsets, and different operations are applied to nodes according to which subset the nodes belong. The two subsets are updated dynamically during the scanning course. Glover et al. (1984 and 1985) adapted this idea and improved it by using a threshold label value to determine the two subsets (queues). One subset, where nodes are removed at the head, only contains the nodes with labels less than, or equal to, the threshold value.

In implementing the shortest-best-search idea, Dijkstra's algorithm does not perform well when the number of links is large; therefore, a great many of the nodes need to be labeled. However, several research efforts have focused on improving the implementation of Dijkstra's algorithm. Replacing the unordered list with an ordered two-way linked-list, for instance, can improve efficiency.

The major difference among the improved implementations of Dijkstra's algorithm is the data structure used to maintain the set of labeled nodes. Dial (1969) proposed an implementation taht maintains an array of buckets to operate the candidate set Q. The *ith* bucket contains all nodes with label value i. When the label of a node is updated, the node is moved from one bucket to another one. The next node to be scanned is always taken from the first non-empty bucket.

The heap structure is another way to implement Dijkstra's algorithm. The most used heap structures include the binary heap (Williams, 1964; EL Johnson, 1972; DB Johnson, 1977), the R-heap (Ahuja et al., 1990), the Fibonacci heap (Fredman and Tarjan, 1987), and the k-array heap (Cormen et al., 1990).

As mentioned before, the shortest path problem has been studied for decades. Many researchers have derived algorithms as well as experiments regarding efficiency and accuracy (Gallo and Pallottino, 1988; Cherkassky, Goldberg, and Radzik, 1993).

The efficiency of an algorithm is designated by its complexity, which is a theoretical reference regarding the performance of the algorithms. The time complexity of an algorithm is commonly expressed using big O notation, which excludes coefficients and lower order terms. For example, if the time required by an algorithm on all inputs of size n is at most $3n^3 + 2n$, the asymptotic time complexity is $O(n^3)$. Apart from time complexity, algorithm's space complexity is also important: This is essentially the number of memory cells which an algorithm needs.

Table 2-1 summarizes the time and space complexities of the shortest path algorithms that are most commonly used. In the table:

n: The number of nodes in the network

m: The number of links in the network

l_{max}: The length of the longest link in the network

Table 2-1: Complexities of Traditional Shortest Path Algorithms

Algorithm	**Time Complexity**	**Space Complexity**
Dijkstra		
Basic implementation	$O(n^2)$	$4n+2m$
Using bucket structure	$O(m + nl_{max})$	$5n+2m+l_{max}$
Using binary heap	$O(nlogn)$ with sparse graph	$5n+2m$
Bellman-Ford-Moore		
Basic implementation	$O(nm)$ $O(N^3)$ with complete graph	$4n+2m$
With parent-checking	$O(nm)$	$4n+2m$
Double-ended queue	$O(2^n)$ $O(n \times 2^n)$ with complete graph	$4n+2m$
Two queues	$O(n^2m)$	$4n+2m$
Threshold	$O(n^2m)$	$5n+2m$

To further explore how the algorithms perform from an empirical perspective, a variety of experiments have been designed in many different studies. Dijkstra's algorithm with double buckets was found to be the best for non-negative arc length (Cherkassky, Goldberg, and Radzik, 1993).

In Gallo and Pallottino's experiment (1988), the threshold algorithm was found to be faster than those with a double-ended queue or with two queues, and these two algorithms behaved very much alike. In terms of variations of Dijkstra's algorithm, the binary heap performed better than the bucket structure. The arc-length range affected the performance of the Dial's algorithm. When the maximum arc length increased, the algorithm slowed down dramatically.

Fifteen shortest path algorithms were tested using real road networks (Zhan and Noon, 1998). The algorithms with a double-ended queue and two queues are the best performing implementations for one-to-many searches in either large or small networks. Dijkstra's algorithm with bucket structure is recommended for one-to-one searches, although performance directly depends on the maximum link length. When the maximum link length is less than 1,500 units, Dijkstra's approximate buckets implementation is the fastest one, and Dijkstra's double buckets implementation is also an appropriate alternative otherwise.

2.2.2 Heuristic Methods

In most cases of real-world vehicle navigation, the "best" route is not necessary (Uchida, Iida and Nakahara, 1994). Instead, the "optimal" route that can be obtained in an acceptable time range is much more valuable. The overall accuracy and time performance are of particular interest in the real-time path finding applications. The traditional path finding algorithms are not suitable for large networks with the real-time processing constraint (Karimi, 1996; Zhan and Noon, 1998). For this reason, heuristic methods get more attention in research related to the real-time shortest path problems.

Heuristic methods utilize known information to reduce the search space, thereby reducing the total computing time. This reduction could happen before or during the running of the algorithms.

The A* algorithm is probably the best-known heuristic method (Hart, Nilsson and Raphael, 1968; Nilsson, 1971; Pohl, 1971; Pearl, 1984) for path finding problems. It is similar to the basic implementation of Dijkstra's algorithm except that the A* uses an evaluation function $F(i)$ as the label of a node i. This label consists of not only the current path length $L(o,i)$ from the origin node to node i, but also an estimation of distance $e(i,d)$ that is from node i to destination node. With this estimation $e(i,d)$, which is usually the Euclidean distance between node i and destination node, the evaluation function could reflect how likely the shortest path could go through a node i. The smaller the function of a node, the more likely the node would be on the shortest path.

The A* algorithm can make the node expansion always move forward toward the destination node. Therefore, the algorithm can avoid examining all intermediate nodes, most of which would turn out to be irrelevant to the shortest path. Sedgewick and Vitter (1986) found that the A* algorithm could find the shortest path with an average $O(n)$ time complexity.

The bi-directional method (Dantzig, 1960) starts the search from origin and destination nodes and proceeds simultaneously; two shortest path trees rooted from origin and destination are built respectively. When the two trees meet at the "middle" between origin and destination, the search procedures stop according to the predefined criterion.

In the bi-directional method, the defined "stopping" criterion is the most critical factor. Nicholson (1966) proposed one such criterion to guarantees to find the shortest path, but the algorithm is too strict to perform better than traditional algorithms. Many other researchers have made great achievements based on this criterion, solving the problem that the two search procedures may pass each other (Pohl, 1971; Fu, 1996).

As most traffic networks remain relatively constant over time, the preprocessing can be done as a trade-off between storage space and computation time. All of the possible shortest paths in the graph could be computed first, and retrieved when needed. Based on this idea, Wagner and Willhalm (2003) tested a network preprocessing technique that builds a subset for each individual edge. These subsets store the nodes that can be reached by a shortest path starting with this edge. When the algorithms are applied, only the nodes stored in the subsets could be visited, which reduces the search area. With its intensive computation and large storage spaces, this method does not impact the correctness of shortest path algorithms. In a number of cases, the method is able to reduce the search space to 5% to 20% and only take 1/10 processing time compared to the base case. Essentially, this technique trades storage space for response time, which may not be suitable for personal/vehicle navigation systems because this kind of device usually does not have enough storage space.

2.2.3 Network Pruning

The objective of network pruning is excluding the partial network where the node search hardly reaches. As a result, the computation can be reduced. Hierarchical network is an often used network pruning method. This method is based on the idea that people, with their knowledge and experience, tend to choose the roads with higher levels and/or roads familiar to them (Liu, 1996).

Car and Frank (1993) proposed a conceptual model of human reasoning process to find the path in a hierarchically structured street network. In this network, roads were grouped into three hierarchical levels according to mobility and accessibility: U.S. interstate highways, U.S. federal highways, and state (local) highways. They examined the examples in real cases of path finding. These facts form the primary hypothesis that the fastest path can be found by just considering the highest appropriate level of the network. The hypothesis contributed to formalize the main consideration of the path finding in the multilevel road network. Although this study did not produce concrete results but the conceptual model only, many other studies have designed and conducted experiments to examine the superiority of the hierarchical concept to the traditional algorithms.

Most experiments used two levels of road network (Chou et al., 1998; Jung and Pramanik, 2002). Within the proximities of origin and destination nodes, the search algorithms are applied to the lower level, which has more details, to make sure the OD nodes can be connected to the network. Between the two proximities, the search algorithms are applied to the higher level that only contains the major roads, so the search space could be reduced efficiently.

The hierarchical concept has proved effective in reducing the complexity of the shortest path problem, although the actual computational savings are influenced by network topology, search rules and trip lengths. A computational study by Car and Frank (1993) found that the hierarchical method could be two times faster than the non-hierarchical method on small networks with fewer than 400 nodes and 600 links. Another test using a larger real network (12,697 nodes and 30,867 links) suggested that the hierarchical method could be 8 to 10 times faster with two levels of road layers (Liu, 1997).

Cho and Lan (2009) conducted a hybrid method on a real network in Taiwan. This method integrates the hierarchical concept into the traditional Dijkstra's algorithm and/or heuristic A* algorithm. A two-level hierarchical network is used. The original problem was broken into three components: two on the lower level within the proximities of origin and destination nodes, and the third on the higher level. Three combinations of algorithms were tested; D-D-D and A-A-A applied Dijkstra's algorithm and A* algorithm for all three components, respectively, and A-D-A ran the A* algorithm for the component on the higher level and Dijkstra's algorithm on the higher level.

There was a "noticeable computation speed-up and memory saving related to the reduction of search space" in the hierarchical scenario, compared to the non-hierarchical scenario (Cho and Lan, 2009). The hybrid algorithms are found to be 20,000 to 80,000 times faster than the traditional Dijkstra's algorithm, with travel time only 5% longer. As a comparison, A* algorithm without network preprocessing is only 1.8 times faster than Dijkstra's algorithm, with comparable extra travel time on found paths. This indicates that the hierarchical network is a critical factor regarding the significant reduction in response time.

Jagadeesh, Srikanthan and Quek, (2002) proposed a similar approach to utilize the hierarchical network. They considered the three components of the problem as a whole, using the heuristic shortest paths on the lower level network as trial bases when looking for the nodes where the search actions switch between the two levels. The combined method was 50 times faster than the pure A* algorithm, and the found paths were only 3.31% longer on average.

The sub-network concept has been proposed as another network pruning method. The objective of the sub-network concept is limiting the node search to a smaller area extracted from the original network before the search algorithms are applied. Obviously, the extracted sub-network can reduce the computations by limiting the search space.

However, there exists a risk that a path cannot be found in the sub-network. For this reason, it is critical to build extraction rules to make the sub-network reduce search space efficiently as well as maintain the effectiveness and accuracy of the paths found.

Karimi, Sutovsky, and Durcik (2008) designed and conducted an experiment on two types of rectangular sub-networks. The experiment contained a variety of tests on different sizes of sub-networks, as well as on the entire network with the same algorithm applied. The results showed that both approaches are about 6 to 10 times faster than the baseline with acceptable errors.

Empirical achievements obtained from this study also showed that the performance and accuracy of sub-networks depend on the characteristics of origin/destination locations and the road network itself. However, the study did not reveal the relationships between these characteristics and the size and orientation of windows. It also did not reveal the underlying patterns of drivers' actual route-choices regarding the features of the network. The investigations on real paths will be useful and critical for identifying these relationships and patterns that help find certain rules to determine the optimal size and orientation of the sub-network once a specific origin-destination pair is given.

2.3 Summary of State-of-the-Art-Review

Most research on path choices lacks sufficient field data to identify factors that may be influential. Moreover, few studies have developed methods to quantify certain impacts.

Heuristic methods, including network pruning, are a better option than traditional algorithms for one-to-one, real-time path finding applications. One of the network pruning methods, the sub-network concept, has not received as much attention as the hierarchic network. Few studies have conducted experiments to test the performance of the sub-network. However, not a single study has explored what kind of method can determine an optimal sub-network for a specific path finding problem.

CHAPTER 3
STUDY APPROACH AND RESEARCH METHODOLOGY

The preceding review of state-of-the-art literature review has provided insights into data, tools and techniques necessary to achieving study goals and objectives. Using these insights, the study approach and methodologies have been developed. These methods include development of tools to extract path information, identify the influence of signals, turns, and road classes on route-choice behavior, and prune the large street network. Appropriate statistical tools are applied to develop the experimental setup and analyze data. Figure 3-1 illustrates the study approach.

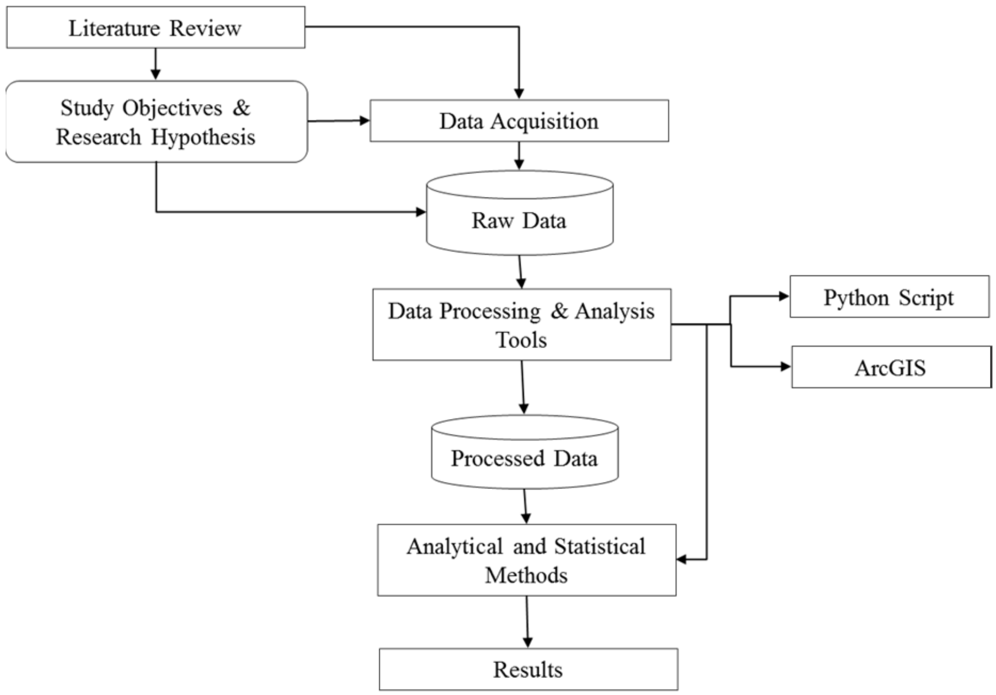


Figure 3-1: Approach of This Study

Information extraction from the large sample of field data is required for examining how signals, turns, and road classes influence path choices, as well as for developing a method for street network pruning to improve path finding algorithms for one-to-one search.

GPS-tracked path trajectories collected for a study in the metropolitan area of Minneapolis-St. Paul (the Twin Cities), Minnesota, during the period from September to December 2008, are acquired. The dataset contains information on trips made by 44 randomly selected volunteers in the 21 to 65 year age group. They commuted alone and made travel choices without any instructions. A GPS device was installed in the vehicle of each study participant. The device recorded the travel trajectories of each vehicle at a frequency of one GPS location point per second. The geographic location and time stamps of each point were documented and projected onto the ArcGIS shape files for post-processing. The GPS data will be matched to a digit street network obtained from the National Geospatial Program of the U.S. Geological Survey's (USGS).

3.1 Research Methods and Tools

Research tools and methodologies used or developed to fulfill the study objectives include the following:

1. Map-matching procedure,
2. Python scripting to perform geometric computations; and identify through, left-turn, and right-turn movements;
3. Python scripting to implement path-finding algorithm;
4. Experimental design to facilitate testing of research hypothesis;
5. Statistical analyses; and
6. Python scripting for network pruning and computational measurements

All trips made by each participant are stored in a single ArcGIS file. A Python script was developed to distinguish individual trips by identifying the start and end points of each trip. The map-matching algorithm is based on a built-in ArcGIS function that is also implemented by a Python script. This script is used to snap the GPS trip points to the digitized street network and thereby facilitate extraction of the real paths traversed on the network.

The shortest time and distance paths are computed using Dijkstra's algorithm, which is also implemented as a Python script. Another script performs geometric computations on network nodes and links to identify left or right turns along each real path or shortest path. The number of turns and signals of the path are counted while performing the geometric computations are performed.

Descriptive statistical analyses are performed to examine how signals, turning movements, and the distribution of road classes influence path choice. Specifically, the experimental setup calls for paired sample t-tests between real paths and theoretical shortest time / distance paths.

An experiment designed to evaluate the computational efficiencies of network pruning methods in the path finding problem was devised. The tool for conducting this experiment also applies Dijkstra's algorithm on a series of sub-networks. Test results are compared to paths obtained using the entire network so as to verify the accuracy and efficiency of sub-networks.

3.2 Research Assumptions

In shortest path computation, the posted speed limits are used as travel speeds of specific road segments. In the shortest path problem, the travel time of each individual road link, as the impedance, is computed from the length and the travel speed of this link. The estimation of the travel speed should be based on statistics, but not the real-time speed of individual vehicles.

CHAPTER 4
STREET NETWORK AND MAP-MATCHING

4.1 Preparation of Street Network

The road network used for this research was obtained from the National Geospatial Program of the U.S. Geological Survey's (USGS). The data cover seven counties in the metropolitan area of Twin Cities, Minnesota. The network contains full paths of all trips in the dataset. The USGS files use a census feature class code (CFCC) to classify streets and indicate some attributes of streets. For example, the USGS files may depict streets with opposing traffic lanes as two distinct lines; in this case, the road is called "separated".

The CFCC is a three-character code that describes road class (such as primary road or local road) and minor category, which indicates whether the street is separated by median, in a tunnel, or on a bridge, etc. For example, the code 'A15,' which represents interstate highways, depicts two distinct lines (opposing traffic direction), and a line coded as "A41" means it is a two-way local street.

The street network contains 23 road classes. Table 4-1 shows the number of links and the total length of links in each class.

Table 4-1: Street Network Links by Census Feature Class Code

Major Category	Minor Category	Code	Number of Links	Total Length of Links (miles)
Primary road with limited access or interstate highway	Unseparated	A11	561	75.575
	Unseparated, in tunnel	A12	5	0.301
	Unseparated, underpassing	A13	162	6.303
	Separated	A15	1488	283.607
	Separated, in tunnel	A17	218	22.645
	Separated, underpassing	A19	26	1.391
Primary road without limited access, US highways	Unseparated	A21	815	105.070
	Unseparated, underpassing	A23	24	1.192
	Separated	A25	1247	205.684
	Separated, in tunnel	A27	71	7.299
	Separated, underpassing	A29	16	0.671
Secondary and connecting road, state highways	Unseparated	A31	3500	386.896
	Unseparated, underpassing	A33	16	0.894
	Separated	A35	1812	260.362
	Separated, in tunnel	A37	26	2.087
	Separated, underpassing	A39	34	1.363
Local, neighborhood, and rural road, city street	Unseparated	A41	175551	16,561.8333
	Unseparated, in tunnel	A42	1	0.351
	Separated	A45	2668	233.835
	Separated, underpassing	A49	170	4.963
Vehicular trail	Unseparated	A51	38	4.704
Access ramp		A63	2339	295.586
Alley		A73	36	2.920

The street network data does not contain information about the existing speed limits on the network links. Since travel speeds are essential for determining travel times along paths, the following Minnesota Department of Transportation speed limits are used for different road categories:

- 10 mph in alleys
- 30 mph on streets in urban districts
- 55 mph on other roads
- 65 mph on expressways

- 65 mph on urban interstate highways
- 70 mph on rural interstate highways.

The travel time on each link can easily be calculated by dividing the link length by the traveling speed.

Also, the USGS files do not contain signal information. It was obtained from the local jurisdiction. The signal data is a point feature on GIS that was attached to the nearest intersection based on the coordinates and added to the network as one of the attributes of the nodes. Tables 4-2 and 4-3 provide a summary of the geometry of the network used in this study.

Table 4-2: Attributes of Link and Node

Element	**Attribute Name and Description**	**Attribute's Data Type**
Link: polyline	*ID*	Long
	From Node: *ID* of starting node of this link	Long
	To Node: *ID* of ending node of this link	Long
	One-Way	Boolean
	Length: length of link in miles	Double
	Time: link travel time in minutes	Double
	Class: functional classification of link	Text
Node: point	*ID*	Boolean
	x	Double
	y	Double
	Signal: if traffic light at this node	Boolean

Table 4-3: Statistics on the Street Network

Number of Nodes		149,916
Number of Links		370,079
Link/Node Ratio		2.47
Number of Signalized Intersection		3,014
Link Length (Miles)	Maximum	2.574
	Mean	0.097
	Stand. Dev.	0.104

4.2 Real Path Identification

A three-step methodology is developed to identify real paths as shown in Figure 4-1. The first step generates individual trips from the dataset of GPS-tracking points. Then these trip points are snapped to the street network by a map-matching algorithm, yielding paths represented by both node sequence and link sequence. In the last step, further path screening eliminates invalid paths, producing the set of real paths for later analysis.

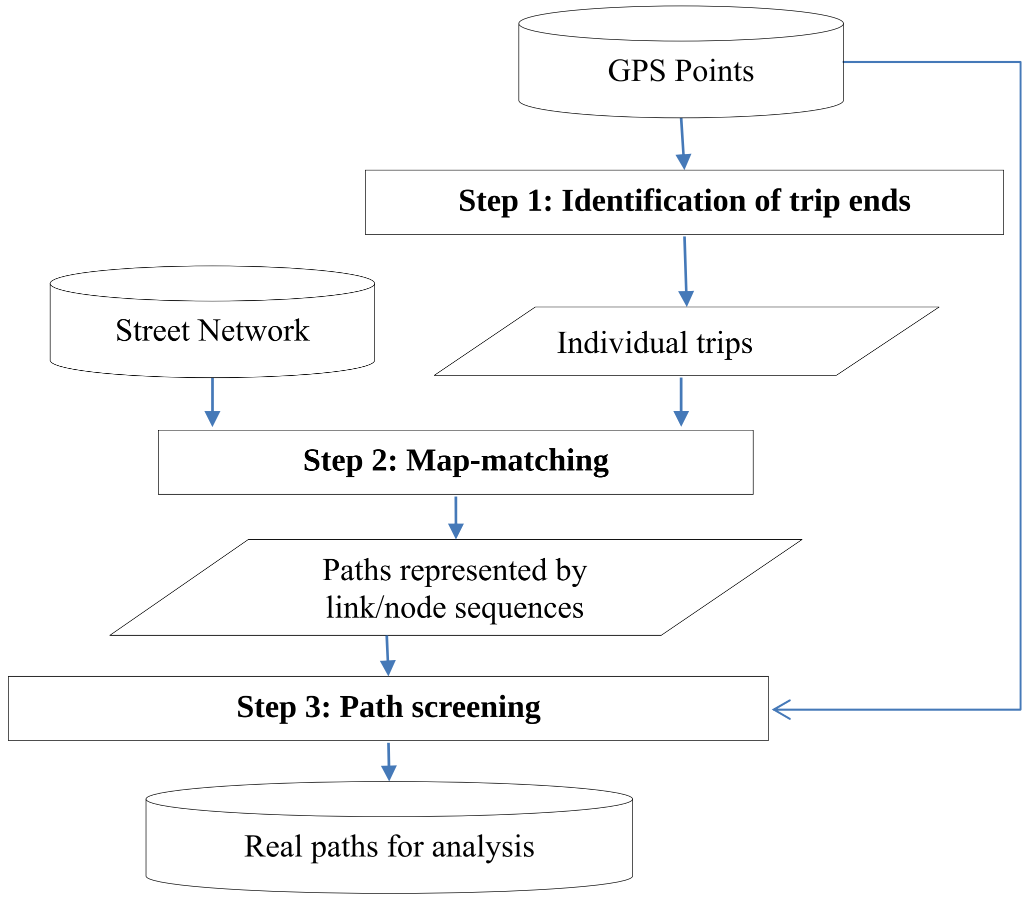


Figure 4-1: Process to Identify Real Paths from GPS-Tracking Data

4.2.1 Identification of Trip Ends

Real world trips used for this study were made by 44 volunteers in the metropolitan area of Twin Cities, MN, during the period from September to December 2008. The path for each trip consists of a sequence of points tracked by GPS devices at one-second intervals. The dataset contains 44 subsets and each of them stores path trajectories of all trips made by the same driver. These trajectories need to be distinguished trip by trip for the next analysis.

Ideally, if the interval between two successive points is more than one second, the two points may be treated as points belonging to two different trips. However, when a satellite signal is lost, while the driver is still on the trip, the data on trip trajectories may be broken. This could lead to erroneously splitting a single trip into more than two trips. Another exception also can occur. If the driver finished a trip when the GPS device was still on, the dwelling time would be included and the two trips would be treated as one.

Processing of trip data to identify individual trips from the large set of GPS data involved the following effort:

1) GPS points that are away from the road network are removed from the dataset.

   It is assumed that the vehicle may not stay on a digitized road at the end of the trip. This assumption is expected to help screen out the data pertaining to trips in which the GPS device is still on when the driver has completed a trip. As the trajectories are not aligned with a network link, a threshold is needed. With "trial and error" and a random manual check, it is determined that a threshold of 30 meters could provide a good measure.

2) Determining the minimum possible time gap requires an identification process for the next trip. Intuitively, there must be a threshold below which a new trip is not possible. However, if the gap is larger than the threshold, further analysis would be needed to determine whether it is because of a trip-end or just signal loss period. Previous research showed that 30 seconds is a good threshold for the minimum time gap (Du and Aultman_Hall, 2007).

3) Distinguishing signal loss from trip-end for the time gap longer than the minimum threshold of 30 second is the third important step in identifying individual trips. If the time gap were caused by signal loss, the average speed of the vehicle during this gap would not be much different from average speeds before and after the gap, if the driving pattern were assumed constant. The average speed during the gap can easily be estimated from the time and distance recorded by GPS devices. The highest free flow speed is 70 mph, and average speeds on most streets are no more than 50 mph. Therefore, it would be reasonable to identify a trip-end when the average speed during the time gap is 50% less than speeds before and after the gap. Du and Aultman-Hall (2007) suggested using 20 points before and after the gap to calculate the assumed driving speed. This proved enough to obtain successful identification.

This methodology has proved to be very effective in identifying individual trips. Certain special situations posed challenges to this methodology. For example, if the vehicle met with traffic congestion or traveled from an expressway to a local road when the GPS device was experiencing a signal loss, a continuous trip would be split because the average speed reduction could be over the threshold. Such instances are assumed to be rare and could not be avoided. A few random checks have shown that this identification process could yield better results than the results obtained only using the original time stamps recorded by GPS devices.

4.2.2 Map-matching Algorithm

A route can be obtained by connecting GPS points according to the time order in which they are recorded. However, the route may not match any links on the street network in many cases, due to either an error in the GPS location or an inaccurate digital road network. It is necessary to snap the GPS points of each trip to the digital street network. This technique is called map-matching. Real trip paths are identified in the format of node and/or link sequences between trips' origins and destinations.

Numerous studies have developed procedures to perform map-matching effectively and accurately. Map-matching algorithms can be classified into three categories (Bernstein and Kornhauser, 1996; Quddus, Ochieng, and Noland, 2007):

a) geometry based algorithm

b) geometry and topology based algorithm

c) probability based algorithm

A geometry based algorithm makes use of only geometric information provided by the digital network, and it does not consider the connectivity between candidate links. It may be the easiest way to do map-matching, but is not considered accurate enough. By introducing topology information, a geometry-and-topology-based algorithm can reduce the set of candidate curves dramatically and improve accuracy (Bernstein and Kornhauser, 1996). A probability-based algorithm selects matched link from multiple candidates within a buffer of the GPS point. The evaluation criteria may include heading, connectivity, and closeness (Quddus and Noland, 2007).

According to the elements involved, map-matching algorithms may also be classified as:

a) point-to-point,

b) point-to-curve, and

c) curve-to-curve.

The curve-to-curve method has rarely been used in GPS data processing because this method is sensitive to outliers (Quddus et al., 2007), and the trip dataset is based on points. The point-to-curve method, which matches a GPS point to a curve with shortest perpendicular distance, is more advanced and accurate compared to point-to-point method (Bernstein and Kornhauser, 1996; Quddus et al., 2007). The point-to-curve approach with consideration of link connectivity is employed in this study for performing map-matching.

ArcGIS software and the embedded tools in ArcGIS are used extensively for map matching. Where necessary, custom tools within or outside the ArcGIS environment were developed. ArcGIS function *GENERATE_NEAR_TABLE* can find the nearest link for each GPS point. Since GPS collects points second by second, multiple points may have a common nearest link, thus, the link will not be kept in the sequence repeatedly if it is the same as its previous link.

The set of candidate links identified by ArcGIS may contain links that are not on real paths used by trips in the dataset. On the other hand, a few links forming the real path may be missing, because neither the GPS device nor the digital street map is 100% accurate and/or compatible. To screen out the incorrect links and find back the missing links, further processing of the digital network is needed. This processing requires examination of connection between successive links to make sure the link sequence is consistent with the real travel route and direction.

Figure 4-2 illustrates three potential cases of incorrect matching. In Case A, one or more links is missed between two successive candidate links because the links are too short or the GPS device has malfunctioned. Case B usually happens on the primary road where two traffic directions are represented by two distinct lines in the digital map. In this case, the GPS point is snapped to the opposing line. The algorithm should replace the wrong link with its counterpart on the correct direction. Case C occurs at intersections where the crossing street line should be eliminated from the sequence.

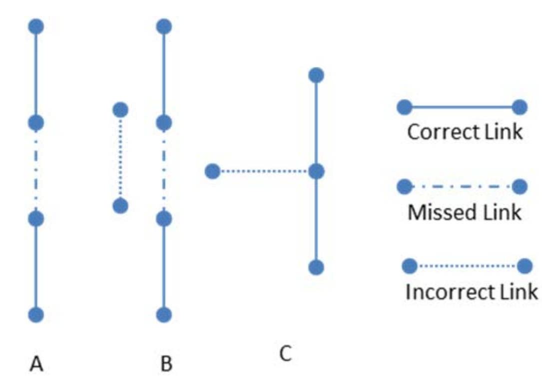


Figure 4-2: Cases of Incorrect Map-Matching

The map-matching algorithm developed in Python for this study eliminates wrong links, while at the same time retrieving missing links. The algorithm also generates a new node sequence that is consistent with the actual trip. The first and last nodes in the node sequence of each trip are identified as the origin and destination of the trip. The following pseudo code (Figure 4-3) describes the map-matching algorithm.

```
function build_new_sequence(oldSequence):
        newSequence := []
        curr := oldSequence[0]
        i = 1
        while i < len(oldSequence):
                next := oldSequence[i]
                if curr.id = next.id:
                        i = i + 1
                else:
                        if connection(curr , next) = True
                                newSequence.append(curr)
                                curr := next
                                i = i + 1
                        else:
                                if curr.street = next.street and shortest_path(curr, next) < threshold:
                                        newSequence.append(shortest_path(curr, next))
                                        curr := next
                                        i = i + 1
                                else:
                                        if len(newSequence) > 0 and curr.street = newSequence[-
                                1].street:
                                                i = i + 1
                                        else:
                                                if len(newSequence) > 0:
                                                        curr = newSequence[-1]
                                                        newSequence.remove(curr)
                                                else: return Node
        return newSequence
```

Figure 4-3: Pseudo Code of the Map-Matching Algorithm

About 8% of total trips (1,668 out of 20,174 trips) failed in map matching due to errors attributable to the digital street network (for example, topological errors, missing links, or incorrect link configuration). Figure 4-3 illustrates an example of successful a map matching.

Figure 4-4: Example of Matched Path

4.2.3 <u>Path Screening</u>

The map-matching process reduced the data to 18,560 trips from 20,174 trips. For identifying traversable paths in the network, the data are further screened by adherence to the following criteria and by eliminating the trips that do not fit the criteria:

a) A path requires at least two links. If a path contains fewer than three nodes, the trip is eliminated from the analysis dataset.
b) Trips with path lengths shorter than one minute in travel time indicate a potentially faulty GPS device and therefore are dropped from the analysis dataset.
c) If multiple paths made by the same driver are identical to each other, they are assumed to be commuting trips. Since such duplicate trips do not provide additional insights into the route-choice behavior of the driver, making those trips, only one of these multiple trips is included in the analysis dataset.

After this screening process is complete, the remaining 5,694 trips were identified with a valid path represented by both a link sequence and a node sequence. The first and last nodes in the path are flagged as origin and destination, respectively. The O-D node set is then used for computing theoretical shortest paths.

Figure 4-5 depicts the distribution of street links that are used by one or more paths. Links used over 100 times by the trips in the analysis dataset (thick red lines) are mostly within downtown areas. In the suburban areas, the most used links are primary or secondary roads, and none of them was used more than 10 times. This map indicates that identified valid paths primarily occurred in the urban areas.

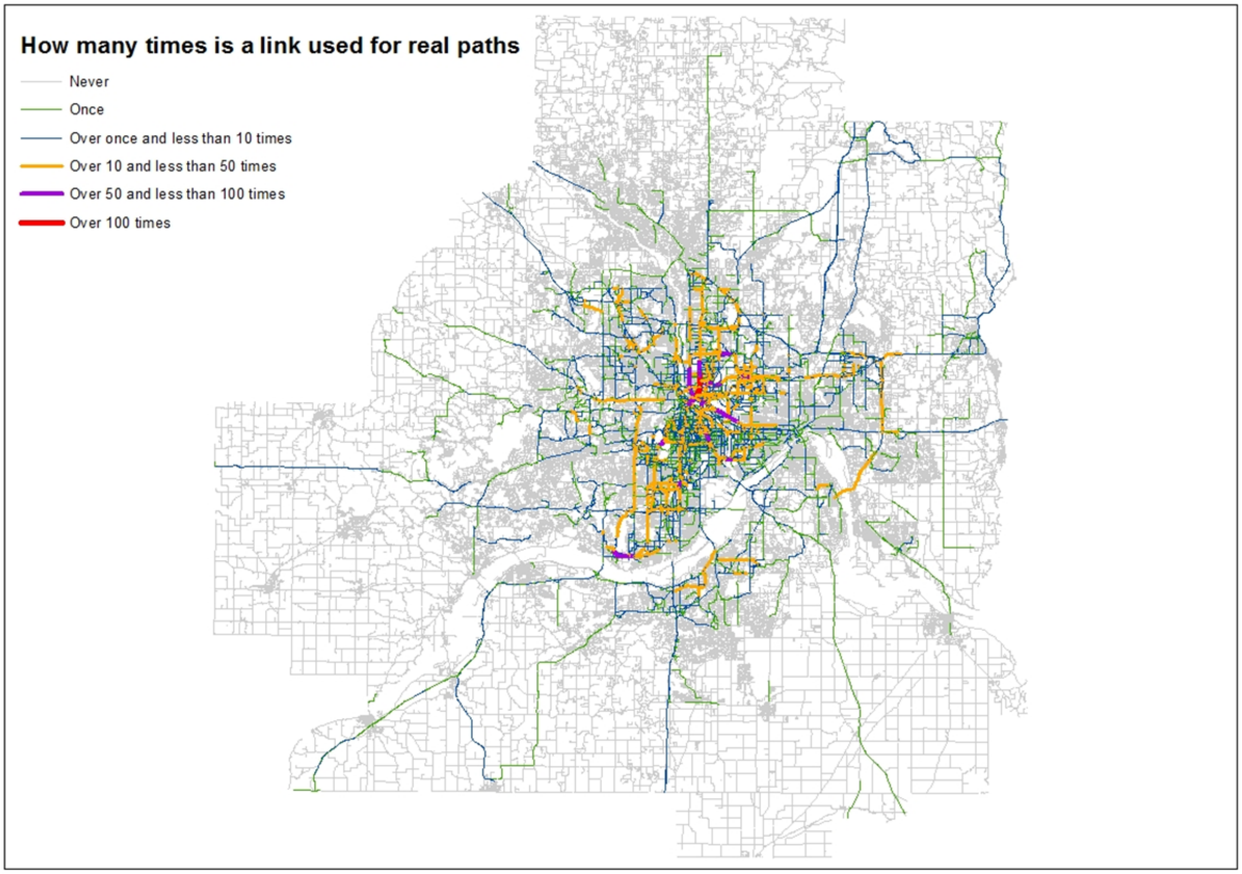


Figure 4-5: Frequency of Link Usage for Routes in the Dataset

4.3 Chapter Summary

This chapter describes the data preparation on real GPS tracked data and the digit street network in the study area. ArcGIS is used to extract necessary attributes of the street network, including link characteristics, signal existence, and network connectivity. Python scripts are developed to distinguish individual trips, and implement the map-matching algorithm. GPS point trajectories are snapped to the street network, generating a set of real paths for following experiment and analysis.

CHAPTER 5
IDENTIFICATION OF NETWORK ATTRIBUTES ALONG THE PATH

This chapter introduces methodologies used to identify possible factors that may influence path choice. These factors include the number of turning movements and signals along a path, the distributions of road classes, as well as travel time and distance.

Figure 5-1 depicts how these factors are identified for statistical analysis. Compared to real paths obtained from data preparation, shortest time paths and shortest distance paths are computed using path finding algorithms. The same factors are identified for all paths. Paired sample t-tests between real paths and shortest paths will be conducted on these factors to analyze their influence on path choice.

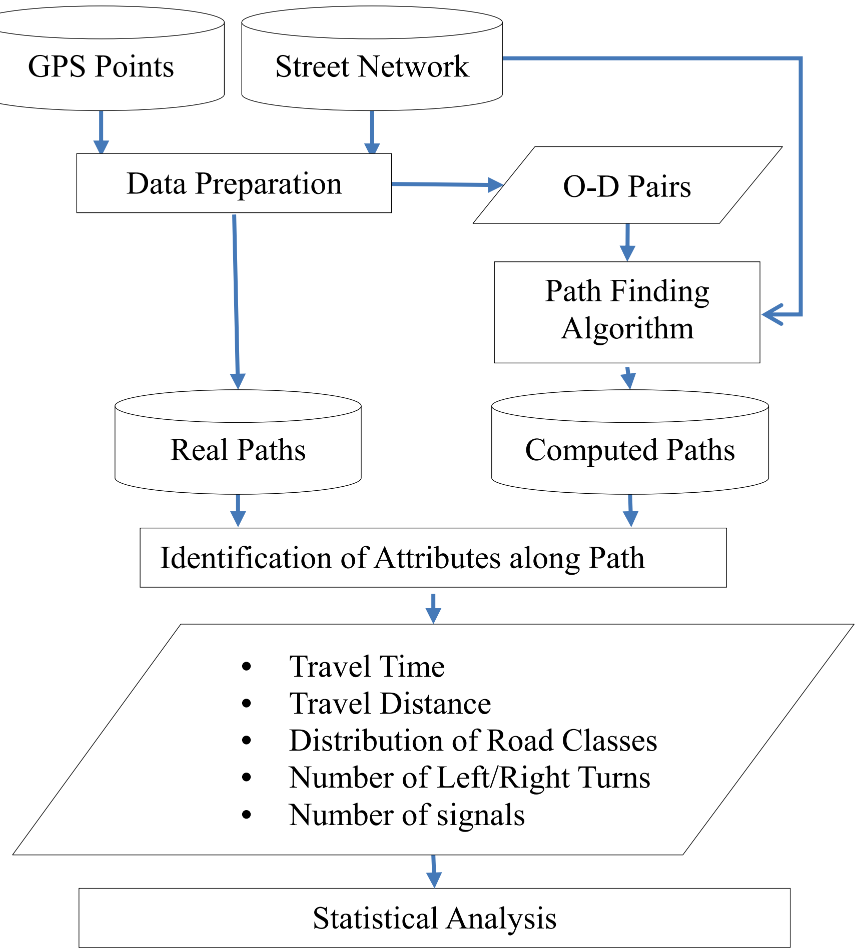


Figure 5-1: Analysis of Factors Influencing Path Choice

5.1 Computation of Shortest Paths

This study analyzes people's path choices by comparing real paths to shortest time/distance paths. Custom Python scripts are developed for accomplishing the study objectives including the implementation of shortest path algorithms. The code uses *Forward Star* notation to represent the street network data. The notation represents the network as an adjacency list:

$$G = \{n \in N, n = Node(id, x, y, signal, FS(n))\}$$

where G is the graph, n is the node that belongs to a set of nodes N, with an identification tag id, coordinates x and y, signal existence, and a link set named forward star. A forward star of a node is a set of all links starting from this node. The forward star notation is a representation of network connectivity.

Dijkstra's algorithm (Dijkstra, 1959), also known as the label-correcting algorithm, is used in the script for finding shortest paths. The algorithm maintains an unordered linked-list for candidate set Q (also known as the priority queue). After the full list is examined, the node with the minimum label in current Q is selected as the next node to be scanned. The algorithm is terminated once the destination node is selected to scan obviating the need to build the full shortest path tree. This algorithm is very appropriate for a one-to-one path search.

The script generates two paths for each trip. The first path is based on travel time, and the second path is based on distance. As in the case of real paths, each of the two computed shortest paths is represented by a node sequence as well as by a link sequence. Thus, for each trip between the same O-D pair, there are three paths:

1. real path,
2. shortest time path, and
3. shortest distance path.

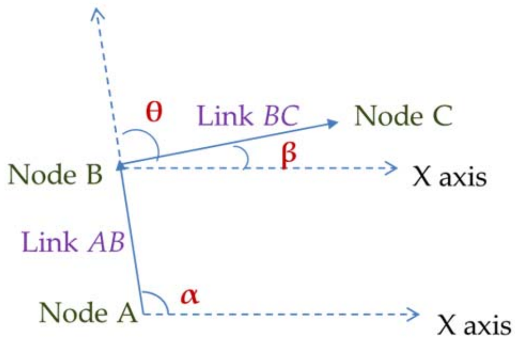
Link sequence or node sequence along these three paths may or may not be the same. Comparing the node/link sequences of two paths can identify whether or not they are identical to each other. Only when two paths have the same number of nodes/links in their sequences, and at every corresponding place of these two sequences, node/link in one sequence, the node/link is exactly the same as each other, can two paths be regarded identical.

5.2 Turns and Signals along Paths

Each node in the network has been tagged with a Boolean attribute to indicate whether a signal exists at that location. Because of this, finding the presence of signals along a trip path is relatively straightforward. The number of signals in each path can be counted by simply going through the node sequence for the path.

The most challenging part of the Python script developed for this study is the procedure that identifies left- or right-turning movements. The routine performs geometric calculations on nodes and the links along the paths (real or computed) under examination.

A turning movement is identified when the angle between two successive links is greater than 45 degrees. This angle, θ, as shown in Figure 5-2, is the absolute value of the difference by subtracting angle α from angle β. As the figure indicates, angle α and angle β are angles between links and the positive direction of x axis. Using coordinates of the three nodes A, B, and C, which form the two links, angle α and angle β can be calculated as follows:



$$\alpha = tan\frac{B_y - A_y}{B_x - A_x}, \beta = tan\frac{C_y - B_y}{C_x - B_x}$$

$$\theta = |\beta - \alpha|$$

Figure 5-2: Identifying Turns along a Path

Starting from the first node in any path sequence, for each consecutive three-node group the procedure computes the angle θ. If θ is greater than 45°, a turning movement is recognized to occur at node B. Depending on the direction of trip and turn, the movement is flagged as a left or right turn. The procedure for identifying left or right turn is illustrated in Figure 5-3.

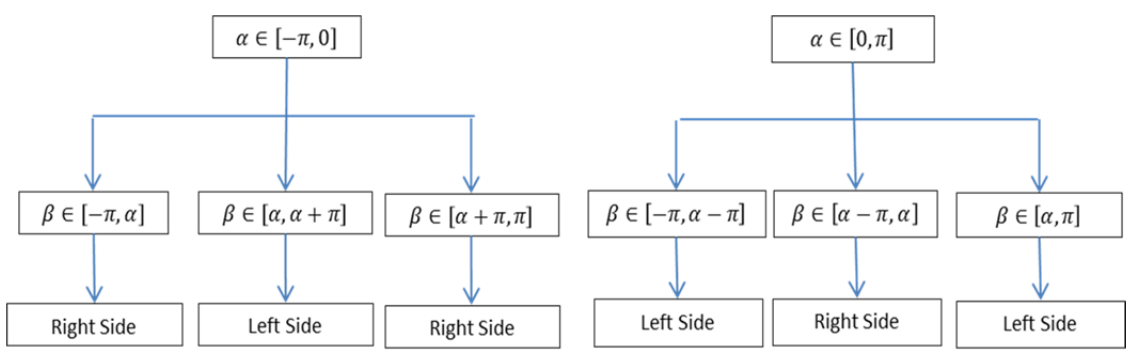


Figure 5-3: Identifying the Direction (Left or Right) of a Turn

The procedure goes through the whole path sequence and counts the number of left and right turns along the path, respectively. Because each of the three paths for a given O-D pair can have different lengths, the turn counts are normalized as number of turns per mile for later comparisons.

5.3 Distribution of Road Classes on Real Path

For analytical convenience, all the 23 road classifications in the network are combined into three major classes (Table 5-1). For both measurements of number of links and total link length, the class of local roads is dominant in the network. The primary road class has values generally higher than but comparable to the secondary road class.

Table 5-1: Road Classification Used in the Study

Classes	CFCC Code	Description	Number of Links	Total Link Length (Miles)
Primary Road	A1, A2	Interstate, US highways and their ramps	6,972	1,005
Secondary Road	A3	State highways and their ramps	5,388	651
Local Road	A4, A51, A73	Rural roads, local streets, and other minor roads	178,464	16,808

To exclude the possibility that drivers have to choose a certain road class merely due to the composition of the network rather than their preferences, a concept of "road availability" was defined. This concept reflects how much a specific class of facility can be chosen by drivers when they are making a trip.

The road availability measure cannot be based on the entire network, because only a small portion of the network is relevant for a specific trip. This small portion, defined as *trip proximity* in this study, is the smallest rectangular area being able to cover the trip path (Figure 5-4). The portion of trip represented in each road class is computed. Road lengths are also summed up by road class within the area of trip proximity. The following two measures are computed from these accumulation counters:

1. percentage of trip length in each road class
2. percentage of trip length in each road class normalized by length of road class within the trip proximity

The Python script examines link sequences of each path and classifies links, summing up the link lengths by CFCC codes. The summed miles of each class are divided by path length to obtain usages for this class.

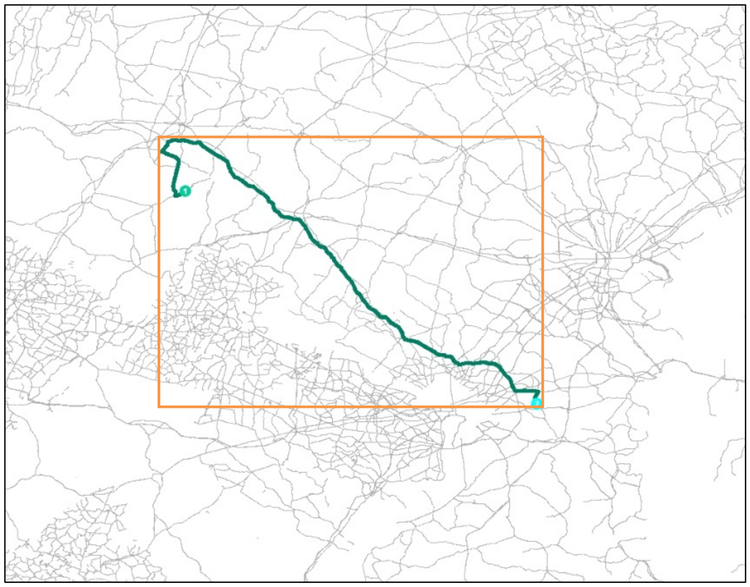


Figure 5-4: Example of Trip Proximity

The comparison between availability and usage of a specific trip can rule out the possible effect of the network composition on path selection, so as to help identify drivers' preferences among various road classifications.

5.4 Chapter Summary

Dijkstra's algorithm is used to find the shortest paths between O-D pairs determined from real paths. For each path, turning movements are identified based on the geometric calculation. The information on signal existences at intersections, as well as the distribution of road classes, is also extracted from the street network. These path attributes will be used in the statistical analysis to examine their impacts on path choices.

CHAPTER 6
FACTORS INFLUENCING ROUTE-CHOICE

Although the best known impact factors of route-choice are travel time and travel distance, in most urban areas, turning movements, traffic lights, and road classifications are hypothesized to have a significant effect on travelers' choices. This chapter identifies the impacts of these factors by comparing real paths to computed shortest paths. Statistical analysis was conducted to reach sound conclusions.

6.1 Simple Comparisons: Real Paths vs. Computed Paths

For analytic conveniences, real paths, shortest distance paths, and shortest time paths are categorized in four groups by their length. First, average travel time and travel distance on real paths and shortest paths are compared.

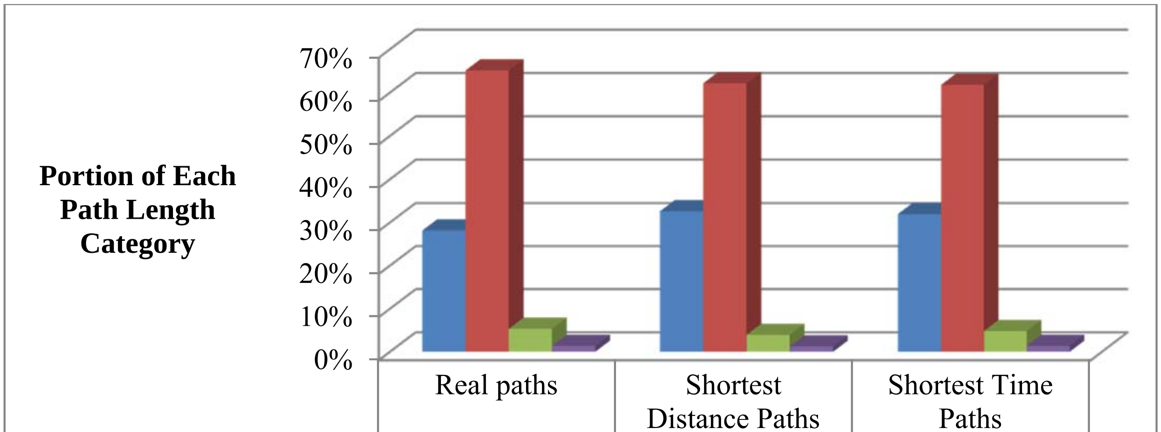
6.1.1 Path Composition by Path Length

On real paths the median value of path length is 1.539 miles and the average length of all paths is 2.184 mile. Thus it is reasonable to put all paths shorter than one mile into one group and paths between 1 and 5 miles in another. The maximum path length in the dataset is 30.545 miles and relatively long paths represent only a small percentage. For this reason, all paths longer than 10 miles can be categorized together. Therefore, the data on real paths are presented in four groups: a) shorter than one mile, b) between one and five miles, c) between five and ten miles, and d) greater than ten miles. Shortest distance and time paths are categorized in the same way.

The Table 6-1 shows that most trips have a path length between 1 and 5 miles. With paths shorter than 1 mile together, over 90% of trips have a path less than 5 miles, no matter if the path is real or computed (Figure 6-1). The reason that longer trips are lacking in the dataset is that some of the longer trips were removed in previous steps because they were repetitive commuting trips made by the same driver.

Table 6-1: Real Paths and Shortest Paths by Length

Path Length Category	Real Paths	Shortest Distance Paths	Shortest Time Paths
less than 1 mile	1,603	1,858	1,820
between 1 and 5 miles	3,708	3,540	3,518
between 5 and 10 miles	304	224	275
more than 10 miles	79	72	81
Total	5,694	5,694	5,694



	Real paths	Shortest Distance Paths	Shortest Time Paths
less than 1 mile	28.15%	32.63%	31.96%
between 1 and 5 miles	65.12%	62.17%	61.78%
between 5 and 10 miles	5.34%	3.93%	4.83%
more than 10 miles	1.39%	1.26%	1.42%

Figure 6-1: Real Paths and Shortest Paths by Length

Table 6-2 shows 1,997 shortest distance paths and 1,951 shortest time paths identical to their real path counterparts. On the other hand, more than 60% real paths (about 3,548 paths) are different from both computed shortest paths.

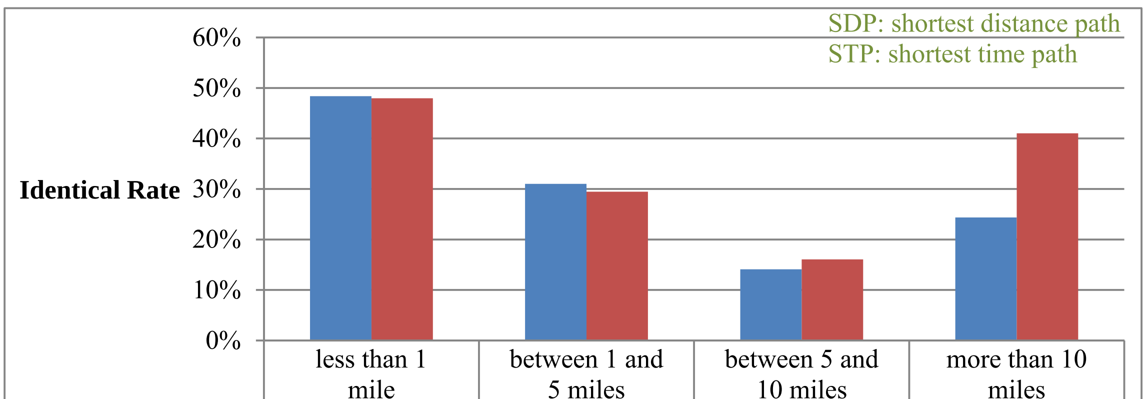
Table 6-2: Shortest Paths Identical to Real Path

Shortest Path	Number of Identical Paths	Percentage of Identical Paths
Shortest Distance Path	1,997	35.07%
Shortest Time Path	1,951	34.26%

Figure 6-2 further illustrates how many shortest paths are identical to corresponding real paths in each path length category. A measure of identical rate was defined as:

$$Identical\ Rate = \frac{number\ of\ identical\ trips}{number\ of\ all\ trips}$$

The highest identical rates occur on paths with fewest miles of length, for both shortest distance and shortest time paths. The rate decreases when the path length becomes longer. The category of “between 5 and 10 miles” has the lowest identical rate, i.e., among all real paths in this category, only 14.10% happened to be the shortest distance paths between their O-D pairs, and 16.07% to be the shortest time paths.



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Identical SDP	48.39%	31.02%	14.10%	24.36%
Identical STP	47.96%	29.46%	16.07%	41.03%

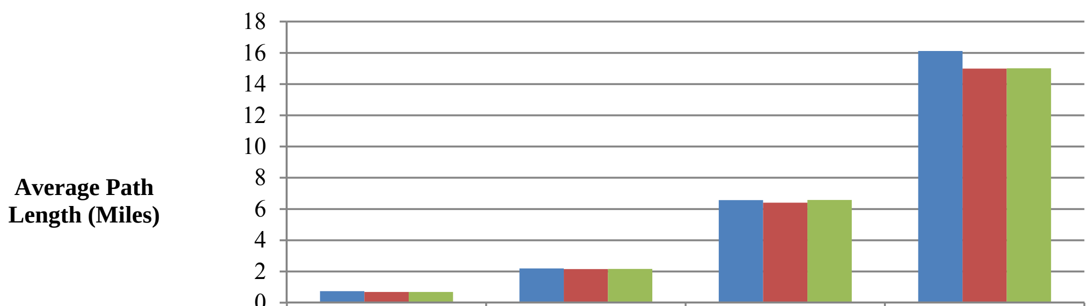
Figure 6-2: Identical Rate by Path Length

The identical rate shows that drivers did not necessarily choose the shortest distance paths or the shortest time paths. This finding means that drivers must be influenced by other factors when they choose their real paths.

6.1.2 Average Travel Distance and Travel Time

Both real and computed paths have a similar average path length (Figure 6-3), especially between two computed paths, which have the same average length for paths shorter than 5 miles. Real paths have an average length slightly longer than theoretic paths, except the trips between 5 and 10 miles. In this category, shortest time paths have the longest average path length.

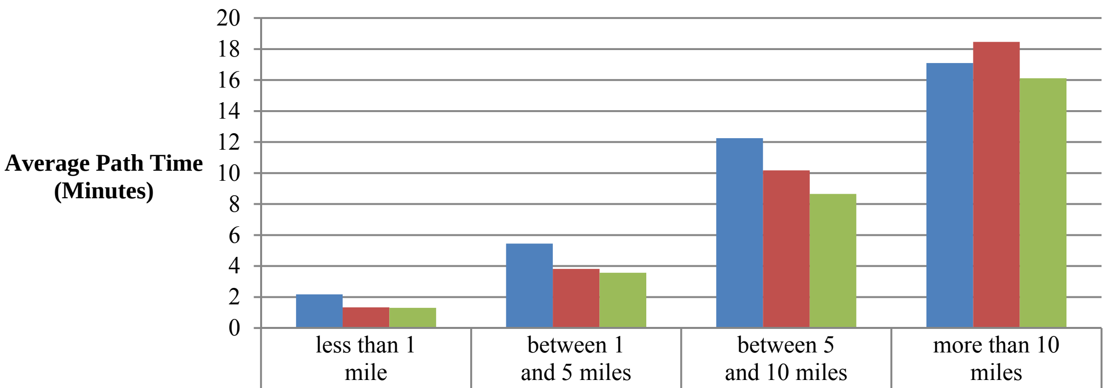
In terms of the category of more than 10 miles, the shortest distance paths take the longest time (Figure 6-4). For all trips, the shortest time paths take less time than the other two types, which is consistent with expectation.



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Paths	0.74	2.19	6.57	16.12
Shortest Distance Paths	0.69	2.16	6.41	15.00
Shortest Time Paths	0.69	2.16	6.58	15.01

Path Length Category

Figure 6-3: Average Path Length



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Paths	2.17	5.45	12.24	17.09
Shortest Distance Paths	1.33	3.81	10.17	18.46
Shortest Time Paths	1.31	3.57	8.65	16.11

Figure 6-4: Average Path Time

6.2 Influence of Road Classifications

This study firstly examines if drivers' choices of route were influenced by the network composition. Then usages of primary and secondary roads on real paths and shortest paths are compared.

6.2.1 Real Paths: Road Availability vs. Road Usage

Figure 6-5 compares road availability to road usage of real paths for four path lengths and three major road classes. Local roads within proximity dominate for all length groups. Although the availability of primary and secondary roads together is only 10% or less, the two higher classes have a bigger portion in the road usage. In spite of the network composition, drivers were willing (not forced) to choose roads with a higher level of functional class.

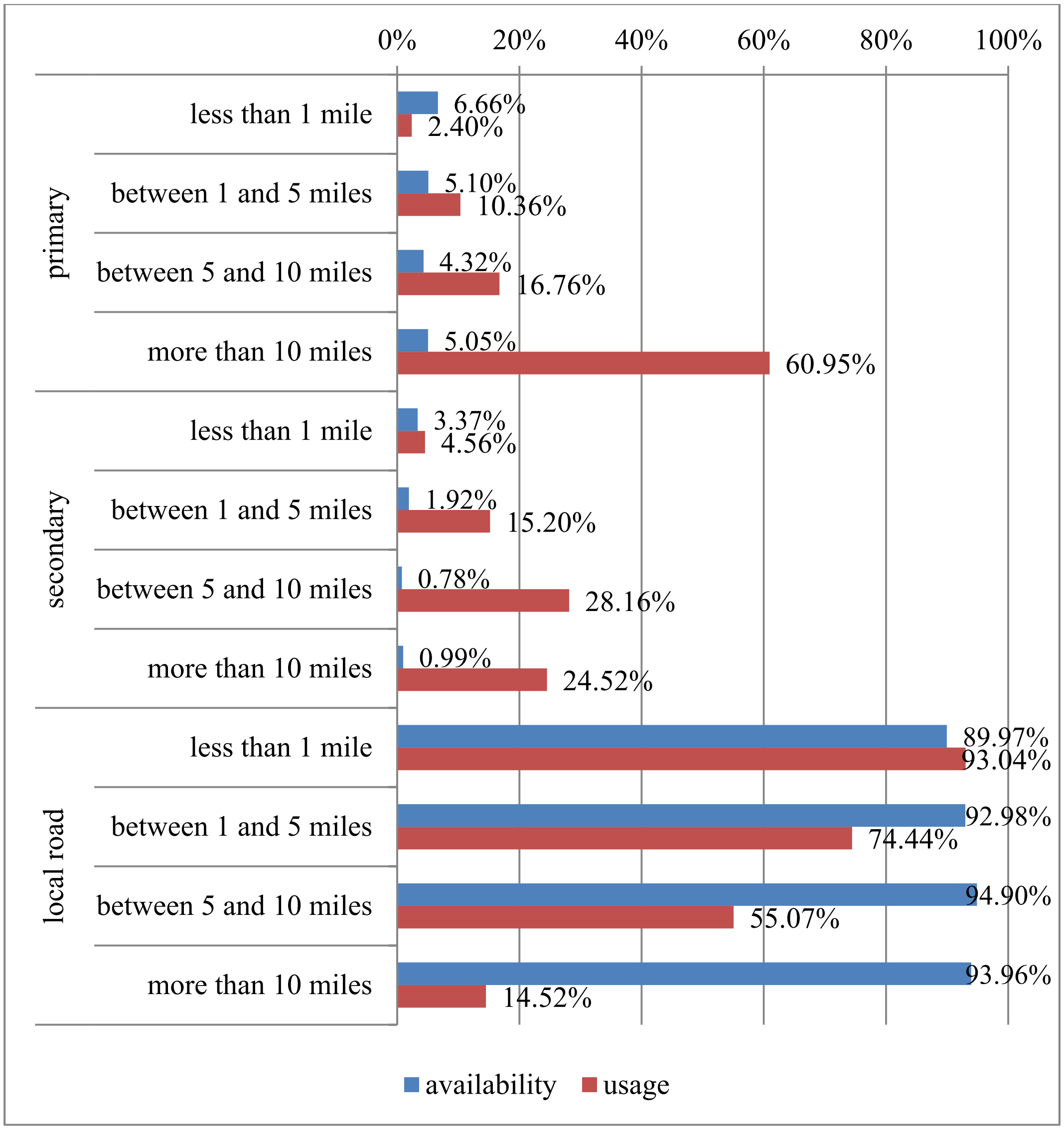


Figure 6-5: Road Availability vs. Road Usage

As path lengths increase, the portion of the primary road in a real path becomes larger, and the portion of the local road becomes smaller. In terms of secondary roads, the percentage increases then decreases. For paths shorter than 10 miles, over 50% usage belongs to local road, and primary road only takes the smallest portion. Only when trip length is longer than 10 miles, does the percent share of local roads become lower than the other two classes. The percent share of primary roads is the highest.

6.2.2 Primary and Secondary Road Usage

This section further explores commonalities and differences between real paths and shortest paths in terms of usage of primary and secondary roads. In each road class, shortest distance paths and shortest time paths trend the same way as real paths when path lengths increase (Figure 6-6). In other words, usage of primary roads increases while that of local roads decreases. The highest usage of secondary roads happens to paths between 5 and 10 miles. However, for trips longer than 10 miles, the smallest percentage is secondary road for shortest distance path., For trips between 5 and 10 miles, the largest percentage is secondary road for shortest time path. .

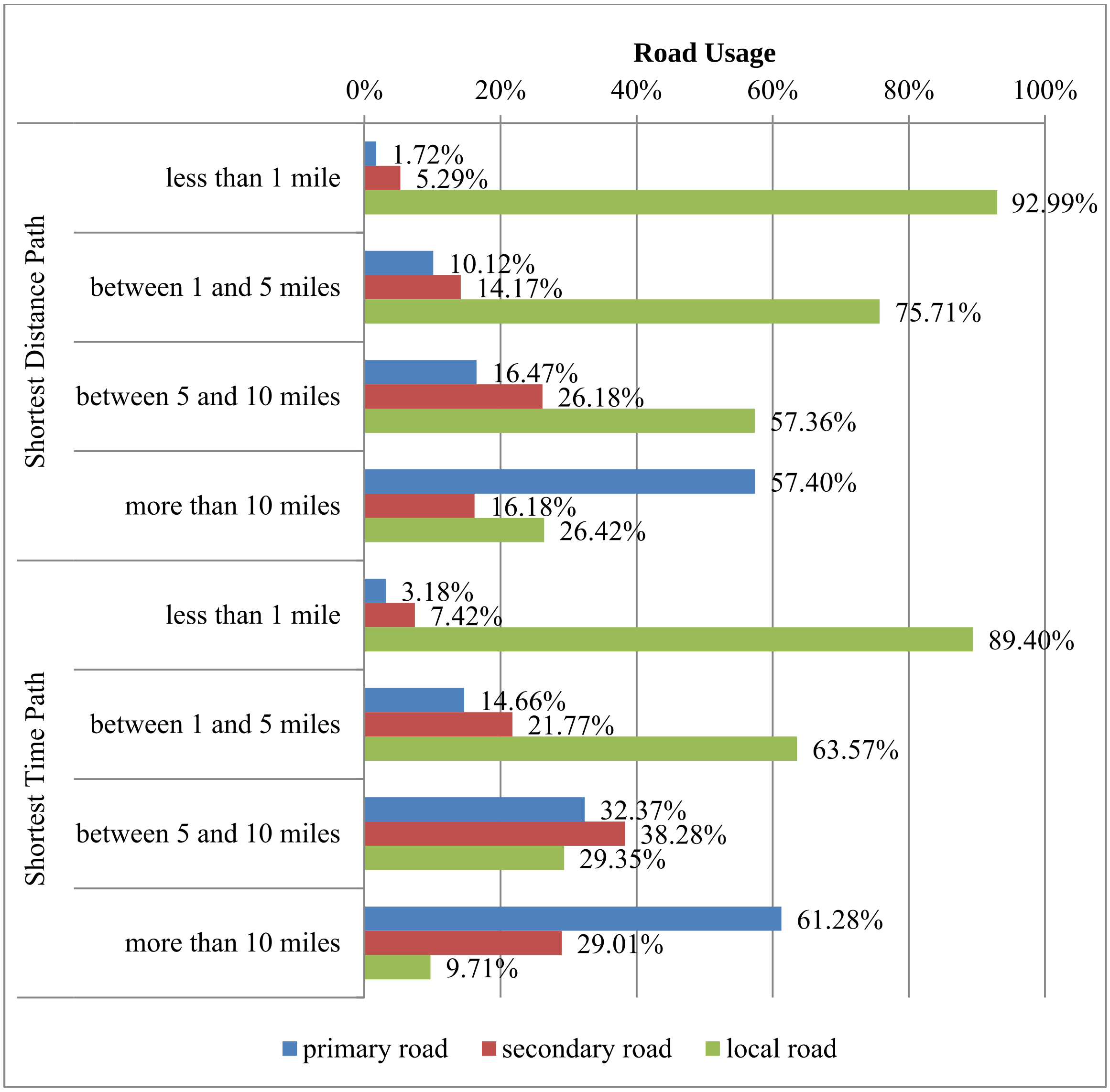
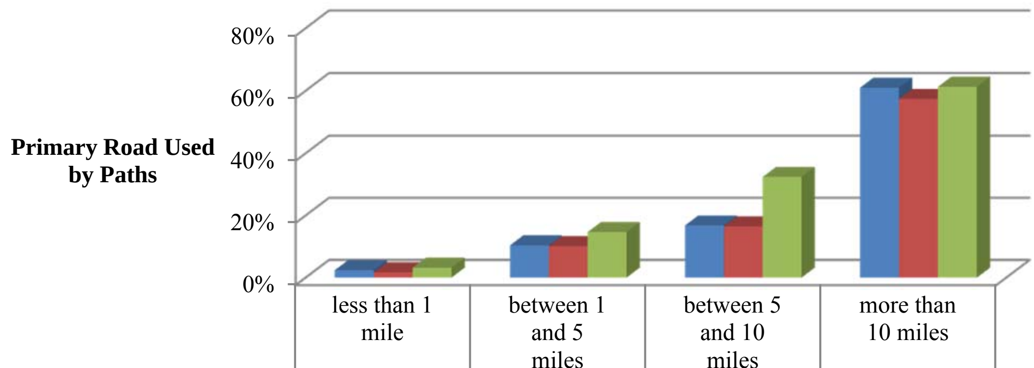


Figure 6-6: Shortest Distance and Time Paths for Roads Classified by Type and Length

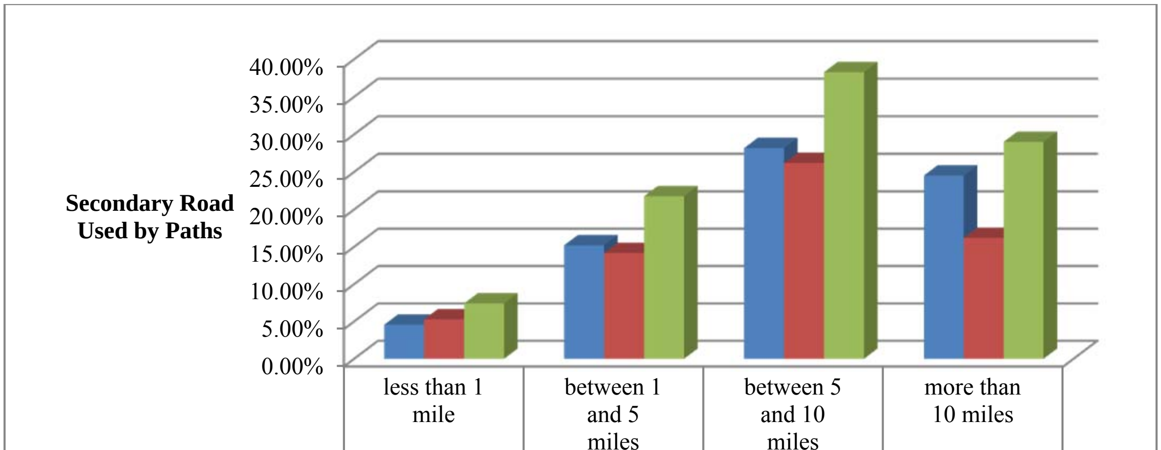
Figures 6-7 to 6-9 compare primary, secondary, and local road usage on real paths and shortest paths. On average, shortest time paths occurred more along primary and secondary roads than did the other two sets and accordingly, fewer local roads for all trips. This is reasonable because primary and secondary roads yield higher average traveling speeds and less travel time accordingly. Furthermore, for the shortest time path, the percentage of local roads declined more sharply than for the other two types of path with path length increasing. Meanwhile, shortest distance paths used less primary road than real paths in all categories.



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Path	2.40%	10.36%	16.76%	60.95%
Shortest Distance Path	1.72%	10.12%	16.47%	57.40%
Shortest Time Path	3.18%	14.66%	32.37%	61.28%

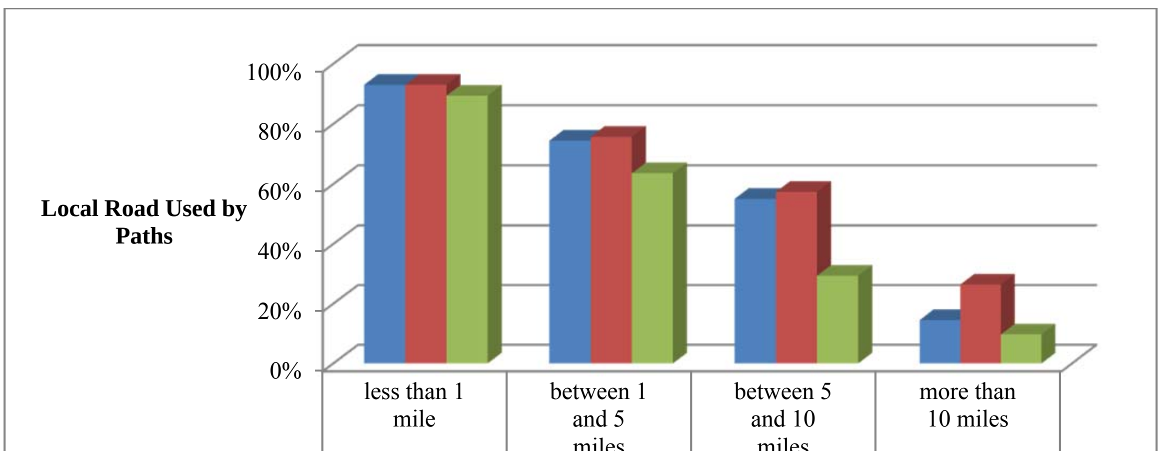
Path Length

Figure 6-7: Usage of Primary Roads along Real and Shortest Paths



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Path	4.56%	15.20%	28.16%	24.52%
Shortest Distance Path	5.29%	14.17%	26.18%	16.18%
Shortest Time Path	7.42%	21.77%	38.28%	29.01%

Figure 6-8: Usage of Secondary Roads along Real and Shortest Paths



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Path	93.04%	74.44%	55.07%	14.52%
Shortest Distance Path	92.99%	75.71%	57.36%	26.42%
Shortest Time Path	89.40%	63.57%	29.35%	9.71%

Figure 6-9: Usage of Local Roads Real and Shortest Paths

Table 6-3 and 6-4 present the results of paired sample t-tests. They show that real paths have higher percentages of primary and secondary roads than shortest distance paths in almost all circumstances. Only for paths shorter than one mile do real paths have less usage of secondary roads than shortest distance paths, where the p-value is bigger than 0.05 (0.123).

Table 6-3: Effect of Primary Road on Path Choice - Real vs. Shortest Paths

Path	**Results of paired t-test**	Path Length			
		Shorter than 1 mile	Between 1 and 5 miles	Between 5 and 10 miles	Longer than 10 miles
Real Path	Mean percentage of primary road along real path (μ_r)	0.021	0.093	0.164	0.602
	Degrees of freedom	1602	3707	304	78
Vs. Shortest Distance Path	Mean percentage of primary road along shortest distance path (μ_s)	0.013	0.088	0.128	0.540
	Mean difference (μ_r - μ_s)	0.0080	0.005	0.037	0.062
	t-statistic	3.667	3.871	3.726	3.534
	p-value (one-tailed)	< 0.001	< 0.001	< 0.001	< 0.001
Vs. Shortest Time Path	Mean percentage of primary road along shortest Time path (μ_s)	0.030	0.123	0.254	0.617
	Mean difference (μ_r - μ_s)	-0.009	-0.030	-0.090	-0.015
	t-statistic	-4.284	-14.610	-8.375	-1.263
	p-value(one-tailed)	1.000	1.000	1.000	0.895

H_0: $\mu_r - \mu_s \leq 0$, H_a: $\mu_r - \mu_s > 0$
At 95% confidence interval of the difference, the p-value less than 0.05 indicates it's safe to reject H_0 and H_a may be accepted.
Values in shaded cells indicate real paths have significantly higher percentage of primary road than shortest paths in statistics.

Paired t-tests also found that shortest time paths have both primary and secondary road usage significantly greater than real paths for all paths. On the other hand, shortest distance paths do not have a higher percentage than real paths regarding road usage with higher classes.

Table 6-4: Effect of Secondary Road on Path Choice - Real vs. Shortest Paths

Path	**Results of paired t-test**	Path Length			
		Shorter than 1 mile	Between 1 and 5 miles	Between 5 and 10 miles	Longer than 10 miles
Real Path	Mean percentage of secondary road along real path (μ_r)	0.040	0.138	0.274	0.260
	Degrees of freedom	1602	3707	304	78
Vs. Shortest Distance Path	Mean percentage of secondary road along shortest distance path (μ_s)	0.038	0.131	0.230	0.187
	Mean difference (μ_r - μ_s)	0.002	0.008	0.044	0.074
	t-statistic	1.162	3.352	3.565	3.985
	p-value (one-tailed)	0.123	< 0.001	< 0.001	< 0.001
Vs. Shortest Time Path	Mean percentage of secondary road along shortest Time path (μ_s)	0.063	0.189	0.375	0.288
	Mean difference (μ_r - μ_s)	-0.022	-0.050	-0.100	-0.028
	t-statistic	-7.666	-18.518	-8.117	-2.077
	p-value(one-tailed)	1.000	1.000	1.000	0.979

H_0: $\mu_r - \mu_s \leq 0$, H_a: $\mu_r - \mu_s > 0$
At 95% confidence interval of the difference, the p-value less than 0.05 indicates it's safe to reject H_0 and H_a may be accepted.
Values in shaded cells indicate real paths have significantly higher percentage of secondary road than shortest paths in statistics.

In summary, people tend to choose roads with higher classes. However, real paths did not have more primary and secondary roads than shortest time paths. The reason why people prefer major roads is that major roads have higher posted speed limits and fewer interruptions, which leads to less traveling time. This feature has been reflected by the impedance function of shortest time paths. Therefore, there is no significant difference between real paths and shortest time paths regarding the distribution of road classifications.

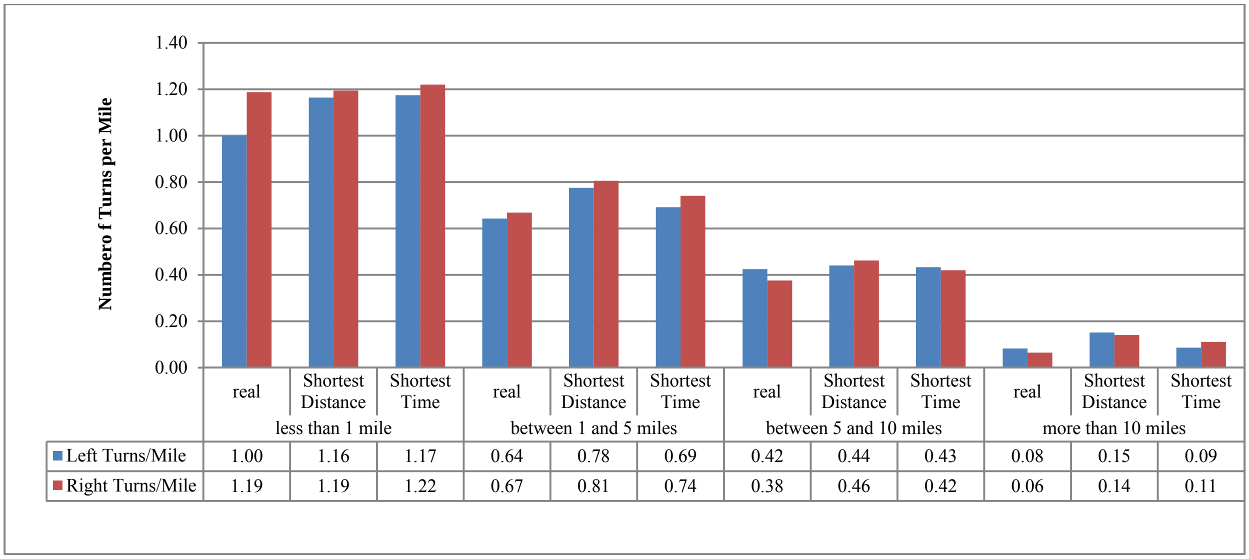
6.3 Effect of Turning Movements on Path Choice

The influence of turning movements on route-choice are studied with regard to: 1) all turns regardless of the existence of a signal, 2) turns at signalized intersections, and 3) turns at non-signalized intersections. Counts of turns along a path are normalized as number of turns per mile due to different path length.

6.3.1 All Turns Regardless of Signal Existence

Figure 6-10 illustrates the number of turns per mile along paths for real paths, for shortest distance paths, and for shortest time paths. The longer the path length, the fewer the number of turns. This pattern appeared for all sets and for both left and right turns.

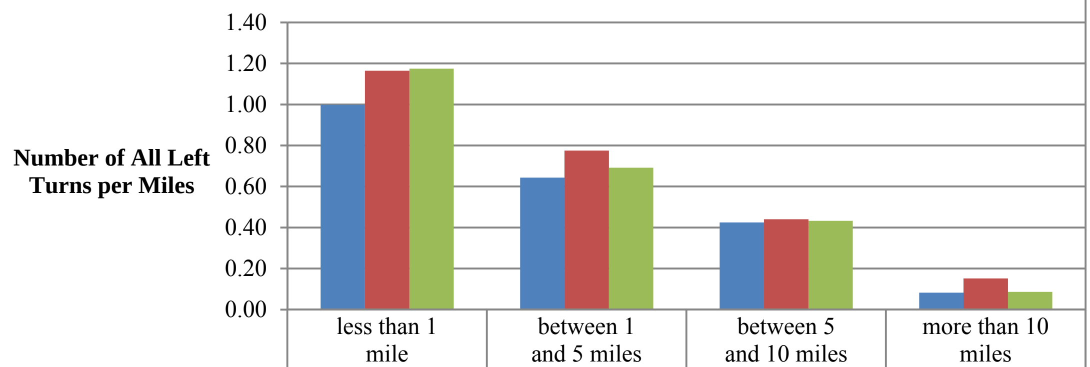
All paths have more right turns than left turns on average for trips shorter than 5 miles. According to analysis of road classification in the previous section, for shorter trips the majority of road usage is in the local road category. Local roads have more intersections, and consequently may generate more turning movements. However, there is no obvious pattern for trips of more than 5 miles because on these trips the influence of turns declined due to the increasing percentage of primary and secondary roads.



	less than 1 mile			between 1 and 5 miles			between 5 and 10 miles			more than 10 miles		
	real	Shortest Distance	Shortest Time	real	Shortest Distance	Shortest Time	real	Shortest Distance	Shortest Time	real	Shortest Distance	Shortest Time
Left Turns/Mile	1.00	1.16	1.17	0.64	0.78	0.69	0.42	0.44	0.43	0.08	0.15	0.09
Right Turns/Mile	1.19	1.19	1.22	0.67	0.81	0.74	0.38	0.46	0.42	0.06	0.14	0.11

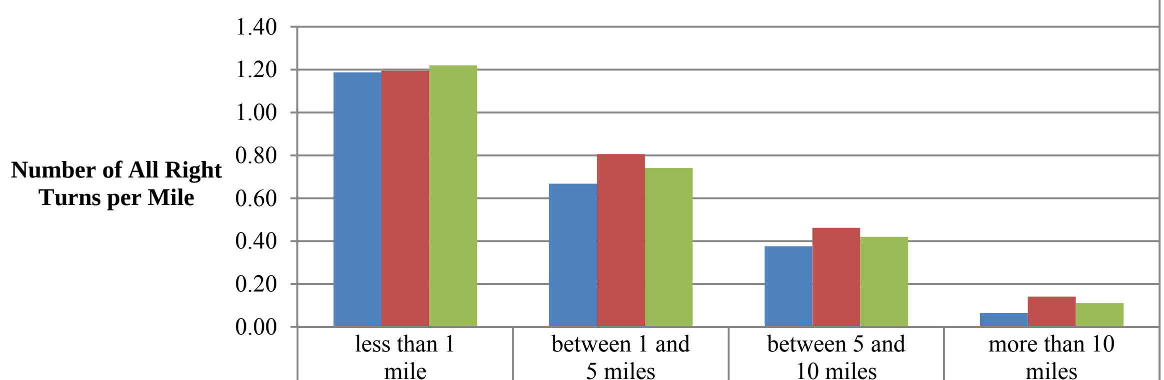
Figure 6-10: Number of Turns per Mile Regardless of Signal Existence

Real paths have a smaller average value in all length categories than computed paths. Figures 6-11 and 6-12 show the comparisons in terms of number of left turns per mile and number of right turns per mile.



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Paths	1.00	0.64	0.42	0.08
Shortest Distance Paths	1.16	0.78	0.44	0.15
Shortest Time Paths	1.17	0.69	0.43	0.09

Figure 6-11: Number of All Left Turns per Mile



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Paths	1.19	0.67	0.38	0.06
Shortest Distance Paths	1.19	0.81	0.46	0.14
Shortest Time Paths	1.22	0.74	0.42	0.11

Figure 6-12: Number of All Right Turns per Mile

The results of paired t-tests in Tables 6-5 and 6-6 support the observations statistically except when paths are longer than 5 miles and compared to shortest time paths. The paired t-tests yielded p-values over 0.05, which means at 95% confidence interval, it is not safe to reject the null hypothesis that real paths have more left turns. The test results show that drivers tend to choose paths with fewer left turns and right turns than shortest path.

Table 6-5: Effect of All Left Turns on Path Choice - Real vs. Shortest Paths

Path	**Results of paired t-test**	**Path Length**			
		Shorter than 1 mile	Between 1 and 5 miles	Between 5 and 10 miles	Longer than 10 miles
Real Path	Mean left turns per mile along real path (μ_r)	1.012	0.678	0.449	0.088
	Degrees of freedom	1602	3707	319	62
Vs. Shortest Distance Path	Mean left turns per mile shortest distance path (μ_s)	1.182	0.859	0.566	0.155
	Mean difference (μ_r - μ_s)	-0.171	-0.181	-0.118	-0.067
	t-statistic	-5.570	-12.818	-3.485	-2.839
	p-value (one-tailed)	< 0.001	< 0.001	< 0.001	0.003
Vs. Shortest Time Path	Mean left turns per mile along shortest time path (μ_s)	1.198	0.785	0.482	0.069
	Mean difference (μ_r - μ_s)	-0.186	0.107	-0.034	0.019
	t-statistic	-6.055	-7.740	-1.032	2.107
	p-value (one-tailed)	< 0.001	< 0.001	0.303	0.980

H_0: $\mu_r - \mu_s \geq 0$, H_a: $\mu_r - \mu_s < 0$
At 95% confidence interval of the difference, the p-value less than 0.05 indicates it is safe to reject H_0 and H_a may be accepted.
Values in shaded cells indicate real paths have fewer left turns per mile than do shortest paths statistically.

Table 6-6: Effect of All Right Turns on Path Choice - Real vs. Shortest Paths

Path	Results of paired t-test	Path Length			
		Shorter than 1 mile	Between 1 and 5 miles	Between 5 and 10 miles	Longer than 10 miles
Real Path	Mean right turns per mile along real path (μ_r)	1.206	0.730	0.397	0.066
	Degrees of freedom	1602	3707	319	62
Vs. Shortest Distance Path	Mean right turns per mile along shortest distance path (μ_s)	1.272	0.865	0.616	0.138
	Mean difference (μ_r - μ_s)	-0.066	-0.136	-0.219	-0.067
	t-statistic	-2.095	-10.802	-7.877	-2.839
	p-value (one-tailed)	0.018	< 0.001	< 0.001	0.003
Vs. Shortest Time Path	Mean right turns per mile along shortest time path (μ_s)	1.305	0.812	0.499	0.081
	Mean difference (μ_r - μ_s)	-0.099	-0.107	-0.102	-0.015
	t-statistic	-3.153	-7.740	-4.307	-1.605
	p-value (one-tailed)	0.001	< 0.001	< 0.001	0.057

H_0: $\mu_r - \mu_s \geq 0$, H_a: $\mu_r - \mu_s < 0$
At 95% confidence interval of the difference, the p-value less than 0.05 indicates it is safe to reject H_0 and H_a may be accepted.
Values in shaded cells indicate real paths have less right turns per mile than do shortest paths statistically.

6.3.2 Turns at Signalized Intersection

For real, shortest distance, and shortest time paths, this study also analyzed turning movements at signalized intersections and at intersections without traffic lights. Figure 6-

13 compares signalized turning movements and shows that neither left turns nor right turns are prevalent. The only clear pattern is that the number of turns per mile decreased with increasing path length.

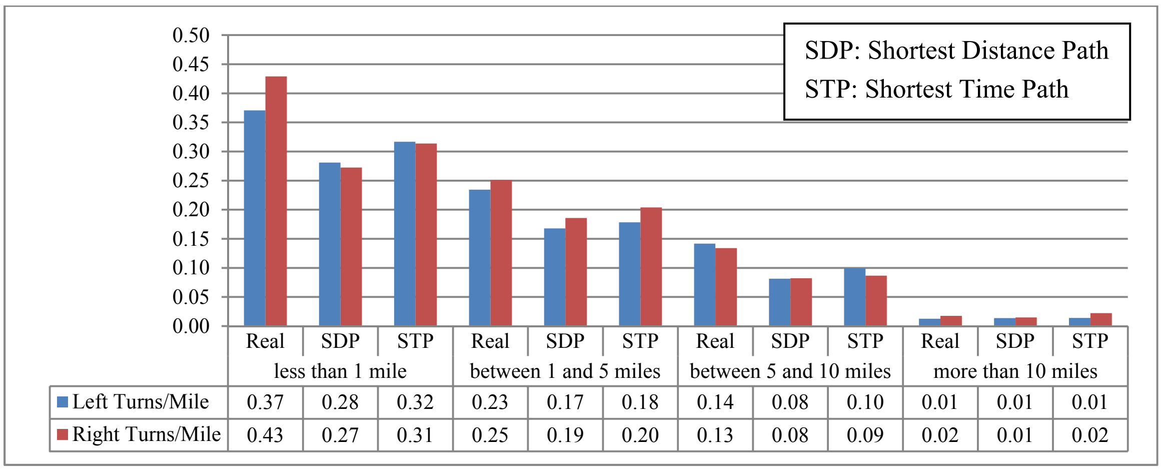
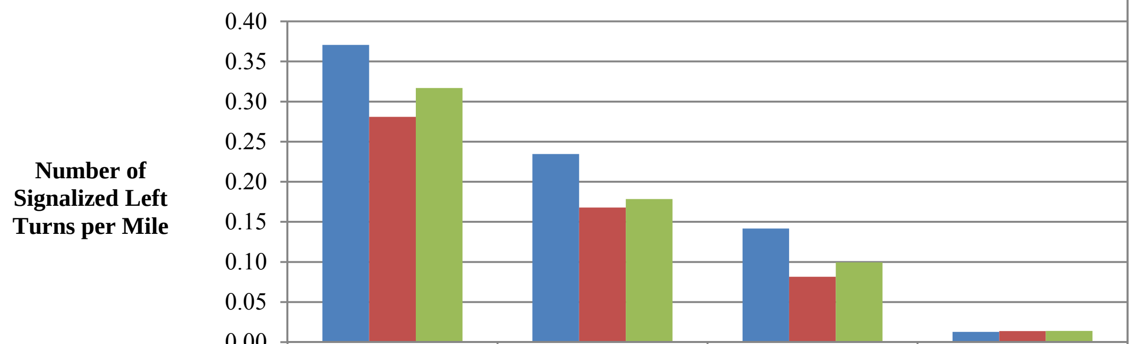


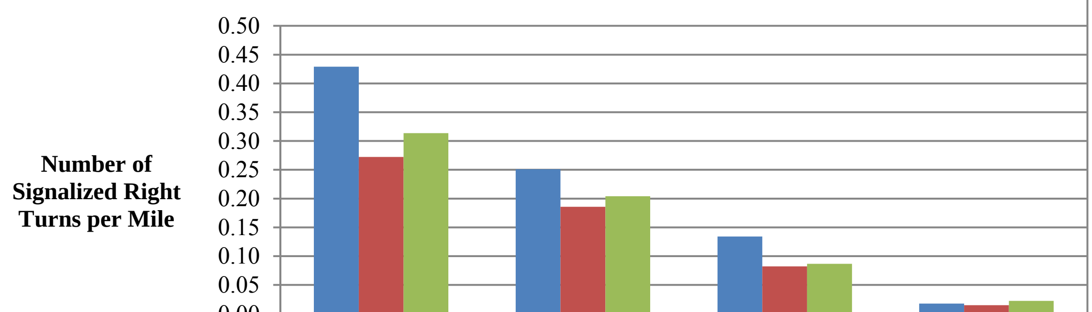
Figure 6-13: Number of Signalized Turns per Mile

Figures 6-14 and 6-15 depict average signalized left turns per miles and right turns per mile respectively. The results of statistical analysis are shown in Table 6-7 and 6-8. None of the paired t-tests yielded the conclusion that real paths have fewer signalized turns.



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Paths	0.37	0.23	0.14	0.01
Shortest Distance Paths	0.28	0.17	0.08	0.01
Shortest Time Paths	0.32	0.18	0.10	0.01

Figure 6-14: Number of Signalized Left Turns per Mile



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Paths	0.43	0.25	0.13	0.02
Shortest Distance Paths	0.27	0.19	0.08	0.01
Shortest Time Paths	0.31	0.20	0.09	0.02

Figure 6-15: Number of Signalized Right Turns per Mile

Table 6-7: Effect of Signalized Left Turns on Path Choice - Real vs. Shortest Paths

Path	Results of paired t-test	Path Length			
		Shorter than 1 mile	Between 1 and 5 miles	Between 5 and 10 miles	Longer than 10 miles
Real Path	Mean signalized left turns per mile along real path (μ_r)	0.376	0.250	0.149	0.013
	Degrees of freedom	1602	3707	319	62
Vs. Shortest Distance Path	Mean signalized left turns per mile along shortest distance path (μ_s)	0.285	0.197	0.089	0.014
	Mean difference (μ_r - μ_s)	0.091	0.053	0.060	-0.002
	t-statistic	5.371	8.118	6.194	-0.404
	p-value (one-tailed)	1.000	0.999	1.000	0.344
Vs. Shortest Time Path	Mean signalized left turns per mile along shortest time path (μ_s)	0.321	0.207	0.089	0.009
	Mean difference (μ_r - μ_s)	0.056	0.043	0.060	0.003
	t-statistic	3.240	6.235	6.561	1.709
	p-value (one-tailed)	0.999	1.000	1.000	0.954

H_0: $\mu_r - \mu_s \geq 0$, H_a: $\mu_r - \mu_s < 0$
At 95% confidence interval of the difference, the p-value less than 0.05 indicates it is safe to reject H_0 and H_a may be accepted.
Values in shaded cells indicate real paths have fewer signalized left turns per mile than do shortest paths statistically.

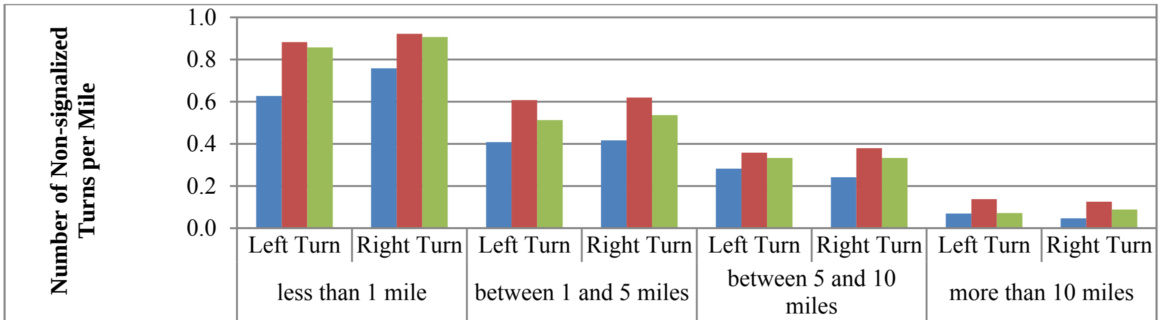
Table 6-8: Effect of Signalized Right Turns on Path Choice - Real vs. Shortest Paths

Path	Results of paired t-test	Path Length			
		Shorter than 1 mile	Between 1 and 5 miles	Between 5 and 10 miles	Longer than 10 miles
Real Path	Mean signalized right turns per mile along real path (μ_r)	0.426	0.273	0.144	0.019
	Degrees of freedom	1602	3707	319	62
Vs. Shortest Distance Path	Mean signalized right turns per mile shortest distance path (μ_s)	0.289	0.199	0.107	0.016
	Mean difference (μ_r - μ_s)	0.137	0.074	0.037	0.003
	t-statistic	7.770	12.207	2.877	0.520
	p-value (one-tailed)	1.000	1.000	0.998	0.698
Vs. Shortest Time Path	Mean signalized right turns per mile along shortest Time path (μ_s)	0.333	0.221	0.103	0.016
	Mean difference (μ_r - μ_s)	0.093	0.052	0.040	0.003
	t-statistic	4.970	8.670	3.358	0.573
	p-value (one-tailed)	1.000	1.000	1.000	0.716

H_0: $\mu_r - \mu_s \geq 0$, H_a: $\mu_r - \mu_s < 0$
At 95% confidence interval of the difference, the p-value less than 0.05 indicates it is safe to reject H_0 and H_a may be accepted.
Values in shaded cells indicate real paths have fewer signalized right turns per mile than do shortest paths statistically.

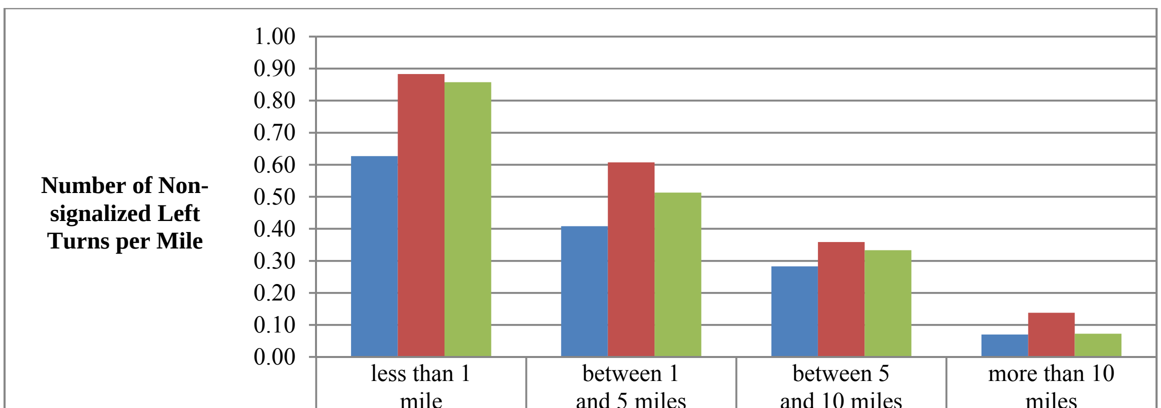
6.3.3 Turns at Non-signalized Intersections

The analysis of non-signalized turns and the analysis of signalized turns led to different conclusions. Both the average value (Figure 6-16 to 6-18) and statistical analysis (Table 6-9 and 6-10) indicate that real paths, compared with shortest paths, have fewer non-signalized turns and these findings are statistically significant. The only exception is the comparison of right turns for real paths and shortest time paths longer than 10 miles.



	less than 1 mile Left Turn	less than 1 mile Right Turn	between 1 and 5 miles Left Turn	between 1 and 5 miles Right Turn	between 5 and 10 miles Left Turn	between 5 and 10 miles Right Turn	more than 10 miles Left Turn	more than 10 miles Right Turn
Real	0.63	0.76	0.41	0.42	0.28	0.24	0.07	0.05
Shortest Distance	0.88	0.92	0.61	0.62	0.36	0.38	0.14	0.13
Shortest Time	0.86	0.91	0.51	0.54	0.33	0.33	0.07	0.09

Figure 6-16: Number of Non-signalized Turns per Mile



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Paths	0.63	0.41	0.28	0.07
Shortest Distance Paths	0.88	0.61	0.36	0.14
Shortest Time Paths	0.86	0.51	0.33	0.07

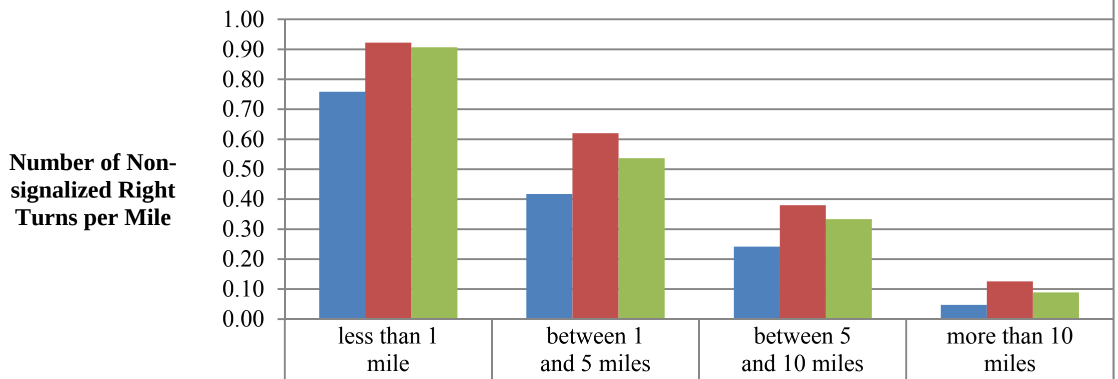
Figure 6-17: Number of Non-signalized Left Turns per Mile

Like the findings for all turns, left turns at intersections without signals are fewer than right turns at intersections without signals when path length is shorter than 5 miles. For other paths, neither direction of turn prevails over the other.

Table 6-9: Effect of Non-signalized Left Turns on Path Choice - Real vs. Shortest Paths

Path	**Results of paired t-test**	**Path Length**			
		Shorter than 1 mile	Between 1 and 5 miles	Between 5 and 10 miles	Longer than 10 miles
Real Path	Mean non-signalized left turns per mile along real path (μ_r)	0.635	0.428	0.299	0.076
	Degrees of freedom	1602	3707	319	62
Vs. Shortest Distance Path	Mean non-signalized left turns per mile shortest distance path (μ_s)	0.897	0.663	0.477	0.141
	Mean difference (μ_r - μ_s)	-0.262	-0.234	-0.178	-0.065
	t-statistic	-8.757	-17.707	-5.669	-2.960
	p-value (one-tailed)	< 0.001	< 0.001	< 0.001	0.002
Vs. Shortest Time Path	Mean non-signalized left turns per mile along shortest time path (μ_s)	0.877	0.578	0.393	0.060
	Mean difference (μ_r - μ_s)	-0.242	-0.149	-0.094	0.016
	t-statistic	-8.069	-11.785	-3.084	1.687
	p-value (one-tailed)	< 0.001	< 0.001	0.001	0.952

H_0: $\mu_r - \mu_s \geq 0$, H_a: $\mu_r - \mu_s < 0$
At 95% confidence interval of the difference, the p-value less than 0.05 indicates it is safe to reject H_0 and H_a may be accepted.
Values in shaded cells indicate real paths have fewer non-signalized left turns per mile than do shortest paths statistically.



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Paths	0.76	0.42	0.24	0.05
Shortest Distance Paths	0.92	0.62	0.38	0.13
Shortest Time Paths	0.91	0.54	0.33	0.09

Figure 6-18: Number of Non-signalized Right Turns per Mile

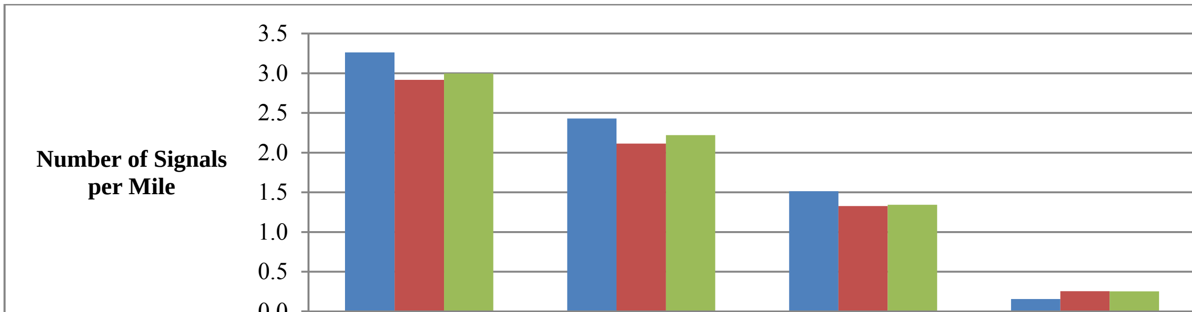
Table 6-10: Effect of Non-signalized Right Turns on Path Choice - Real vs. Shortest Paths

Path	Results of paired t-test	Path Length			
		Shorter than 1 mile	Between 1 and 5 miles	Between 5 and 10 miles	Longer than 10 miles
Real Path	Mean non-signalized right turns per mile along real path (μ_r)	0.780	0.457	0.253	0.047
	Degrees of freedom	1602	3707	319	62
Vs. Shortest Distance Path	Mean non-signalized right turns per mile shortest distance path (μ_s)	0.983	0.667	0.509	0.122
	Mean difference (μ_r - μ_s)	-0.203	-0.210	-0.256	-0.075
	t-statistic	-6.681	-16.465	-9.581	-3.396
	p-value (one-tailed)	< 0.001	< 0.001	< 0.001	< 0.001
Vs. Shortest Time Path	Mean non-signalized right turns per mile along shortest time path (μ_s)	0.972	0.591	0.395	0.065
	Mean difference (μ_r - μ_s)	-0.192	-0.135	-0.142	-0.018
	t-statistic	-6.398	-11.344	-6.035	-1.730
	p-value (one-tailed)	< 0.001	< 0.001	< 0.001	0.044

H_0: $\mu_r - \mu_s \geq 0$, H_a: $\mu_r - \mu_s < 0$
At 95% confidence interval of the difference, the p-value less than 0.05 indicates it is safe to reject H_0 and H_a may be accepted.
Values in shaded cells indicate real paths have fewer non-signalized right turns per mile than do shortest paths statistically.

6.4 Effect of Signals on Path Choice

Figure 6-19 shows that both shortest distance paths and shortest time paths have fewer signals than real paths when the length is shorter than 10 miles. The paired t-test results are consistent with this observation. Most of p-values in Table 6-11 are bigger than 0.05, which means the null hypothesis that real paths have more signals than shortest paths cannot be rejected. The only exception is in the comparison to shortest distance paths regarding trips longer than 10 miles. Considering the small portion of long paths, this p-value could be regarded as the result of outliers. Therefore, it can be concluded that the number of signals is not a factor that significantly impacts drivers' path choice.



	less than 1 mile	between 1 and 5 miles	between 5 and 10 miles	more than 10 miles
Real Paths	3.26	2.43	1.51	0.16
Shortest Distance Paths	2.92	2.11	1.33	0.25
Shortest Time Paths	3.00	2.22	1.34	0.25

Figure 6-19: Number of Signals per Mile

Table 6-11: Effect of Signals on Path Choice - Real vs. Shortest Paths

Path	Results of paired t-test	Path Length			
		Shorter than 1 mile	Between 1 and 5 miles	Between 5 and 10 miles	Longer than 10 miles
Real Path	Mean signals per mile along real path (μ_r)	3.296	2.535	1.582	0.160
	Degrees of freedom	1,601	3,707	319	62
Vs. Shortest Distance Path	Mean signals per mile shortest distance path (μ_s)	3.059	2.247	1.471	0.252
	Mean difference (μ_r - μ_s)	0.236	0.288	0.111	-0.092
	t-statistic	7.390	16.137	2.290	-2.423
	p-value (one-tailed)	1.000	1.000	0.989	0.009
Vs. Shortest Time Path	Mean signals per mile along shortest time path (μ_s)	3.129	2.352	1.343	0.165
	Mean difference (μ_r - μ_s)	0.167	0.183	0.239	-0.005
	t-statistic	4.924	9.430	4.015	-0.288
	p-value (one-tailed)	1.000	1.000	1.000	0.387

H_0: $\mu_r - \mu_s \geq 0$, H_a: $\mu_r - \mu_s < 0$
At 95% confidence interval of the difference, the p-value less than 0.05 indicates it is safe to reject H_0 and H_a may be accepted.
Values in shaded cells indicate real paths have fewer signals per mile than do shortest paths statistically.

6.5 Summary

When they choose their paths, drivers prefer roads with higher classes. The percentages of primary and secondary roads along real paths are much higher than their percentages in the network composition. The longer the trip, the higher the percentage of roads with higher classes.

The real paths do not contain more primary or secondary roads than computed shortest time paths statistically. This result means that the road classification is not the reason why real paths deviate from shortest time paths.

The comparison regarding signal numbers along paths reveals that drivers do not try to avoid traffic lights. This finding was reached because the statistical analysis does not show that shortest paths have more signals per mile. In other words, the number of signals along path is not a significant factor influencing drivers' path choice.

Paired sample t-tests show that fewer turning movements for real paths than for both shortest distance and time paths. Thus, it maybe concluded that the number of turning movements significantly impacts the paths drivers choose. Turns at signalized intersections did not influence drivers much, but drivers did try to minimize the number of non-signalized turns along their path. Statistically, computed shortest paths have more non-signalized turns than do real paths and fewer signalized turns. Therefore, one can conclude that the number of turning movements, especially at non-signalized intersections, is a significant factor affecting drivers' path choice.

The exceptions in the analysis are few and they all happen with paths longer than 10 miles. Since such paths have much smaller sample size and fewer turns and signals per mile than shorter paths, these exceptions do not change the final conclusion.

CHAPTER 7
IMPACT OF TURN PENALTIES ON PATH ALGORITHMS

Analysis in previous chapters revealed that drivers tend to minimize turns, especially left turns, along the route. Presented in this chapter is a methodology for incorporating turn-penalties as part of the network data structures and path finding algorithm. A turn penalty is the time taken to negotiate the indicated turn. It is used as an impedance variable within path finding algorithms. The example in Figure 7-1 illustrates different path finding results before and after the turn penalties are taken into consideration.

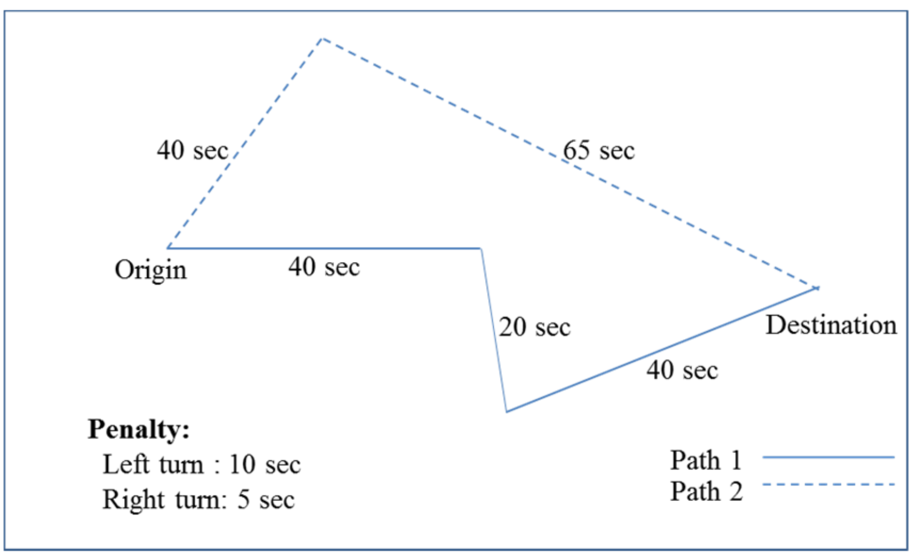


Figure 7-1: Influence of Turn Penalties on Path Finding

In this simple network, when turn penalties are not taken into consideration, the time impedances from the origin to the destination through Path 1 are 100 seconds and through Path 2 are 105 seconds. There are one right turn and one left turn along Path 1, and only one right turn along Path 2. Assume that the penalties for a left turn and for right turns are 10 seconds and 5 seconds, respectively. When turn penalties are considered, the total impedance along Path 1 and Path 2 becomes 115 seconds and 110 seconds, respectively. Consequently, Path 2 would be the result of a search for shortest path when turn penalties are considered.

7.1 Calculation of Turn Penalties

Commercially available tools for travel demand modeling include algorithms to account for turn penalties. However, literature on development and implementation of these algorithms has been fairly limited. Very few studies have focused on appropriate values for turn penalties and on how to quantify these penalties. Thériault, Vandersmissen, Lee-Gosselin and Leroux (1999) has chosen penalties of 24 seconds for left turns and 12 seconds for right turns. In another study, 30 seconds and 7.5 seconds were thought more suitable penalties for left and right turns, respectively (Yiannakoulias Bland and Svenson, 2013). Both these studies were based on empirical and qualitative analysis.

With the advantage of a large-size dataset that contains trajectories of real-world trips, this study has focused on finding a reasonable quantitative method to determine turn penalties. The analysis is based on calculation of time taken to negotiate the turn.

The data on trip paths are represented by a GPS point trajectory and are then conflated with network data as a sequence of network links. The time taken for turns at intersections is calculated from the time stamps associated with each GPS point. Each turn is associated with a deceleration and acceleration zone at the approaches to an intersection. Depending on the signal status and traffic conditions, each turn may also be associated with stopped delay. The total delay associated with an intersection is the time differential between free-flow travel times on the approach links and the time taken to negotiate the turns.

As shown in Figure 7-2, the turning time should be the time taken to travel through the whole distance represented by the dashed line.

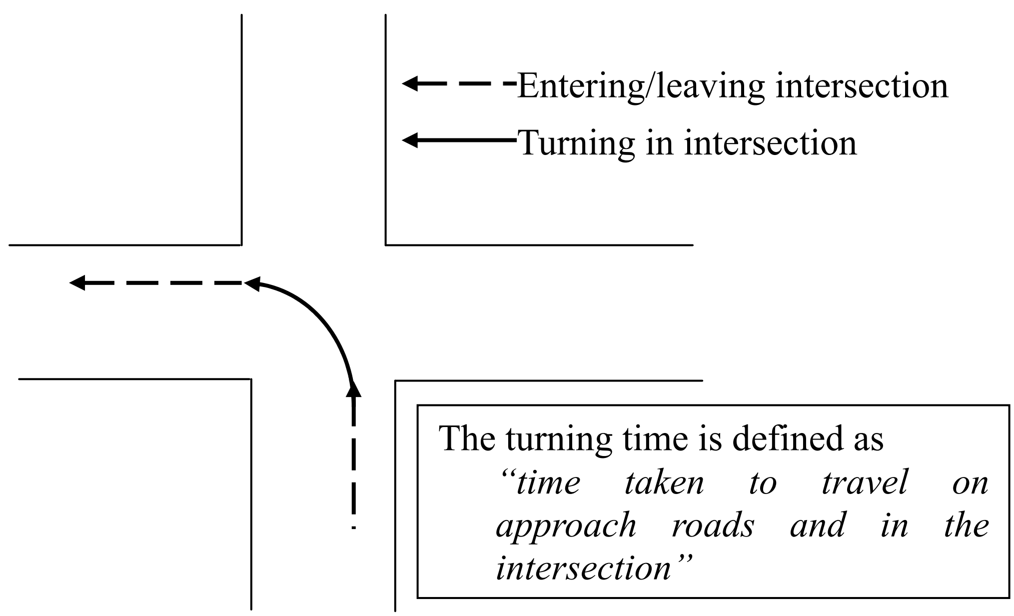


Figure 7-2: Definition of a Turning Movement

Red parts of the dashed line represent lengths traveled out of the intersection in the process of completing a turning movement. The length of the approach, where negotiating the turn-movement is expected to alter travel speeds, cannot be too long or too short. It should include adequate distance for deceleration and acceleration caused by intersection conditions, such as waiting for a traffic signal or yielding to vehicles with right-of-way. At different intersections, the distance cannot be the same because various speed limits, signal statuses, and traffic conditions together determine the initial and final speeds, as suggested by Table 7-1.

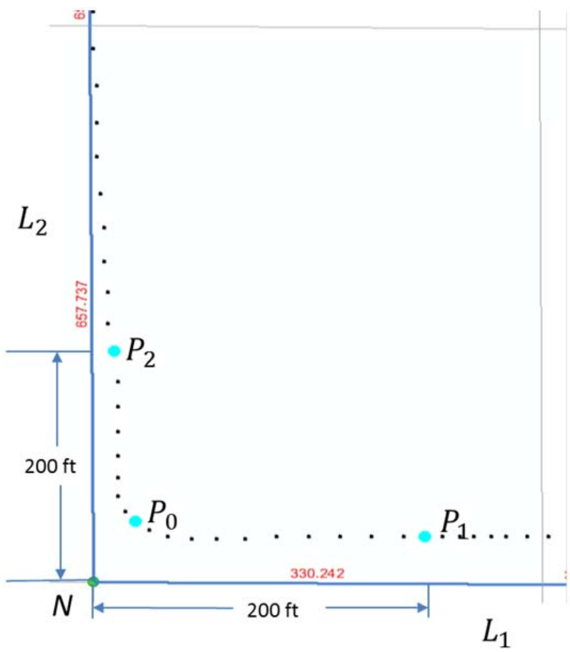
Table 7-1: Deceleration Distance for Typical Passenger Car

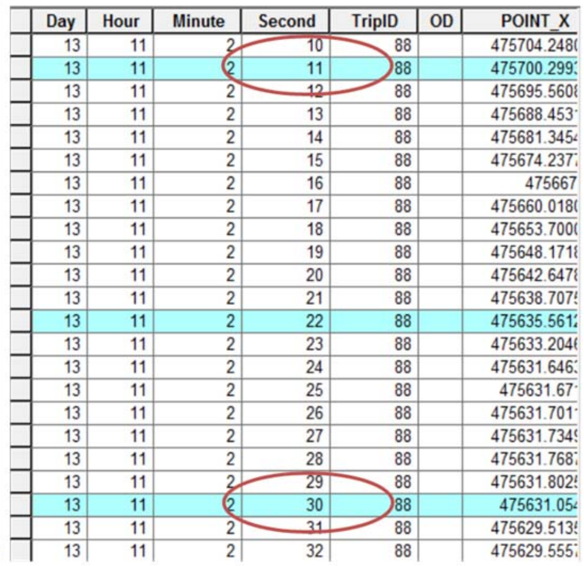
Deceleration Distance (feet)		Final Speed (mph)			
		15	10	5	0
Initial Speed (mph)	55	268	280	287	290
	45	172	184	192	194
	35	96	108	115	117
	25	38	50	57	60

Note: Calculations are based on AASHTO's standard deceleration rate of 11.12 ft/s^2, and for 0% grade only

As the primary purpose of this analysis is to develop a method for quantifying the influence of turning movements, a uniform length of 200 feet on both approaches to the intersection (entering and leaving) is assumed to represent adequately the deceleration and acceleration zones.

Figure 7-3 illustrates a simple turn penalty computation from the GPS data points. In this example, the green dots represent network nodes and node *N* is an intersection where the vehicle is making a right turn. GPS points tracking the vehicle movement are represented by black dots and the blue lines represent the network links on the path(s).





	Day	Hour	Minute	Second	TripID	OD	POINT_X
	13	11	2	10	88		475704.248
	13	11	2	11	88		475700.299
	13	11	2	12	88		475695.560
	13	11	2	13	88		475688.453
	13	11	2	14	88		475681.345
	13	11	2	15	88		475674.237
	13	11	2	16	88		475667
	13	11	2	17	88		475660.018
	13	11	2	18	88		475653.700
	13	11	2	19	88		475648.171
	13	11	2	20	88		475642.647
	13	11	2	21	88		475638.707
	13	11	2	22	88		475635.561
	13	11	2	23	88		475633.204
	13	11	2	24	88		475631.646
	13	11	2	25	88		475631.67
	13	11	2	26	88		475631.701
	13	11	2	27	88		475631.734
	13	11	2	28	88		475631.768
	13	11	2	29	88		475631.802
	13	11	2	30	88		475631.05
	13	11	2	31	88		475629.513
	13	11	2	32	88		475629.555

Figure 7-3: Calculation of Turning Time from GPS Points

The calculation of actual turning time follows the steps outlined below:

1) Find the nearest GPS point P_0 to the node N where the turning movement occurred;
2) Start from P_0, traverse points along the trip trajectory backward, locate the first point P_1, which is over 200 feet from P_0, and record the time stamp of P_1 ;
3) Repeat Step 2 except traversing forward along the trajectory, identifying P_2.

The duration between two time stamps at P_1 and P_2 is the actual turning time. In this example it took 19 seconds for the driver to complete the turning movement.

The calculation of estimated turning time is similar. Starting from the turning node and along the real path, the approach links (entering and leaving) to the node are examined one-by-one. The travel time on these links was accumulated until the total link length reaches 200 feet. Usually the last examined link of forward and backward directions is segmented into two parts to get exactly 200 feet long link. Segments closer to node *N* are assumed to fall within the turn-movement influence zone (200 feet). It is assumed that the remaining portion of the link is not influenced by the turn movement. Expected travel times in each link-segment are calculated in proportion to the segment length. As a result, only the time duration corresponding to 200 feet is calculated for forward and backward directions, respectively.

In Figure 7-3, both links L_1 and L_2 are longer than 200 feet. Table 7-2 presents the simple mathematical calculation.

Table 7-2: Calculation of Turning Time from Link Sequence

Link	Link Length (feet)	Speed Limit (mph)	Time for traversing whole link (sec)	**Time for traversing 200 feet segment (sec)**
L_1	330.2	30	7.5	**4.56**
L_2	657.7	30	14.9	**4.56**
The turning time at this intersection can be easily calculated as 4.56 + 4.56 = 9.12 seconds				

With both actual and estimated turning times, the right turn penalty at this intersection can be easily calculated as 19 – 9.12 = 9.88 seconds.

A special situation is presented when two or more turning movements happen at multiple successive intersections, and the road segments connecting them to each other are shorter than 200 feet. Under this scenario, the multiple turns would be considered together (Figure 7-4). The measured road length should start 200 feet before the first intersection and end 200 feet after the last one. Therefore, the turning time would include the travel time spent on the 400 feet, then within these intersections, and road segments between them. Then the time would be split equally as turning time of the multiple turning movements respectively.

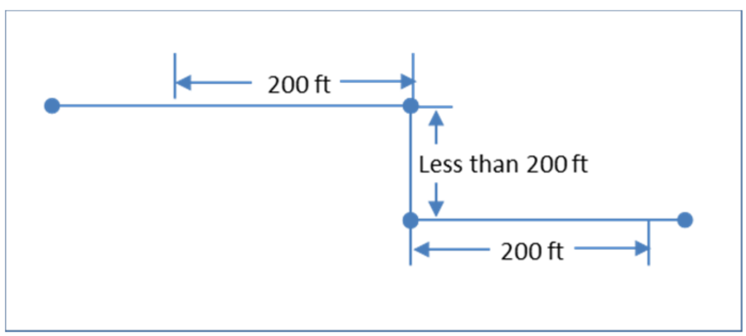


Figure 7-4: Example of Two Adjacent Turns

All turning movements of real paths are grouped into four categories by direction of turns and signal existence of intersections (Table 7-3). Turn penalties are calculated for each individual turning movement, and are averaged for each category .

Table 7-3: Turn Penalties

Turning Movements	Number of Turns	Average Turn Penalties (second)
Signalized Left Turn	3,299	29.41
Signalized Right Turn	2,960	15.06
Non-signalized Left Turn	7,105	14.71
Non-signalized Right Turn	5,465	12.51

7.2 Implementation of Turn Penalties in Path Finding Algorithm

Incorporating turn penalties into path finding algorithms is not very straightforward. The labeling method calculates and updates costs from the start node to other nodes until the destination is reached with the minimum impedance. During this process, the algorithm keeps track of "scanned" nodes, nodes with costs that will not be updated or improved anymore. The candidate set is used to store "labeled" nodes that are neighbors of one or more of the "scanned" nodes. During the execution of the algorithm, the cost from the origin to each labeled node is minimized constantly, and the corresponding scanned node is flagged as the precedent to this labeled node.

If turn penalties are included in the calculation of impedances, for each labeled node, not only its preceding node but also the node that preceded the precedent, need should be recorded. Figure 7-5 provides a simplified example to illustrate the process of updating impedance to labeled nodes with and without turn penalties.

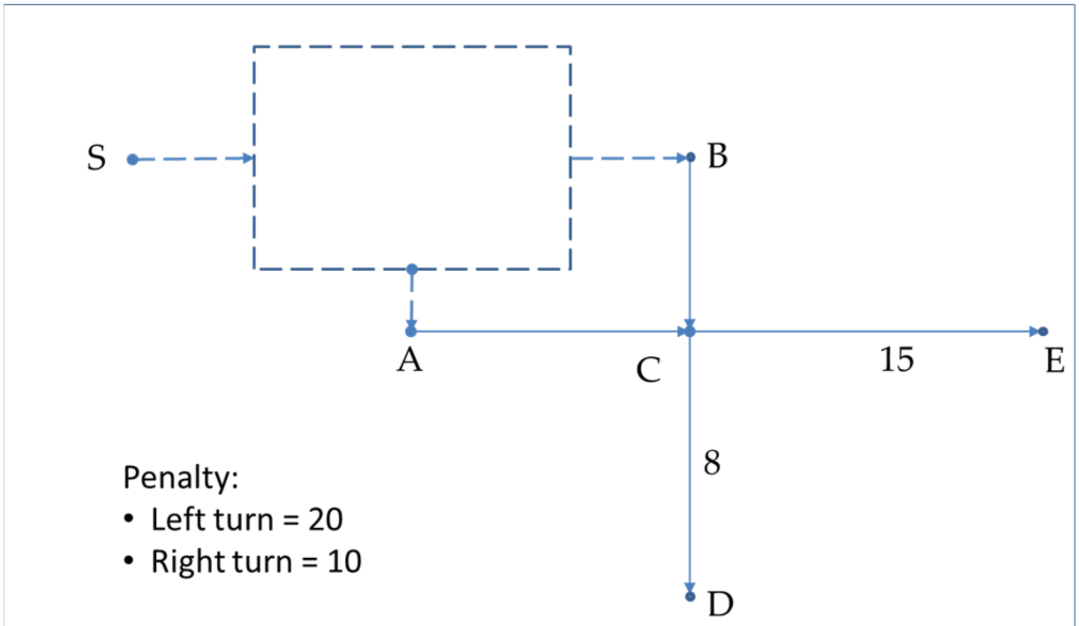


Figure 7-5: Example of Integrating Turn Penalties into Path Finding Algorithm

In this example, node S is the origin, and the dashed rectangle represents the street network that has already been scanned. At this point in the execution of the algorithm, two known paths from node S to node C include one path through the node A and the other through node B. Assume the impedances of two paths are as follows:

$$p(S, \ldots, A, C) = 100$$

$$p(S, \ldots, B, C) = 105$$

If no turn penalty is considered, node A is flagged as the precedent of node C because this path has less impedance. In the following step that updates impedances to nodes D and E, the impedances of links CD and CE are added to this current cost respectively. Now the path impedances from node S to nodes D and E are:

$$p(S, \dots, A, C, D) = 100 + 8 = 108$$

$$p(S, \dots, A, C, E) = 100 + 15 = 115$$

So far, node *B* does not appear on the solution space of paths.

The calculation becomes more involved if turn penalties are taken into account. Both paths reaching node *C* need to stay in the candidate set before nodes *D* and *E* are updated because both node *A* and node *B* must be included to identify the turning movements. The resulting four paths and their impedances are:

$$p(S, \dots, A, C, D) = 100 + 8 + 10 = 118 \text{ (Right turn)}$$

$$p(S, \dots, A, C, E) = 100 + 15 = 115 \text{ (Through movement)}$$

$$p(S, \dots, B, C, D) = 105 + 8 = 113 \text{ (Through movement)}$$

$$p(S, \dots, B, C, E) = 105 + 15 + 20 = 140 \text{ (Left turn)}$$

7.3 Verification

To verify the influence of turning movements on path choice, the turn penalties are applied to the path finding algorithm. The following four scenarios are examined:

1) No turn penalties are applied;
2) Penalties are applied on signalized turns only;
3) Penalties are applied on non-signalized turns only; and
4) Penalties are applied on all turns regardless of signal status.

Four sets of paths, one each for the scenarios described above, are generated. Since the turn penalties are represented as part of time impedance, only travel time is used as path costs. The first scenario, "no penalty applied", would yield the same set as the set of theoretic shortest time paths in Chapter 5.

The generated paths with turn penalties are then compared to real paths in the trip dataset. For each trip, if node sequence along a generated path is the same as its counterpart in the base case, the path is then recognized as identical to the real path. By examining trips one by one, the identical rate could be identified for each scenario. The rate is defined as:

$$Identical\ Rate = \frac{number\ of\ identical\ trips}{number\ of\ all\ trips}$$

Table 7-4 lists the identical rates under different turn penalties applied.

Table 7-4: Summary of Identical Rates with Turn Penalties Applied

Applied Penalties	Identical Rate
No Turn Penalty	34.26%
Signalized Turn Penalties Only	28.04%
Non-signalized Turn Penalties Only	36.14%
All Turn Penalties	28.04%

These results show that considering turn penalties into path finding did not improve the rate of identical paths between real world paths and theoretical paths. One possible reason for this is that the mere existence of a turn, rather than the delay associated with the turn, may have a real influence on the path choice. Also, it is conceivable that generalized turn penalties computed in this study may not be true representations of field conditions. It can be also concluded from Table 7-4 that, with incorporation of non-signalized turn penalties, the path finding algorithm could have more chances to generate the path identical to people's choice in the real world. On the other hand, when all turn penalties are applied (signalized or non-signalized), the algorithm could not improve the rate of identical paths compared to the scenario with no penalty.

The results also show that the optimal path yielded by incorporating turn penalties has not significantly increased the chance of matching theoretical paths to actual paths. The primary reason for this may be that the dataset lacks additional information regarding attributes of turning movements. Descriptions are not sufficiently specific or satisfactory on how a turning movement occurs. For example, the dataset did not indicate presence of exclusive turning lane(s). The other possible attributes to include are if a turn is from a major road to a minor road, or vice versa, and the type / nature of other traffic devices used at the specific intersections. These attributes could yield many more combinations, which are expected to produce more accurate turn penalties.

CHAPTER 8 NETWORK PRUNING

The analysis presented in Chapters 4 to 7 provide insights into the influence of street network variables, especially intersection control and turns, on route-choice. In this chapter, a detailed study on the effect of network size on path-finding algorithms for a one-to-one search is presented.

8.1 Sub-network Concept

When the search is limited to finding a path between a single origin-destination pair, only a part of the entire street network is relevant for computations; most nodes and links in the network could be irrelevant to the process of finding a path. Therefore, it would be reasonable to limit the search to a sub-network within which feasible solutions exist. Network pruning is the process of extracting or flagging this sub-network from the original network. In the sub-network, a feasible set of best alternative paths exists.

Figure 8-1 presents a simple example of network pruning. Network pruning reduces the number of nodes from 19 in the full network to 9 in the sub-network and number of links in the full network from 29 to 10 in the sub-network.

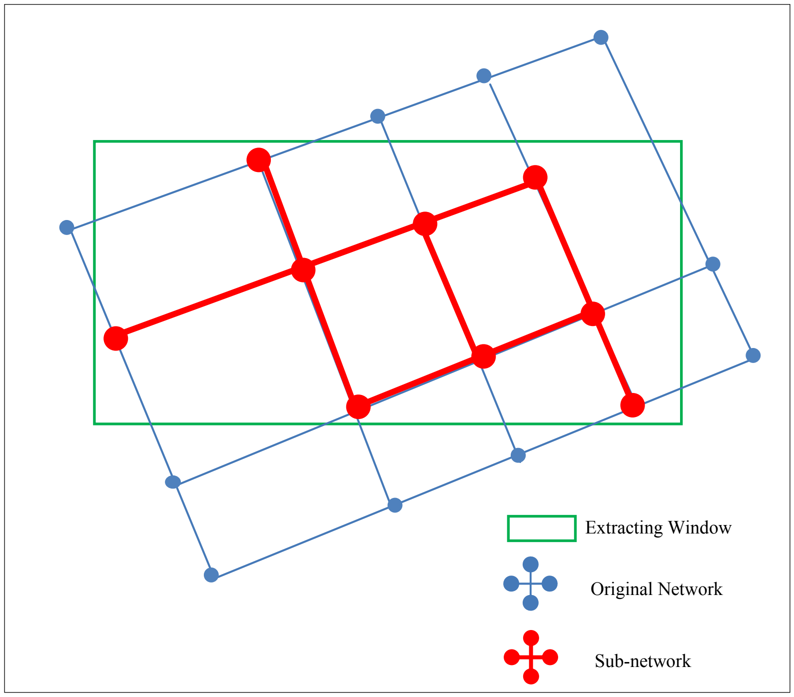


Figure 8-1: An Example of a Sub-network

The smaller the size of the sub-network, the shorter will be the computing times for the search. The risk of a sub-network is that a valid path may not connect the origin with the destination because the sub-network may have excluded necessary connectivity between the O-D pair. Also, due to exclusion of some portion of the network, the path search in the sub-network may not yield the same unique solution as did by the search in the full network. Therefore, the sub-network must be of sufficient size to account for possible deviations. The determination of sub-network size is important on achieving the balance between efficiency and accuracy.

Thus, an ideal pruned network should exclude irrelevant nodes and at the same time contain the nodes and connectivity necessary for finding a path between the O-D pair. Given the propensity of drivers to deviate from the intended direction, selection of a candidate sub-network for finding the path is a challenging task. Most commercially available GPS navigation systems contain path finding routines that prune a larger street network into a sub-network for arriving at route recommendations quickly. However, by their very nature such commercially adopted algorithms tend to be proprietary and their built in methodologies are seldom published in popular literature.

The typical path, “as the crow flies” between a given pair of origin and destination in a network, is rarely a straight line. However, drivers tend to choose a path as close to a straight line as possible. The degree of connectivity, hierarchy of roadway facilities, or pure personal preferences of the driver may influence the negotiated path and deviate substantially from the straight-line path. Though counter-intuitive in nature, a driver’s familiarity with the network may result in a path choice that has the initial and/or final stretches of the path going in the opposite direction of the intended direction of the travel.

8.2 Bounding-box Method

To explore the patterns related to deviations in real-world paths, a bounding box method is introduced. The bounding box is a rectangle that is determined by the location of given O-D pair. All four sides of the bounding box are parallel to the X or Y axis. The origin and destination nodes are located at the diagonal corners of the rectangle, and the straight line connecting them divides the rectangle into two triangular parts.

Enlarging the bounding box with a buffer forms a rectangular extracting window (something like a cookie cutter) to overlay the original network for obtaining a sub-network. Empirical research on appropriate buffer width to obtain the most appropriate size of sub-network is currently not available. Therefore, simple rules are laid out for methodically selecting a sub-network within which a path can quickly be found for a given O-D pair. The basis for establishing these rules is the analysis of real paths chosen by the Twin Cities drivers. The methodology focuses on first analyzing real-world paths to study deviation patterns from straight-line paths in the intended direction of travel. Then rule-sets are adopted and the efficiency of the path finding algorithm for each rule-set is studied.

8.3 Exploration on Real Paths

Analysis of real-world paths aims to reveal the relationship between the path and relative O-D locations. The analysis examines the patterns of path deviation from the straight-line connecting the O-D pair. The statistical results provide the foundation for the methodology of extracting the most appropriate sub-network.

8.3.1 Definitions of Deviations

For every point on the path we can measure the perpendicular distances from the closer sides of the bounding box, which are parallel to the X and Y axis, respectively. The two distances are defined as x and y deviation of this path-point. If the point is located within the bounding box, as P_1 in Figure 8-2, both x and y deviations are recognized as zero.

In the same figure both P_2 and P_3 are out of the bounding-box. However, P_2 is located between the extended lines of two opposite sides, so its x deviation is zero. Only P_3 is deviated from the bounding box at both directions of x and y axes.

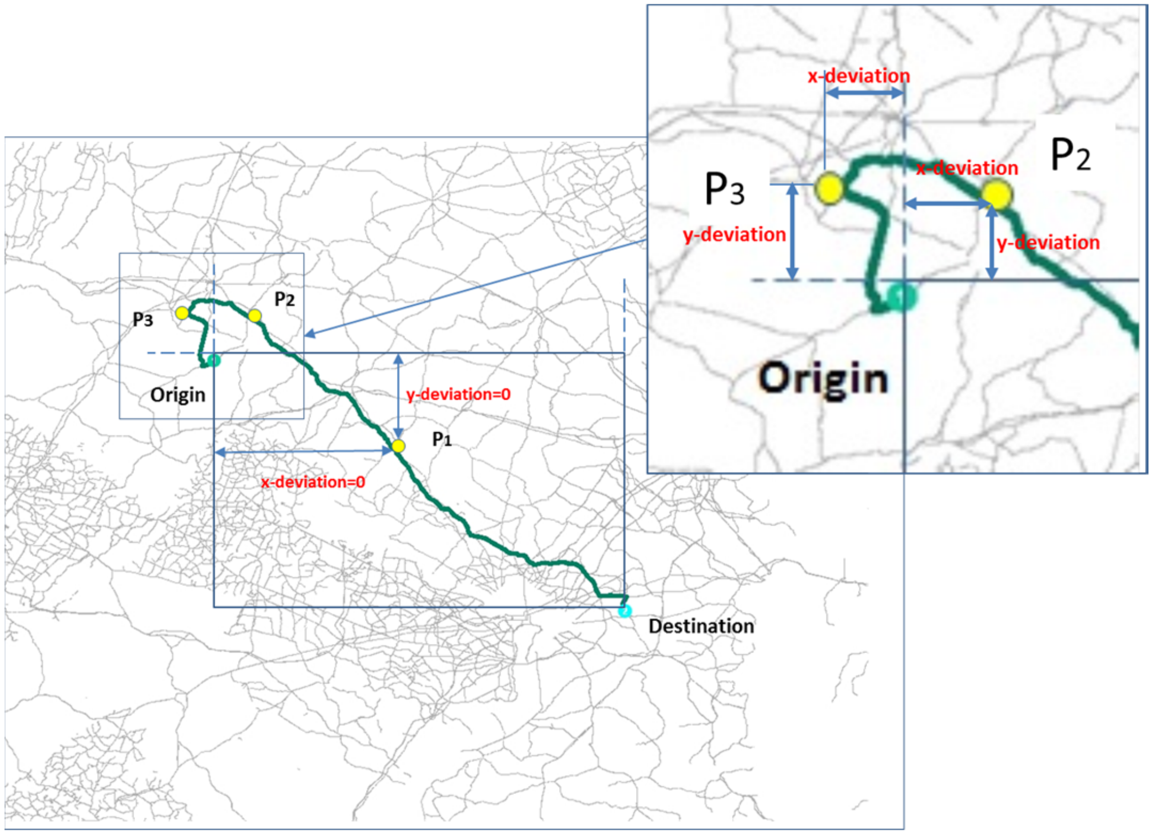


Figure 8-2: Illustration of Point Deviations

8.3.2 <u>Statistics on Real Paths</u>

Comparing deviations of every point on the path results in the maximum deviations for each path. Among 5,694 trips studied, 256 trips are found to contain the entire path within the bounding box determined by origin and destination. Only 4 paths have had a maximum deviation of more than 5 kilometers. Most paths (4,323) have partial segments out of the box and the maximum deviation is shorter than 1 kilometers.

The slopes of the straight lines connecting the O-D pairs were examined to explore for other characteristics given by O-D locations that may impact the potential path deviations. The value of the slope represents the orientation of the destination with respect to the origin to some extent. When the absolute value of the slope is great than 1, the *y*-parallel sides of the bounding box are longer than the *x*-parallel sides. On the other hand, the *x*-parallel sides are longer when the slope's absolute value is less than 1.

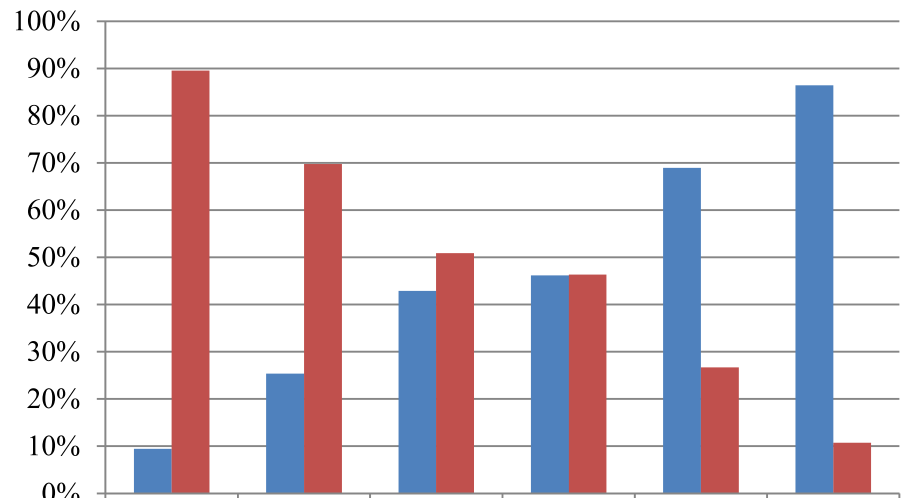
Table 8-1 shows that path deviations are correlated to the slope. The table aggregates real trips by slope value and relationship between values of *x* and *y* deviations, listing the number of trips for each scenario. Without considering the circumstances of equal deviations (only a small portion), it can be seen that when the absolute value of slope is less than 1, there are more cases that the *x* deviation is smaller than *y* deviation. As the slope value increases, the cases when the *x* deviation is longer are more and more frequent.

Table 8-1: Number of OD Pairs by Various Circumstances

Absolute Value of Slope	Comparison between *x* and *y* Deviations		
	$x > y$	$x = y$	$x < y$
[0, 0.1]	65	7	617
[0.1, 0.6]	397	76	1091
[0.6, 1]	262	38	311
[1, 1.7]	277	45	278
[1.7, 10]	1023	65	396
[10, ∞]	637	21	79

Figure 8-3 illustrates the comparison between two scenarios with various slope values. In the first, the *x* deviation is larger than the *y* deviation; in the second, vice versa. When the absolute value of the slope is close to one, the case numbers of these two scenarios are very close to each other, which means the *x*- and *y*-deviation tend to be same.

When the slope is far away from one, the closer the slope value is to zero, the likely the *y* deviation is larger than the *x* deviation. The change trend is opposite as the slope value gets bigger and bigger.



	[0, 0.1]	[0.1, 0.6]	[0.6, 1]	[1, 1.7]	[1.7, 10]	[10, ∞]
X Dev > Y Dev	9.43%	25.38%	42.88%	46.17%	68.94%	86.43%
X Dev < Y Dev	89.55%	69.76%	50.90%	46.33%	26.68%	10.72%

Figure 8-3: Percentages of Two Scenarios Comparing *x* and *y* Deviations

The relationship between *x* and *y* deviations for each slope scenario is quantified in Table 8-2. These relationships form the basis for experimental design, the goal of which is identifying the bounding-box dimensions.

Table 8-2: Relationship between X and Y Deviations

Absolute Value of Slope	Average Deviation (m)		$\frac{x\ Deviation}{y\ Deviation}$
	x	y	
[0, 0.1]	24	170	7.01
[0.1, 0.6]	42	151	3.59
[0.6, 1]	65	98	1.51
[1, 1.7]	82	87	0.94
[1.7, 10]	150	42	3.60
[10, ∞]	163	17	9.70

8.3.3 Summary

The path-deviation analysis has revealed relationships between the size of the sub-network and O-D locations. Very few paths have deviations more than one kilometer, and therefore, the minimum buffer width is set as one kilometer. The analysis also revealed that the relationship between *x*- and *y*-deviations is correlated to the slope of the straight line connecting origin and destination. When the absolute value of the slope is less than 0.1, the *y*-deviation is about seven times the *x*-deviation. The multiple is reduced to 3.6 times when the slope value is between 0.1 and 0.6 (Table 8-2). When the slope value belongs to two symmetric ranges [1.7, 10] and [10, ∞], the multiples are about 3.6 and 9.7, respectively.

The analysis of path deviations asserts an intuitive assumption that the paths drivers choose are not far from the proximity area determined by their origin and destination. The procedure to extract sub-network would purge the nodes and links (alternatively, flag the nodes and links for processing exclusion) that are supposed to be irrelevant to the potential path.

8.4 Experimental Setup

An experiment is conducted to examine the performance of network pruning (the sub-network concept) in terms of path finding. Dijkstra's algorithm is applied to find the shortest paths between computer-generated O-D node pairs. With different buffer types, two groups of sub-networks are extracted from the real-world street network using the bounding-box method. Measurements regarding accuracy and efficiency are used to evaluate the test results.

8.4.1 <u>Generation of the Experimental Dataset</u>

The revealed relationships between path deviations and O-D locations are based on paths with Euclidean distances shorter than 40 kilometers. Using a 40-kilometer separation as a surrogate, the experiment first randomly generated 500 pairs of origin and destination nodes with Euclidean distances between O-D nodes of no more than 50 kilometers. These O-D pairs thus generated are grouped as shown in Table 8-3.

Table 8-3: Statistics of Random O-D Samples

Group by Euclidean Distance	Number of Trips	Euclidean Distance between O-D Nodes (m)			
		Average	Std. Dev	Minimum	Maximum
00-05	7	2,706	1,343	757	4,749
05-10	25	7,816	1,674	5,108	9,979
10-15	47	12,938	1,334	10,172	14,970
15-20	61	17,497	1,574	15,015	19,958
20-25	65	22,245	1,548	20,026	24,911
25-30	71	27,586	1,574	25,030	29,989
30-35	66	32,361	1,421	30,028	34,976
35-40	57	37,654	1,362	35,056	39,793
40-45	55	42,646	1,477	40,063	44,902
45-50	46	47,216	1,428	45,003	49,634
Total	500	281,88	11,897	757	49,634

The efficiency of the path finding algorithm for the full network and for the sub-network must be compared. Dijkstra's algorithm was first applied to the entire network. This yielded a set of paths and the associated computational speeds as the baseline for comparison. Then a set of paths is generated for each buffer width using Dijkstra's algorithm. Paths generated for the sub-network with varying buffer widths are compared for both accuracy and computational efficiency to the paths in the baseline scenario. When finding the paths for full network and sub-networks, the algorithm also incorporated turn penalties calculated in Chapter 7. Since turn penalties are represented as time cost, the algorithm used only travel time as impedance.

8.4.2 Size of Sub-network

The experiment tests two different types of sub-networks, differentiated by their buffer types:

1. Buffer deviations based on field data (proportional buffer)
2. A uniform buffer

The first type of sub-network used information about deviations from the field data. The statistics on real paths has revealed that the slope of the straight line connecting O-D pairs can determine the proportional relationship between paths' *x*- and *y*-deviations. Therefore, a narrow buffer can be set for the two shorter sides of the bounding box. A wider buffer for the other two sides can be obtained by multiplying a coefficient based on the straight-line slope. As the narrow buffer changes, the wider buffer changes proportionally (so called "proportional buffer"). Figure 8-4 illustrates two examples of the proportional buffer.

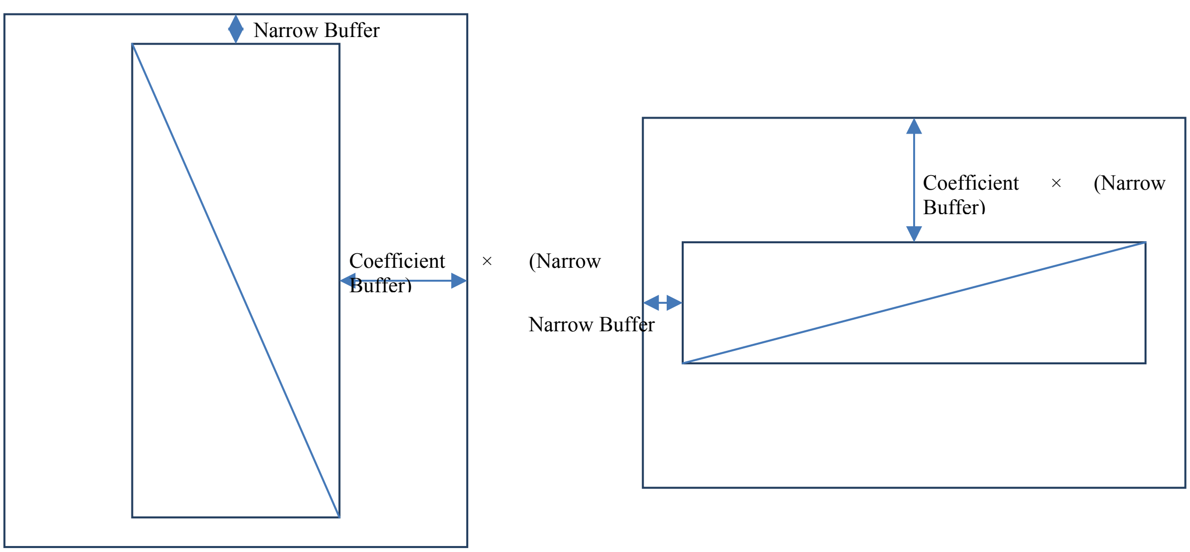


Figure 8-4: Two Examples of the Proportional Buffer

The experiment contains a series of tests on different sizes of sub-networks. In each test series, the narrow buffer widths for all O-D pairs are the same and the wider buffer widths are determined by the straight-line slopes. The narrow widths start at one kilometer and increase at one kilometer increments. The wider buffer widths increase based on corresponding buffer coefficients. As in the previous section, three buffer coefficients - one, two and four - are employed for six ranges of the slope's absolute value (Table 8-4).

Table 8-4: Buffer Widths Based on O-D Locations

Absolute Value of the Slope	**Buffer Width (km)**	
	x	***y***
[0, 0.1]	*Narrow Buffer*	4× *Narrow Buffer*
[0.1, 0.6]	*Narrow Buffer*	2 × *Narrow Buffer*
[0.6, 1]	*Narrow Buffer*	*Narrow Buffer*
[1, 1.7]	*Narrow Buffer*	*Narrow Buffer*
[1.7, 10]	2 × *Narrow Buffer*	*Narrow Buffer*
[10, ∞]	4 × *Narrow Buffer*	*Narrow Buffer*

The second type of sub-network uses a uniform buffer, which has the same width on all four sides of the bounding box and it does not vary for different O-D pairs. This type of sub-network has been used in some previous studies (Karami, Sutovsky and Durcik, 2008). The uniform buffer is used to generate results as control variables.

The buffer widths used in tests also start at one kilometer and increase at one kilometer increments. The test results are used to compare to those with proportional buffer.

8.4.3 Evaluation Approaches

In the experiment the most important consideration is accuracy –(how close paths within the sub-network are to their counterparts in the entire network). The second most important consideration is efficiency (how fast the paths can be found in the sub network relative to the search in the full network).

Two types of errors are defined to evaluate the accuracy of sub-networks. The first is a measure of incomplete path. An incomplete path indicates that the pruning algorithm failed to find a valid path connecting origin and destination. An incomplete path error would be unacceptable under any circumstances.

The second error measures non-similarity of paths obtained in the full network and paths obtained in the pruned networks. The way to identify non-identical paths is similar to the way to compare real paths and shortest paths (see Chapter 5). The sequences of paths between any given O-D pair yielded by each of the sub-networks and their counterpart in the entire network are compared on a node-by-node basis. The two paths are considered identical when every node in the path-sequence derived from the pruned network is the same as the sequence in the full path. The error rate is the percentage of total paths that are not identical. Figure 8-5 shows examples of two paths with different sub-network sizes.

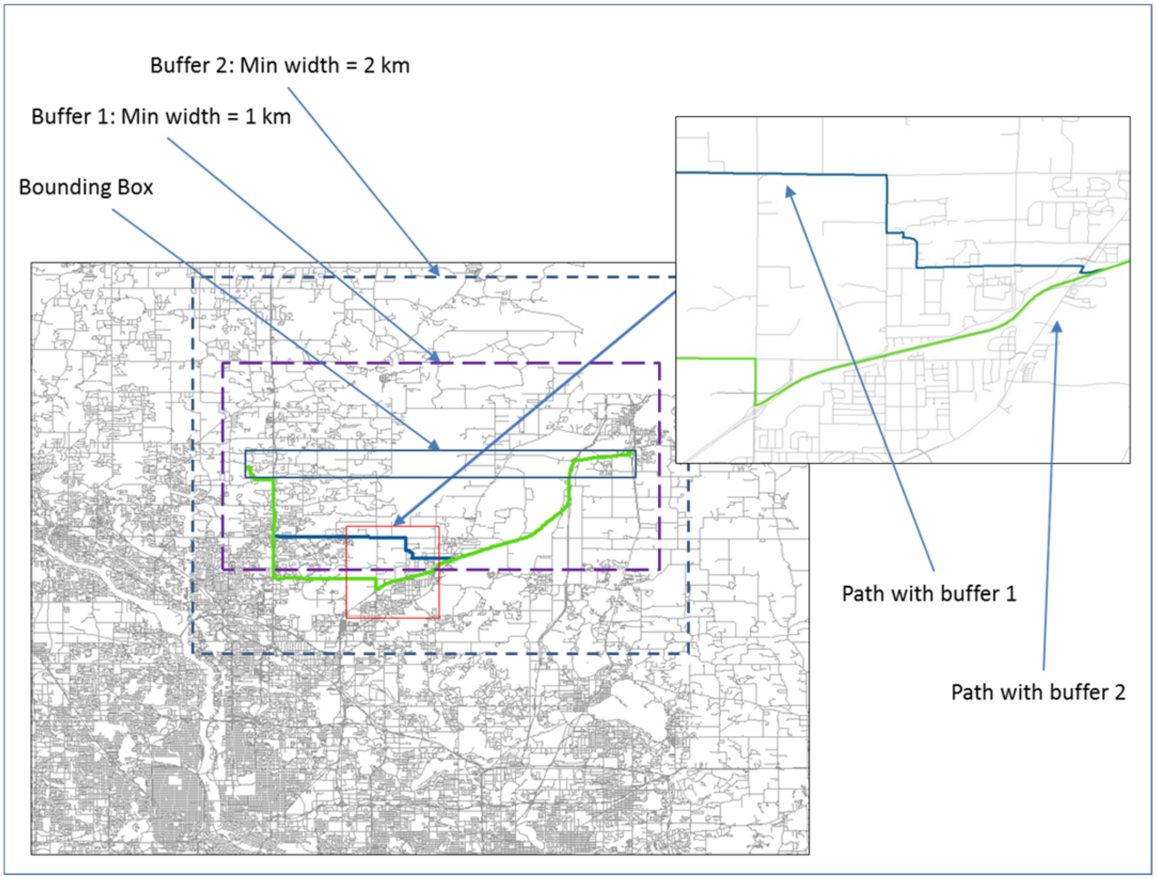


Figure 8-5: Examples of Paths with Different Sub-network Sizes

The Python script developed to implement this experiment was executed on a full network and all pruned networks using a workstation configured with 2.13GHz Intel(R) Xeon(R) processor with 12.0 GB RAM. The computing times (in seconds) taken by each run are recorded. The average computing time per path is used as a normalized measure to evaluate the efficiency of the sub-network.

8.5 Results and Findings

Presented in Table 8-5 is a summary of experimental results. It should be noted that a large enough sub-network can always yield valid and accurate paths. For both proportional and uniform buffer types, the error rates of incomplete paths are reduced to zero as the sub-network is enlarged. For example, when the width of the uniform buffer is 10 kilometers and the narrow width of the proportional buffer is 6 kilometers; all O-D pairs can be connected by a valid path within the sub-network. Although there are a few paths which are different from their counterparts obtained with the entire network, the error rates are very low (1.61% for the uniform buffer and 0.81% for the proportional buffer) and therefore are deemed acceptable.

The sub-networks obtained using uniform and proportional buffers are also expected to improve the computing efficiency of path finding. When the algorithm is applied to find a path in the entire network, it took on average 125 seconds per path. In a sub-network extracted using the proportional buffer, it took only 30.91 seconds per path when the number of incomplete path is zero. The sub-network with uniform buffer needs slightly more time (35 seconds per path). Figures 8-6 and 8-7 show reductions in computing time of all sizes of sub-networks. Both types of sub-networks can reduce by more than 70% the computation time compared to using the entire network.

Table 8-5: Error Rates and Computing Time of Sub-networks with Various Buffer Widths

Buffer Width		Number of Paths		Total Error Rate	Average Computing Time per Path (sec)
		incomplete	non-identical		
Uniform Buffer	1 km	70	191	52.62%	9.73
	2 km	19	169	37.90%	11.97
	3 km	11	86	19.56%	14.03
	4 km	6	60	13.31%	16.85
	5 km	3	42	9.07%	22.50
	6 km	2	30	6.45%	23.13
	7 km	2	22	4.84%	24.54
	8 km	1	18	3.83%	27.51
	9 km	1	11	2.42%	33.17
	10 km	0	8	1.61%	33.50
Proportional Buffer	Narrow Buffer =				
	1 km	19	130	30.04%	11.18
	2 km	5	61	13.31%	14.76
	3 km	4	30	6.85%	18.74
	4 km	3	16	3.83%	19.22
	5 km	1	7	1.61%	27.06
	6 km	0	4	0.81%	30.91

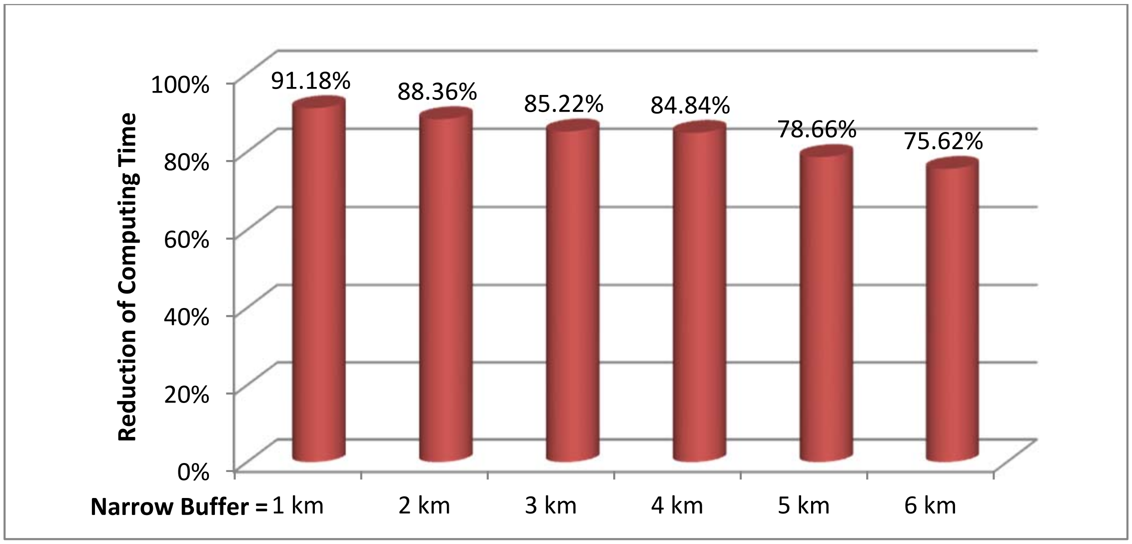


Figure 8-6: Reduction in Computing Time: Proportional Buffer

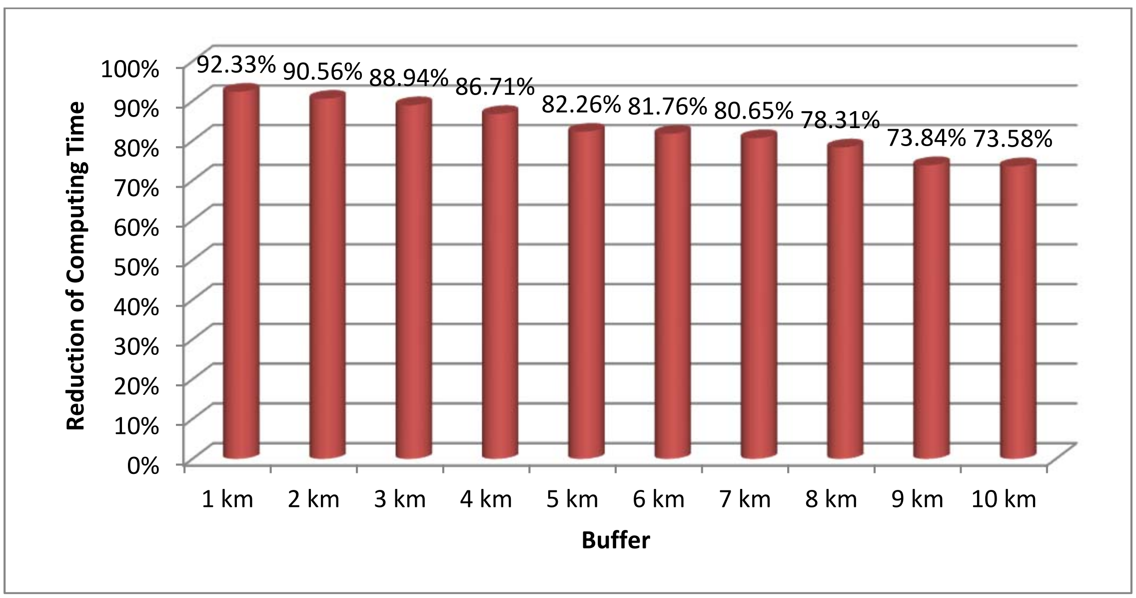


Figure 8-7: Reduction in Computing Time: Uniform Buffer

Tests are also conducted to compare the efficiency and accuracy of proportional and uniform buffers. Four sizes of sub-network are selected for both buffer types, respectively, forming four pairs for comparison. Each pair of two buffers generates the same number of incomplete paths, as shown in Table 8-6.

Table 8-6: Buffer Widths of Selected Sub-networks

Number of Incomplete Paths	Width of Uniform Buffer (km)	Narrow Width of Proportional Buffer (km)
19	2	1
3	5	4
1	9	5
0	10	6

Figures 8-8 and 8-9 illustrate that the proportional buffer has advantages on both evaluation measurements. While the rates of incomplete path are the same, the proportional buffer always has lower rates of non-identical path and shorter computing time.

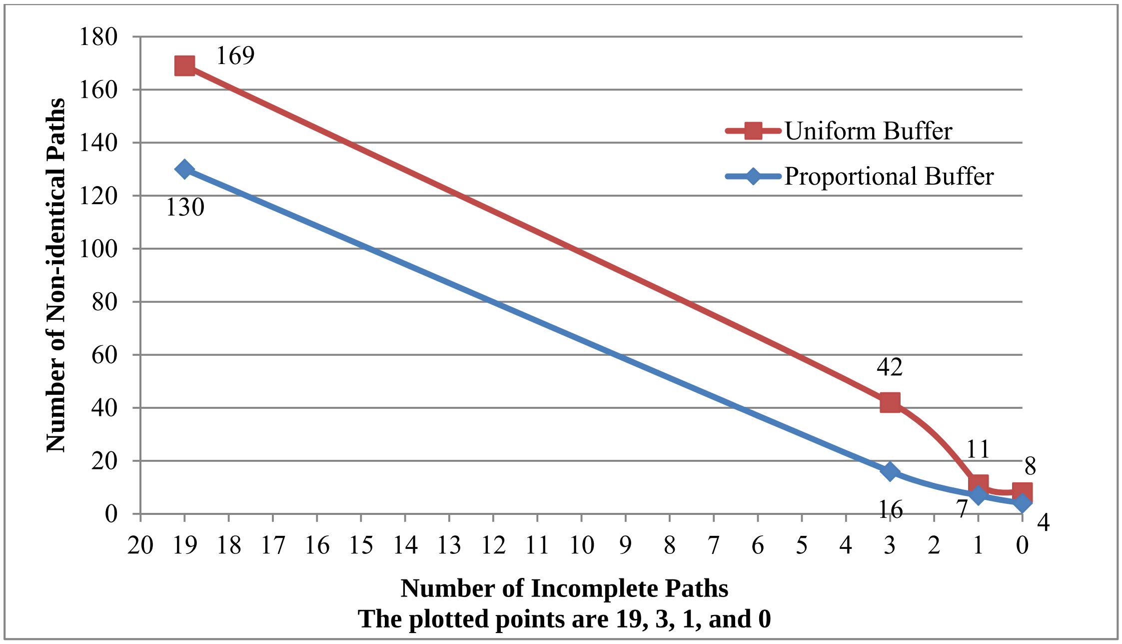


Figure 8-8: Comparison of Two Buffers on Result Accuracy

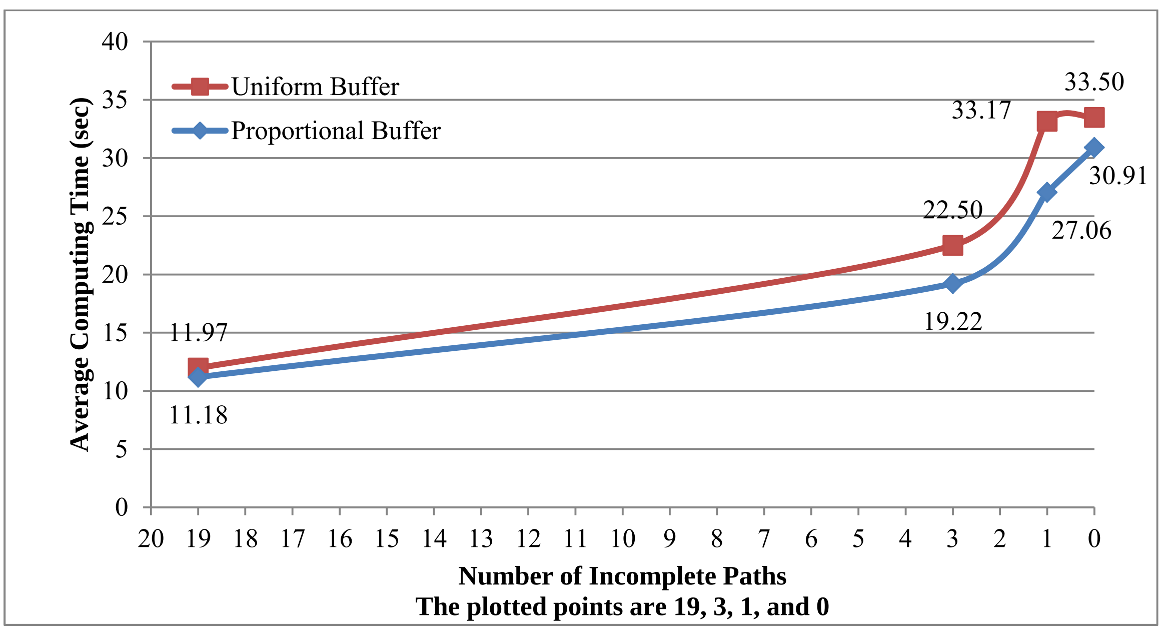


Figure 8-9: Comparison of Two Buffers on Computation Efficiency

8.6 Summary

The sub-network greatly improves the efficiency of path finding algorithms without losing much accuracy. When both uniform and proportional buffers are used to extract the sub-network, the average computing times are 10% to 30% shorter than the time to generate the paths in the entire network. The proportional buffer is more efficient than the uniform buffer, with the same level of accuracy. Useful rules for network pruning can be derived from this investigation. These rules can be incorporated in path finding algorithms for use in a variety of situations.

CHAPTER 9 CONCLUSIONS

9.1 Summary of Research Findings

This research examined the influence of certain street network and path variables on drivers' path choices in a metropolitan urban area (the Twin Cities of Minneapolis-St. Paul, Minnesota). Computed shortest distance paths and computed shortest time were compared to actual paths chosen in the real world. The analysis indicated that drivers do not necessarily choose the theoretical shortest paths. Instead, they are willing to spend longer time or travel longer distances on paths that have fewer turning movements.

Generally, drivers tend to avoid turning movements during their travel. There is statistical evidence to indicate that real paths have fewer turns per mile than both shortest time and shortest distance paths. When they must make a turn (left or right) to complete their trips, drivers seem more prone to making the turn at a signal controlled intersection, while at the same time trying to minimize the number of turns occurring at non-signalized intersections..

The research also explored the influence of traffic lights on path choice. The number of traffic signals is found not to be a significant factor during path choice processing. Statistical results also showed that in terms of the number of signals per mile, theoretical paths are not different from real paths.

Among various classes of the street network in the urban area, drivers prefer choosing roads with higher classes, even though the local road has a much bigger portion in the network composition. The longer the trips, the more primary and secondary roads the paths have.

Through the investigations on real paths, the study found the orientation of the straight line connecting the O-D locations is relevant to the relationship of x and y deviations of the real path. This relationship can help to develop the extracting method for the sub-network.

The experiment on two types of sub-network, with the proportional buffer and with the uniform buffer, showed that both of them can reduce computing time significantly. The ability to achieve expected accuracy is limited by the buffer width around the bounding box. Results of experiments also showed that the new concept of proportional buffer has advantages over the usually used uniform buffer in terms of balance between accuracy and efficiency of the algorithm. For a given error rate, the proportional buffer can save more computing time than the uniform buffer.

9.2 Research Contributions

This research used a large dataset of paths with trajectories tracked by GPS to identify the impacts of certain network and path characteristics on drivers' route-choice. Prior to this effort, most studies relating path choice behavior to network and path characteristics were based on stated preference surveys. Compared to stated preference surveys, the GPS tracking data are a better representation of how drivers choose route in actual practice. The findings from this influential analysis will make it easier to find paths more consistent with drivers' real choices and consequently provide more sound and solid solutions to traffic assignment problems and other problems in transportation planning.

This research not only revealed the impacts of turning movements on path choice, it also developed a methodology to quantify these impacts as turn penalties. The method computes turn penalties based on the difference between the time recorded by the GPS device and the time estimated from network link attributes. Individual turn penalties are averaged by directions of turns (left or right) and signal existence at intersections (signalized or non-signalized).

This research defined the concept of road availability to represent how much a specific class of road can be chosen by drivers when they are making a trip. This concept can help exclude possible impacts of the network composition on drivers' actual choice, so as to help identify drivers' preferences among different road classes.

This research also proposed a novel proportional buffer method to extract the sub-network for a specific pair of origin and destination. The experimental results showed that the proportional buffer is superior to the uniform buffer considering the balance between efficiency and accuracy. Even though certain commercially available route guidance systems / solutions have successfully addressed network pruning methods for faster and real-time solutions to shortest path algorithms, those solutions are proprietary in nature, and hence the literature available on this subject is very limited. This research has filled the gap in the literature on the methodologies for efficient network pruning.

9.3 Recommendations for Future Work

Expansion of the data available for analysis would improve the significance and robustness of the results of this research. The dataset analyzed is large but does have some limitations. Hence, certain observations made in this study may be applicable only to the trip data collected from 44 volunteers in St. Paul-Minneapolis, Minnesota. While the trends observed from these data may be applicable to other population groups and areas, care must be taken in generalizing the results or using the numerical values in traffic impact and transportation planning studies. More real-world observation data from other metropolitan areas and rural areas would provide more statistically significant results and findings.

Another recommendation for future research is to categorize the time-periods for impacting path choice. For example, trips can be grouped by occurring time, exploring the influences of signals and turns during peak and non-peak periods. Because of vehicle queues at intersections, the turning time is expected to be longer during peak periods than during non-peak periods. However, this assumption needs further verification. In addition, turning movements can be further classified.

REFERENCES

Abdel-Aty, M.A., R. Kitamura, P.P. Jovanis, and K.M. Vaughn (1994) "Understanding Commuters' Attitudes, Uncertainties, and Decision-making and Their Implications of Route Choice" Institute of Transportation Studies Research Report UCD-ITS-RR-94-5, University of California at Davis, Davis CA.

Ahuja, R.K., K. Mehlhorn, J.B. Orlin, and R.E. Tarjan (1990) "Faster algorithms for the Shortest Path Problem". Journal of Association of Computing Machinery, 37:213 – 223.

Antonisse, R., A. Daly, and M. Ben-Akiva (1989) "A Highway Assignment Method Based on Behavioral Models of Car Drivers' Route Choice". Transportation Research Record, 1220:1-11.

Bellman, R.E. (1958) "On A Routing Problem" Quarterly of Applied Mathematics, 16:87 – 90.

Ben-Akiva, M. and D.Bolduc (1996) "Multinomial probit with a logit kernel and a general parametric specification of the covariance structure". Paper presented at the 3rd Invitational Choice Symposium, Columbia University.

Ben-Akiva, M., A.D. Palma and I. Kaysi (1991) "Dynamic network models and driver information systems" Transportation Research Part A, 25A (5):251 – 266.

Bernstein, D. and A. Kornhauser (1996) "An introduction to map matching for personal navigation assistants". Technical report, New Jersey TIDE Center, Princeton University, USA.

Car, A., A.U. Frank (1993) "Hierarchical Street Networks as A Conceptual Model for Efficient Way Finding". In Proceedings of 4th European Conference and Exhibition on Geographical Information Systems (EGIS'93).

Cascetta, E., F. Russo, F.A. Viola and A. Vitetta (2002) "A Model of Route Perception in Urban Road Networks". Transportation Research Part B, 36:577 – 592.

Chen, W.H. and P.P. Jovanis (2003) "Driver En Route Guidance Compliance and Driver Learning with Advanced Traveler Information Systems: Analysis with Travel Simulation Experiment". Transportation Research Record, 1843:81 – 88.

Cherkassky, B.V., A.V. Goldberg and T. Radzik (1993) “Shortest Paths Algorithms: Theory and Experimental Evaluation”. Technical Report 93-1480, Computer Science Department, Stanford University.

Cho, H.J. and C.L. Lan (2009) “Hybrid shortest path algorithm for vehicle navigation” Journal of Supercomputing, 49:234 – 247.

Chou, Y.L., H.E. Romeijn, R.L. Smith (1998) “Approximating shortest paths in large-scale networks with an application to intelligent transportation systems”. INFORMS Journal on Computing, 10 (2): 163 – 179.

Cormen, H.T., C.E. Leiserson, and L.R. Rivest (1990) “Introduction to Algorithms”. MIT Press, Cambridge, MA

Correa, J. (2011) “Wardrop equilibria”. Wiley Encyclopedia of Operations Research and Management Science, 8 Volume Set, Wiley

Daganzo, C. and Y. Sheffi (1977) “On stochastic models of traffic assignment”. Transportation Research 14B: 243 – 255.

Dantzig, G.B. (1960) “On the shortest route through a network”. Management Science, 6:187–90.

Dial, R.B. (1969) “Algorithm 360: Shortest path forest with topological ordering”. Communications of the Association for Computing Machinery, 12:632 – 633.

Dijsktra, E.W. (1959) “A note on two problems in connexion with graphs”. Numerische Mathematik, 1:269 – 271.

Du J. and L. Aultman-Hall (2007) “Increasing the accuracy of trip rate information from passive multi-day GPS travel datasets: Automatic trip end identification issues”. Transportation Research Part A, 41: 220 – 232.

Fielding, G.J., M.E. Brenner and K. Faust (1985) “Typology for bus transit”. Transportation Research Part A, 19(3):269 – 278.

Ford, L.R. (1956) “Network flow theory”. Rand Corporation Report P-923, Santa Monica, CA.

Ford, L.R, and D.R. Fulkerson (1962) “Flows in networks”. Princeton University Press, Princeton, NJ.

Fredman, M.L. and R.E.Tarjan (1987) “Fibonacci heaps and their uses in improved network optimization algorithms”. Journal of the ACM (JACM), 34(3): 596 – 615.

Fu, L. (1996) “Real-time vehicle routing and scheduling in dynamic and stochastic traffic networks”. Ph.D. Dissertation, University of Alberta, Edmonton, Alberta

Fu, L., D. Sun and L.R. Rilett (2006) “Heuristic shortest path algorithms for transportation applications: state of the art”. Computers and Operations Research, 33:3324 – 3343.

Galante, F., F. Mauriello, A. Montella, M. Pernetti, M. Aria and A. D’Ambrosio (2010) “Traffic calming along rural highways crossing small urban communities: Driving simulator experiment”. Accident Analysis and Prevention 2010, 42(6):1585 – 1594.

Gallo, G., and S. Pallottino (1988) “Shortest Path Algorithms”. Annals of Operation Research, 13: 3 – 79.

Glover, F., R. Glover and D. Klingman (1984) “Computational study of an improved shortest path Algorithm”. Networks, 14: 25.

Glover, F., D. Klingman and N. Philps (1985) “A new polynomially bounded shortest path algorithm”. Operation Research, 33: 65.

Hart, E.P., N.J. Nilsson and B. Raphael (1968) “A formal basis for the heuristic determination of minimum cost paths”. IEEE Transaction, System Science and Cybernetics, SSC-4(2):100 – 7.

Jagadeesh, G.R., T. Srikanthan and K.H. Quek (2002) “Heuristic techniques for accelerating hierarchical routing on road networks”. IEEE Transactions on Intelligent Transportation Systems, 3(4):301 – 9.

Jackson, W.B. and J.V. Jucker (1981) “An Empirical Study of Travel Time Variability and Travel Choice Behavior”. Transportation Science, 16(4).

Jan, O., A.J. Horowitz and Z.R. Peng (2000) “Using Global Positioning System Data to Understand Variations in Path Choice”. Transportation Research Record, 2007: 37 – 44.

Janssen, J. and W. Laatz (2010) “Statistische Datenanalyse mit SPSS (7.Aufl.)”. Springer, Heidelberg.

Johnson, D.B. (1982) “A priority queue in which initialization and queue operations take O(log log D) Time”. Math. Systems Theory, 15: 295 – 309.

Johnson, E.L. (1972) “Shortest path on sorting”. In Proceedings of the 25th A.C.M. Annual Conference, pp.510 – 517.

Jung, S. and S. Pramanik (2002) “An efficient path computation model for hierarchically structured topographical road maps”. Knowledge and Data Engineering, IEEE Transactions on, 14 (5): 1029 – 1046.

Karimi, H.A. (1996) “Real-time optimal route computation: a heuristic approach”. ITS Journal, 3(2):111 – 27.

Karimi, H.A., P. Sutovsky and M. Durcik. (2008) “Accuracy and Performance of a Window-Based Heuristic Algorithm”. Map-based Mobile Service: Design, Interaction and Usability, Springer-Verlag Berlin Heidelberg.

Karlaftis, M.G. and P. McCarthy (2002) “Cost Structures of Public Transit Systems: A Panel Data Analysis”. Transportation Research E, 38(1):1 – 18.

Lawler, E.L. (1976) “Combinatorial optimization: Networks and Matroids”. Holt, Rinehart and Winston, New York.

Li, H. (2004) “Investigating Morning Commute Route Choice Behavior Using Global Positioning Systems and Multi-day Travel Data”. Doctoral Dissertation, Georgia Institute of Technology, Atlanta, GA.

Liu, B. (1996) “Intelligent route finding: combining knowledge, cases and an efficient search algorithm”. In Proceedings of 12th Int. Conf. Artificial Intelligence, Cases and an Efficient Search Algorithm, pp. 380 – 384.

Liu, B. (1997) “Route finding by using knowledge about the road network”. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 27(4):436 – 48.

Mardia, K.V., J.T. Kent and J.M. Bibby (1979) “Multivariate Analysis”. Academic Press, Elsevier, Amsterdam, Netherlands.

McFadden, D. (1981) “Econometric Models of Probabilistic Choice”. STRUCTURAL ANALYSIS OF DISCRETE DATA, MIT Press: Cambridge, pp. 198 – 272.

Montella, A., M. Aria, A. D’Ambrosio, F. Galante, F. Mauriello and M. Pernetti (2011) “Simulator evaluation of drivers’ speed, deceleration and lateral position at rural intersections in relation to different perceptual cues”. Accident Analysis and Prevention, 43(6):2072 – 2084.

Moore, E.F. (1957) “The shortest path through a maze”. In Proceedings of the international symposium on theory of switching, pp. 285 – 92.

Myers, C.S. and L.R. Rabiners (1981) “On the use of dynamic time warping for word spotting and connected word recognition”. Bell System Technical Journal, 60: 303 – 325.

Nicholson, J.A.T (1966) “Finding the shortest route between two points in a network”. Computer Journal, 9:275 – 80.

Nilsson, J.N. (1971) “Problem-solving methods in artificial intelligence”. McGraw-Hill, NewYork.

Pallottino, S. (1979) “Adapting the algorithm to D’Esopo-Pape for the determination of all shortest paths: improvements and simplifications”. Technical Report 136, Centre de Recherchesur les Transports, University de Montreal.

Pallottino, S. (1984) “Shortest path methods: Complexity, interrelations and new propositions”. Networks, 14: 257.

Pape, U. (1974) “Implementation and efficiency of Moore algorithms for the shortest route problem”. Math. Progr., 7: 212 – 22.

Pape, U. (1980) “Algorithm 562: Shortest path lengths”. A.C.M. Transactions on Mathematical Software, 6: 450.

Pape, U. (1983) “Remark on algorithm 562”. A.C.M. Transactions on Mathematical Software, 2:260.

Pearl, J. (1984) “Heuristics: intelligent search strategies for computer problem solving”. Addison-Wesley Publishing Company.

Pohl, I. (1971) “Bi-directional search”. Machine Intelligence, 6:127 – 40.

Polydoropoulou, A, M. Ben-Akiva and I. Kaysi (1994) “Influence of Traffic Information on Drivers’ Route Choice Behavior”. Transportation Research Record, 1453:56 – 65.

Quddus, M.A., W.Y. Ochieng and R.B. Noland (2007) “Current map-matching algorithms for transport applications: State-of-the art and future research directions”. Transportation Research Part C: Emerging Technologies, 15 (5): 312 – 328.

Sedgewick, R. and J.S. Vitter (1986) “Shortest path in euclidean graphs”. Algorithmica, 1:31 – 48.

Srinivasan, K.K. and H.S. Mahmassani (2000) “Modeling Inertia and Compliance Mechnisms in Route Choice Behavior Under Real-Time Information”. Transportation Research Record 1725:45 – 53.

Stern, E., E. Holm and M.V. Maarseveen (1993) “Information and Commuters Behaviour: A Comparative Analysis”. Europe on the Move, pp.141 – 155.

Tan, P.N., M. Steinbach and V. Kumar (2006) “Introduction to Data Mining”. Pearson Addison Wesley.

Thériault, M., M.H. Vandersmissen, M. Lee-Gosselin and D. Leroux (1999) “Modelling Commuter Trip Length and Duration within GIS: Application to an O-D Survey”. Journal for Geographic Information and Decision Analysis, 3(1): 41 – 55.

Uchida, T., Y. Iida and M. Nakahara (1994) “Panel survey on drivers’ route choice behavior under travel time information”. In Proceedings of Vehicle Navigation and Information Systems Conference, pp. 383 – 388.

Vovsha, P. (1997) “Cross-Nested Logit Model: An Application to Mode Choice in the Tel-Aviv Metropolitan Area”. Transportation Research Record, 1607: 6 – 15.

Wagner, D. and T. Willhalm (2003) “Geometric speed-up techniques for finding shortest paths in large sparse graphs”. In Proceedings of 11th European Symposium on Algorithms (ESA), Springer, LNCS 2832: 776 – 787.

Williams, J.W.J. (1964) “Algorithm 232: Heapsort”. Communications of the Association for Computing Machinery, 7(6): 347 – 348.

Yiannakoulias, N., W. Bland and L.W. Svenson (2013). “Estimating the effect of turn penalties and traffic congestion on measuring spatial accessibility to primary health care”. Applied Geography, 39:172 – 182.

Zhan, F.B. and C.E. Noon (1998) “Shortest path algorithms: an evaluation using real road networks”. Transportation Science, 32(1):65 – 73.

Zhuang, L., J. Gong, Z. He and F. Xu (2012) “Framework of Experienced Route Planning Based on Taxis’ GPS Data”. 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, Alaska, USA.

BIOGRAPHY

Ms. Xi Zhou graduated from Beijing Jiaotong University, China in 2000 with a Bachelor's degree in Traffic and Transportation Engineering. She had worked as a Research Engineer in China Academy of Railway Sciences before she came to George Mason University in 2005. She earned her Master's degree in Civil and Infrastructure Engineering in 2006. After that, Ms. Zhou joined Daniel Consultants, Inc. as a Traffic Engineer. Since 2010 she has focused on her doctoral research and received her doctoral degree in Civil and Infrastructure Engineering in 2014.