Center for Advanced Multimodal Mobility Solutions and Education

Project ID: 2020 Project 10

USING COMPUTATIONAL BIOLOGY TO MITIGATE PATH OVERLAP IN TRANSIT ASSIGNMENT

Final Report

by

Timothy James Becker, Ph.D. (ORCID ID: https://orcid.org/0000-0001-9784-5664)
Assistant Professor, Department of Computing Sciences
University of Hartford
200 Bloomfield Ave, West Hartford, CT 06117
Phone: 1-860-768-4502; Email: tbecker@hartford.edu

Nicholas Lownes, Ph.D., P.E. (ORCID ID: https://orcid.org/0000-0002-3885-2917)
Associate Professor and Associate Head, Department of Civil and Environmental Engineering
University of Connecticut
Castleman Building Room 301, Storrs, CT 06269
Phone: 1-860-486-2717, Email: nicholas.lownes@uconn.edu

for

Center for Advanced Multimodal Mobility Solutions and Education
(CAMMSE @ UNC Charlotte)
The University of North Carolina at Charlotte
9201 University City Blvd
Charlotte, NC 28223

October 2021

ACKNOWLEDGEMENTS

This project was funded by the Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE @ UNC Charlotte), one of the Tier I University Transportation Centers that were selected in this nationwide competition, by the Office of the Assistant Secretary for Research and Technology (OST-R), U.S. Department of Transportation (US DOT), under the FAST Act and by the University of Connecticut. Additional project resources were donated by the University of Hartford, who provided several networked multi-core computer workstations for prototyping and testing the LCSWT search algorithms and python3 software with the CSTS dataset (outlined in the results section). Special thanks to Dr. Lownes' Ph.D. student Asadul Tanvir whom had interest in path overlap mitigation and assisted with the completion of the literature overview and manually checked the sanity of the resulting paths from the analysis output in the results section.

DISCLAIMER

Table of Contents

List of Figures

List of Tables

EXCUTIVE SUMMARY

In the past thirty years, transit assignment has grown from a niche theoretical exercise to a rigorous field of study with a variety of advanced models and methodologies designed to incorporate the unique decision-making process of transit trips in the planning process. A significant boon to the transit modeling community came with the advent of the General Transit Feed Specification (GTFS) in 2006. Prior to 2006, information on transit networks was difficult to obtain and was inconsistently formatted meaning that software developed for one network may need significant modifications to be applied elsewhere. The motivation of having transit systems searchable on Google Maps proved great enough to convince hundreds of transit systems around the globe to develop their own GTFS feed and make it openly available. This opened the door to a vast amount of research potential, not least the transit assignment community.

GTFS organizes the transit network as a collection of linked spreadsheets that organizes the system as stops, trips and routes. Trips are a sequence of stops visited by a transit vehicle at a particular time of the day. It is these trips, and their associated stop times that are used by FAST-TrIPs in the path generation phase. Because GTFS presents these trips as sequences, it presents an opportunity to leverage work with sequences to potentially improve on the path generation phase of a dynamic transit passenger assignment program (DTPA).

One particular DTPA program named FAST-TrIPs; improved upon prior computational performance by utilizing the static components of the GTFS structure to find feasible transfers or links. The program takes a detailed (or enhanced) version of GTFS and transit demand data and simulates the transit choices people might make by generating many possible paths for every trip a simulated traveler wants to make and then assigns them to paths probabilistically, incorporating the capacity of the transit vehicles along the way. In this path generation phase, one issue identified by researchers working to take DTPA from research to practice was the presence of path overlap. Path overlap occurs when two or more paths in a path choice set contain many similar features with slight variations, violating independence assumptions in any logit-based path choice model and resulting in poor representation of transit path choice behavior. Sometimes nonsensical paths are generated in this result as well with multiple transfers between two routes, alighting and then reboarding to the same route, which waste computational resources and have the potential for selection in the DTPA process.

These paths which can have some or more shared stop sequences can be modeled as mathematical sequences, whereby sequence simultaneity metrics can be used to effectively corrective the path selection and reduce the search space at runtime. In computational molecular biology efficient comparison of sequences has become the foundation of genomics and transcriptomics, with decades of research and algorithm development having been already completed and applicable to this path overlap problem (Gusfield 1997). Specifically, we made use of a transit-specific variation on the Longest Common Subsequence (LCS) algorithm to include spatio-temporal components while providing an optimally efficient run-time. In our work we separate computation structures that can be calculated independently from any demand and those that must be computed with specific origin-destination spatio-temporal points in mind. We

first pre-process the network to find all the viable links or potential transfer points given passenger waiting, walking and access/egress limitations. Next the potential links are searched for viable paths which are then scored for cost using a per-mode offset coefficient transfer penalty. All of these variable paths are compared to each other for overlap and assigned a numeric value for similarity against all other paths. Taken together the total trip cost given the penalty model along with the similarity measure can be used to select representative paths of a potential large group which have the best and most unique trips.

Using the CTTransit Hartford area GTFS file and the CSTS demand dataset along with the needed walking and drive access files as used in Fast-TrIPs we were able to employ our LCS based viable path reduction protocol using a parameter k. This parameter is applied to each transfer level and provides up to k paths out of the potential n, which include the lowest cost path and most unique path and k-2 paths that lie somewhere between those extremes. Using this strategy we were able to contain an exponential number of paths with respect to number of transfers into a constant number of paths. Once the k-dissimilar paths are selected using our framework, a system like Fast-TrIPs could be altered to use the reduced path set and not need to full search the full set of viable trips. We provide the preliminary efforts written in python/cython in open-source form for the community to share, contribute to and learn from at: https://github.com/timothyjamesbecker/transit_alignment.

1 Introduction

1.1 Problem Statement

The longest common sub sequence (LCS) is defined as those symbols that appear in the same order between two input sequences of possibly varying sizes. For our work, we formulate trips from a transit network as a sequence, while stops on the trip can be thought of as the symbols. Trips that have many of the same stops appearing with a similar ordering will then have a larger LCS than those trips with larger amounts of unique stops. By proceeding to calculate the LCS for all possible trip pairs we arrive at a natural clustering where alternate routes and express versions of trips become easily identifiable by the LCS signatures they contain. The LCS can drive more than just pairwise comparisons as well, which we formulate as a multiple sequence alignment problem. When multiple sequences undergo a distance calculation such as the LCS, edit symbols are added to extend a sequence to another causing the LCS components to move to the same column position. Once these aligned trips are in this symbol extended form, we can use further tools to look at the proportions of dominant LCS signatures, providing a computationally efficient macro scale view of the entire route/sub-route/express network structure. Because the LCS can be quickly computed on all trips due to the limited length of the strings filtered by domain knowledge, we have explored polynomial time solutions via dynamic programming to generate similar sub sequences on only those trips that have at least one match using a linear-time hash-map solution. This means that for large transportation networks, the applications of the linear-time filter will result in a polynomial reduction with respect to the number of trips that will receive the more expensive pairwise LCS calculation, since without a match the LCS is 0. The same techniques do not have such efficient run times when generalized to arbitrary repeated symbol directed graphs.

For a transportation network, sequences are only a fraction of the possible paths that could be utilized to find a valid user trip that has minimal transfers and the desired time cost. But since the network is fixed and not a dynamic run-time consideration to a transit assignment simulation framework, we argue that a priori network trip clustering and alignment could not only improve the realism of simulated transit assignment tools, but could also attach the perceptual element of similarity into the path-weight calculations themselves. Every origin/destination pair within a time range has a fixed constant number of trips, each of which has a preexisting amount of similarity to all other trips in that constant set. By including similarity and correcting for it in path-weight calculations, we can prune a large graph search space to greatly reduced size of feasible trips that contain edge-bundling/similarity information

1.2 Objectives

The goals of the proposed research are broadly to apply the methods of sequence alignment and matching from computational biology, focusing on LCS, to the DTPA problem in three contexts:

(1) A priori characterization of the network

(2) Elimination of nonsensical paths during path generation phase

(3) Mitigation of path overlap through development of metrics to be integrated with the path choice model.

1.3 Expected Contributions

(1) Finalize an ideal scoring mechanism and metrics (should have more than one!)

(2) Explore the effects of LCS with symbol-time matching (currently we have only employed symbol matching). A symbol-time matching LCS would offer an additive weight for when time matches and the symbol matches.

(3) Develop and explore route/sub-route/express tailored symbol-time LCS signatures

(4) Develop and explore a transit specific route/sub-route/express tailored symbol-time multiple trip alignment algorithm using (3)

(5) Explore and develop pathweight model for incorporating trip similarity

(6) Develop and implement a proof of concept system that can perform the LCS based alignment and path-weight model on a large network such as SF.

1.4 Report Overview

The rest of the report is structured with a short literature review of transit assignment and path overlap. That is followed with a technical method section that sets to formal define the similarity metrics used for the novel components to the work. Traditional penalty modeling using a per mode structure and the algorithm application to our CT GTFS and CSTS demand files are included in the results section.

2 Literature Review

Transit assignment approaches are broadly divided into two types: frequency-based and schedule-based transit assignment. The frequency-based approach assumes each transit line operates on a constant frequency without any reliable schedule. Frequency-based models are commonly used in static transit assignments. It is effective in describing bus networks operating in congested roads with varied travel times (Oliker & Bekhor, 2018). In contrast, schedule-based approach uses detailed departure or arrival times of transit vehicles for assignment purposes. This makes schedule-based models more suitable for dynamic transit assignment in a sense that these models must incorporate the time component explicitly (Nuzzolo & Crisalli, 2004). Schedule-based methods require faster computer and larger database unlike the frequency-based approaches as they can model the within-day dynamics in passengers' route choices (Liu, Bunker, & Ferreira, 2010). Tong and Richardson (1984) were the first in adapting Dijkstra's shortest path algorithm to find the minimum cost paths in public transit systems running on fixed schedules. Based on this work, Tong and Wong (1999) generated the time-dependent optimal path using a specially developed branch-and-bound algorithm. They then formulated a dynamic stochastic transit assignment model using a schedule-based approach with the given time-dependent O-D matrix.

Spiess and Florian (1989) described an optimal strategy for frequency-based transit networks where passengers are allowed to reach their destination at minimum expected cost. The model assumes that the waiting time for a passenger before boarding the first arriving vehicle depends on the combined frequencies of the transit services considered by the traveler. Nguyen and Pallattino (1988) developed a graph-theoretic framework (hyperpath) to deal with the common lines problem in transit network route choice. The "common lines" here refers to the decision problem faced by the passengers at a transit stop shared by several competitive line, whether to board an arriving bus or to wait for a faster one to minimize the total travel time. Nguyen, Pallotino, and Gendreau (1998) later introduced the concept of efficient hyperarc and investigated the logit splitting of the demand on efficient hyperpaths. In a transit system where frequencies are low and timetables are reasonably reliable, the departure time and route choice can be equally important to the users. In a later study (Nguyen, Pallotino, & Malucelli, 2001), a transit assignment model was developed incorporating both departure time and route choice simultaneously. Schmöcker, Bell, and Kurauchi (2008) presented a first dynamic frequency-based transit assignment model for overcrowded high-frequency transit networks. A "fail-to-board" probability is introduced in this study to capture the passengers' inability to board the first arriving vehicle due to overcrowding. The model also assumes that on-board passengers get the priority, waiting passengers mingle on the platform, and Markov-based process is used for loading. Schmöcker et al. (2011) later extended this model by adapting a "fail-to-sit" probability with travel costs based on the likelihood of travelling seated or standing.

Nuzzolo, Russo, and Crisalli (2001) presented a schedule-based path choice model for high-frequency transit networks which considers both day-to-day and within-day dynamic of passenger path choices. Hamdouch and Lawphongpanich (2008) took into account both transit schedules and vehicle capacity constraints in their schedule-based UE transit assignment model. One of the distinctive properties of schedule-based networks is that the adjacent stops vary over time, and the cost of reaching an adjacent stop is a function of service available at a given time. This time-varying property of transit stops makes schedule-based approaches more complicated than the static road networks (Khani et al., 2015). Noh, Hickman, and Khani (2012) introduced a link-based time-expanded (LBTE) network for transit schedules where each link represents a sched-

uled vehicle trip with time information between two consecutive stops. A logit-based hyperpath model integrated with the LBTE network was also proposed in this study for stochastic transit assignment. Although this LBTE model reduces the effort to build network on transit schedules because of the expansion of network with scheduled links, the size of the network increases significantly requiring significant computational effort to solve path assignment problems. To improve the computational efficiency, a new network representation, the Trip-based Network has been introduced by Khani et al. (2015). This trip-based network uses Google's General Transit Feed Specification (GTFS) data provided by the transit agencies to represent the transit network (GTFS Static Overview, 2019). Moreover, it uses transit vehicle trips as network edges and also takes into consideration the transfer stop hierarchy in transit networks. Using this hierarchical trip-based transit network format, a set of path algorithms including algorithms for the shortest path, a logit-based hyperpath, and a transit A* were developed. Khani et al. (2013) extended these models by incorporating boarding priority and strict capacity constraints to formulate the schedule-based dynamic transit assignment model named FAST-TrIPs. Khani, Bustillos, Noh, Chiu, and Hickman (2014) then developed a joint model consists of a dynamic traffic assignment, a schedule-based transit assignment, and a park-and-ride choice model for comprehensive modeling of transit and intermodal tours in a dynamic multimodal network.

Path overlap is a common issue in multimodal route choice modeling. It affects the path generation and path probability estimation in schedule-based transit assignment models such as Fast-Trips (Zorn & Sall, 2017). When paths in a choice set overlap with one another, the logit based hyperpath model is affected by the Independence of Irrelevant Alternatives (IIA) property. This IIA property can be stated as if all else being equal, a person's choice between two alternatives is unaffected by other available choices (Cheng & Long, 2007). Transit vehicle trips running on the same route and having similar characteristics in schedule-based network may be considered as independent paths because of this property. Several modifications to the logit model such as C-logit (Cascetta et al., 1996), link-nested logit (Vovsha & Bekhor, 1998) were proposed for overcoming this route overlap issue. Ben-Akiva and Bierlaire (Ben-Akiva & Bierlaire, 1999) proposed a path-size logit model incorporating a path size factor to reduce alternative's disutility in case of overlap.

Ramming (2002) suggested path size logit gives better empirical fit than C-logit, and overlap variable based on travel time yields better results than the overlap variable based on distance for road networks. For multimodal networks, Hoogendson-Lanser, Nes, and Bovy (2005) analysed three definitions of overlap using path size estimators- number of legs, time, and distance. The results found that path size defined on number of legs producing better results than the other definitions, and the weighting parameter for path size variable should be set equal to 1.

Because we formulate the path overlap as a sequence similarity problem, we will provide some short background in this field and then expand on our spatio-temporal variation in the method section. Pairwise sequence comparison can be viewed as a more restricted endeavor from acyclic graphs which in turn are restrictions on directed graphs. If individual trips restricted to the acyclic directed graph form (also known as a tree), then they are a path in that tree that can be represented as a sequence which in turn can be compared to any other sequence. Because all trips contain timestamps and ordering, all trips in the network can be represented in this form due to the linearity of time. One of these pairwise sequence comparison methods called longest common subsequence (LCS) has been adopted to measure and assess the level of inter-route overlap. Historically, pairwise sequence comparison methods such as edit distance (ED) and LCS have been used to study everything from language, biochemistry, music theory, human behavior to travel

trajectories (Abbott, 1995; Hirschberg, 1975; Mongeau & Sankoff, 1990; Smith & Waterman, 1981; Thompson, 1994; Wilson, 2008). In each of these domains the order of the symbol and its match plays an important role in finding larger scale patterns and trends.

3 Methodology

Transportation trips can take many forms and have many modes of travel. We focus on the readily available public transportation in Connecticut which is bus, but utilize methods and terms that are broadly applicable and functionally without loss of generality. Access and egress to this transportation network is made by either driving to a park and ride which we define as the drive mode of access/egress, walking, waiting or finally the most desirable mode in this framework which is when in the bus making progress which we call vehicle. We use the now ubiquitous GTFS file format for our open source implementation which makes use of a few key concepts and terms that relate to the spatio-temporal properties that must be navigated in order to perform any trip search. First and , trip searching is made from the perspective of a particular origin or destination date and time and from this starting point a search can be made in terms of optimization whereas the rider can locate and order trips based on the time it will take to get to the destination which can involve transfers along the way. We will formally define in a mathematical framework the concept of trip search as a tree search problem and will make use of sequences to the address issue that naturally arise when considering the reduction of the large numbers of viable trips.

We define the set of all unique stops in a transportation network $\mathbb{S}$ (which are contained in the GTFS stops.txt file). A trip we will term a sequence whose alphabet is $\mathbb{S}$. Sequences in the context rarely have repeating symbols but can in fact repeat for looping style trips example of loop route. If we consider the set of all trips as $\mathbb{U}$ then searching for a viable trip will involve the use of spatio-temporal selection on $\mathbb{U}$, which is to say we need to find a time range around the desired origin or destination point and well as a spatial range around the walk or drive access points. Both of these ranges act as a time and space buffer that will produce a very small subset of $\mathbb{V} \subset \mathbb{U}$ which is the set of viable trips that are within the active spatio-temporal search bands.

We proceed with the details of our Random Tree Search (RTS) algorithm as follows:

(1) find all feasible origin stops that are within the transportation analysis zone (TAZ) using first walk distances and then driving distances

(2) search all trips that will pass through any of the feasible stops within the time buffer

(3) find feasible origin trips that contain a stop that is within the walking distance buffer or a driving distance buffer, if (3) doesn't not find a suitable stop (which would make this search have a direct trip), then a recursive solution involves (optional 4) finding all linking trips.

In our implementation, we speed this basic search up in several ways: (1) preprocess the network so that all trip to trip transfer links are cached in a dictionary and will have $O(1)$ search time on average, (2) use a constant number of transfers since these are undesirable, (3) preprocess the origin and destination stop and trips which we term as candidate trips: $\mathbb{C} \subset \mathbb{V}$, (4) heading limiting which calculates for each sequence X in $\mathbb{C}$: the vector in mph from the access stop to the closest stop point and then orders the search in descending order while filters trips that have negative walking speed headings, (5) depth limiting of the search so that when direct trips are encountered the search can terminate early, (6) random importance sampling where as the transfer number increases, the branches are down-sampled more and more aggressively using heading limits from (4). Taken all together these simplistic measures lead to efficient and low cost trip searches on very large networks. We call the set of all returned sequences (trips) from the RST algorithm: $\mathbb{P}$

3.1 K-dissimilar Paths

The paths $\mathbb{P}$ that are returned from RST are also in fact spatio-temporal sequences where each point along the trip the traveler is at some stop $s \in \mathbb{S}$ using one of the modes: vehicle, waiting, walking, drive. At each spatial stop point in this framework, there is the time value or the number of elapsed seconds in the day. Although the set $\mathbb{P}$ does in fact represent all viable options for a trip, it often contains large numbers of poorly performing trips as well as large numbers of near-duplicate or very similar trips. A more reasonable alternative is desirable for analysis and network simulation that favors low-cost trips as well as trips that have reduced similarity to the others. Using these desired components, we detail a information-criterion based reduction of the size of $|\mathbb{P}|$ to a constant value we call k. This value can be set by the user and will be calculated for each transfer level separately since a two-transfer trip by nature will have very different stops than a direct express. Because similarity in this context is spatial an temporal we explore two dissimilarity metrics that can be used to objectively select the k sequences in $\mathbb{P}$ that are low cost and also very dissimilar: (1) Jaccard set dissimilarity (J) and (2) longest common subsequence with transfer (LCSWT) dissimilarity.

3.2 Jaccard Dissimilarity Metric

A very simplistic approach is to simply count the number of shared stops between two trips and divide by the total number and then take the complement: $\mathbb{J}(X,Y) = 1 - \frac{|\mathbb{X} \cap \mathbb{Y}|}{|\mathbb{X} \cup \mathbb{Y}|} : X, Y \in \mathbb{P}, \mathbb{X}$ is the set of stops in sequence X, and $\mathbb{Y}$ is the set of stops in sequence Y.

The main issue here is that stops have to be identical in this formulation and the spatial proximity is not considered meaning that with the $\mathbb{J}$ metric two stops could be five feet away from each other and not be counted as similar. A second more drastic issue here is that the order of the stops is not considered since set metrics have no ordering. A novel and more accurate metric follows that incorporates both ordering and spatial matching components, which we call LCSWT. LCWST is a spatio-temporal variant of the longest common subsequence algorithm which seeks to compare how close two sequences are to each other.

3.3 Sequence Comparison Metrics

Sequence similarity measures can be used to automatically identify and score unique paths among graphs and tree structures. Historically, pairwise sequence comparison methods such as edit distance (ED) and longest common subsequence have been useful in powering numerous query tools and analysis systems across a wide range of applications in domain like language, biochemistry, music theory, behavior and more recently in travel trajectories (Hirschberg 1975)(Smith & Waterman 1981)(Thompson 1994)(Mongeau 1990)(Abott 1995) (Wilson 2008). We first start with a basic LCS and then detail and expanded form design specifically for trip paths that involve transfer options.

3.3.1 LCS

LCS (denoted below as **L**) traditionally measures the number of common subsequence elements and so requires some transformations to become a metric. First we outline a traditional recursive definition and then we will expand this simple version to one that incorporates space and time:

$$\forall X, Y \in \mathbb{U}: \quad \mathbf{L}(X\_i, Y\_j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ \mathbf{L}(X_{i-1}, Y_{j-1}) + 1 & \text{if } i, j > 0 \text{ and } X_i = Y_j \\ \mathbf{max}\{\ \mathbf{L}(X_i, Y_{j-1}),\ \mathbf{L}(X_{i-1}, Y_j)\ \} & \text{if } i, j > 0 \text{ and } X_i \neq Y_j \end{cases}$$

3.3.2 LCSWT

This classic formulation doesn't quite work for our purposes since an exact stop and time match would be implied by: $X_i = Y_j$. Instead we simply apply time selection and do not use $\mathbb{U}$ or $\mathbb{V}$ but $\mathbb{P}$. Instead of the exact stop, we will use its proximity or transfer potential via a stop to stop distance matrix:

$$\sigma(a, b) = \sqrt{(a_{lat} - b_{lat})^2 + (a_{long} - b_{long})^2} : \forall a, b \in \mathbb{S}$$

We then use a diffusion function to zero the matching space component so only those stops that are within the walking distance buffer denoted as d will get partial matching credit [0.0,1.0] instead of the 1 for an exact match. In this formulation an exact match still gets a 1-stop credit, but stops that are close will have approximate value contributed proportional to its . Using this variation we then have our LCSWT (denoted as $\mathbb{L}$):

$$\forall X, Y \in \mathbb{P}: \quad \mathbb{L}(X\_i, Y\_j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ \mathbb{L}(X_{i-1}, Y_{j-1}) - \sigma(X_{i-1}, Y_{j_1}) d^{-1} + 1 & \text{if } i, j > 0 \text{ and } \sigma(X_i, Y_j) < d \\ \mathbf{max}\{\ \mathbb{L}(X_i, Y_{j-1}),\ \mathbb{L}(X_{i-1}, Y_j)\ \} & \text{if } i, j > 0 \text{ and } \sigma(X_i, Y_j) \geq d \end{cases}$$

3.3.3 LCSWT Dissimiliarity Metric

We now transform the $\mathbb{L}(X, Y)$ value into a symmetric distance using the harmonic mean over both numerators which creates symmetry between small and large sequence comparisons:

First we have the X to Y version: $\frac{\mathbb{L}(X,Y)}{|X|}$ and then the Y to X version: $\frac{\mathbb{L}(X,Y)}{|Y|}$

recalling that the harmonic mean of any two metrics: m_a, m_b can be obtained by: $\frac{2m_a m_b}{m_a + m_b}$,

we then derive our LCSWT dissimilarity metric:

$$\delta(X, Y) = 1 - 2\frac{\left(\frac{\mathbb{L}(X,Y)}{|X|}\right)\left(\frac{\mathbb{L}(X,Y)}{|Y|}\right)}{\left(\frac{\mathbb{L}(X,Y)}{|X|}\right) + \left(\frac{\mathbb{L}(X,Y)}{|Y|}\right)} = 1 - \frac{2\left(\frac{\mathbb{L}(X,Y)^2}{|X||Y|}\right)}{\left(\frac{\mathbb{L}(X,Y)(|X|+|Y|)}{|X||Y|}\right)} =$$

$$1 - 2\left(\frac{\mathbb{L}(X,Y)\mathbb{L}(X,Y)}{|X||Y|}\right)\left(\frac{|X||Y|}{\mathbb{L}(X,Y)(|X| + |Y|)}\right) = 1 - 2\left(\frac{\mathbb{L}(X,Y)}{|X| + |Y|}\right)$$

3.4 Normalized Sum of Pairs Scoring

This provides the needed criterion for comparing one sequence to another and when we wish to answer objectively which sequence $Q \in \mathbb{P}$ is the most dissimilar to the others we can make use of one to many group comparison operations such as the well establish sum of pairs score (SP). We will derive a normalized SP score here using the $\delta(X, Y)$ metric but it should be noted that the $\mathbb{J}(X, Y)$ will work just as well and is available in our implementation.

Here we will define the normalized SP score:

$$\forall X \in \mathbb{P}: \quad \Delta(X) = \sum_{Y \in \mathbb{P} \setminus \{X\}} \left(1 - 2\frac{\mathbb{L}(X, Y)}{|X| + |Y|}\right)\left(\frac{1}{|\mathbb{P}|}\right) = \quad \frac{1}{|\mathbb{P}|} \sum_{Y \in \mathbb{P} \setminus \{X\}} \delta(X, Y)$$

3.5 Trip Cost and Penalties

The second major aspect of analysis of a set of return trips $\mathbb{P}$ from RST is the cost in terms of time and in our implementation under a penalty model. Penalties are used here as a way to model behavioral elements of passengers that prefer to have direct routes and to not walk excessively. In our work we incorporate a coefficient and fixed offset penalty model for each mode of travel. This allows the user to search in a restricted movement situation or in a case of a highly mobile but time-crunched passenger that is willing to walk several miles for a transfer.

We will use the notation of $\alpha(X)$ to denote the total cost in seconds for sequence $X \in \mathbb{P}$. Here we use penalty functions in seconds as well to allow easy incorporation of both the actual measured time and that time we are modeling that may be felt by the passenger (time with penalties).

let t denote the time in seconds for a sequence segment, m_w denote the coefficient for a mode and m_o denote the fixed cost of a mode:

$$\forall X \in \mathbb{P}: \quad \alpha(X) = \sum_{i=1}^{|X|} X_{i,t}\ X_{i,m_w} + X_{i,m_o}$$

To normalize this so that is is compatible with our $\Delta(X)$ metric we use the minimum and maximum cost sequences in $\mathbb{P}$ to denote a normalized version of α we will call β:

$$\forall X \in \mathbb{P}: \quad \beta(X) = \left(\frac{\alpha(X) - \min\limits_{Y \in \mathbb{P}} \{\alpha(Y)\}}{\max\limits_{Y \in \mathbb{P}} \{\alpha(Y)\} - \min\limits_{Y \in \mathbb{P}} \{\alpha(Y)\}}\right)$$

Notice here in the normalized cost form, the best trip will have the value of 0.0 which is also the numeric orientation of our Δ value where 0.0 will denote the sequence that is most dissimilar to all others in $\mathbb{P}$.

3.6 Cost-Dissimilarity Optimization

We can combine our two metrics in many ways but in order to achieve a meaning analysis we must go back to the importance of returning the lowest cost path from $\mathbb{P}$ since this is the most desired result and also or returning the sequences that are very different. This means that a simple additive or linear hybrid optimization over cost and dissimilarity will not work since it could return sequences that have an average similarity and also average meaning. This would lead to selected paths being good at both when we are actually looking for some of the best cost sequences and some of the most dissimilar sequences in our space reduction of $\mathbb{P}$. A natural conclusion then

is to use an set of incremental iterative weights that feature the importance of the lowest cost first and makes progress up to the kth sequence is selected to then select the sequence that is most dissimilar to all other in $\mathbb{P}$

We use a linear monotonically decreasing mixing function that starts at 1.0 and goes to 0.0:

$\omega(i) = 1 - \left(\frac{i-1}{k-1}\right) :$ for $i = 1$ to $k+1$

For example, when $k = 5$: $\omega = \{1, \frac{3}{4}, \frac{1}{2}, \frac{1}{4}, 0\}$

We then proceed to use ω to mix the cost with the dissimilarity (using the complement of ω_i) as we pick the k sequences:

$\mathbb{K} \leftarrow \phi$

for $i = 1$ to k :

$$\mathbb{K} \leftarrow \mathbb{K} \cup \left\{ \min_{X \in \mathbb{P} \setminus \mathbb{K}} \{\omega_i \beta(X) + (1 - \omega_i)\Delta(X)\} \right\}$$

4 Results

4.1 CTTransit GTFS Dataset for Hartford

For our work, we used the 2016 GTFS network dataset taken from:https://espace.library.uq.edu.au/view/UQ:732753/GTFS_CTTransit_Hartford_2016.zip, so that it would closely correspond with our CSTS datset we used for looking at demand. It comprises bus routes across CT and includes the incoming work trips that we were looking to study and apply our LCSWT based k-dissimilarity algorithm upon detail in the method section above.

4.2 CSTS Demand Dataset for Hartford

Because DTPA also requires to have demand information in a specific form, we made use of actually collected data that was part of our prior work: https://portal.ct.gov/-/media/DOT/documents/dTDAQM/FinalReportCTStatewideHouseholdTransportationStudyNoAppendicespdf.pdf. We use unidentified person trips for those people in the survey that were headed into work to select appropriately complex trips to compare and contrast against our LCSWT base k-dissimilar path selection. We focused on those trips that had at least one transfer and in the process were able to obtain person trips with very long and transfer-filled commutes. We were interested to see if users picked very low cost trips that were closure to the best or if they were picking some other higher-cost trip that had components that were unique when compared to the best trip possible. We selected all in bound weekday trips from the CSTS dataset that were going to work in Hartford that had reported needing one or more transfers. This resulted in 39 unique persons and 56 trips in total.

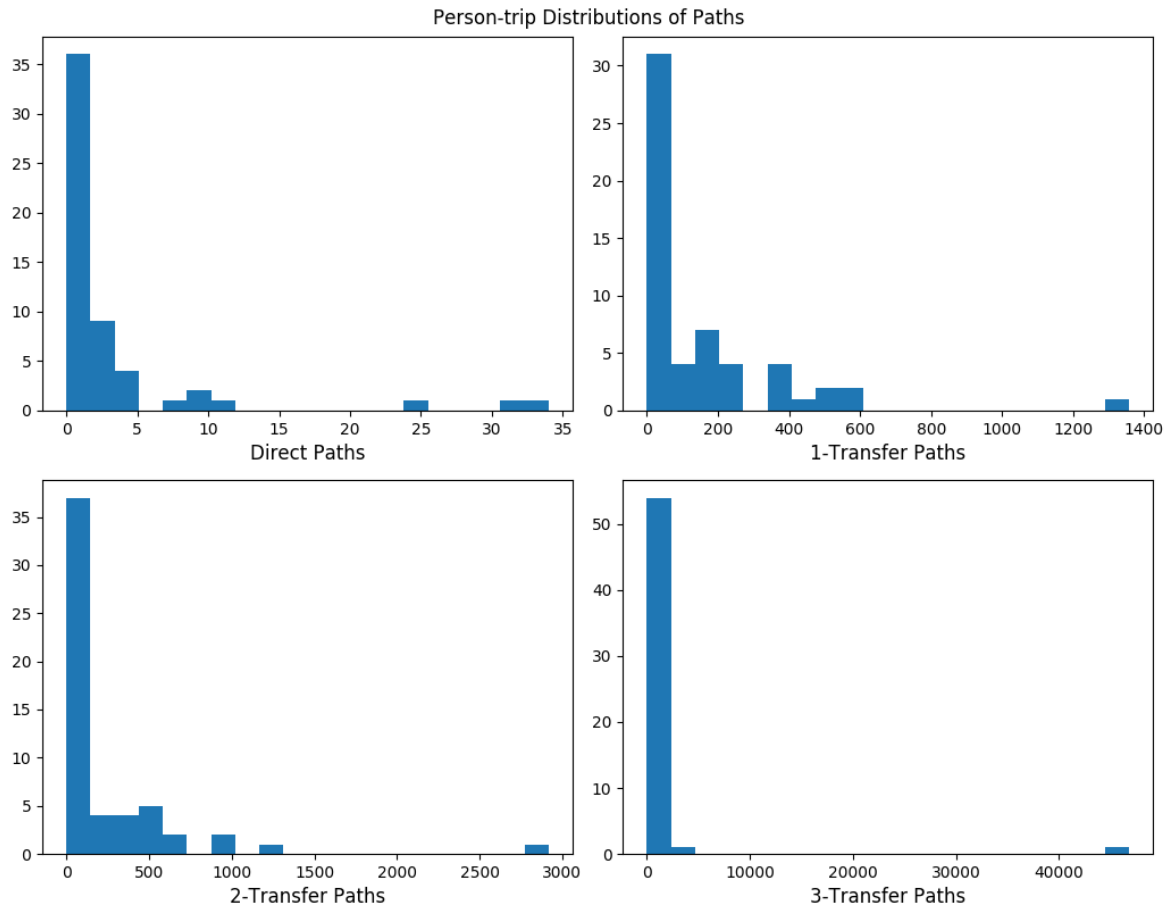

Figure 1 The number of potential paths generated by the RST phase of our algorithm when sampling is at its full scale leaves a very large pool of possible paths. Some 3-Transfer searches yielded over 40000 potential ways to travel.

To explore the effectiveness in the selected k trips out for the total potential paths (that Fast-TriPs would normally sample from) we first plotted the cost of each trip against the similarity measure. The k-dissimilar algorithm takes the lowest cost as the first selected value always and then proceeds to select paths in this plot that are in the lower left corner. The set of the k-paths tend to encompass the most relevant and also some of the most unique among what is an initial set of many self-similar clusters. To further show the unique paths have been selected, we also plotted the multi-dimensional scaled (MDS) version of the pair-wise distance matrix of the LCTWT SP scores for each path. Here, points appear very close when then contain the same stops and appear very far away when the trip represents a truly different path from the origin to destination.

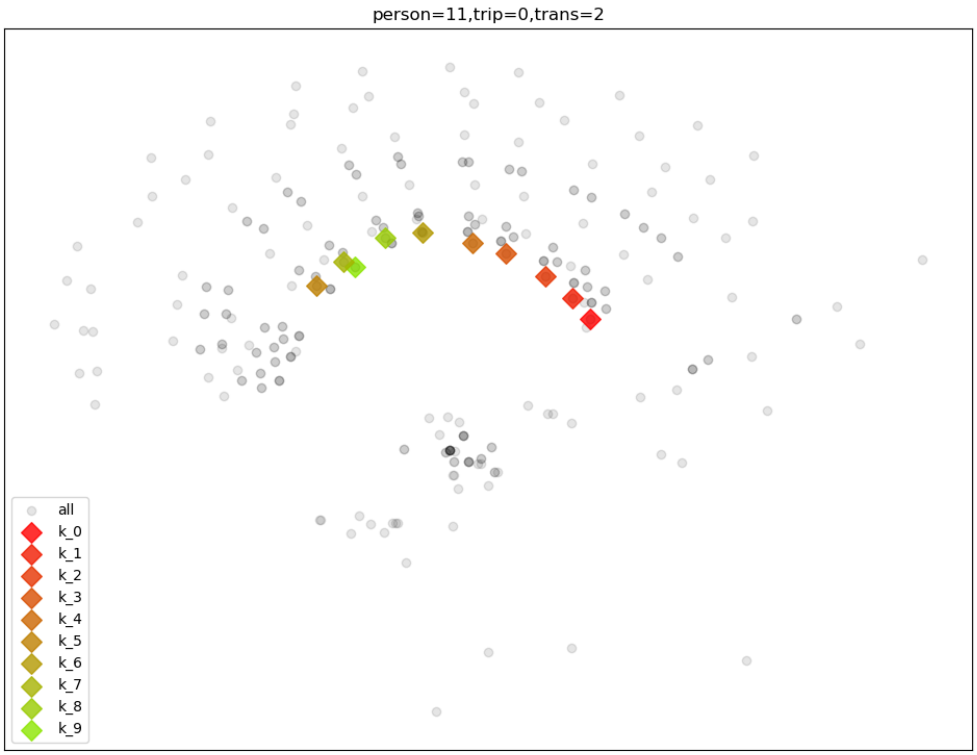

Figure 2 Person 11 Trip 1 requires 2 transfers to complete but the search space is very large. The MDS of the LCSWT distance matrix shows clear clusters of similar options in a fan like position at the top. The k-paths that are selected closely align to 10 of the 15 naturally occurring clusters.

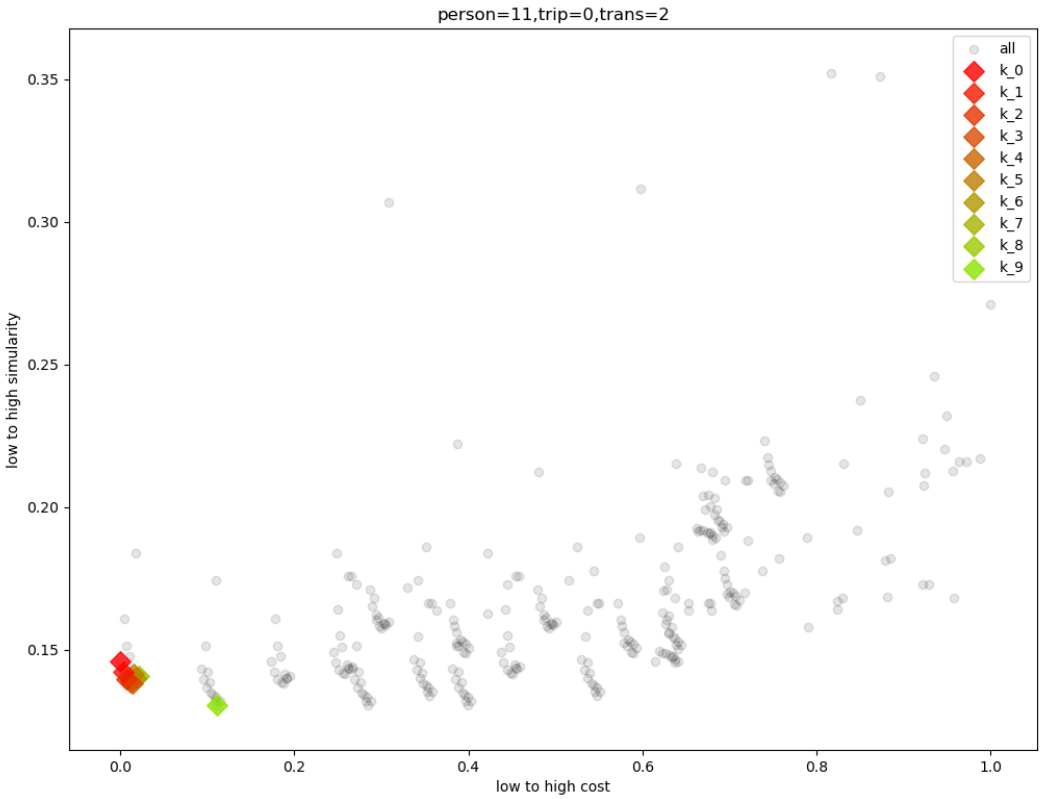

Figure 3 The same Person 11 Trip 1 with one transfer is depicted here with cost on the X-axis versus the SP-score. Trips that are selected by the algorithm are the left and bottom most points since these are the dual optimal positions. The distribution or positioning from the lowest cost to the least similar can be further manipulated by the interpolation function given to the optimization function where a linear function was used here. There are no lower-cost trips in the total search space that in our k-path set, while there are no more unique trips taken either which would show as lower y-axis values.

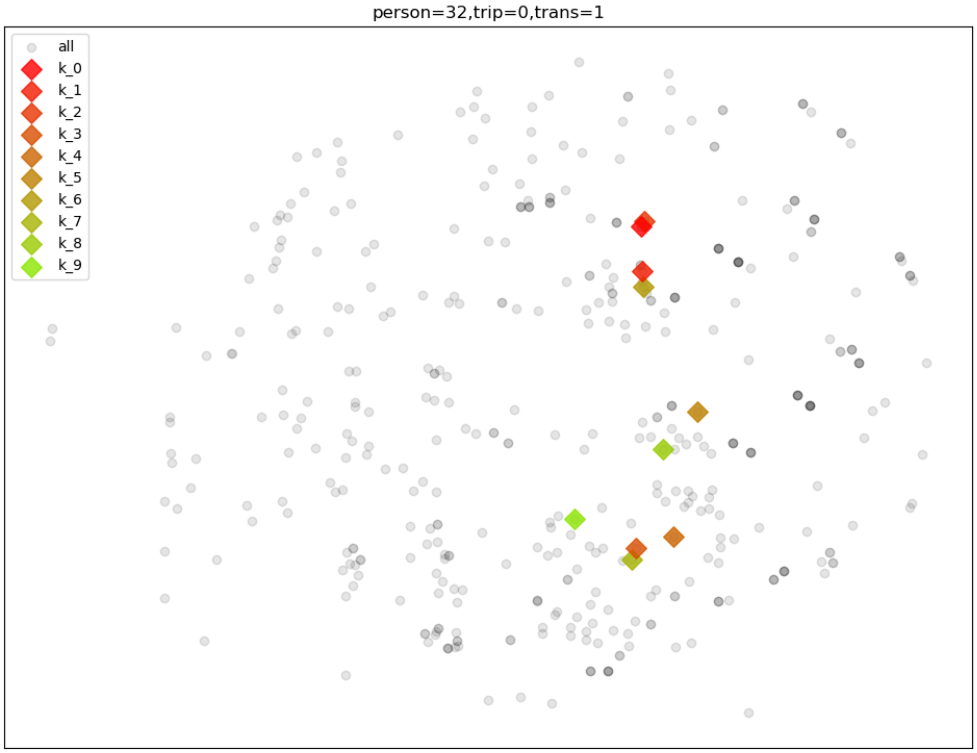

Figure 4 Person 32 Trip 1 with a single transfer has a very large search space. When looking at the MDS here, many natural clusters emerge most of which are very high cost and would lead to a very undesirable trip for the passenger. The k-path trips seems for form a small ellipse at the center of the MDS that defines the most information rich section in possible paths.

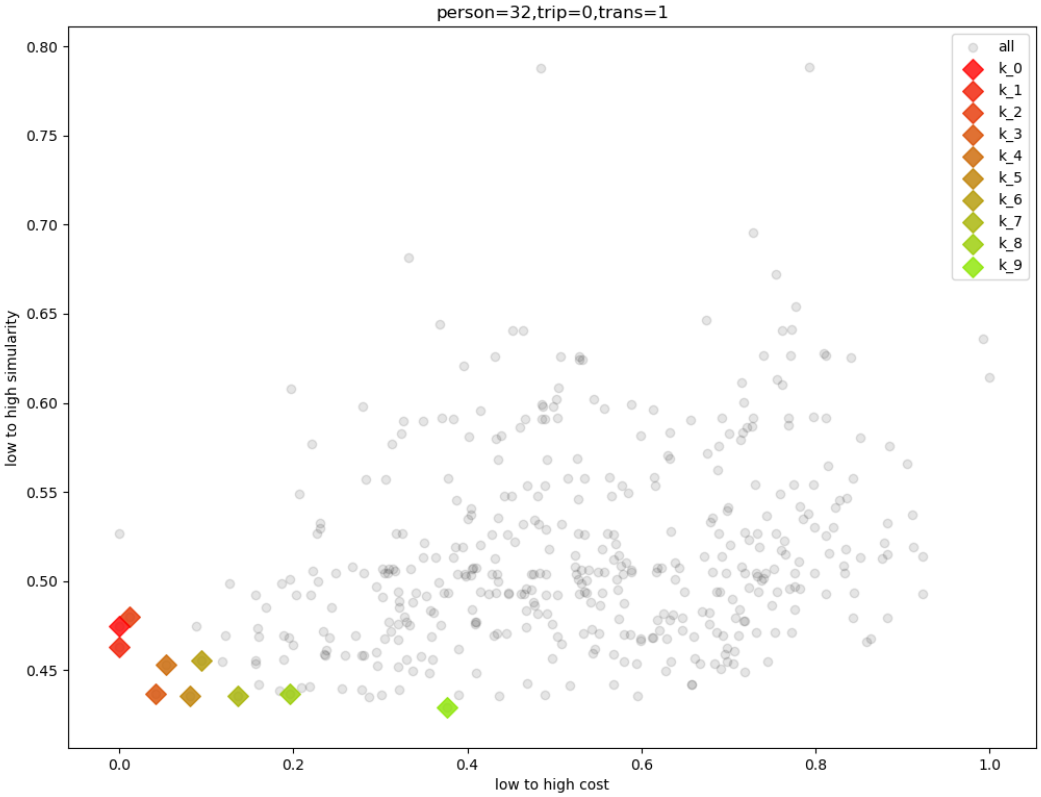

Figure 5 When the same Person 32 Trip 1 is examined from cost to similarity we can see the same clear lower left portion of paths are selected that are both very unique and low cost.

5 Conclusion

We have developed a novel way to determine path overlap for transit assignment and transit system analysis in our project. Our LCSWT method provides a secondary way to score potential paths and when employed in conjunction with the standard penalty function for cost, can be employed to complete an information-rich approximation on large system path potentials. We have successfully adapted the LCS algorithm to be useful in transportation assignment by extending it to the spatio-temporal domain in the form of stop sequences or trips from GTFS. We are now able to pre-compute all trip-to-trip LCSWT measures to automatically find related routes, subroutes and express relationships. We are able to find the lowest cost and most unique k trips out of a large number and every trip that lies between those extremes using interpolation. We tested our performance on the Hartford CT network which is sufficiently large to demonstrate our proof of concept and have made our software implementation open-source and availible to the community as part of our project fulfillment.

The practical implications for this work are simple and utilitarian. The k paths can reduce link complexity of systems that have 40,000 or so potential paths to a smaller more manageable number like 1000 which would result in faster simulation. For simulations systems like Fast-TriP which uses random sampling, this could greatly increase speed on the largest networks that exist. When large potentials are randomly sampled there is a higher chance of not sampling a low cost. By selecting the k paths ahead of a sampling procedure, that sample procedure is guaranteed to already have some of the best choices to choose from and so the randomization has some bounds that would align to human behavior.

Our method is a natural extension to the SP problems where dynamic programming can be used to keep local optimum and be built upon to encompass a large system. Instead of the single highest scoring path our k-paths could be used instead thereby retaining k-1 potential links between each set of points, instead on the normal single value (or all values that tie for the lowest cost). A natural future direction is a full implementation of this algorithm integrated into the standard Dijkstra, so that any possible demand could be precomputed and cached in a database like structure for fast searches that would yield many unique paths between any points.

References

Abbott, A. (1995). Sequence analysis: new methods for old ideas. Annual Review of Sociology, 21(1), 93-113.

Ben-Akiva, M., & Bierlaire, M. (1999). Discrete choice methods and their applications to short term travel decisions. In Handbook of Transportation Science (pp. 5-33). Springer, Boston, MA.

Bonizzoni, P., & Della Vedova, G. (2001). The complexity of multiple sequence alignment with SP-score that is a metric. Theoretical Computer Science, 259(1-2), 63-79.

Cascetta, E., Nuzzolo, A., Russo, F., & Vitetta, A. (1996). A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks. In Transportation and Traffic Theory. Proceedings of The 13th International Symposium On Transportation And Traffic Theory, Lyon, France, 24-26 July 1996.

Cheng,S., & Long, J. S. (2007). Testing for IIA in the multinomial logit model. Sociological Methods & Research, 35(4), 583-600.

CHTS_fasttrips_demand_v0.5. (2018). Retrieved May 23, 2019, from https://mtcdrive.app.box.com/s/ag6d9dosbfrya3u6sq0l6sk65g9im0ue

Dyno-Demand. (2015). Retrieved July 30, 2019, from https://github.com/osplanning-data-standards/dyno-demand

GTFS-PLUS. (2015). Retrieved July 30, 2019, from https://github.com/osplanning-data-standards/GTFS-PLUS

GTFS Static Overview. (2019). Retrieved July, 2019, from https://developers.google.com/transit/gtfs/

Hamdouch, Y., & Lawphongpanich, S. (2008). Schedule-based transit assignment model with travel strategies and capacity constraints. Transportation Research Part B: Methodological, 42(7-8), 663-684.

Hirschberg, D. S. (1975). A linear space algorithm for computing maximal common subsequences. Communications of the ACM, 18(6), 341-343.

Hoogendoorn-Lanser, S., van Nes, R., & Bovy, P. (2005). Path size modeling in multimodal route choice analysis. Transportation Research Record, 1921(1), 27-34.

Jones, N. C., & Pevzner, P. (2004). An Introduction to Bioinformatics Algorithms. Retrieved July 19, 2019, from http://www.cs.ukzn.ac.za/~hughm/bio/docs/IntroToBioinfAlgorithms.pdf

Khani, A., Bustillos, B., Noh, H., Chiu, Y. C., & Hickman, M. (2014). Modeling transit and intermodal tours in a dynamic multimodal network. Transportation Research Record, 2467(1), 21-29.

Khani, A., Hickman, M., & Noh, H. (2015). Trip-based path algorithms using the transit network hierarchy. Networks and Spatial Economics, 15(3), 635-653.

Khani, A., Sall, E., Zorn, L., & Hickman, M. (2013). Integration of the FAST-TrIPs person-based dynamic transit assignment model, the SF-CHAMP regional, activity-based travel demand model, and san francisco's citywide dynamic traffic assignment model (No. 13-4601).

Khani, A. Route Choice based on Hyperpath in Schedule-based Transit Networks. University of Minnesota – Twin Cities, Minneapolis, 2017.

Kim, J., Pramanik, S., & Chung, M. J. (1994). Multiple sequence alignment using simulated annealing. Bioinformatics, 10(4), 419-426.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics, 34(18), 3094-3100.

Liu, Y., Bunker, J., & Ferreira, L. (2010). Transit Users' Route-Choice Modelling in Transit Assignment: A Review. Transport Reviews, 30(6), 753-769.

Mongeau, M., & Sankoff, D. (1990). Comparison of musical sequences. Computers and the Humanities, 24(3), 161-175.

Mount, D. W. (2004). Bioinformatics: sequence and genome analysis. 2nd (Vol. 692). Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press. xii.

Moving Fast-Trips from Research to Practice...a three-agency experience. (2015). Retrieved July 19, 2019, from http://fast-trips.mtc.ca.gov/

network_draft1.14_fare. (2018). Retrieved May 23, 2019, from http://fast-trips.mtc.ca.gov/library/

Nguyen, S., & Pallottino, S. (1988). Equilibrium traffic assignment for large scale transit networks. European journal of operational research, 37(2), 176-186.

Nguyen, S., Pallottino, S., & Gendreau, M. (1998). Implicit enumeration of hyperpaths in a logit model for transit networks. Transportation Science, 32(1), 54-64.

Nguyen, S., Pallottino, S., & Malucelli, F. (2001). A modeling framework for passenger assignment on a transport network with timetables. Transportation Science, 35(3), 238-249.

Noh, H., Hickman, M., & Khani, A. (2012). Hyperpaths in network based on transit schedules. Transportation Research Record, 2284(1), 29-39.

Nuzzolo, A., & Crisalli, U. (2004). The schedule-based approach in dynamic transit modelling: a general overview. In Schedule-based dynamic transit modeling: theory and applications (pp. 1-24). Springer, Boston, MA.

Nuzzolo, A., Russo, F., & Crisalli, U. (2001). A doubly dynamic schedule-based assignment model for transit networks. Transportation Science, 35(3), 268-285.

Oliker, N., & Bekhor, S. (2018). A frequency based transit assignment model that considers online information. Transportation Research Part C: Emerging Technologies, 88, 17-30.

Ramming, M. S. (2001). Network knowledge and route choice. Unpublished Ph. D. Thesis, Massachusetts Institute of Technology.

Schmöcker, J. D., Bell, M. G., & Kurauchi, F. (2008). A quasi-dynamic capacity constrained frequency-based transit assignment model. Transportation Research Part B: Methodological, 42(10), 925-945.

Schmöcker, J. D., Fonzone, A., Shimamoto, H., Kurauchi, F., & Bell, M. G. (2011). Frequency-based transit assignment considering seat capacities. Transportation Research Part B: Methodological, 45(2), 392-408.

Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. Journal of molecular biology, 147(1), 195-197.

Spiess, H., & Florian, M. (1989). Optimal strategies: a new assignment model for transit networks. Transportation Research Part B: Methodological, 23(2), 83-102.

Thompson, J. D., Higgins, D. G., & Gibson, T. J. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic acids research, 22(22), 4673-4680.

Tong, C. O., & Richardson, A. J. (1984). A computer model for finding the time-dependent minimum path in a transit system with fixed schedules. Journal of Advanced Transportation, 18(2), 145-161.

Tong, C. O., & Wong, S. C. (1999). A stochastic transit assignment model using a dynamic schedule-based network. Transportation Research Part B: Methodological, 33(2), 107-121.

Vovsha, P., & Bekhor, S. (1998). Link-nested logit model of route choice: overcoming route overlapping problem. Transportation research record, 1645(1), 133-142.

Wilson, C. (2008). Activity patterns in space and time: calculating representative Hagerstrand trajectories. Transportation, 35(4), 485-499.

Zorn, L., & Sall, E. (2017). Dynamic Passenger Assignment Challenges (No. 17-05722).