

12-17-2015

Designing Transit Networks for Equity and Accessibility

Kelly L. Bertolaccini
kelly.bertolaccini@uconn.edu

Follow this and additional works at: <https://opencommons.uconn.edu/dissertations>

Recommended Citation

Bertolaccini, Kelly L., "Designing Transit Networks for Equity and Accessibility" (2015). *Doctoral Dissertations*. 974.
<https://opencommons.uconn.edu/dissertations/974>

Designing Transit Networks for Equity and Accessibility

Kelly Layne Bertolaccini, Ph.D.

University of Connecticut, 2015

The equitable provision of public transportation services is a major concern for transit planners and service providers around the world. However, very few tools are available to planners seeking to incorporate equity concerns into their transit network designs. One approach taken by equity researchers is to map the access provided by existing and proposed services and explore how this access varies over time and across communities. The first chapter of this dissertation discusses a comprehensive accessibility mapping method which is able to identify variations in access over time and space.

This research also proposes innovative models for directly incorporating equity into the stop sequencing and stop grouping components of the Transit Network Design Problem (TNDP) and develops a genetic algorithm (GA) to solve these models. First, a single route model is developed to test the effects of designing routes according to different definitions of equity. This research explores nine possible inequity minimizing objective function formulations, drawing from horizontal, vertical, and intermodal equity perspectives. The single route model is largely based on the traveling salesman problem (TSP); every stop must be visited exactly once by a single circulating route. The primary difference is that the single route model replaces the TSP's cost minimizing objective function with an inequity minimizing function. The Sioux Falls, SD and Willimantic, CT networks were used to test the single route model and to develop the GA. These experiments narrowed the list of possible equity objective functions from nine to six.

Kelly Bertolaccini – University of Connecticut, 2015

Extensive testing was conducted on the GA, on both its algorithmic structure and its input parameters, to validate its quality and efficiency.

After testing the single route model and developing the GA, the model was expanded into a multiple route model which includes the stop grouping component of the TNDP. This model also considers route transfers, walking connections to transit stops, demand zones, multiple paths between demand zones, and idle time. This model was applied to a subset of the University of Connecticut's (USA) shuttle bus system and solved using an expanded and updated version of the GA applied to the single route model.

Designing Transit Networks for Equity and Accessibility

Kelly Bertolaccini

B.S., Civil Engineering, University of Connecticut, 2010

B.A., English, University of Connecticut, 2010

M.S., Civil Engineering, University of Connecticut, 2013

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2015

Copyright by
Kelly Layne Bertolaccini

2015

APPROVAL PAGE

Doctor of Philosophy Dissertation

Designing Transit Networks for Equity and Accessibility

Presented by

Kelly Layne Bertolaccini, M.S., B.S., B.A.

Major Advisor

Nicholas E. Lownes

Associate Advisor

Karthik Konduri

Associate Advisor

Amy Burnicki

University of Connecticut
2015

*This dissertation is dedication to my brother and first colleague in engineering,
Matthew Steven Bertolaccini (1990-2004).*

Acknowledgements:

I would like to thank my major adviser Dr. Nick Lownes for all of his guidance and support over the past 5 years. Thank you for challenging and encouraging me to explore ways to integrate the topics I am most passionate about into my research.

Thank you to Dr. Osleeb, who may not be a member of my committee on paper, but provided extensive feedback throughout the research process. I am grateful for his input and guidance.

Thank you to the other members of my committee, Dr. Konduri and Dr. Burnicki, for reviewing my research and providing feedback.

Thank you to the many other professors who have influenced my growth as an academic and a professional, especially Dr. Norman Garrick, Dr. S. Travis Waller, Dr. John Ivan, Dr. Carol Atkinson-Palombo.

Thank you to my family, Mom, Dad, Bailey, Christy, and my extended family, who have always provided me with the emotional and financial support I needed to pursue my academic goals.

Thank you to my boyfriend, Ned Eskew, for the love and support – especially in the past few months as I scrambled to finalize my research. I promise to do the same for you when you are writing your dissertation. I also want to thank his family, Kate, Ron, and Camilla, for their encouragements.

Thank you to Dr. Abigail Osei-Asamoah and Dr. Melissa Duell for coaching me through the doctoral process and being my co-women in engineering. You are wonderful friends.

Thank you to my other colleagues in the UConn Transportation and Urban Engineering program whom I have had the opportunity to work with over the past 5 years – Dr. Sha Mamun, Dr. Hamed Anghari, Nick Hart, Dr. Chris McCahill, Jason Zheng, Jim Mooradain, Jerry Klein, Jessica Ratch, Brett Decker, Mike Kowalzyck, and Garrett Bolella.

Thank you to the many, many other friends who have encouraged and supported me in myriad ways over the years – Valeri Kolev, Shana Clarke, Elise Dunham, Julia Werth, Alyssa McDonagh, Steve Samela, Steve Byrne, Franz Maier, Jess Stanley, Dr. Caitlin Martin, Alyssa Pepper, Dr. Laura Lewis, Dr. Kelly Holbrook, Dr. Lisa Oransoff, and Amy Law.

Finally I would like to thank the organizations that have supported me financially throughout my graduate studies: FHWA, NEUTC, CTDOT, CT-ITE, and WTS-CT Valley.

Table of Contents:

Chapter 1. Measuring and Mapping Transit Accessibility using Public Data.....	1
Chapter 2. Optimizing Transit Routes for Equity.....	25
Chapter 3. A Genetic Algorithm Approach to Solving the Equitable Traveling Salesman Problem.....	79
Concluding Remarks and Future Work.....	98
Appendices.....	104
References.....	108

Chapter 1: Measuring and Mapping Transit Accessibility using Public Data

1. Background.....	2
2. Methodology.....	5
2.1. Transit Opportunity Index.....	6
2.1.1. Transfers.....	8
2.1.2. Time of Day Analysis.....	8
2.2. Data Sources.....	9
2.3. Procedure.....	11
3. Discussion of Results.....	17
3.1. Time of Day Analysis.....	21
4. Conclusions.....	23

1. Background

Improving access to transportation services is a concern of citizens, planners, and policymakers worldwide. The broad concern for transportation access derives from concerns about access to economic and social opportunities. Transportation systems designed to move people provide individuals with the mobility necessary to be active participants in their communities and the economy, especially in places with spatially segregated land use like much of the modern United States (Hanson, 2004; Grava 2003). Traditionally, most urban areas were designed so that stores, schools, jobs, and healthcare services were within walking distance of peoples' homes. However, as private automobile ownership increased in the mid twentieth century, planners began to decentralize land use (Knox, 2010) leading to cases of spatial mismatch¹. In this context, limited access to transportation services can severely limit peoples' abilities to meet their needs.

Research in the social sciences confirms existing inequalities in access to healthcare (Syed, 2013), supermarkets offering healthy food (Walker, 2010; Farber et al, 2014), and higher education (Kenyon, 2011). These studies specifically point to a lack of physical access, caused by a combination of segregated land use and a lack of transportation options, as one of the causes for these inequalities. In particular, the studies suggest that locating services in areas where there is poor or no public transportation service can be a barrier to access.

While transportation researchers also agree that having access to a reliable mode of transportation is essential to finding and maintaining employment, there is some disagreement on the effectiveness of investing in public transportation in certain cases. Several studies suggest

¹ John F. Kain (1968) is generally credited with coining the term spatial mismatch. He used the term to describe the spatial separation of unemployed African American workers from potential jobs which arose from the combination of housing and employment discrimination against African Americans 1960s Chicago. The spatial mismatch hypothesis has since been applied to a broader range of scenarios in which people are unable to live where they can find jobs.

that improving access to jobs by public transportation is an effective way to improve access to jobs and improve overall levels of employment (Zhang et al, 1998; Yi, 2006; Alam 2009; Rotger and Nielson, 2015). Others suggest that providing cars to the unemployed is a more effective strategy for improving access to jobs, particularly in rural and job-poor neighborhoods (Blumenberg and Ong, 2001; Blumenberg and Shiki, 2003; Sandoval et al, 2011). Despite the apparent contradiction, all of the studies agree that a public transportation system can improve access to jobs, if the system provides reliable, frequent service at times when people need it to commute to work. The studies which suggested that public transportation was not the most effective method for providing access to jobs were case studies of American cities with transit systems that did not provide reliable or frequent enough service for the unemployed population. This suggests that providing reduced fares for low income riders or re-routing bus lines to provide more stops in low income neighborhoods alone is not enough to improve job accessibility. People need service they can rely on at all hours of the day, including late at night, for public transportation to effectively improve their employment opportunities and ability to maintain jobs.

A good public transportation system is particularly important in creating a transportation system where all people have equal access to jobs and services. While the private automobile has certainly increased the personal mobility of some segments of the population, it is important to remember that not everyone is equally able to use private auto as a means of transportation (Knox, 2010). Children, older adults, and people with certain physical disabilities are unable to personally operate private vehicles and therefore their physical access is limited when living in a region with poor public transportation services. People with disabilities which prevent them from driving are particularly dependent on public transportation and cite poor transit services as

major barriers to both employment and fuller participation in their communities (Jansuwan, 2013; Samuel et al, 2013). Older adults, particularly those living in rural communities, also point towards infrequent and untimely transit services as a challenge for accessing medical services (Mattson, 2011). Low income families and individuals who cannot afford a vehicle or the associated maintenance and fuel costs will also find themselves at a disadvantage if living auto-dominated areas with decentralized land use. Public transportation systems provide mobility options for people who cannot afford to own or operate a private auto, providing them with physical access to jobs, education, healthcare, supermarkets, and other services. Similar to the other vulnerable populations mentioned, low income communities would greatly benefit from increasing transit service frequency, especially during off-peak hours and in reverse commute directions (Giuliano, 2005). Improving public transportation services between residential areas and non-local activities, in addition to locating more activities locally, is one important way to create more equitable physical access for everyone (Newman and Kenworthy, 1999).

While many researchers and organizations have recognized the importance of public transportation access and equitable transportation services, they also voice the need to better quantify, analyze, and plan for the concepts of accessibility and equity (Hanson 2004).

Quantifying access can be challenging because there are many types of barriers, including spatial, temporal, financial, social, etc., that may limit a person's access, not all of which can be quantified. The tool presented in this paper quantifies access by public transportation using the transit opportunity index (TOI) developed in Mamun et al (2014). The TOI considers three major aspects of access: spatial, temporal, and trip coverage. To put it into the perspective of transit riders, the TOI assigns scores based on travelers' abilities to access transit stops from their homes, the frequency of busses at these local stops, and the number of other stops they can reach

within a reasonable amount of time. The TOI will be discussed in further detail in the following section. Other measures of accessibility focus almost exclusively on either spatial accessibility (CITE) or temporal accessibility (CITE). (For a comprehensive literature review of other accessibility measures, refer to Mamun et al (2014)) However, as discussed earlier in the introduction, the frequency of services and the ability of the transit services to get people to places they need to go in a reasonable amount of time are vitally important to transit dependent people. Any transit accessibility tool seeking to inform transportation policy and provide transportation planners with a useful tool for improving the equity and quality of public transportation services should therefore include these dimensions of accessibility.

The tool for measuring accessibility discussed in this paper was developed as a part of the ongoing t-HUB project at the University of Connecticut. t-HUB seeks to provide public transportation planners and service providers in Connecticut with practical tools for analyzing the equity of services and meeting Title VI requirements (Lownes et al, 2013). While the tool described in the paper does not explicitly consider equity, it was created to allow comparisons between proposed service changes to access for different communities.

Another goal of the t-HUB initiative is to use easily obtained, publically available data, such as General Transit Feed Specification (GTFS) data and census data. The accessibility tool presented in this paper relies exclusively on GTFS data for transit system information. Further information on data sources is provided in the methodology section.

2. Methodology

The purpose of the research presented in this chapter is to create a Python-based tool which automates the calculation of the Transit Opportunity Index (TOI) for a transit service area using only general transit feed specific (GTFS) data and basic census data. This will allow transit service providers and planners to generate accessibility data quickly and create maps to visualize transit accessibility with minimal data requirements. This tool can also be used to examine changes in the transit service area's TOI distribution over the course of a day or a week.

To validate the script, it was applied to the bus systems operated by Connecticut Transit (CT-Transit) which includes the Hartford, Meriden, New Britain, New Haven, Stamford, and Waterbury systems, as well as Atlanta's MARTA system. The weekday TOI was calculated for each block group in these transit system's service areas. Additionally, TOI was calculated for Hartford for four distinct time periods (morning peak, inter peak, evening peak, and off peak) to explore whether the TOI might be a useful tool for exploring temporal changes in access.

The script was written in Python 2.7.2 for use with ArcGIS 10.1 and is compatible with later versions of ArcGIS.

2.1 The Transit Opportunity Index

The Transit Opportunity Index (TOI) is a comprehensive measure of transit accessibility and connectivity. For the purposes of this application, TOI is represented as a single score for each block group in the service area. All scores will be between 0 and 1 with a higher score indicating greater accessibility. The TOI includes three major components: a spatial coverage score, a temporal coverage score, and a trip coverage score. Figure 1.1 below outlines each of the calculations of the TOI.

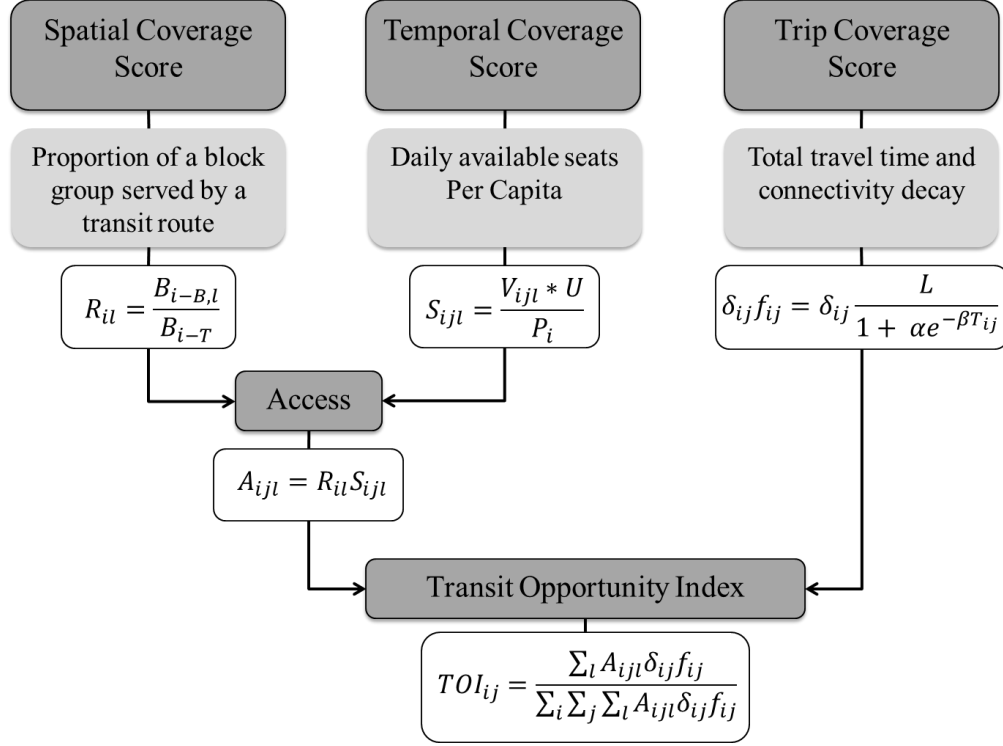


Figure 1.1: Transit Opportunity Index Calculations

The spatial coverage score (R_{il}) is calculated as the proportion of the area within a census block group that lies within 400 meters of bus stop on route l . The temporal coverage score (S_{ijl}) is the daily transit capacity per capita. It is calculated as the vehicles traveling from origin i to destination j on route l (V_{ijl}) multiplied by the capacity of the vehicle (U) divided by the population of origin i (P_i). The trip coverage score is calculated using a decay function which indicates a decreasing level of “coverage” for trips with longer travel times. The trip coverage score is multiplied by a connectivity factor (δ_{ij}) which equals 1 if i and j are connected and 0 otherwise. It also includes total trip time from origin i to destination j . Trip time is calculated using equation 1 shown below:

$$T_{ij} = T_{access} + T_{wait} + T_{In\ Vehicle} + T_{egress} \quad (1)$$

The other elements of the connectivity decay functions are set parameters. See Mamun et al. (2014) for further details on the connectivity decay function and setting the function parameters. The TOI of each origin destination pair is found by calculating the product of these scores and standardized across the system. The TOI of each origin is calculated from the pair scores using equation 2 below:

$$TOI_i = \sum_j TOI_{ij} \quad (2)$$

2.1.1 Transfers

Initial TOI scores are calculated using directly connected stops only. However, transfers between routes can be incorporated into the TOI scores. Transfers must be considered before the scores are simplified to origin only scores. If two stops are not directly connected, the first step is to identify a possible transfer stop k . The identification of transfer stops will be discussed in great detail in the procedure section. Once a transfer stop is identified, the trip travel time T_{ij}^k is calculated using equation 3. Note that 20 minutes are added to the trip travel time as a transfer penalty.

$$T_{ij}^k = T_{ik} + 20 + T_{kj} \quad (3)$$

The new TOI score is then calculated using equation 4:

$$TOI_{ij}^k = \frac{1}{2} \left(\frac{TOI_{ik}}{f_{ik}} + \frac{TOI_{kj}}{f_{kj}} \right) * f_{ij}^k \quad (4)$$

2.1.2 Time of Day Analysis

To analyze and visualize changes to the TOI over the course of a day, some minor alterations were made to the code. For this application, the day was split into four time periods for analysis: Morning Peak (6 am to 10 am), Inter Peak (10 am to 4 pm), Evening Peak (4 pm to 8 pm), and Off Peak (8 pm to 6 am). Because these time periods are of varying lengths, the TOI must be adjusted using equation 5 for the purpose of comparison.

$$TOI_{TOD,adj} = TOI_i * \frac{24}{n} \quad (5)$$

Where n is the number of hours in the analysis period.

The only other alteration to the code is that TOI is no longer normalized by the sum of the TOI scores across the system for the given analysis period. This will allow for an exploration of changes in access over the various time periods.

2.2 Data Sources

This script calculates TOI using readily available and, in many cases, public data. The necessary transit system information was drawn from the system's GTFS data. GTFS is the data specification required of any service provider who wishes to make their data searchable on Google Transit (Google Developers, 2015). Despite its relatively recent development, many researchers are developing tools which utilize GTFS to analyze public transportation systems and is quickly becoming a standard within the industry (Nassir et al, 2011; Nazem et al, 2013; Wong, 2013; Bertolaccini and Lownes, 2014; Liu and Cirillo, 2015). GTFS comes in the form of a zip file containing multiple comma delimited text files. From GTFS files, users can determine network topology, vehicle frequencies and headways, in-vehicle travel time, and stop locations. This script presented in this paper uses only GTFS files required by Google Transit. Descriptions of the necessary files and their purposes in the script are outlined below:

- “stops.txt” – provides stop locations, required for calculating spatial score and finding transfers
- “route.txt”- identifies routes, required for organizational purposes
- “trips.txt” – identifies each time a vehicle makes a trip, required for calculating temporal component
- “stoptimes.txt” – identifies each time a vehicle makes a stop, required for calculating length of trip and assessing variations in access by time of day
- “calendar.txt” – defines service ids which indicate which day(s) of the week certain trips operates, required for calculating temporal component

These files are connected to each other with unique stop, trip, and route ids, as shown in figure 1.2 below:

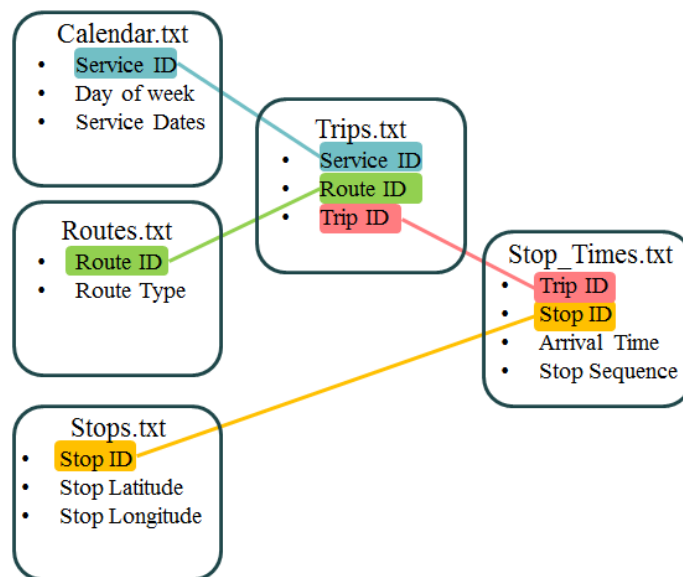


Figure 1.2: GTFS file structure

Some transit service providers may also include the optional transfer file in their GTFS. If this file is available and inclusive of the entire transit system, it should be used in place of the portion

script used to identify transfer stops. However, users must carefully inspect the transfer file prior to use as transit providers may choose to include only a subset of the available modes in the transfer file. For example, Boston only includes transfers on the T system in their transfer file while not providing information on transfers by other modes. The GTFS files used to develop this script were provided by CT Transit (CT Transit, 2015).

In addition to the transit system data, the TOI script requires a shapefile with basic population data at the preferred geographic scale. The population data is required to calculate the temporal score of the TOI. This data is readily available from either the US Census Bureau or state data centers. The shapefile of CT population data at the block group scale used to develop this script uses American Community Survey (ACS) 5-year estimates from the CT State Data Center (CT State Data Center, 2015).

2.3 Procedure

This section of the paper will outline the major steps of the script, emphasizing how to use GIS tools in conjunction with GTFS data. Figure 1.3, found on the following page, shows the flow of this script.

Step 0: Input GTFS and block group population data. Set coordinate system.

Step 1: Organize Input Data

Organize the GTFS files into useful formats, including nested dictionaries and lists.

Step 2: Create Stop and Route-Stop shapefiles using GTFS data

Create a shapefile of all stops using the coordinates provided in the GTFS stops file. Next create a Route-Stop shapefiles which identifies stops by both stop id and route id. If a stop is

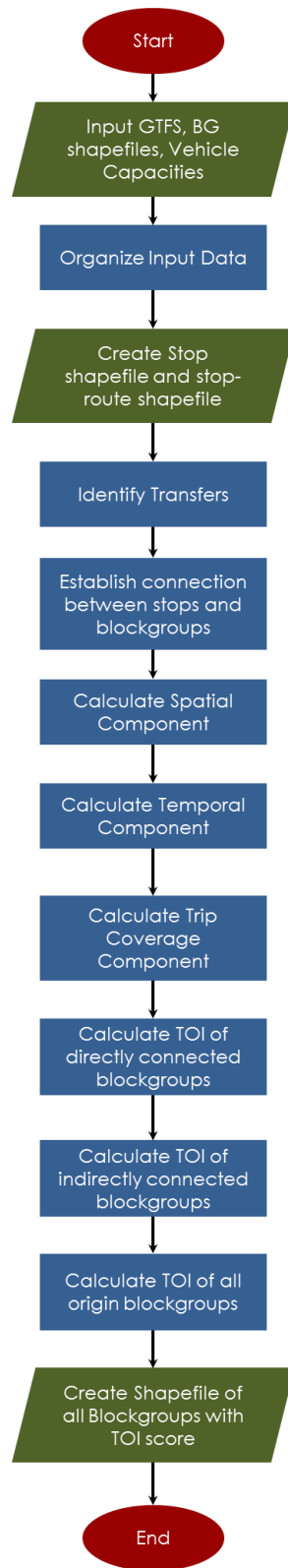


Figure 1.3: Flowchart of automated TOI Calculation Procedure

used on multiple routes, then a separate point will be included for each stop- route pair. This shapefile will also include vehicle frequency data obtained from the trips file. Frequency data is separated by both stop and route.

Step 3: Identify Transfers

The purpose of this step is to identify possible transfer stops between routes. The suggested method only considers physical proximity of stops, assuming temporal concerns are not significant. This is a reasonable assumption given the hub-and-spoke structure of the CT Transit systems used to develop this code but will not always be the case. (Most transfer stops were located in the central business districts with frequent enough transit service that temporal considerations are unlikely to present a serious barrier to transfer). Transit service providers can recommend transfer stops in the optional transfer GTFS file but it is not required. If a transfer file is provided, use the information provided in the file and skip this step.

To begin identifying transfer stops, the Route-Stop shapefile must be dissolved by route. Dissolving by route will cluster the set of stops associated with a specific route into a single multipoint object. Next, a 200 meter buffer is created around each route object. This script assumes that 200 m is the maximum distance people are willing to walk to make a transfer. To the knowledge of the authors, there is no formal research indicating how far people are willing to walk to make a transfer. However, in the experience of the authors, people are generally only willing to cross the street or walk at most city block to make a transfer. Figure 1.4 below shows the 200 meter buffers around the stops of two routes running parallel to each other. Even though the stops along a route appear to be separate, they are actually represented as a single multipoint object in the script.

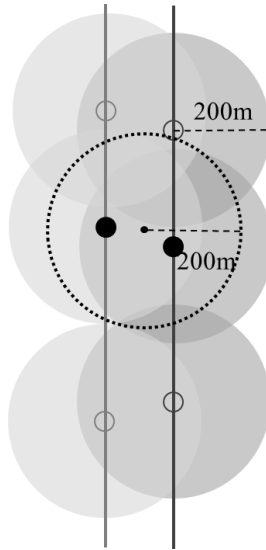


Figure 1.4: Identifying Transfers for Parallel Routes

Next, identify the intersection of a route's buffer with another route's buffer. Find the centroid of this intersection. The centroid must lie within the intersection area. In the example shown in Figure 1.4, the centroid of the intersection is shown as a small black dot. Finally, select the stops within 200 m of the centroid. The selected stops are the possible transfer stops. The purpose of using the centroid is to avoid identifying multiple transfer locations between routes. Many routes share segments or have parallel segments that extend many stops. It unnecessarily complicates the script to include all of these potential transfer points. For the purpose of calculating TOI, it does not matter how many transfer stops are available between a pair of routes or exactly where a person chooses to transfer. All that matters for the purpose of calculating TOI is whether or not a transfer between routes is possible.

Step 4: Establish connections between stops and block groups

Identify the block groups within the 400 meter service area of each stop. This requires using the simple, but relatively time consuming, search by location function.

Step 5: Calculate Spatial Component of TOI (for directly connected stops)

Calculate spatial score R_{il} for each origin i and route l . The score is simply the ratio of the area within a block group that is within 400 meters (approximately ¼ mile) of a bus stop to the total area of the block group. To find this value, first create a 400 meter buffer around each of the multipoint route objects (created in Step 3) as shown in figure 1.5 below.

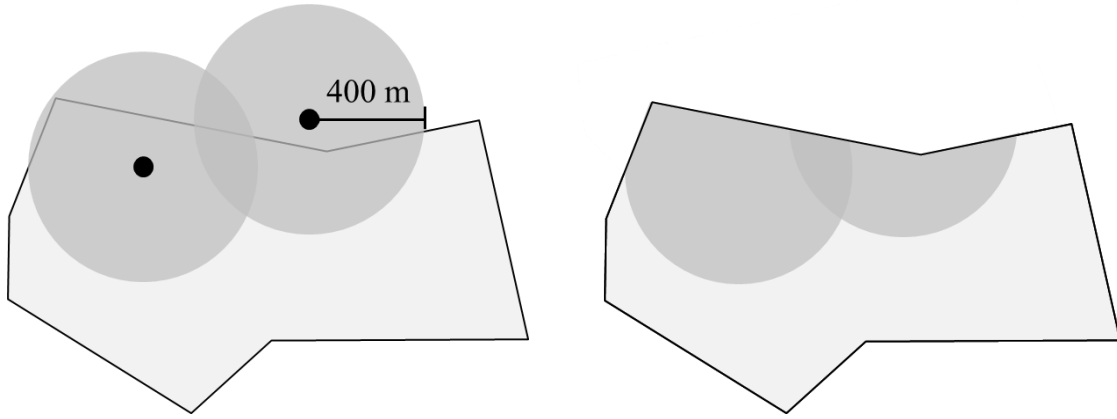


Figure 1.5: Spatial Score Calculations (a) Creating buffers around stops (b) Intersecting buffers with block group boundaries. For this example, the spatial score would be approximately 0.30.

Then find the area of the intersection between the route buffer and the block group boundary.

Divide this value by the total area of the block group to find the spatial score of the block group for that particular route.

Step 6: Calculate Temporal Component of TOI (for directly connected stops)

The temporal score, S_{ijl} , is the daily available seats per route per capita. The daily vehicle frequency of trips from i to j along route l can be determine from the data provided in the GTFS trips file. In its current form, the script uses average weekday frequency. With simple alterations,

the script could find vehicle frequency for other days of the week, during different periods of the day, or over the course of the week.

Determining the daily seats per route per capita also requires knowing the capacity of vehicles and the population at origins. The population information comes from the block group population shapefile. The vehicle capacity must either be estimated or collected directly from the service provider because it is not provided in the GTFS files. The original formulation of the TOI assumes vehicle capacity is constant across routes, trips, and times of day. This is not always the case. For this particular application, vehicle capacity was assumed to equal 34 people. This is the number of seats on a standard CT Transit bus though some routes operate with higher capacity vehicles during peak hours.

Step 7: Calculate Trip Coverage Component of TOI (for directly connected stops)

The key component of the trip coverage score that is calculated using the script is the in-vehicle travel time. All of the other parameters in the trip coverage score are constant. Calculating vehicle travel time between origin i and destination j uses data from the GTFS stop times file. Determining the exact value can be tricky considering there may be multiple stops with block groups i and j and the length of the trip may vary throughout the day. The current script uses the first stop of a trip within block group i and the first stop of a trip within block group j to calculate in-vehicle travel time. This should give a middle range value for in-vehicle travel time between stops serving the origin and destination. The script also calculates the in-vehicle travel time during off peak hours when calculating a weekday TOI score.

Step 8: Calculate TOI of directly connected stop pairs

Use the spatial coverage score, temporal score, and trip coverage score, calculated in steps 5, 6, and 7 respectively, to calculate the TOI of directly connected stop pairs.

Step 9: Calculate TOI of indirectly connected stop pairs

Use the transfer stops found in step 3 (or identified in the GTFS transfer file, if available) to incorporate transfers into the TOI scores found in Step 8.

Step 10: Calculate TOI of all origin block groups and add to block group shapefile.

Calculate the TOI of origins using equation 2. Then add the TOI score as a field to the block group shapefile for further analysis and mapping.

3. Discussion of Results

The script was applied to the six transit systems operated by CT Transit Hartford: Hartford, Meriden, New Britain, New Haven, Stamford, and Waterbury, as well as the multimodal MARTA system. Table 1.1 below shows some important characteristics of these transit systems and their service areas alongside the total time it took for the automated TOI tool to run for each system. These characteristics will be useful for comparing the TOI maps and the script execution times.

Table 1.1: Characteristics of CT Transit Systems

		MARTA	Hartford	Meriden	New Britain	New Haven	Stamford	Waterbury
System Info	Routes	96	59	6	12	26	18	24
	Stops	8881	4239	204	751	3194	864	904
	Trips	8622	3031	125	408	1969	990	802
Area Info	Area (sq. km)	1541	1466	69	271	668	218	320
	Pop (1000s)	1375.3	56.0	4.3	13.7	36.8	14.7	13.0
	Block Groups	5088	569	48	147	404	156	136
Run Time (min):		84.8	19.7	0.5	1.4	8.2	1.8	1.8

Figure 1.6 and 1.7 below show the distribution of TOI scores throughout the systems. Darker colored areas indicate higher levels accessibility. Gray areas lie outside of the area served by the transit system. The meaning of block group TOI scores can be difficult to interpret on their own but are useful to look at in comparison to other block groups within the system service area. The distribution of scores as visualized by the maps in Figures 1.6 and 1.7 can help transit planners and service providers understand where access is greatest and where it may need to be improved. Though these maps are very useful for comparisons between block groups within systems, it is important to be cautious when comparing the TOI maps between different systems, especially considering that TOI scores are standardized by total system scores.

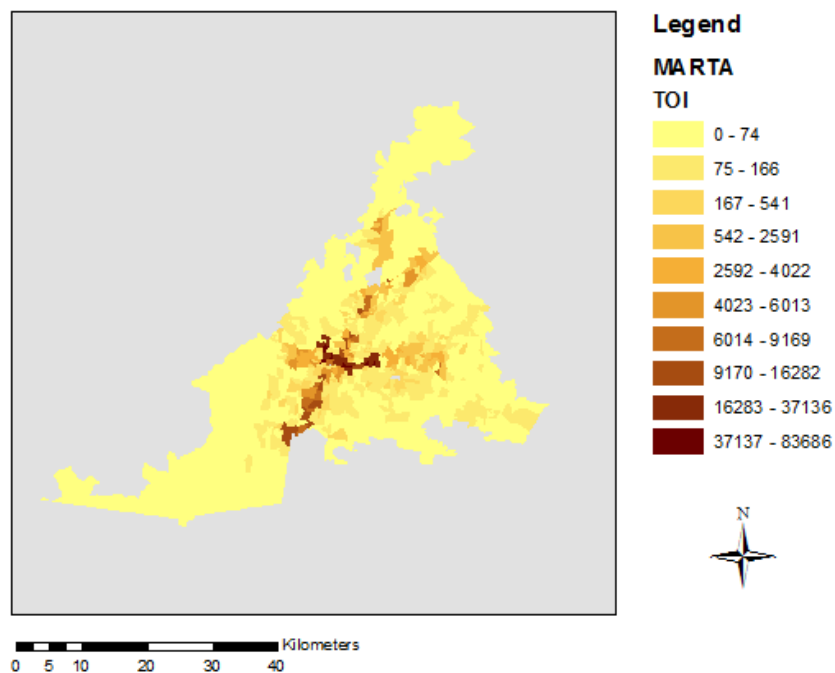


Figure 1.6: TOI Map for MARTA (Atlanta, GA). Darker colors indicate greater transit access

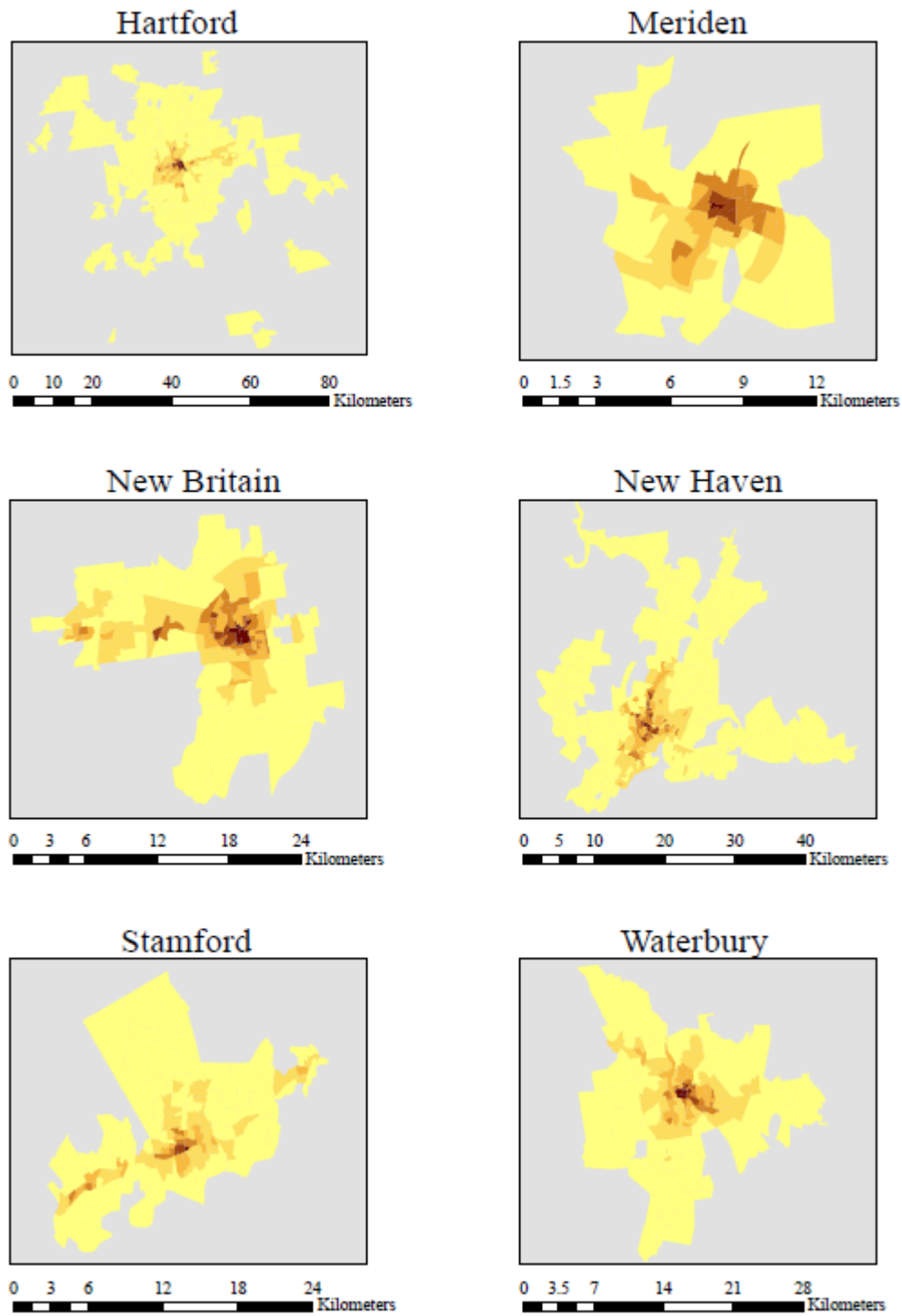


Figure 1.7: TOI Maps for CT Transit Systems. Darker colors indicate greater transit access.

This script has provided substantial time savings for the calculation of the TOI. In Mamun et al (2014), the authors calculated the TOI manually for CT Transit New Haven without transfers within the boundaries of the city. Even this smaller, simpler version of the New Haven TOI took 8 to 10 hours to calculate manually, as estimated by the authors. This script found the TOI for the entire CT Transit New Haven service area, including transfers, in just over 8 minutes. Table 2 below outlines how long it took to run each step of the script for each of the systems in seconds. For reference, the script was run a computer with a quad core processor with a speed of 2.5 GHz and 6.0 GB of RAM.

Table 1.2: Script Running Times (in seconds)

Step:	Hartford	Meriden	New Britain	New Haven	Stamford	Waterbury
Organize Data	11.6	4.7	5.3	10.3	5.7	5.2
Create Stop Shapefile	6.7	0.9	1.4	4.4	2.0	1.7
Identify Transfers	94.8	8.3	15.8	53.3	20.7	20.7
Relate Stops-BGs	344.8	10.4	41.3	231.0	49.4	51.4
Spatial Component	75.1	3.9	11.7	45.5	14.9	16.5
Temporal Component	9.0	0.2	0.7	8.3	0.9	0.5
Trip Coverage	14.7	0.1	0.8	17.0	1.0	0.6
TOI Directly Connected	2.8	0.1	0.2	1.3	0.3	0.2
TOI Indirectly Connected	626.8	0.2	3.6	118.8	9.9	11.8
Complete TOI Shapefile	0.8	0.3	0.3	0.5	0.3	0.3
Total:	1186.9	29.0	81.2	490.5	105.0	108.7

The script for CT Transit Hartford, by all measures the largest CT transit system considered in this research, ran in just under 20 minutes. In contrast, the script for Meriden, the smallest system, ran in 29 seconds. While 20 minutes seems like a reasonable amount of time for the script to run for a system of this size, the drastic increase in time raises some concerns about the scalability of this script to systems larger than Hartford. The Hartford system has about 21 times the stops, 24 times the trips and serves nearly 12 times the block groups of Meriden, yet the

script takes more than 40 times longer to run. A similar trend appears when comparing the Hartford system to the MARTA system. The Hartford system is roughly half the size of the MARTA system yet the script took over 4 times longer to run on the MARTA system. The script running time increases at a rate greater than the increasing size of the system which may become problematic as the TOI tool is applied to larger, more complex transit system.

Certain portions of the script were more consistently time consuming than others. Identifying transfer stops and establishing the relationship between stops and block groups are relatively time consuming steps. The reason these steps are costly is because of time consuming GIS functions, not inefficient coding. There is not much that can be done to improve the speed of these steps. However, these are also steps that do not need to be repeated if a service provider was interested in exploring changes to TOI scores under a series of different scenarios. Transfer stops and the relationship between stops and block groups can be stored for repeat use.

Calculating the spatial component was also relatively costly due to required GIS functions, though its scores would likely need to be recalculated with any changes in services. By far the most inefficient step was incorporated transfers into the TOI score. This may be due to inefficient coding. This portion of the script is written as a series of nested for-loops which could easily be parallelized for further time savings.

3.1 Time of Day Analysis

In addition to running the script on the 6 CT Transit systems and MARTA, the script was also adjusted to explore changes in accessibility in Hartford over the course of a normal weekday. The results are shown in Figure 5 below.

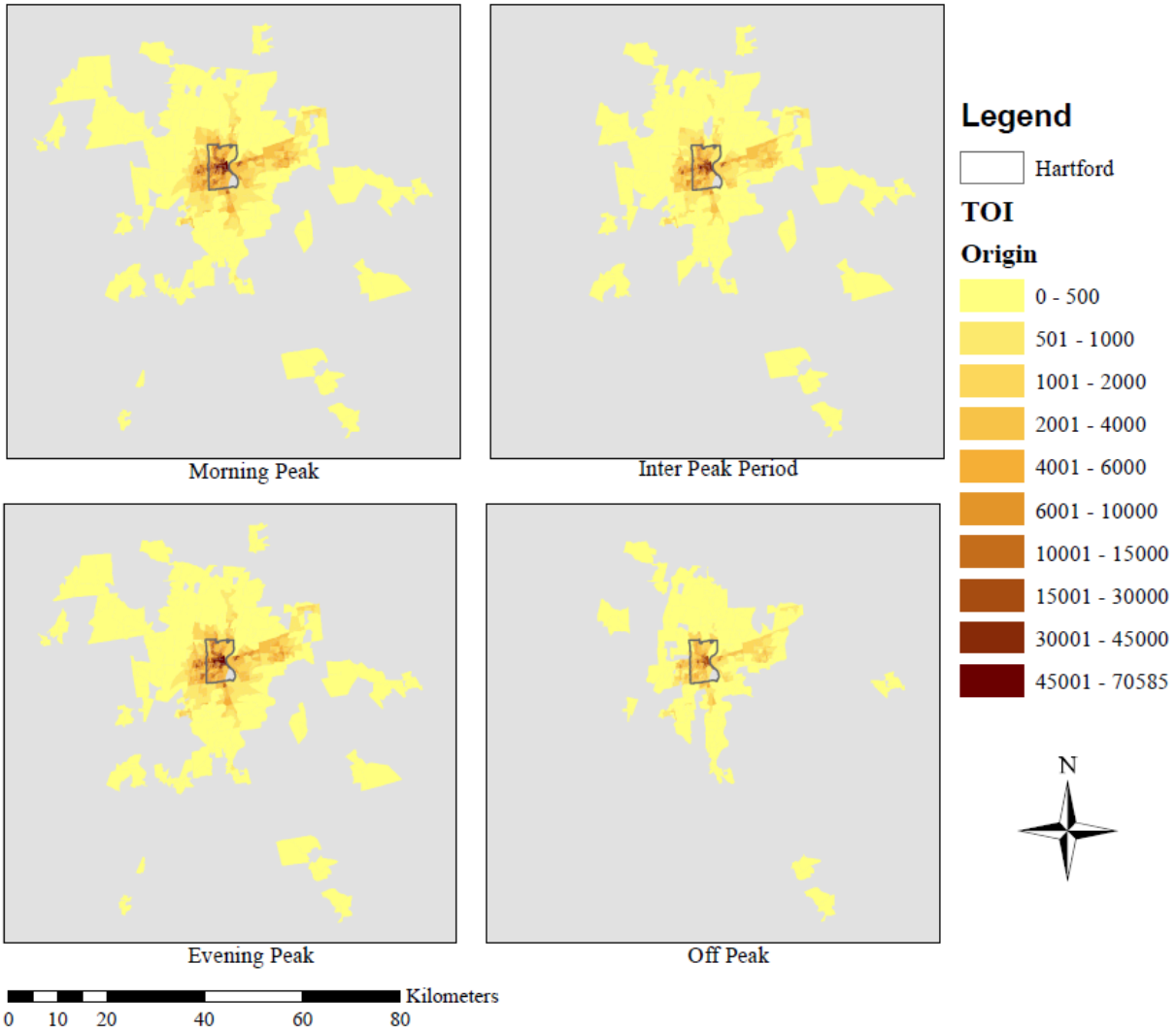


Figure 1.8: CT Transit Hartford: TOI by Time of Day (for a normal weekday)

The maps in Figure 5 match the expected pattern, with the greatest level of access during the peak period, slightly less service during the inter peak period, and greatly constricted service during the off peak hours. Certain transit services connecting the northern suburbs and southwestern region (including New Haven) to Hartford are only available during peak hours, while connections to the east (including East Hartford and Manchester) are available nearly 24 hours a day. This pattern makes sense considering the demographics of the region. East Hartford and Manchester have larger transit dependent populations than the other surrounding regions and

a higher proportion of the residents in those areas are employed in jobs with unconventional hours. The adjusted time of day script was useful for visualizing the changes in access throughout the day which is very important to improving accessibility for transit dependent populations.

4 Conclusions

The purpose of this research was to create a python script to automate the calculation of block group TOI scores using only GTFS and census data. By using easily accessible, publically available data, the script can be applied to a larger number of systems by a wider range of users. The script successfully calculates TOI using only publically available data while making minimal assumptions. Automating the TOI calculations resulted in substantially time savings. Mamun et al (2014), the original developers of the TOI, estimate that it took between 8 and 10 hours to calculate block group TOI for a limited version of the New Haven system without considering transfers. The TOI script was able to calculate TOI for the entire New Haven system with transfers in just over 8 minutes. This script is also capable of calculating TOI for larger systems, such as Hartford and MARTA, though further research is required to determine the impact of going to much larger systems.

Future research should consider the improving efficiency of the final step of the script which incorporates transfers into the TOI. This was both the costliest step of the script and the section most likely to be improved through more efficient coding. This step could be significantly improved through parallel processing. The tool could also be improved by developing a more nuanced approach to incorporating transfers. It is possible to use the GTFS data to develop a method for identifying transfers which considers both the spatial and temporal components to transfers. Research is currently underway to allow more user control over the treatment of

transfers within the system, especially those between different modes. The paucity of research on transfers between modes makes it difficult to know whether a transfer between modes (eg. bus to light rail) should be treated differently from a transfer on the same mode (eg. bus to bus) but it seems reasonable that a transit operator may want to implement different transfer penalties for different types of transfers.

This research is currently being used by the Capital Region Council of Governments (CRCOG) in their regional transit study. It is also being adapted for a study by the Boston Metropolitan Planning Organization (MPO).

CHAPTER 2: OPTIMIZING TRANSIT ROUTES FOR EQUITY

1. Introduction.....	26
2. Defining Equity.....	30
3. Single Route Model.....	33
3.1. Methodology.....	33
3.1.1. Model Formulation.....	33
3.1.2. Solution Methodology.....	34
3.2. Test Networks and Experiments.....	41
3.3. Discussion of Results.....	43
4. Multiple Route Model (without stop grouping)	51
4.1. Methodology.....	51
4.1.1. Model Formulation.....	51
4.1.2. Solution Methodology.....	53
4.2. Test Networks and Experiments.....	54
4.3. Discussion of Results.....	57
5. Multiple Route Model (with stop grouping)	58
5.1. Methodology.....	59
5.1.1. Model Formulation.....	59
5.1.2. Solution Methodology.....	61
5.2. Test Networks and Experiments.....	69
5.3. Discussion of Results.....	70
6. Conclusions and Future Work.....	76

1. Introduction

The equitable provision of public transportation services is a major concern for transit planners and service providers from around the world. In the United States, any organization receiving federal funding must meet the non-discrimination requirements outlined in Title VI of Civil Rights Act of 1964 (U.S. Congress 1964) and Executive Order 12898 on environmental justice (Clinton 1994) . Though the Federal Transit Administration (FTA) has published multiple circulars (FTA 2012) attempting to clarify how Title VI and environmental justice requirements apply to transit services, many planners and service providers still voice confusion on how to measure incorporate equity and non-discrimination requirements into the planning process (Lownes et al. 2013; Amekudzi et al. 2012). Researchers have also expressed concern about the post hoc nature of existing equity analysis methodologies and have suggested that equity needs to be actively considered during the technical stages of the planning process which have traditionally focused solely on maximizing efficiency and capacity (Deka 2004).

Several post hoc methods have been developed to assess the equity of existing public transportation systems. Some researchers have explored the application of Lorenz curves to public transportation systems (Delbosc and Curry 2011; Bertolaccini and Lownes 2014). Delbosc and Curry used Lorenz curves to assess how well the transit supply matched up with transit demand in Melbourne, Australia. The Lorenz curves were then used to calculate a single coefficient reflecting the equity of a region's transportation system. Bertolaccini and Lownes expanded the application to several U.S. cities. Lei et al developed thorough indicators of transit and auto accessibility which consider the spatial and temporal changes in accessibility. They were then able to map both transit and auto accessibility and use these maps to identify areas in

need of transit improvements (Lei et al. 2012). Similarly, the research begun by Mamun et al. (2013) and continued in the first chapter of this dissertation is being used in equity applications. The transit opportunity index (TOI) can be used to compare the transit access provided to different neighborhoods both for the existing system and for any proposed changes to the system. These methods are intended for the analysis of existing systems. They also have the potential to be applied to a limited set of proposed redesigns. However, these methods do not provide method for incorporating equity concerns into the development of new transit routes and networks. The research presented in this chapter develops a methodology which incorporates equity directly into the transit network design process.

Though the topic of designing transportation networks for equity has been explored to some extent in the field of highway design and traffic assignment (Duthie and Waller 2008), network maintenance scheduling (Boyles 2015), project selection (Joshi and Lambert 2007), and congestion pricing (Wu et al. 2008; Eliasson and Mattsson 2006), few tools exist to help public transportation planners incorporate these concerns into the design of public transportation networks. Bowerman et al. (1995) created a multi-objective optimization model for designing urban school bus routes which explicitly considered the equitable distribution of trip lengths among the students in one of its objectives. Though school bus routing is certainly different from general transit system design due to its limited and fixed set of users, this model was a first attempt at incorporating equity into the vehicle routing problem (VRP); the problem at the foundation of many transit routing models. Fan and Machemehl (2011) formulated and developed a method for solving a public transportation network redesign problem which accounts for spatial (or horizontal) equity concerns. Their paper however does not consider the varying needs of different segments of the population, a concept known as social (or vertical)

equity. Ferguson et al. (2012) proposed a method for addressing equity concerns in the formulation of the transit frequency setting problem. The objective function of their formulation minimizes the coefficient of variation of the difference between auto accessibility and public transit accessibility while prioritizing improvements to residential areas with larger proportions of protected persons. The term protected persons, as used by the authors, refers to persons protected by Title VI and the previously mentioned environmental justice executive orders. These populations include low income families, and racial and ethnic minorities. To the knowledge of the author, no published research has attempted to incorporate equity into the stop sequencing or stop grouping components of the transit network design problem.

The transit network design problem (TNDP) is an incredibly complex problem with several interrelated components (Farahani et al., 2013; Desaulniers and Hickman, 2007). Figure 2.1 below visualizes the TNDP and the four primary components of the problem: selecting stop location, grouping stops into routes, sequencing stops within routes, and setting route frequency. The research presented in this chapter focuses on incorporating equity into the stop sequencing and stop grouping components of the TNDP. Together this chapter will refer to this as transit route design, or routing.

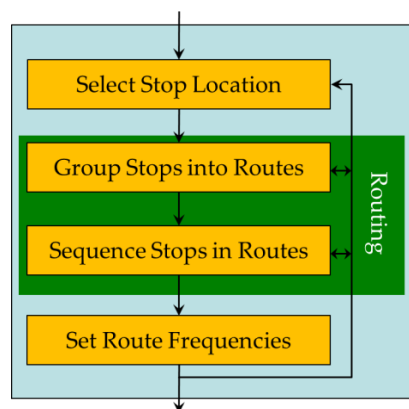


Figure 2.1 The Transit Network Design Problem

Defining equity can be a challenging task as the concept of equity is subject to a person's political and philosophical views. Three types of equity will be considered in this paper: horizontal (or spatial) equity, vertical (or social) equity, and intermodal equity. The goal of horizontal equity is to achieve equity of inputs. All people are treated the same without consideration for existing inequalities or differing levels of need. The vertical equity perspective considers these existing inequalities and treats people in accordance with their level of need, striving for equity of outcomes over equity of inputs. These two perspectives often conflict with one another. For example, if transit planners design a transit system to prioritize the needs of people who cannot afford private vehicles, they are not treating all members of the community equally. For further discussion of horizontal and vertical equity philosophies, as well as others, from the perspective of a transportation planner, please refer to Khisty (1996). Intermodal equity provides a different perspective. A transit planner taking an intermodal equity approach will attempt to ensure equal levels of mobility across various modes, particularly public and private modes, rather than an equitable distribution of service among the population. This research experiments with designing transit routes to optimize different definitions of equity based on the horizontal, vertical, and/or intermodal equity perspectives.

Three models will be explored in this chapter: a single route model, a multiple route model without a stop grouping component, and a multiple route model with a stop grouping component. The single route model (EqTSP) focuses solely on stop sequencing. The single route model is an intentionally simplistic model designed to test the nine proposed equity objective functions. Testing the objective functions on this model allowed some of them to be eliminated prior to

application on the more complex models. The multiple route model (MEqTSP) without stop grouping expands on the single route model and allows users to optimize the stop sequences of multiple routes simultaneously. In both models, stops have been assigned to routes a priori. Stops may not be assigned to a new route in the course of finding a solution. The primary purpose of developing the multiple route model without stop grouping was to expand the solution method, a genetic algorithm, to have the ability to store and process multiple route solutions and account for transfers. The third and final model is a multiple route model (EqTNDP) with stop grouping. In this model, stops are not assigned to specific routes before the model is implemented. The solution method implements the stop sequencing and stop grouping components of the model simultaneously.

2. Defining Equity

In this chapter, nine inequity minimizing objective functions are considered. All of the objective functions minimize a function of travel time rather than maximizing a function of access.

Though travel time is a less comprehensive measure than access, travel time maintains properties such as continuity and convexity that access does not, making it easier to use in the formulation of objective functions. This formulation can be updated in the future to reflect a broader definition of access.

The nine objection functions considered in this paper are shown below:

$$\text{Minimize } Z_1 = \sum_{i \in I} \sum_{j \in I} t_{ij} D_{ij} \quad (1)$$

$$\text{Minimize } Z_2 = \sum_{i \in I} \sum_{j \in I} t_{ij} D_{ij} E_i \quad (2)$$

$$\text{Minimize } Z_3 = \sum_{i \in I} \sum_{j \in I} T_{ij} D_{ij} \quad (3)$$

$$\text{Minimize } Z_4 = \sum_{i \in I} \sum_{j \in I} T_{ij} D_{ij} E_i \quad (4)$$

$$\text{Minimize } Z_5 = \left[\frac{1}{\bar{T}} \left[\frac{1}{|i||j|} \sum_{i \in I} \sum_{j \in I} D_{ij} (T_{ij} - \bar{T})^2 \right]^{1/2} \right] \quad (5)$$

$$\text{Minimize } Z_6 = \left[\frac{1}{\bar{T}} \left[\frac{1}{|i||j|} \sum_{i \in I} \sum_{j \in I} D_{ij} E_i (T_{ij} - \bar{T})^2 \right]^{1/2} \right] \quad (6)$$

$$\text{Minimize } Z_7 = \left[\frac{1}{T^*} \left[\frac{1}{|i||j|} \sum_{i \in I} \sum_{j \in I} D_{ij} (T_{ij} - T^*)^2 \right]^{1/2} \right] \quad (7)$$

$$\text{Minimize } Z_8 = \left[\frac{1}{T^*} \left[\frac{1}{|i||j|} \sum_{i \in I} \sum_{j \in I} D_{ij} E_i (T_{ij} - T^*)^2 \right]^{1/2} \right] \quad (8)$$

$$\text{Minimize } Z_9 = \max_{\forall i, j \in I} (D_{ij} T_{ij}) \quad (9)$$

Where $T_{ij} = t_{ij} - c_{ij}$ and $\bar{T} = \frac{1}{|i||j|} \sum_{i \in I} \sum_{j \in I} T_{ij}$

t_{ij} is the travel time by transit from origin i to destination j . Transit travel time between origin and destination nodes is largely dependent upon the order in which the nodes are visited. The order of the nodes (or stops) and the associated arcs connecting these nodes are the decision variables. D_{ij} is the proportion of the total system demand making trips from origin i to

destination j . For the purpose of this research, D_{ij} does not account for income or any other demographic variable related to vulnerable populations. E_i is the proportion of the total protected population residing at origin i . c_{ij} is auto travel time from origin i to destination j . In this study, auto travel time is assumed to be the travel time along the shortest path between two nodes which assumes no congestion in the network. In future research, a traffic assignment model could be incorporated into this model to improve the accuracy of auto travel times. However, this is beyond the scope of this research. T_{ij} is the difference between auto travel time and transit travel time between i and destination j , \bar{T} is the mean difference between transit and auto travel time between all node pairs, and T^* is a constant representing a target difference in the travel time between car and transit.

The first two objective functions (Z_1, Z_2) minimize the sum of travel time by transit between all origin-destination (O-D) pairs. Objective function 1 is also the mean trip travel time. The second two objective functions (Z_3, Z_4) minimize the sum of the difference between travel time by transit and by auto between all O-D pairs. Objective function 3 is also the mean difference between trip travel time by transit and by car. The third pair of objective functions (Z_5, Z_6) minimize the coefficient of variation (COV) in the difference between travel time by transit and by auto. By minimizing the COV, this objective function pushes for all riders to experience the same difference in service between transit and auto travel rather than trying to minimize the system wide difference across modes. The next two objective functions (Z_7, Z_8) also minimize the COV in the difference between travel times by transit and car. However, rather than calculate variation from the mean, these objective functions calculate variation from a target value for the difference between trip travel times by mode. By calculating the COV from a target, an

exogenous variable, rather than the mean, an endogenous variable, the objective functions will avoid the problem of non-convexity, as discussed in a paper by Boyles (2015). For this objective function to work properly, the user must set an optimistic target. The final objective function (Z_9) minimizes the maximum difference between transit and auto travel times for all O-D pairs. This effectively minimizes the worst case scenario.

The objective functions including the E_i factor (Z_2, Z_4, Z_6, Z_8) are measures of vertical equity which explicitly prioritize the demand of protected populations, while the others (Z_1, Z_3, Z_5, Z_7, Z_9) are measures of horizontal equity and do not prioritize any population. The final seven objective functions ($Z_3, Z_4, Z_5, Z_6, Z_7, Z_8, Z_9$) also incorporate an intermodal equity perspective by minimizing functions of the difference between transit and auto travel.

These inequity minimizing equations will be used as objective functions in the models described in the remaining sections of the chapter.

3. Single Route Model (EqTSP)

3.1 Methodology

3.1.1 Model Formulation

Determining which sequences of stops within routes will optimize system equity is a combinatorial optimization problem. The various quantitative definitions of equity will serve as objective functions. This research begins by experimenting with the various objective functions on a simple scenario; a single transit route which must visit each stop once and only once before returning to the first stop. This scenario is very similar in structure to the classic combinatorial

optimization problem commonly known as the Traveling Salesman Problem (TSP). The TSP seeks a minimum cost tour of a set of nodes, assuming fixed arc costs. Each node is visited exactly once. Miller et al. (1960) formulated the TSP as shown below:

$$\text{Minimize } z = \sum_i \sum_j \tau_{ij} x_{ij} \quad (10)$$

Where τ_{ij} is the cost of travel from node i to node j and x_{ij} is a binary variable which equals 1 if a connection exists and 0 otherwise.

The objective function is constrained by the following:

$$\sum_{j:j \neq i} x_{ij} = 1 \quad \forall i \in I \quad (11)$$

$$\sum_{i:i \neq j} x_{ij} = 1 \quad \forall j \in I \quad (12)$$

$$u_i - u_j + |I|x_{ij} < |I| - 1 \quad (13)$$

Equations (11) and (12) ensure that each stop is visited once and only once. Equation (13) is a subtour elimination constraint where u_i is a real number associated with node i . This ensures that the solution will be a single continuous route rather than multiple disconnected cycles. The only difference between the TSP and the proposed single route equity optimizing model is the objective function. The equitable variant of the TSP will be referred to as EqTSP throughout the remainder of this dissertation

3.1.2 Overview of Solution Methodology

Due to the complexity of the TSP and the TNDP, exact solution methods can be very time consuming and therefore impractical to apply to large networks. Potvin (1996), Rego and Glover (2002), and Leguizamon et al (2007), among others, have shown that heuristic and metaheuristic methods are preferred to exact methods for solving NP-hard and NP-complete problems, such as the TSP, in large instances. While the experiments conducted in this paper are on relatively small networks, this method is being developed with the necessity of future expansion in mind.

A genetic algorithm (GA), a type of metaheuristic, was developed to find good solutions to the proposed models. To use the terminology developed by Blum and Roli (2003), genetic algorithms are population based, evolutionary algorithms. GAs are based on Darwinian natural selection. Haupt and Haupt (2004) describe classic genetic algorithms as creating “chromosomes” (solution arrays of binary decision variables) by randomly turning “genes” (individual decision variables) on and off. Chromosomes representing infeasible solutions are thrown out while the remaining chromosomes compete to determine which solutions are the fittest. The fittest solutions are then subject to crossover and mutation functions, allowing them to create new solutions, or “offspring.”

When applying GAs to TSPs, researchers typically choose to encode the chromosomes as permutations rather than binary arrays. Permutations are a more natural, more compact way to represent TSP solutions as chromosomes. Consider example figure 2.2 below:

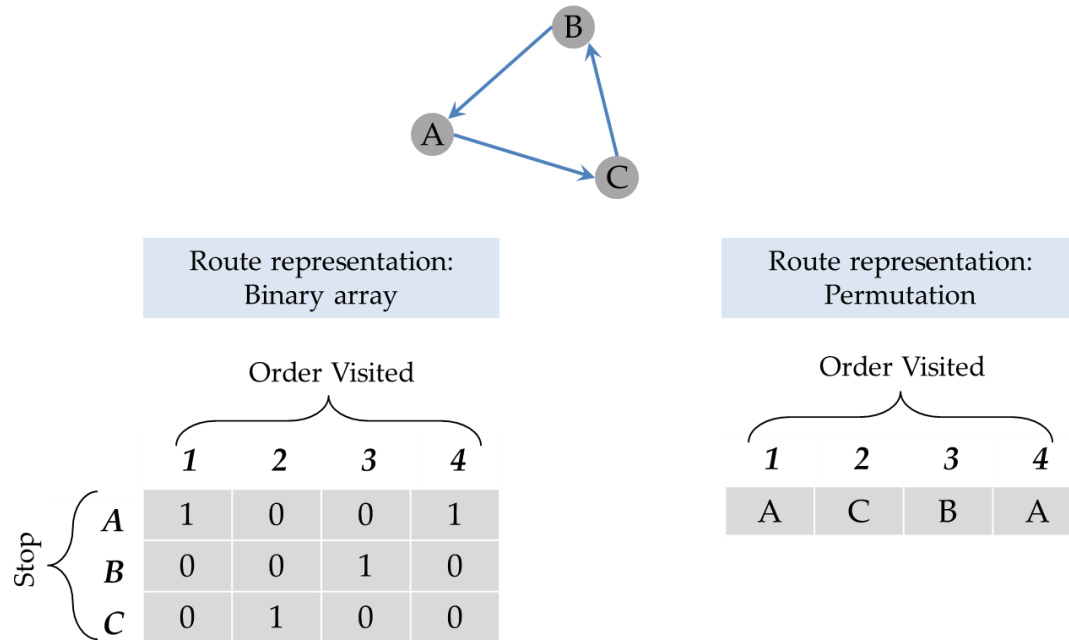


Figure 2.2 Example route and chromosomal representations.

Figure 2.2 shows the difference between encoding a solution as a binary array versus encoding it as a permutation. The binary array representation requires a two dimensional matrix of size $N \times N+1$ (where N is the number of nodes) while the permutation only requires a one-dimensional matrix of size $N+1$. The genetic algorithm used in this research encodes solutions (routes) as permutations.

The flowchart below outlines the broad steps of the GA developed to solve this model. Each of the steps will be described in following paragraphs. A more detailed pseudocode describing the algorithm will be provided in Chapter 3.

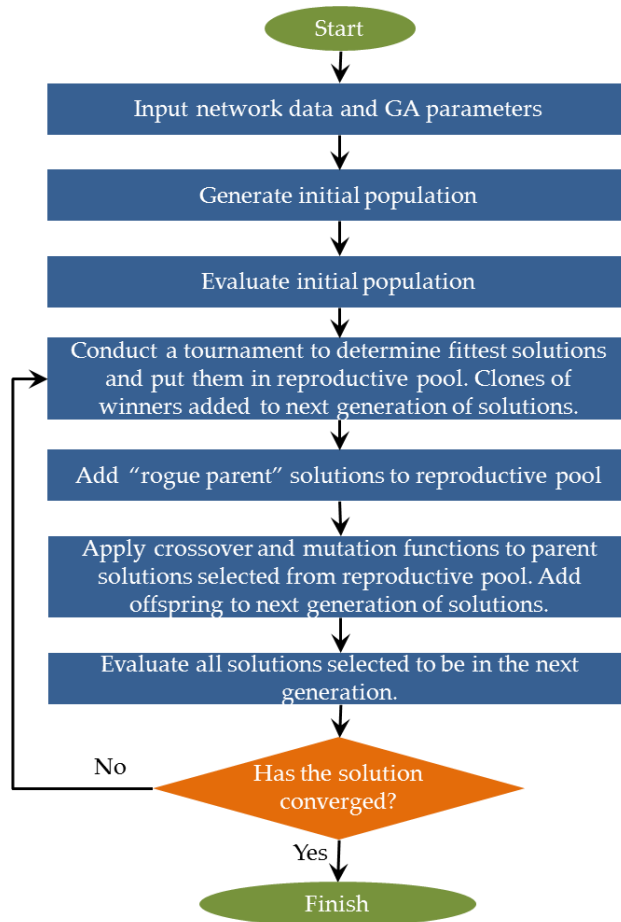


Figure 2.3: Flow chart of genetic algorithm.

Inputs

The GA developed for this research requires knowing the travel time along the shortest path between all O-D pairs, as well as demand and protected population parameters. The Floyd-Warshall algorithm (Ahuja, Magnanti, and Orlin 1993) was used to find the travel time along the shortest path between all O-D pairs from network configuration data. The GA also requires the user to set the following parameters: initial population size, tournament size, number of rogue parents added to the reproductive pool, mutation rate, and number of generations with identical best solution objective function values to achieve convergence. For the purpose of this

application, the parameters were set to the following values: $p = 300$, $t = 6$, $numRogues = 5$, $mutationRate = 0.1$, and $genConverge = 40$.

Generating Initial Solutions

Generating good initial solutions is important to converging on an optimal or near-optimal final solution in a reasonable amount of time. The first step to generating a new solution is randomly selecting a starting node and marking it as the current node, n_i . Each of the remaining nodes, n_j , is then assigned a probability of being selected as the next node, $p_{sel,j}$, using equation 14 below:

$$p_{sel,j} = \frac{\frac{1}{c_{ij}^\alpha}}{\sum_j \frac{1}{c_{ij}^\alpha}} \quad (14)$$

The probability of selecting a node is inversely proportional to the travel time between it and the previous node. The α parameter allows users to give different importance to the nearness of a node; the higher the value of α , the greater the import given to nearness. In this analysis, α was set equal to 1. This process is repeated until all nodes have been included in the route. The final node of the route is then connected to the first node of the solution via the shortest path, completing the TSP solution.

Evaluating Solutions

A chromosome's, or solution's, fitness is determined by the value of the selected inequity objective function. In the case of the previously described objective functions, which seek to

minimize inequity, solutions with lower objective function values equate to chromosomes with more fit solutions.

Determining Fit Solutions

A tournament with rounds of size t is used to find fit solutions. The best solution in each round of the tournament is added to the reproductive pool while the other solutions are tossed out. A clone of each of the tournament winners will be automatically included in the next generation of solutions, alongside the offspring generated in the following steps.

Reproduction: Crossover, Mutation, and Rogue Parents

Encoding chromosomes as permutations rather than binary arrays required the development of specialized crossover and mutation procedures. This GA uses a modified version of the EAX crossover function developed in Nagata and Kobayashi (1999) for TSP applications and in Nagata (2007) for broader vehicle routing applications. The EAX crossover generates offspring from parent solutions in the reproductive pool. Figure 2.4 below shows an example of the EAX applied to a simple six node network.

Two parent solutions are selected from the reproductive pool at random. The parents are used to generate an AB-cycle, a closed loop with alternating arcs coming from different parents. From this AB-cycle, an offspring solution is generated. Not all parent pairings will be able to produce offspring using this method. Whether or not a pair of parents can reproduce, they remain in the reproductive pool. This GA uses non-monogamous reproduction instead of establishing monogamous parent pairs to create a more diverse set of offspring and address the issue of parent pairs that cannot produce offspring. Many parent pairs simply cannot generate multiple offspring

solutions using this crossover method. This may be due to the particular configuration of the networks used in this portion of the research or a result of the way initial solutions are generated.

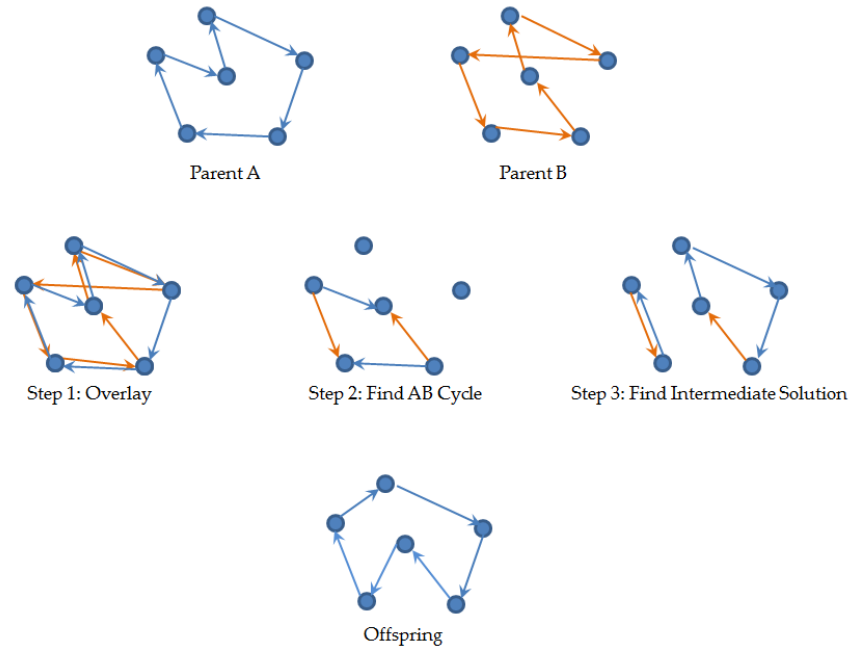


Figure 2.4: Example Application of Edge Assembly Crossover

Using the EAX crossover function alone generally led to suboptimal solutions because the algorithm became trapped at local minima. To maintain a more diverse population of solutions and avoid becoming stuck at local minima, the GA developed in this research uses two techniques: the addition of randomly generated solutions to the reproductive pool each iteration and a standard mutation function to make random alterations to a small proportion of the offspring. The randomly generated solutions will be referred to as rogue parents throughout the rest of the dissertation. To the knowledge of the authors, this is a novel approach to maintaining population diversity in GAs. This approach will be validated and explored more thoroughly in Chapter 3. The mutation function selects a random portion of the route and randomly inserts it

into another part of the route. The length of portion, where the selected portion is inserted, and whether or not the portion is inversed are all selected at random. The mutation function cannot alter more than 20% of a route.

Converging on a solution

The process of finding the fittest solutions, allowing them to reproduce, and evaluating the new generation of solutions is repeated until the algorithm converges on a solution. The algorithm achieves convergence when the best solution has not improved for the specified number of generations (*genConverge*).

Before applying this GA to the EqTSP, it was tested on the TSP for the Sioux Falls and Willimantic test networks described in the following section and compared to the known optimal solutions found using a branch and cut method. For both networks, the GA found the optimal solution 100 out of 100 times. A more thorough investigation into the accuracy and efficacy of the GA for TSPs and the single route model can be found in Chapter 3. Chapter 3 also includes experiments on the effect of the various parameters and algorithm structures.

3.2 Test Networks and Experiments

The GA discussed in the previous section is used to find good solutions to the proposed formulations of the EqTSP on two test networks, Sioux Falls and Willimantic, shown in figure 2.5 below. Sioux Falls is a commonly used test network with 24 nodes and 76 arcs (Transportation Network Test Problems, 2013). The arc travel times for the network were assumed to be the travel time under user equilibrium conditions. The Willimantic network, loosely based on the WRTD-operated Willimantic City bus route operating in Willimantic, CT,

is a network with 28 nodes and 93 arcs (WRTD, 2015). Arc travel times were estimated under free flow travel conditions using Google maps. Willimantic is a rare real-world example of single circulator route operating in near isolation, making it an excellent candidate for these experiments.

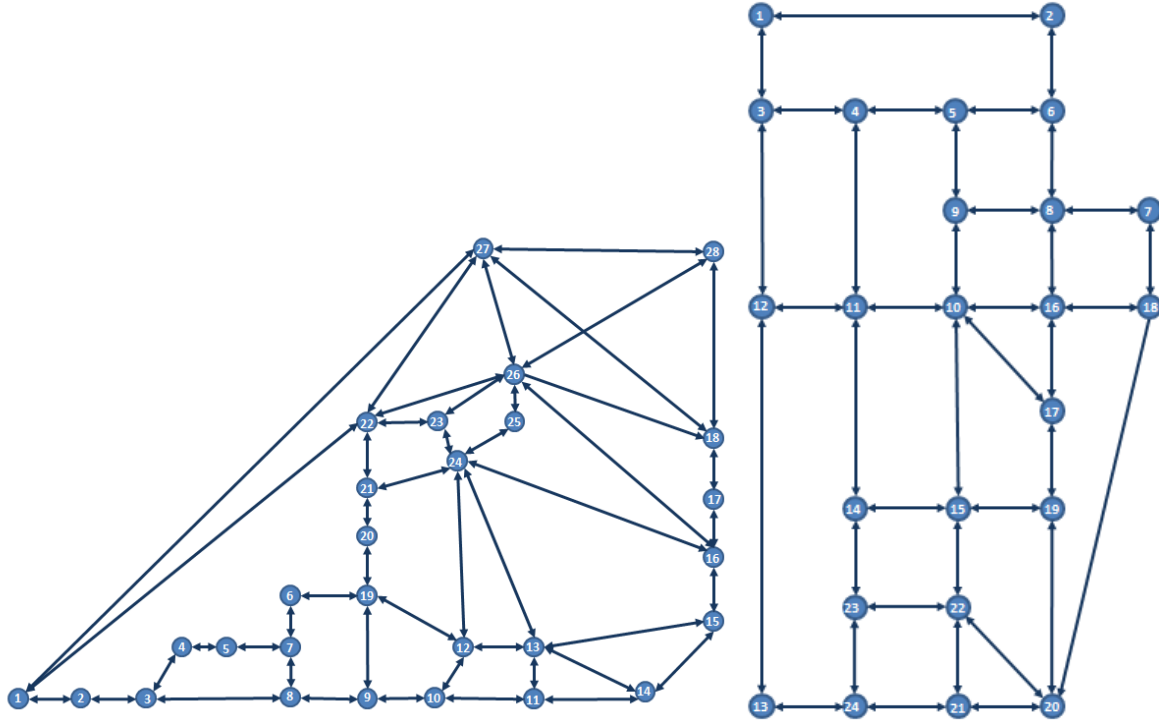


Figure 2.4 (a) Willimantic Network (b) Sioux Falls Network

In these experiments, the nodes were treated as if they were directly connected to every other node in the network by an arc with a cost equivalent to the shortest path cost. This allows for a bus to pass by a stop, possibly on the other side of the road, without stopping the vehicle to pick up passengers. Two demand scenarios (D_{ij} inputs) and two population distribution scenarios (E_i inputs) were generated for each network. One demand and one equity scenario were randomly generated while the other demand and equity scenarios were structured intentionally unevenly to

highlight features of the objective functions. The T^* values for objective functions Z_7 and Z_8 were set to the minimum mean difference between trip travel time by bus and by car as found using Z_3 on the various demand scenarios. Setting optimistic targets is necessary for Z_7 and Z_8 to produce desirable solutions (Boyles 2015).

3.3 Discussion of Results

Thirty replicate runs were conducted for each of the equity objective functions. The series of figures below show some of the best solutions to the objective functions. Figures 2.6 (a-d) show the best solutions found for the single route model on the Sioux Falls Network under demand scenario 2 and equity scenario 5. Demand scenario 2 was intentionally designed to create an uneven distribution of demand across the network and equity scenario 5. Objective function Z_9 did not converge on a single solution and therefore is not included in any of the figures below.

The nodes shown in green produce a large proportion of the system demand. The nodes shown in red attract a large proportion of the system demand. Nodes nested in the yellow boxes contain have a large vulnerable population.

Due to the location of vulnerable populations relative to general demand, the solutions shown in Fig 2.6 are particularly extreme. The distribution of trips by people in vulnerable population centers is almost the exact opposite of the trip distribution of the general population. This led to the common situation of routes including similar arcs but in opposite directions.

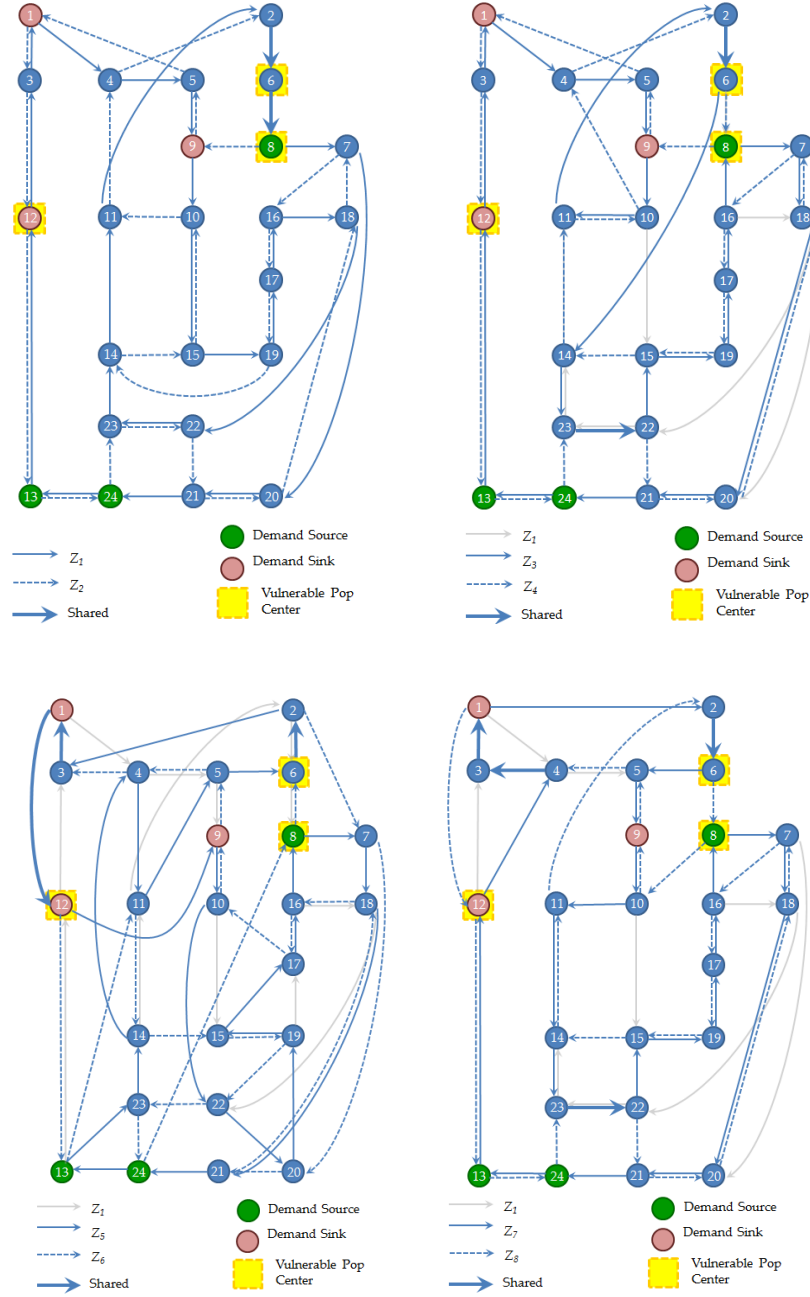


Figure 2.6 Best Solutions, Demand Scenario 2, Equity Scenario 5 (a) Z_1 and Z_2 (b) Z_3 and Z_4 (c) Z_5 and Z_6 (d) Z_7 and Z_8

The solutions shown in the table below are the best solutions, as determined by the respective objective function values, found in the 30 runs. Tables 1 and 2 below summarize the results of

the experiments on the Willimantic and Sioux Falls networks, respectively. The “D” column indicates which demand scenario, or set of origin-destination demand data sets, was used in the experiment. For each network, one demand scenario was randomly generated (demand scenarios 4 and 5). The other demand scenarios (2 and 3) were intentionally designed to highlight the differences between demand weighted and equity weighted objective functions. The “E” column indicates which population distribution scenario, or set of origin protected population data, was used in the experiment. If no value is indicated in this column, the function was not weighted by E_i . All four population scenarios (2, 3, 4, and 5) were randomly generated. The “Average TT” column gives the average travel time for trips by transit on the best route for the indicated objective function and population distribution, \bar{t} . The “Average Difference in TT” gives the average difference in trip travel time between transit and car for the indicated objective function and population distribution, \bar{T} . For ease of interpretation, the table also includes a column assigning each row a reference number. These reference numbers, found in the second column on the left, will be used in the discussion of results.

Table 2.1 shows that the objective functions minimizing trip travel time (Z_1, Z_2) and the objective functions minimizing the difference in trip travel time by mode (Z_3, Z_4) resulted in similar, and in many cases identical, solutions. For example, row 2 which displays results for the objective function minimizing trip travel time under demand scenario 2 and equity scenario 2 contains identical results to row 8 which displays results for the objective function minimizing the difference in trip travel time by modes for the same scenarios. There were no differences in the average travel time or average difference in travel time on the Willimantic network for any of the population distribution scenarios for objective functions 1-4. The largest increase in average travel time between objective functions 1-4 on the Sioux Falls network was only 3.8% (demand

Table 2.1 Average Travel Time (TT) and Average Difference in Travel Time (TT) for best routes on the Sioux Falls and Willimantic networks.

		Sioux Falls Network				Willimantic Network			
		D	E	Avg TT (\bar{t})	Avg Difference in TT (\bar{T})	D	E	Avg TT (\bar{t})	Avg Difference in TT (\bar{T})
Minimizing Trip Travel Time (Obj. Function 1 and 2)	1	2	-	74.60	47.79	3	-	5.06	3.22
	2	2	2	74.60	47.79	3	3	5.06	3.22
	3	2	5	95.10	68.29	3	4	5.06	3.22
	4	5	-	45.10	21.21	4	-	4.53	2.72
	5	5	2	48.26	24.37	4	3	4.59	2.78
	6	5	5	141.72	117.83	4	4	4.55	2.74
Minimizing Difference in Trip Travel Time across Modes (Obj. Function 3 and 4)	7	2	-	74.60	47.79	3	-	5.06	3.22
	8	2	2	74.60	47.79	3	3	5.06	3.22
	9	2	5	95.10	68.29	3	4	5.06	3.22
	10	5	-	45.10	21.21	4	-	4.53	2.72
	11	5	2	46.40	22.50	4	3	4.59	2.78
	12	5	5	141.86	117.97	4	4	4.55	2.74
Minimize COV of Difference in Travel Time (Obj. Function 5 and 6)	13	2	-	311.34	284.53	3	-	23.24	21.39
	14	2	2	225.99	199.17	3	3	19.22	17.37
	15	2	5	194.56	167.75	3	4	13.21	11.36
	16	5	-	282.89	258.99	4	-	22.73	20.92
	17	5	2	187.87	163.97	4	3	16.33	14.53
	18	5	5	261.89	238.00	4	4	15.13	13.32
Minimize COV from Target Difference in Travel Times (Obj. Function 7 and 8)	19	2	-	83.91	57.10	3	-	5.43	3.58
	20	2	2	86.95	60.14	3	3	5.06	3.22
	21	2	5	95.19	68.38	3	4	6.68	4.84
	22	5	-	52.02	28.12	4	-	5.31	3.50
	23	5	2	57.96	34.06	4	3	5.41	3.60
	24	5	5	137.23	113.33	4	4	5.34	3.53
Minimize Maximum Difference (Obj 9)	25	2	-	79.59- 103.55	52.78- 76.74	3	-	7.51- 7.63	5.66- 5.79
	26	5	-	45.42- 51.20	21.53- 27.30	4	-	2.42- 3.22	2.74- 3.62

5, equity 2) while the largest increase in average difference in travel time between modes was 7.7% (demand 5, equity 2). The increase in average travel time \bar{t} is calculated by comparing the \bar{t} value of an objective function on a specific network for a given demand and equity scenario to the \bar{t} value for the travel time minimizing objective functions (Z_1, Z_2) on the same network for the same demand and equity scenarios. For example, the 3.8% increase in average travel time was found by comparing the Sioux Falls \bar{t} values in rows 5 and 11. A similar procedure was used to find the increase in average difference in travel time between modes \bar{T} except the objective functions minimizing the difference in travel time between modes (Z_3, Z_4) are now used as the basis of comparison. The differences between the solutions of objective functions 1-4 may have been more pronounced if a more complex measure of access had been used in place of travel time. Another reason for the similarity may be the assumption that travel time by auto is always equivalent to the travel time along the shortest path between an origin destination pair.

The solutions for the objective functions minimizing the coefficient of variation (Z_5, Z_6) varied substantially from each other and from the best solutions found using the other objective functions. These solutions increased the average trip travel time by between 84% (Sioux Falls, demand 5, equity 5) and 527.2% (Sioux Falls, demand 5) and increased the average difference in travel time by mode by between 102.0% (Sioux Falls, demand 5, equity 5) and 1121.3% (Sioux Falls, demand 5). The COV minimizing objective functions are clearly generating undesirable and unreasonable solutions. This happens because the objective functions push for solutions in which every rider experiences the same difference in travel time between bus and auto without controlling for the magnitude of that difference, creating a race to the bottom scenario. The results of these experiments suggest that minimizing the COV of the difference between bus and auto travel is not a reasonable or appropriate objective function for the EqTSP and therefore,

more broadly, this pair of objective functions is not recommended for use in equitable route design models. This aligns with the recent finding by Boyles (2015) that variance minimizing objective functions are non-convex and therefore unsuitable for use in an optimization program.

In contrast, formulating objective functions to calculate COV from a set goal, an exogenous variable, rather than the mean, an endogenous variable, created more reasonable solutions and eschewed the race to the bottom scenario. For these experiments, the set goal was the minimum average difference between modes as calculated for the best solutions found for objective 3. The solutions found using the objective functions minimizing the COV from a set goal (Z_7, Z_8) experienced an increase in average travel time between 0% (Willimantic, Demand 3, Equity 3) and 32.1% (Willimantic, Demand 3, Equity 4). The increase in the average difference in travel time ranged from 0% (Willimantic, Demand 3, Equity 3) to 50.5% (Willimantic, Demand 3, Equity 4). These increases seem much lower than those found for the COV calculated from the mean. However, only the engineer or planner of a particular transit system will be able to decide whether she or he thinks these increases in trip travel time and the difference in trip travel time between modes are worth the improvements to this particular definition of equity.

The final objective function which minimized the maximum difference between transit and car travel (Z_9) was unable to converge on a single solution. While it was able to converge onto a single value, many alternative routes shared this same objective function value. One of the demand scenarios for Sioux Falls produced 26 unique solutions with the same value for the minimax objective function (Z_9). Considering that the 26 unique solutions were found in only 30 runs, it seems likely that even more unique solutions exist with this value. The solutions had one or more critical sub-sequences in common, depending on the demand scenario. The order of the

stops outside of the critical sub-sequences varied substantially. Without a secondary objective to guide the sequencing of stops outside of the critical sub-sequences, this function is not recommended for use as an objective in equitable route design models. This function, however, may be valuable as a constraint.

The effects of including E_i , the coefficient representing the proportion of the protected population served by zone i , are difficult to discern from Table 2.1. Figure 2.7 below highlights the differences between objective functions which include E_i and those that do not. Figure 2 contains 4 plots, one for each pair of objective functions. The y axis of the graph is the percent improvement in the equity-weighted objective function value when E_i is included, as compared to when it is not included. The x axis of the graph is the opposite, representing the percent improvement to the unweighted objective function value when E_i is not included, as compared to when it is included. This can also be interpreted as the cost, in terms of unrealized benefits to the general population, of including parameter E_i in the objective function. The resulting plots allow for a comparison of the benefits and costs of including the E_i coefficient. If a point lies above the neutral line, shown as a gray line in the plots below, then it is more beneficial to the vulnerable population to include E_i than it is costly to the overall population. A point lying below the neutral line indicates the opposite, that including E_i is more costly to the overall population than it is beneficial to the vulnerable population.

From Table 1 and Figure 2.7, it is clear that objective functions including parameter E_i found different solutions from the objective functions that did not include E_i . However the effects of this inclusion were inconsistent. In some cases, including E_i provided more benefits to the vulnerable population than costs to the overall population, while in others it did not. The one

exception to this is the pair of objectives minimizing COV which were improved by E_i in most cases. However, due to the previously discussed issues with these objective functions, it is difficult to interpret the meaning of this finding. Whether or not it is appropriate to use the E_i parameter will depend on the specific situation and the goals of the transit planner.

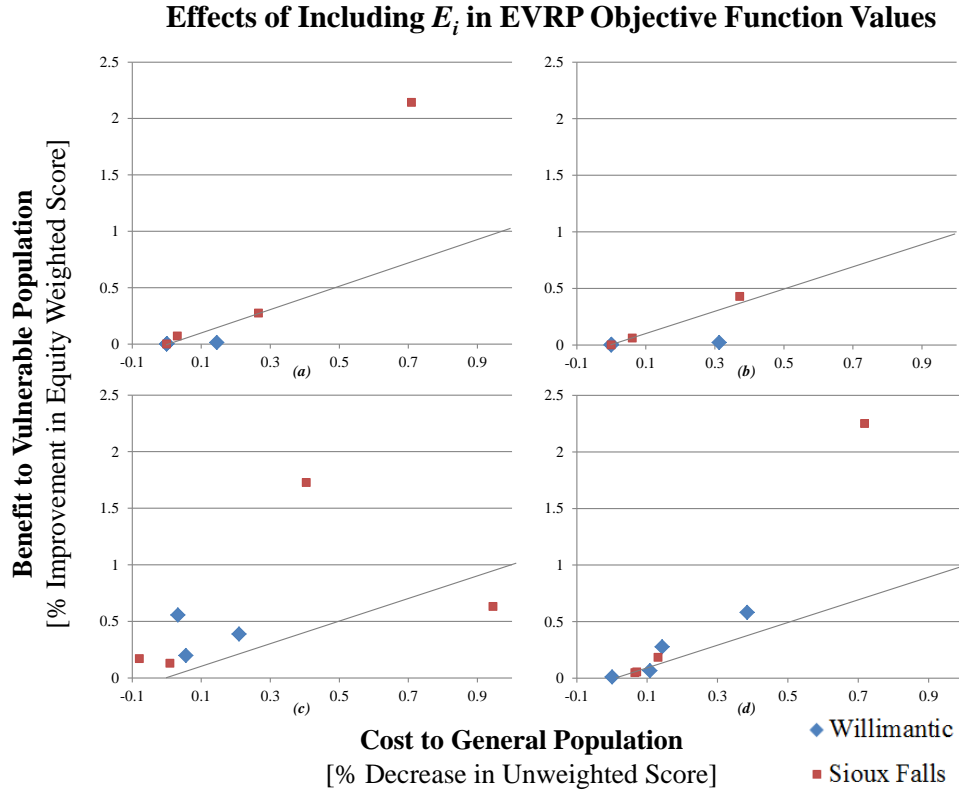


Figure 2.7: Comparison of Costs and Benefits of including E_i in the objective functions²

(a) Minimize Trip Travel Time (Z_1, Z_2) (b) Minimizing Difference in Trip Travel Time across Modes (Z_3, Z_4) (c) Minimize COV of Difference in Travel Time (Z_5, Z_6) (d) Minimize COV from Target Difference in Travel Times (Z_7, Z_8)

² Note that some extreme points are not shown in the figure. They do not add to or take away from the interpretation of the figure.

In its current form, including the E_i parameter alters the objective function such that only members of the protected population are considered in the solution. For real world systems, planners may want to alter the E_i parameter so that it prioritizes, but does not exclusively plan for, protected populations. Another possibility is to use demand data which has already accounted for the differing needs of different communities rather than accounting for the protected population separately. This will have the same effect as the E_i parameter but is applied at a different stage of the transit planning process.

4. Multiple Route Model (without stop grouping)

The primary purpose of developing the multiple route model without stop grouping (MEqTSP) was to create the data structures and methods necessary to handle multiple route networks using the proposed solution method. This model is presented as an intermediate step between the first model, the single route model, and the final model, a multiple route model with a stop grouping component.

4.1 Methodology

4.1.1 Model Formulation

The primary difference between the single route model and the multiple route model without stop groupings is the calculation of transit travel time between origin and destination nodes. In the case of the single transit route, exactly one path exists between each origin- destination pair and the path does not require transferring between routes. However, in a multiple route transit system, several paths may exist between origin-destination pairs and the paths may require transferring between routes. In the multi route objective function formulations, t_{ij} is replaced by

the shortest path transit travel time between origin i and destination j . The shortest path travel time includes a 15 minute penalty for transfers between routes. While researchers agree that transfers should incur a penalty cost, there does not appear to be a consensus on exactly how high that cost should be set. Transfer penalties between 15 and 20 minutes have been used by several researchers working in the field of transit accessibility and modelling (Currie 2004; Mamun et al. 2014).

The model constraints also need to be adjusted for the multiple route model. Each route in the multiple route system is required to adhere to the standard TSP constraints, meaning each route will be a single, circulator route. The adjusted model shown below:

Minimize $Z =$ Inequality

Subject to:

$$\sum_{i; \forall i \in R^k} x_{ij}^k = 1 \quad \forall j \in R^k; \quad \forall k \in K \quad (15)$$

$$\sum_{j; \forall j \in R^k} x_{ij}^k = 1 \quad \forall i \in R^k; \quad \forall k \in K \quad (16)$$

$$u_i - u_j + |I|x_{ij}^k < |I| - 1 \quad \forall i, j \in R^k, i \neq j; \quad \forall k \in K \quad (17)$$

$$x_{ij}^k = \text{Binary}$$

The objective functions remain largely unchanged with the exception that demand and travel time are no longer associated with individual nodes but with demand zones. Demand zones are

sets of stops in close proximity to each other which can be thought of as serving the same demand. Further details on demand zones will be provided later in the chapter. Adjusting the model constraints for multiple routes requires defining a route set K which will be represented by index k . In this particular model, stops (nodes) were assigned to route(s) k a priori. Set R^k represents the set of nodes assigned to route k and maintains the following properties: $R^k \subseteq I$ and $\bigcup_k R^k = I$.

4.1.2 Overview of Solution Methodology

The MEqTSP can be solved using the genetic algorithm described in section 3.1.2 with minor adjustments. The overall flow of the algorithm remains the same (see figure 2.3). Some of the inputs, however, have changed. As previously mentioned, demand is now associated with demand zones rather than individual stops or nodes. A demand zone may include two or one stop(s). For the purpose of this research, stops located within 50 meters of each other are assumed to serve the same demand even though they are treated as separate stops for the purposes of determining the route path and calculating route travel times. This distance was selected to capture stops on opposite sides of a street and assumes that crossing the street does not impose a serious impediment to able-bodied riders. In addition to a differently structured demand matrix, the modified GA also requires the user to input the set of nodes R^k assigned to each route k and a file which maps stops to their respective demand zones.

Because stops cannot switch route assignments, each route in the multiple route model can be treated independently for the purposes of the reproductive functions. This means the only difference in the reproduction phase is that when two parent solutions are selected for crossover,

the crossover is actually performed multiple times; once on each route in the solution. Crossover can only occur between routes containing the same set of stops.

As previously mentioned, one of the differences between the single route and multiple route models is that several paths may exist between stops, or demand zones, in the multiple route model. Several approaches were considered for identifying the shortest path travel times on the set of routes represented by a solution. Ultimately, the best method for topology of the test network was to first identify the shortest path between all directly connected zone pairs and then to enumerate all of the paths between zone pairs requiring route transfers. This makes an implicit assumption that paths without transfers will always have a lower cost than those with transfers. While this may not always be the case, it was true for the test network and for the chosen transfer cost of 15 minutes. In the future, it would be appropriate to replace this shortest path approach for determine transit travel time with a transit assignment model but this was outside the scope of this research. The method also assumes that all zones are reachable by transit with only one transfer. Again, this happens to be true for the test network but is unlikely to be true for other networks. This assumption will be addressed in the final multiple route model with stop grouping.

4.2 Test Networks and Experiments

The multiple route model was tested on a subset of the University of Connecticut's (UConn) shuttle bus system. Figure 2.8 below shows the relevant portion UConn's road network with bus stops. The five selected routes connect on-campus student housing and commuter parking lots to the core of UConn's campus which includes academic buildings, laboratories, offices, and other facilities. The routes connecting to secondary campuses and off-campus apartments

were not considered in these experiments. The core of UCONN's campus is easily walkable for an able bodied person. Most streets shown in the map below have frequent pedestrian crossings in which pedestrians have the right of way. The exceptions are the major roads labelled with black arrows which have less frequent pedestrian crossings and signalized crossings at intersections.

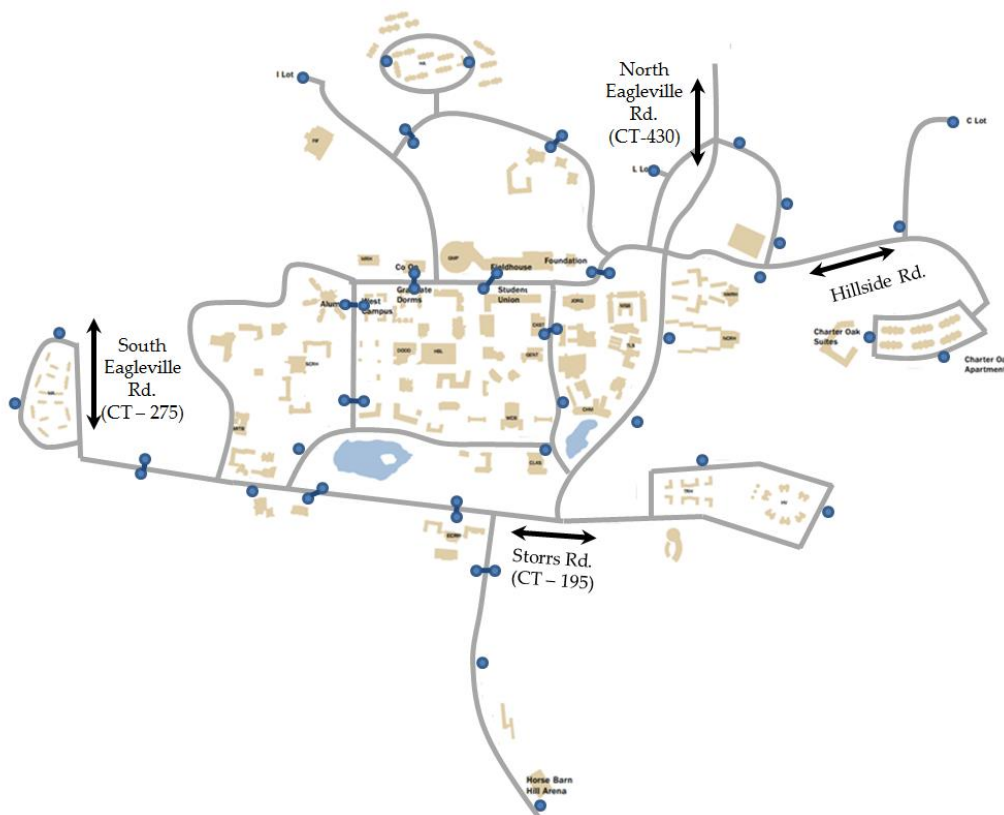


Figure 2.8: UCONN Bus stops and road network

The stops, routes, and road network associated with the five selected routes will be referred to as the UCONN network throughout the remainder of the chapter and are shown in figures 2.9(a-e) below:

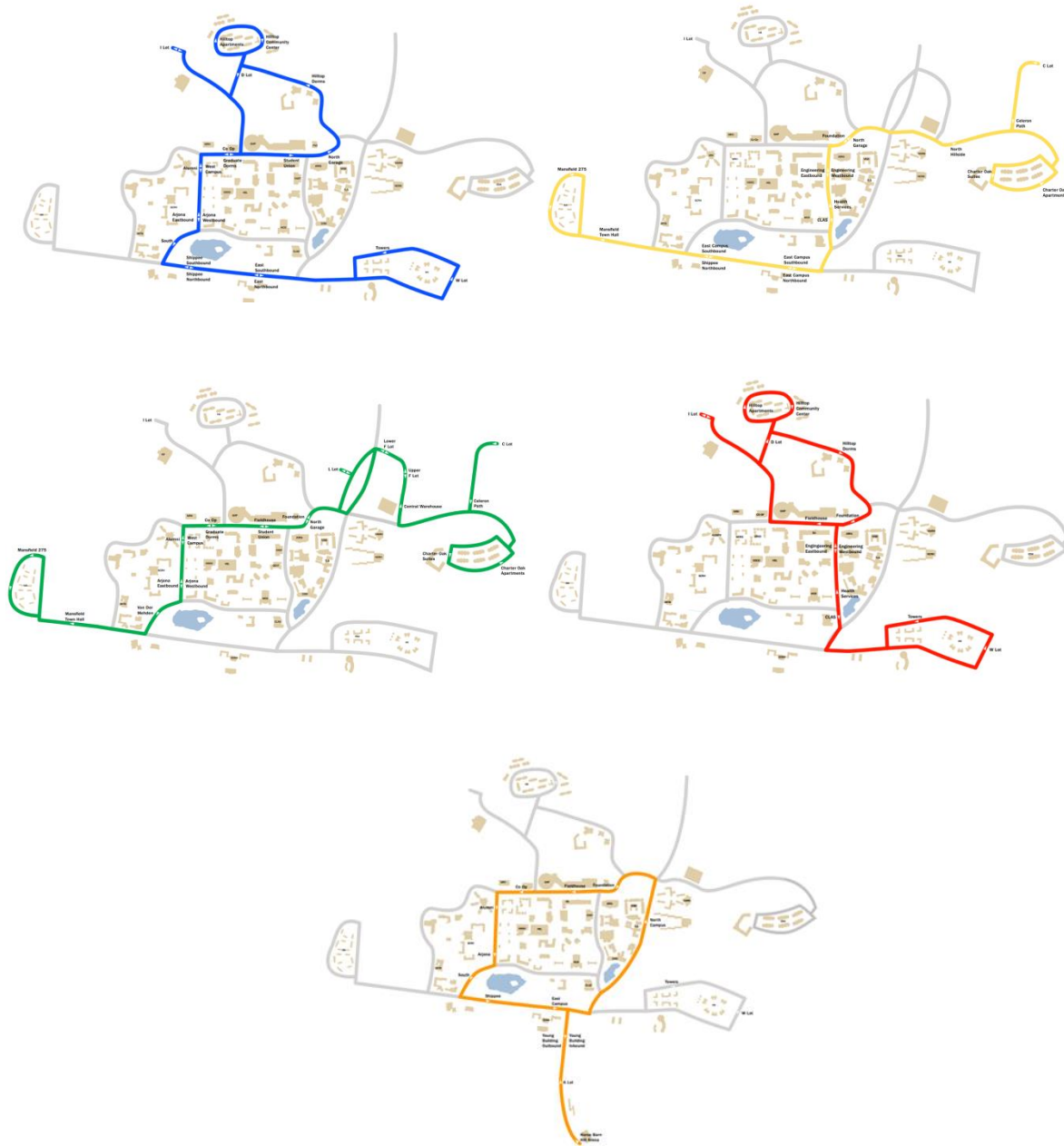


Figure 2.9: UCONN Bus Routes (2015) (a) Blue Line (b) Yellow Line (c) Green Line (d) Red Line (e) Orange line

The UCONN network consists of 48 nodes and 67 arcs. Additionally, 34 of the 48 nodes service more than one route and therefore are potential transfer nodes. The UCONN network includes 38 demand zones, 10 of which are served by a pair of stops. In figure 2.8, a dark blue line

connects the paired stops. Each arc has an associated travel time. These travel times were pulled from the UCONN shuttle bus system's general transit feed specification (GTFS) data using a Python script. The GTFS data was developed from data provided by the University of Connecticut's Transportation, Logistics, and Parking Services. Artificial demand data was generated to reflect the demand as well as possible based on the authors' understanding of the system.

Based on the results of the single route experiments, a limited number of the equity objective functions were selected for testing in the experiments on the MEqTSP model. The coefficient of variation minimizing objective functions (Z_5, Z_6) and the objective function minimizing the worst case scenario (Z_9) were eliminated due to their poor performance in the single route model. The objective functions containing parameter E_i (Z_2, Z_4, Z_6, Z_8) were not considered in this stage of model development due to the inconsistent effect of the parameter on the single route model. Further applications of the objective functions with the E_i parameter will be considered in the final model. This left the objective functions which minimized the average transit trip travel time (Z_1), average difference in travel time between transit and automobile (Z_3), and variation in difference in travel time between modes from a target difference (Z_7).

4.3 Discussion of Results

A hundred replicate runs of the GA were conducted for each of the three equity objective functions. The best solution found in the thirty runs for the three objective function formulations were identical to each other. Interestingly, the solutions were also identical to the actual sequence

of stops in the UCONN shuttle bus system. The GA was able to find this solution in 39%, 39%, and 51% of runs for objective functions Z_1 , Z_3 , and Z_7 , respectively. While there is no guarantee that the actual bus network represents an optimal solution, this outcome does suggest that the MEqTSP model and GA are capable of finding reasonable solutions that align with the decisions of experienced transportation planners and service operators.

In the case of the UCONN network there was no substantial difference between the results of the GA on MEqTSP using the different equity formulations. Ultimately, the objective functions found the same best solution in roughly the same number of generations. This is almost certainly due to the relatively small solution space. Two issues led to this constrained solution space: the strict assignment of stops to routes and UCONN's sparse network topology. In the final model, discussed in the next section, stops are not assigned to route sets and may switch between route sets during the reproduction step of the GA. This creates a much larger solution space meaning the different equity objective functions will likely return different results.

Though the MEqTSP experiments did not provide any insights into application of the inequity minimizing objective functions to multiple route problems, conducting these experiments did accomplish the stated goal of developing the data structures and solution methods necessary to solve multiple route problems.

5. Multiple Route Model (With Stop Grouping)

The final model presented in this chapter considers both the order in which stops are visited (stop sequencing) and how the stops are grouped together into routes (stop grouping). By considering

both stop sequencing and stop grouping simultaneously, the model moves away from a classic travelling salesman problem and becomes more similar to the transit network design problem (TNDP). Most TNDP models assume routes will take the form of bi-directional linear routes (Baaj and Mahmassani, 1995; Chakroborty, 2003; Mauttone, 2009). This relies on representing roads with two way traffic as single, bi-directional arcs. Mauttone (2009), who proposes the most complex method for generating initial routes, intentionally avoids creating any loops. This makes sense given the assumption that a vehicle must travel and forth along a selected path. However this assumption seems unnecessarily restrictive. The model proposed in this section takes a different, more flexible approach. Routes in this model are unidirectional cycles and roads with two way traffic are represented with two separate arcs. This means that part or all of a route may be linear while other portions are more circular. Hybrid routes, with both linear and circular segments, are common in practice and therefore worth considering in the model. The proposed TNDP model will be referred to as the EqTNDP throughout the remainder of the chapter.

5.1 Methodology

5.1.1 Model Formulation

The EqTNDP expands on the MEqTSP model proposed in the previous section. Like the MEqTSP model, each route must meet the TSP constraints. However, unlike the MEqTSP, the EqTNDP needs to consider adding an additional constraint or constraints to the model to ensure the routes do not become so large as to become unserviceable. Some TNDP models solve this by adding a constraint which sets a maximum route length. However, formulating the constraint in this way is unnecessarily restrictive and does not address the real concern, being able to provide the desired frequency of service along routes with the available fleet of vehicles. For this reason,

the EqTNDP model proposes a constraint which sets a maximum average headway H for a system with a vehicle fleet of size V . The full model is shown below.

Minimize $Z = Inequality$

Subject to:

$$\sum_{i; \forall i \in R^k} x_{ij}^k = 1 \quad \forall j \in R^k; \quad \forall k \in K \quad (18)$$

$$\sum_{j; \forall j \in R^k} x_{ij}^k = 1 \quad \forall i \in R^k; \quad \forall k \in K \quad (19)$$

$$u_i - u_j + |I|x_{ij}^k < |I| - 1 \quad \forall i, j \in R^k, i \neq j; \quad \forall k \in K \quad (20)$$

$$\frac{\sum_k \sum_i \sum_j x_{ij}^k c_{ij}}{V} \leq H \quad (21)$$

$$x_{ij}^k = Binary$$

In addition to the adjustments made to the constraints, the travel time between demand zones was calculated differently from the MEqTSP model. Idle time is considered in the calculation of travel time in this model. For each stop along a route, 30 seconds is added to the trip travel time. This model also addresses the possibility of walking connections. The MEqTSP model assumed that if two zones were not directly connected by transit, then the only other connection between the two zones was through a transfer. This is unrealistic, especially for the UCONN network, much of which is located in walkable campus area. It seems highly unlikely that people would

choose to wait for bus and go through the hassle of a transfer, which incurs a fifteen minute penalty in the model, when they could choose to take a three minute walk instead. This was addressed by allowing walking connections between stops located within 400 meters (Euclidean distance) of each other. 400 meters, or a $\frac{1}{4}$ mile, is the standard distance people planners assume people are willing to walk to bus stops. This is about a 5.5 minute walk using the FHWA recommended average pedestrian speed of 1.2 m/s. (FHWA, 2006) Walking is only allowed between zones that are not directly connected by transit and can only be used at the beginning and/or end of a transit trip. Walking cannot be the only mode used to make a connection between zones in this model. While this may not be an accurate depiction of traveler behavior on campus, the purpose of this aspect of the model is to find transit paths involving short walks that are likely to be chosen by users over paths requiring transferring between routes. The purpose is not to accurately model the behavior of all transportation system users.

If zones cannot be reached by a combination of short walks and a single transit trip, then the algorithm will look for the shortest path using a single transfer between routes. In the case that two zones cannot be connected using any of these methods, then the demand is considered to be unmet. Unmet demand has a cost equivalent to 90 minutes of travel time. While other models choose to include a possibility of a second transfer, this seemed unreasonable for the small UCONN network. It could easily be incorporated into this method for future, larger scale applications.

5.1.2 Solution Methodology

Genetic algorithms are a widely used and accepted metaheuristic for solving the transit network design problems (Pattniak and Mohan 1998; Chien et al, 2001; Chakroborty and Dwevedi, 2002;

Fan and Mechemehl, 2006; Beltran et al., 2009; Nayeem et al, 2014) The overall flow of the GA used to solve the EqTNDP model is similar to those of the previous models (see figure 2.3). However, the execution of each step varies significantly from that of the previous two models. This is due to the variations in stop groupings opening up the solution space and changing both the way solutions look and the way crossover functions operate. The structure of the population generating, crossover, and mutation functions in this genetic algorithm were largely drawn from the research in Chakroborty and Dwivedi (2002), Chakroborty(2003), and Fan and Machemehl (2006). Each of the functions which have been changed from the original GA will be discussed in detail below.

Generating Initial Solutions

Generating initial solutions for the EqTNDP model is different from the previous models because stops are not assigned to specific route sets. Additionally, the route sets do not have a defined or permanent size. The only constraint to route size is the average system headway. Users are able to input the available number of vehicles and their desired average headway. This algorithm required developing an original routine for population generation due to the unique definition of routes as cyclical rather than linear.

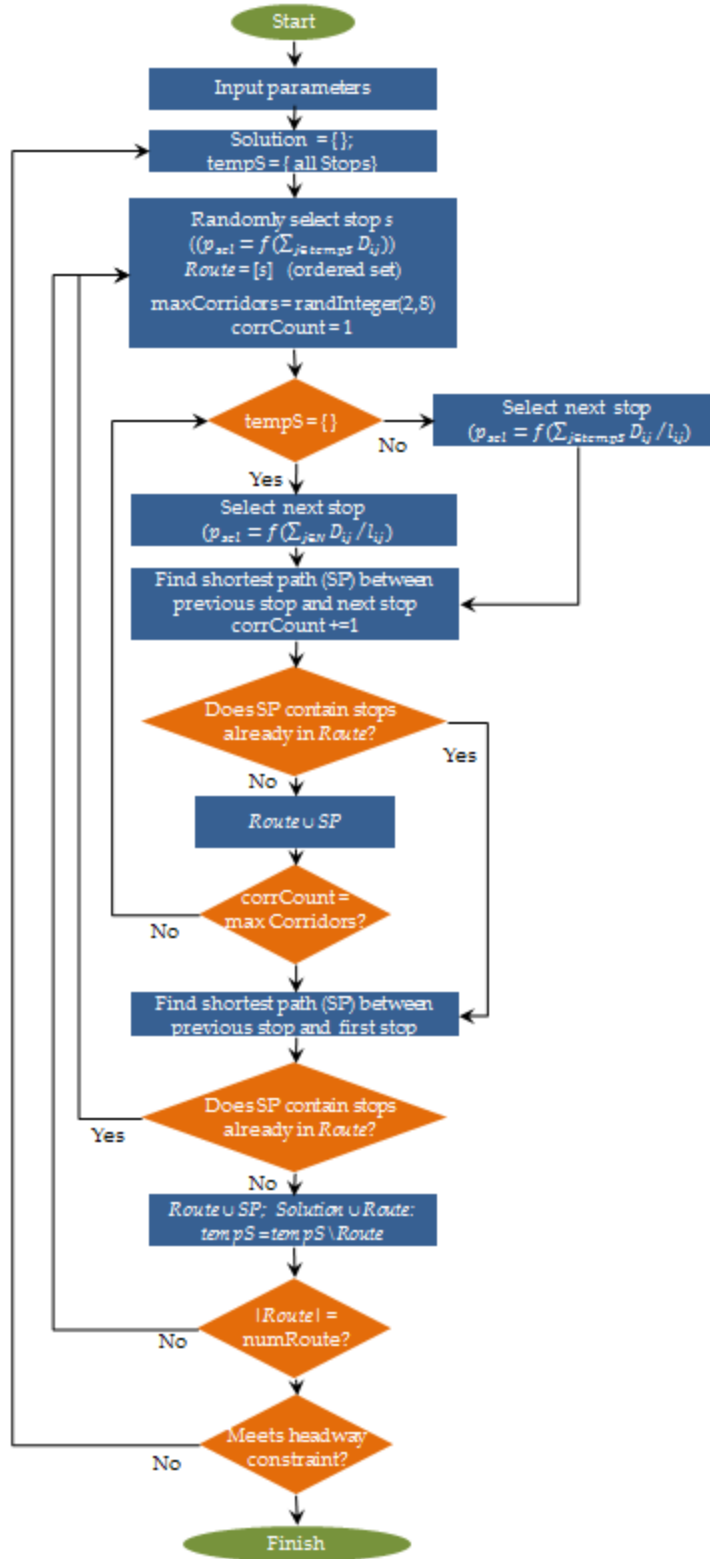


Figure 2.10 Flowchart of solution generation routine

Figure 2.10 shows the procedure used to generate one new solution. One solution will contain the predefined number of routes *numRoutes*. The routine begins by selecting the first node at random. The probability of a node being selected is directly proportional to the demand originating at that node. The routine then grows the route by selecting a high priority next node. The probability of an unselected node being selected is directly proportional to the demand between them and inversely proportional to the distance between them. The first node and the selected node are then connected via the shortest path (Dijkstra, 1959). This is analogous to designing for a demand corridor. This process of connecting a high priority node to the route via the shortest path is repeated until either the shortest path selected for connection overlaps with the existing route or the maximum number of high priority nodes is reached. Once either of these conditions is met, the routine attempts to connect the last node in the route to the first via the shortest path. If it is able to do so without revisiting any previous stops, then the route is complete and can be added to the new solution. Otherwise, the route is thrown out and the procedure begins again. This is repeated until the solution contains the specified number of routes.

Reproduction: Crossover

The reproduction procedure used in this genetic algorithm requires two different crossover functions: an interstring crossover and an intrastring crossover. Figure 2.11 below provides a detailed description of how the two crossover functions work together to produce new offspring.

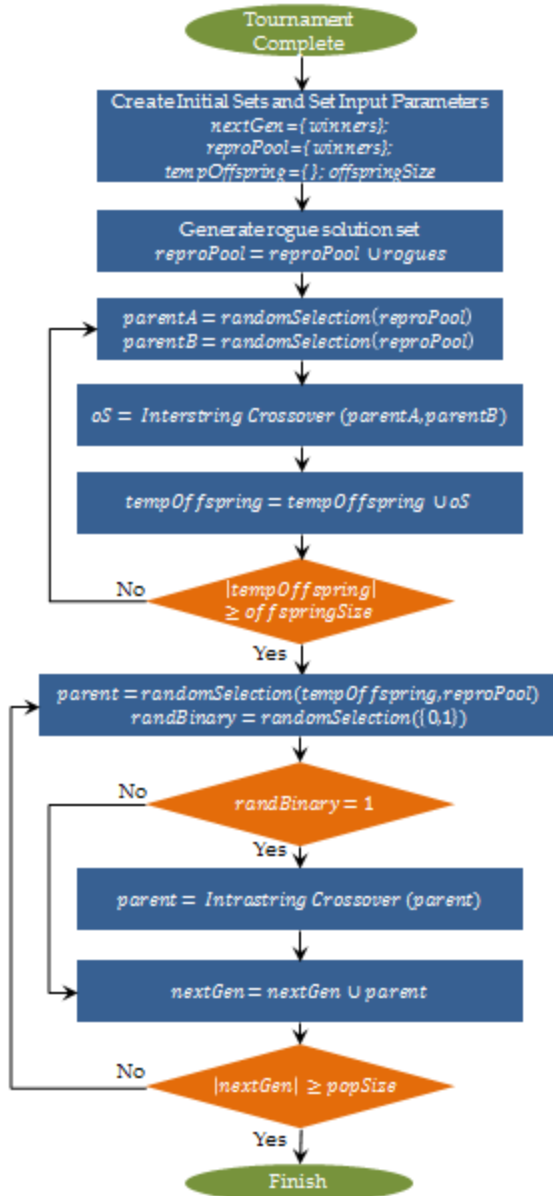


Figure 2.11 Reproduction Procedure: Crossovers

Like in the previous GA, all tournament winners (parents) are cloned directly into the next generation. Offspring can be created through either interstring crossover, intrastring crossover, or a combination of both. The GA also applies a mutation function to the offspring at a user-defined rate (*mutRate*). The crossover and mutation functions are described in greater detail below.

Interstring Crossover

The interstring crossover function is nearly identical to the one proposed in Chakroborty and Dwivedi (2002). The only difference is the function ensures the maximum average headway is met. This function requires two parents. One route is randomly selected from each parent and the routes are swapped to produce two new offspring. Figure 2.12 below shows two parents undergoing an interstring crossover. In this case, the blue and purple routes are swapped.

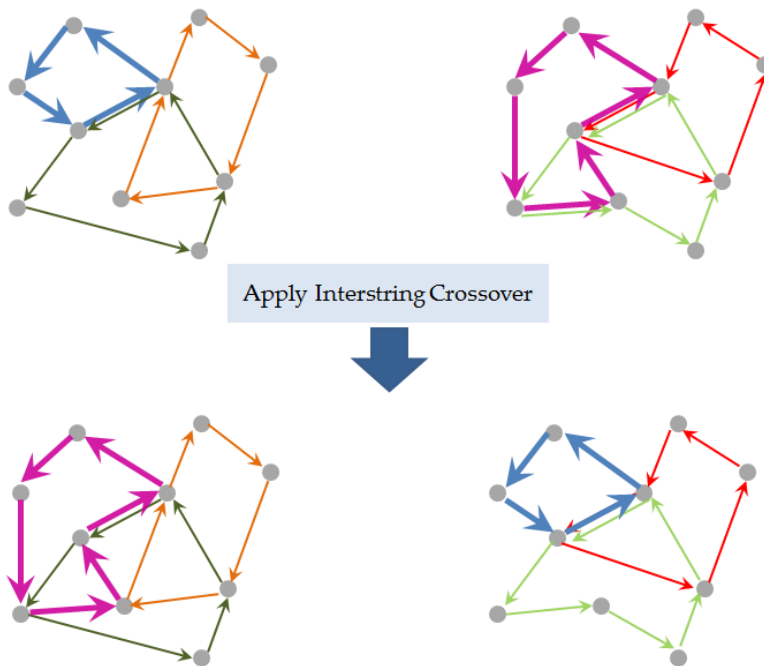


Figure 2.12 Interstring Crossover Function

Intrastring Crossover

While the idea of the intrastring crossover also originates with Chakroborty and Dwivedi (2002), significant changes were made to the procedure to allow for implementation on cyclical routes. Unlike interstring crossovers, intrastring crossovers only require one parent. Two routes within

the same parent swap segments. The intrastring crossover function is illustrated more clearly in Figure 2.13.

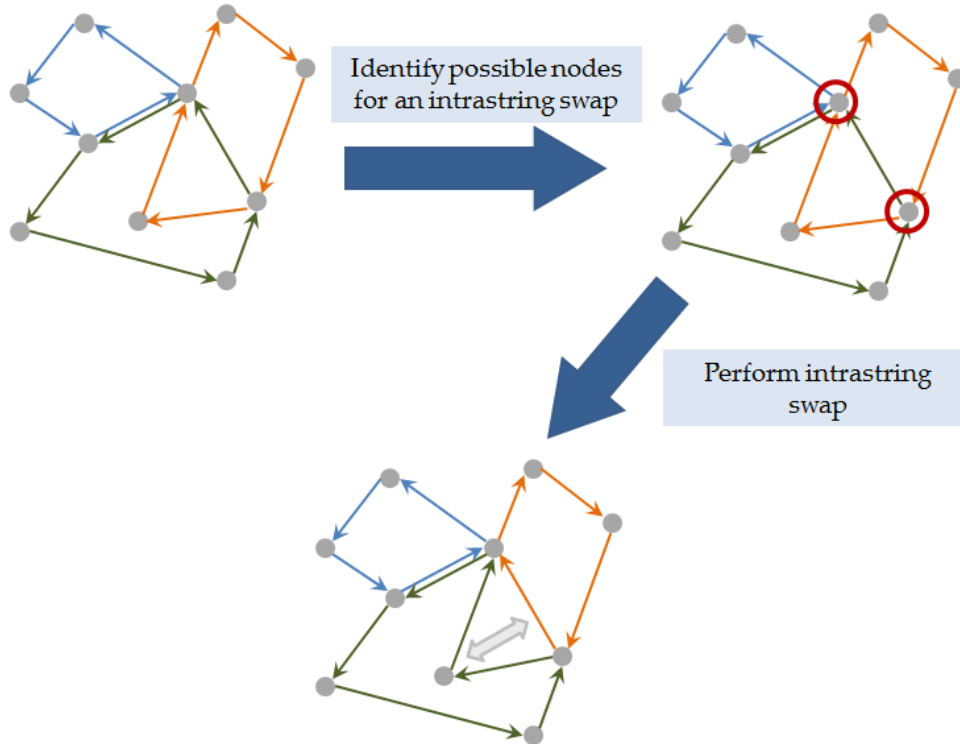


Figure 2.13 Intrastring Crossover

For two segments to be eligible for a swap, they must have the same origin node and the same destination node. Additionally, for the swap to produce a unique offspring, the segments should not be identical to one another.

Mutation

The mutation function is randomly applied to the offspring created by the crossover functions at mutation rate *mutRate*. The purpose of this, or any, mutation function is to insert small, random

changes into the population. In this case, the function makes a small change to one of a solution's routes. Before the algorithm is run, a k -shortest paths algorithm (Yen, 1971) is used to create a dictionary (a Python data structure) of short alternative paths or possible mutations. For a path to be considered a possible mutation path, it needs to be less than five stops long. In the case of the relatively sparse UCONN network, creating the dictionary of possible mutation paths was not a time consuming task.

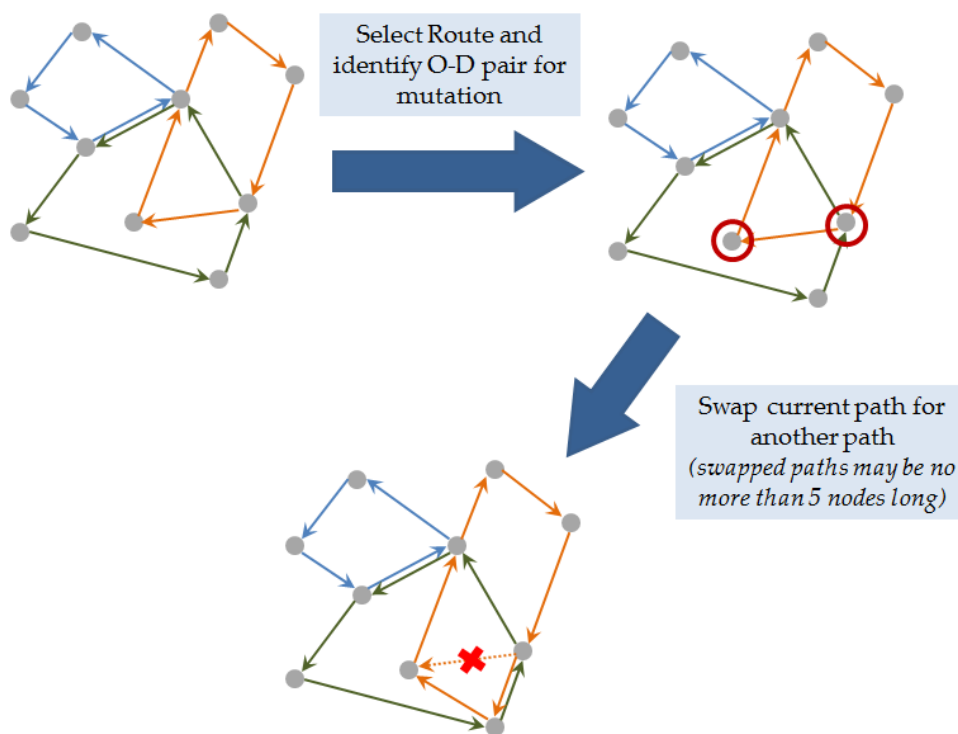


Figure 2.14 Mutation Function

Figure 2.14 shows the application of the mutation function to an example route. The mutation function identifies a stop pair with a possible mutation path and swaps out the old path for the new path.

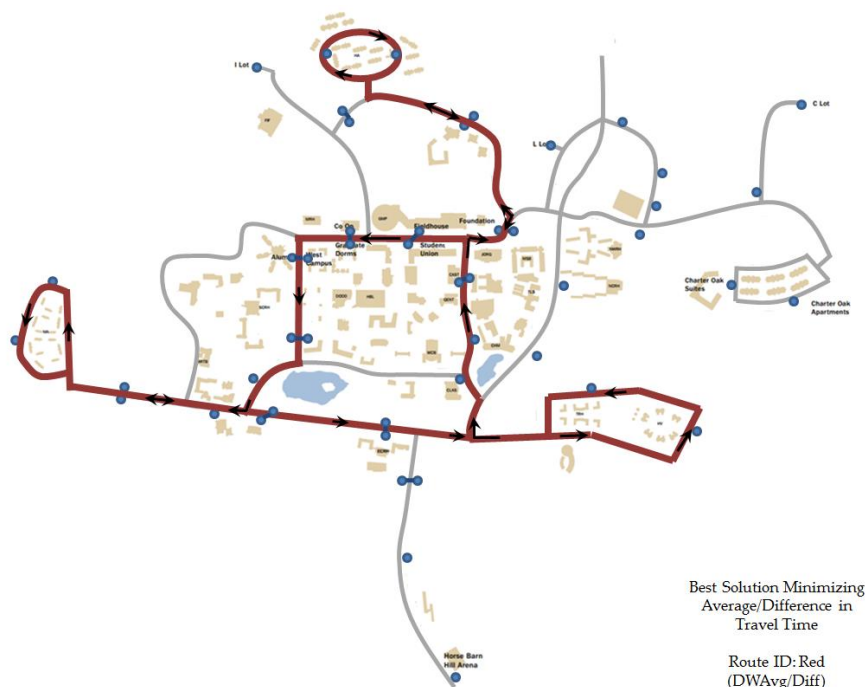
5.2 Test Networks and Experiments

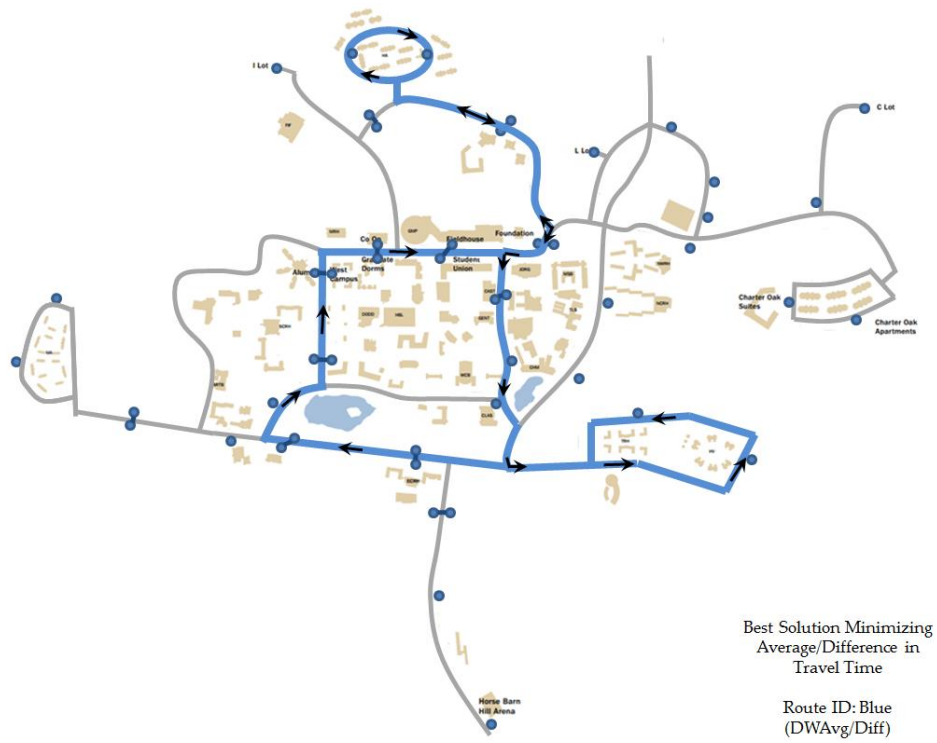
The EqTNDP was developed and tested on the UCONN network described in section 4.2. The same demand estimates were used in this model. However, unlike the previous set of experiments, the EqTNDP was tested for the all six of the inequity minimizing objective functions not eliminated in the single route experiments ($Z_1, Z_2, Z_3, Z_4, Z_7, Z_8$). This includes those with parameter E_i .

Though parameter E_i was originally intended to represent vulnerable or protected populations, such as low income people, ethnic and racial minorities, or older adults, it can be used to give priority to any subgroup within the population. In the case of a university campus, where the only residents are college students who have limited choice in where they live, it does not make sense to plan for traditional protected populations. For the purpose of these experiments, E_i will represent the proportion of commuters entering the bus system at a particular stop. This is not intended to imply that commuters are a vulnerable population or should be given priority in UCONN's transit planning. It is an exercise intended to measure the potential effects of including parameter E_i . E_i values were generated based on the UCONN (2015) shown in figure 2.15 below. Stops located near parking lots used by employees, graduate students, and commuters were assigned values proportional to the size of the parking lot.

variation from a target value were less successful (Z_7 , Z_8). None of the experiments returned solutions that improved upon the existing UConn transit system. It is unclear why these objective functions performed so poorly. It may be due to a combination of not selecting an optimistic enough target value and the way the initial solutions are generated. If the target value is not optimistic enough, the algorithm will push towards inefficient, “race to the bottom” type solutions. The initial solutions are generated by connecting a series of shortest paths meaning they will be relatively high quality solutions. If the algorithm requires poorer solutions in order to improve the objective function value, the initial solution generation procedure may pose a serious impediment to converging on a near optimal solution.

Figures 2.16(a-e) shows the best solution found to the EqTNDP model using the objective function which minimizing average user travel time (Z_1). This happens to be the same best solution found for the objective function which minimizes the difference in travel time between modes (Z_3).





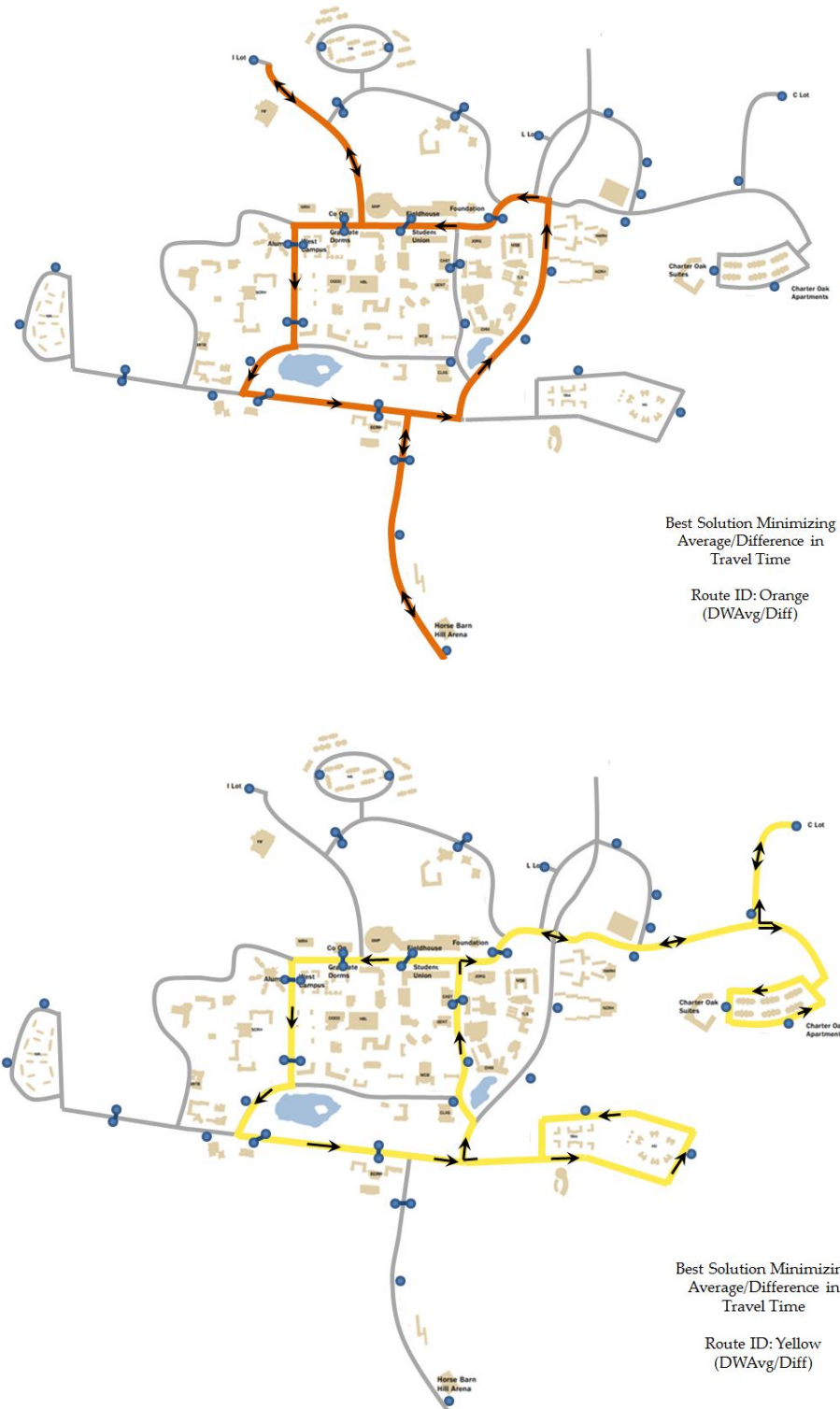


Figure 2.16 EqTNDP Solutions (Z_1) (a) Red Line (b) Green Line (c) Blue Line (d) Orange Line (e) Yellow Line

One of the most noticeable differences between the actual UConn transit system and the solutions found by the EqTNDP model is the way the route is structured around the core of campus. With the exception of Orange line, none of the actual UConn bus routes make a complete cycle around center of campus. Instead, each of the routes goes out and back along the same edges of the core creating more linear routes. In contrast, all of the routes in the EqTNDP solution create a loop (with branches) around the core of campus, opting to take different paths to get around the center campus rather than traversing the same roads in the opposite direction. Considering the walkable nature of the campus and the high cost of transfers between routes, it seems logical the GA would push towards solutions which loop around the campus core over bidirectional linear routes which only traverse some edges of the core. The more cyclical routes will allow more people to reach their destinations without transferring routes. Though not shown in the figures above, some of the objective functions did produce linear routes. Linear routes were more likely to appear in solutions for models using an E_i weighted objective function. This suggests the linear routes are more helpful to commuters, who are traveling from the far fringes of campus to the core, than to the general population. While specific policy suggestions should not be made based on this analysis due to limiting assumptions about traveler behavior, this analysis does support the previous claim of the need for a more flexible transit network design model. Bidirectional linear routes are not always the best solution and it is problematic and unnecessarily limiting to require routes to adhere to this structure.

Table 2.2 compares the best solutions found by the EqTNDP to each other. Each row in the table represents the best solution found to the EqTNDP when designing for the specified inequity minimizing objective function. Each column represents the value of that solution as evaluated by

each of the tested objective functions. The average headway of vehicles operating on these routes, assuming ten available vehicles, is also shown in the table (first column).

Table 2.2: Comparison of best solutions found using the inequity minimizing objective functions

		Evaluation of best solutions						
		Average Head- way	Average TT (min)	Avg Diff TT (min)	COV of Diff in TT	Average TT(min) Commuter	Avg Diff TT(min) Commuter	COV of Diff in TT Commuter
Best solution found using objective function	Z ₁	15.7351	11.0384	0.4548	0.0743	10.8415	1.0220	0.4755
	Z ₃	15.7351	11.0384	0.4548	0.0743	10.8415	1.0220	0.4755
	Z ₅	17.1794	11.9359	1.3523	0.0696	11.4213	1.6018	0.4713
	Z ₂	16.2213	12.3615	1.7778	0.0826	10.5161	0.6966	0.4505
	Z ₄	17.5520	11.7503	1.1667	0.0776	10.5578	0.7383	0.4922
	Z ₆	15.8367	12.3352	1.7516	0.0791	11.0334	1.2138	0.4338

As previously mentioned, Z_1 and Z_3 returned identical solutions. Z_2 and Z_4 , their E_i weighted counterparts also return very similar solutions. The Z_2 solution shown in Table 2.2 actually outperforms Z_4 in minimizing both average commuter trip travel time (Z_2) and minimizing average difference in commuter travel by bus than by car (Z_4). This may be due to some systematic flaw in the objective function but more likely the two objective functions are driving towards the same solution and the Z_2 happened to find a solution that is slightly closer to optimal. Due to the poor performance of the Z_7 and Z_8 in the experiments, little weight should be placed on the values attached to these solutions in Table 2.2.

Table 2.2 also allows for a closer look at the effect of the E_i parameter. For objectives Z_1 and Z_2 , designing the transit network for commuters added 1.33 minutes to the average trip travel time for the general population while saving commuters only 0.33 minutes. For objectives Z_3 and Z_4 , designing the transit network for commuters increased the difference in travel time between modes by 0.72 minutes while decreasing it for commuters by 0.28 minutes. The effect of this parameter is dependent on the specific configuration of the network and the location of the prioritized population. In the particular case of designing the UConn shuttle bus system, it seems to cost the general population more than it benefits commuters to design the bus network for their specific needs.

6. Conclusions and Future Work

In this chapter, three models were developed: a single route model (EqTSP), a multiple route model without stop grouping (MEqTSP), and a multiple route model with stop grouping (EqTNDP). The first model provided the testing environment for nine different inequity minimizing objective functions and for the development of a new genetic algorithm. After the first set of experiments on the EqTSP, some of the objective functions were able to be eliminated from further consideration. The COV minimizing objective functions lead to a “race to the bottom” scenario in which the optimal solution was every rider having an equally long, circuitous route. The objective function minimizing the maximum difference in travel times between modes was also eliminated. Though the solution was able to converge on reasonable quality solutions, the best solutions had too many alternative route configurations for this to be

used as a single objective function. Any of the rejected objective functions could easily be changed into constraints for EqTSP or other models.

The second model, the MEqTSP, provided the opportunity to build the capacity of the solution method and associated data structures to handle multiple routes. Though extensive testing was not done on this intermediary model, the experiments conducted on the UConn test network showed the model was capable of finding solutions equivalent to those developed by experienced transit planners.

The final model, the EqTNDP, considered both stop grouping and stop sequencing simultaneously like other TNDP models. However, this particular model creates some flexibility in route structure that other models do not. To the best of the authors' knowledge, all other TNDP models assume that routes must follow a bidirectional linear structure. In this model however, a route must be cyclical. Given that the network contains separate arcs for opposite sides of the street, a cycle may have segments appear very linear in structure or it may appear more loop-like. The reason this flexibility is important is for cases like the UConn network where passengers may experience greater benefits from routes that are a hybrid of linear and circulator routes than from linear routes alone.

In the future, this model should be tested on a larger, real world network to see if the patterns found on the UConn network are seen elsewhere. When scaling up to a larger network, care should be taken to develop or incorporate more sophisticated methods for calculating travel time by walking and car. It would be particularly interesting to explore how the E_i parameter can be used in the future to design for particular segments of the population. The way it is currently incorporated into the objective function, when E_i is used the prioritized population is the only

population being considered. More realistic this should be incorporated into a function that balances the needs of different groups.

Another interesting direction for this research would be to use the experiments to identify important trunk lines or subroutes. When conducting the EqTNDP experiments, very few of the replicate runs returned the exact same solution. However, the solutions were very similar.

Developing a method which reads through the experiments and determines which route segments or groups of stops are consistently included in the best solutions would be a valuable line of research. This would allow planners and practitioners more flexibility in implementing the results of the model.

Chapter 3: A Genetic Algorithm Approach to Solving the Equitable Traveling Salesman Problem

1. Introduction.....	80
2. Overview of the GA.....	80
2.1. Generating Initial Solutions.....	82
2.2. Tournament.....	83
2.3. Reproduction.....	85
2.3.1. Crossover.....	85
2.3.2. Mutation.....	85
2.4. Convergence.....	86
3. Description of Experiments.....	87
4. Discussion of Results.....	88
4.1. Results of Algorithmic Structure Testing.....	88
4.2. Results of Parameter Testing.....	92
4.3. Larger Networks.....	96
5. Conclusions.....	98

1. Introduction

The research discussed in this chapter is a complement to the research presented in Chapter 2. It contains a much more detailed discussion and validation of the genetic algorithm (GA) used to solve equitable Traveling Salesman Problem (EqTSP) described in Chapter 2, Section 3. First, each step of the GA will be discussed, focusing on the procedures and algorithmic structures which are unique to this specific algorithm. Then experimental evidence will be provided to validate decisions regarding its algorithmic structure, including the decision to clone winning solutions into the next generation and the use of rogue parents. Finally, a sensitive analysis is conducted on the five input parameters: population size, tournament size, number of rogue parents, mutation rate, and convergence criteria. All of the preliminary experiments and sensitive analysis were conducted on the Sioux Falls network. Once the best algorithmic structure and input parameters were determined, the GA was tested on the Qatar national network.

2. Overview of the GA

The GA described in this section is the same as the one described in Chapter 2, Section 3.1. This section will describe the GA in more technical terms and focus on the components which make it unique. Most of the decisions made in the GA can be explained in terms of either intensification or diversification. The purpose of a GA is to drive closer and closer to the optimal solution with each generation, a process known as intensification (Blum and Roli 2003) However, the population of solutions must maintain enough diversity to overcome local minima and find the true optimal solution. Maintaining diversity means that some individual solutions may move further from the optimal solution even as the best solutions continue to improve.

The pseudocode for the primary routine is shown below. Pseudocode will also be provided for the more unique subroutines later in the chapter.

PRIMARY ROUTINE: Genetic Algorithm

Subroutines indicated in bold, italics.

```

begin
   $R := \emptyset$ 
  while  $|R| < p$  do
    begin
       $r := \textbf{Generate Solution} (N, c_{ij}, \alpha)$ 
       $R := R \cup r$ 
    end;
  genCount := 1; convergenceVal :=  $\infty$ 
  while genCount < genConverge do
    begin
      WIN := Run Tournament ( $R, t, \text{fit}(r)$ )
      pop := WIN; reproPool :=  $\emptyset$ 
      while  $|\text{reproPool}| < \text{numRogues}$  do
        begin
           $r := \textbf{Generate Solution} (N, \delta_{ij})$ 
          reproPool := reproPool  $\cup r$ 
        end;
      reproPool := reproPool  $\cup$  WIN
      while  $|\text{pop}| < p$  do
        begin
          PA := randomly selected  $r$  from reproPool
          reproPool := reproPool  $-$  PA
          PB := randomly selected  $r$  from reproPool
          reproPool := reproPool  $\cup$  PA
          if EAX Crossover(PA, PB) produces a solution do:
            offspring := EAX Crossover(PA, PB)
             $n :=$  random number between 0 and 1
            if  $n > \text{mutationRate}$  then pop := pop  $\cup$  offspring
            else do
              offspring := Mutate(offspring)
              pop := pop  $\cup$  offspring
          end;
        bestSolution := Find Best Solution ( $R, \text{fit}(r)$ )
        if bestSolution  $\geq$  convergenceVal then genCount := genCount + 1
        else do:
          convergenceVal :=  $\text{fit}(\text{bestSolution})$ ; genCount := genCount + 1
          solution := bestSolution
        end;
      end;
    end;
  end;

```

Notation:

i	origin nodes
j	destination node
c_{ij}	travel time between i and j along the shortest path
r	route (solution) as an ordered set nodes. Subscripts refer to the position of an element (node) in ordered set r .
N	set of nodes
R	set of routes (solutions) r
p	population size
t	tournament size
$numRogue$	number of rogue parents
$genConverge$	number of generations to convergence
$mutationRate$	proportion of offsprings mutated

2.1 Generating Initial Solutions

Generating good initial solutions is necessary for the algorithm to converge on optimal or near-optimal solutions. Given the size of the solution space for this problem, it is possible to generate an initial population that contains both diverse and good solutions. The solution space contains all possible permutations of the nodes within a network. The size of the solutions space can therefore be calculated as shown below:

$$P(n, k) = \frac{n!}{(n - k)!} \quad (1)$$

$$Size\ of\ Solution\ Space = \frac{|N|!}{(|N| - |N|)!} = |N|! \quad (2)$$

Consider the Sioux Falls network which contains only 24 nodes. Even though it is a relatively small network, it contains 6.2045×10^{23} possible solutions. This is why it is important to start with a reasonably good population.

The pseudocode below was used to generate initial solutions.

SUBROUTINE: Generate Random Solution**Notation in primary routine: GENERATE SOLUTION** (N, c_{ij}, α) **Output: A route (solution)****begin** $S := N; r := \emptyset$ $i :=$ randomly selected element from set S $S := S - i$ $r := r \cup i$ $firstNode := i$ **while** $S \neq \emptyset$ **do****begin**Assign probability of selection to each node in $S, p_{sel, j \in S} = f\left(\frac{1}{c_{ij}^\alpha}\right)$ $j :=$ element selected from S according to probability distribution p_{sel} $S := S - j$ $r := r \cup i$ $i := j$ **end;** $r := r \cup (i, firstNode)$ Output r **end;**

The α parameter allows the user to adjust the importance of the nearness of nodes in generating initial solutions. A higher α value places greater emphasis on the nearness of nodes. For the Sioux Falls experiments, α was set equal to 1. For larger networks, it is necessary to increase α .

For larger networks, it was helpful to apply a second strategy of not allowing initial solutions which exceeded a certain threshold. The TSP solution cannot exceed twice the cost of the minimum spanning tree (MST). The cost of the MST can be found *a priori* using Kruskal's algorithm (Kruskal 1956). Solutions that do not meet this threshold are not added to the initial population.

2.2 Tournament

The tournament determines which solutions will be allowed to enter the reproductive phase of the GA and which will be discarded. The population is split into smaller groups of size t and the best solution from each group is then added to the reproductive pool. While it may seem most sensible to simply rank the solutions and pick the top solutions for inclusion in the reproductive pool, most GAs implement an indirect process, such as tournament, in an effort to maintain a diverse set of good solutions. Note that this process does not guarantee the best solution, will be included in the reproductive pool.

The pseudocode below was used to conduct the tournament.

SUBROUTINE: Conduct a tournament on set of routes R given tournament size t . Each route r will be evaluated using fitness function, $fit(r)$.

Notation in primary routine: RUN TOURNAMENT ($R, t, fit(r)$)

Output: Set of “fittest” solutions

begin

$WIN := \emptyset; POP := R$

while $POP \neq \emptyset$ **do**

begin

$T := \emptyset$

while $|T| < t$ **do**

begin

$r :=$ randomly selected route from R

$POP := POP - r$

$T := T \cup r$

end;

$WIN_VAL := \infty$

for each r **in** T **do**

if $fit(r) < WIN_VAL$ **then** $WIN_VAL := fit(r)$ **and** $WIN_R := r$

$WIN := WIN \cup WIN_R$

end;

 Output WIN

end;

2.3 Reproduction

The purpose of the reproduction phase is to create a new generation of solutions which drives the algorithm closer to the optimal solution while providing new diversity. This GA automatically clones a copy of the tournament winners into the next generation, a decision which will be further investigated later in the chapter. The purpose of this step is to ensure that the population maintains a certain level of quality. Then several newly generated solutions, or rogues, are added to the reproductive pool. To the best of the author's knowledge, this is a completely unique procedure. This is a method for adding diversity to the population and must be implemented in moderation. Its effectiveness will be discussed extensively later in the chapter. A crossover function then generates new solutions from the solutions in the reproductive pool. This function uses pieces from two good solutions and therefore, will hopefully create new good solutions which contribute to both the intensification and diversification processes. Finally, a mutation function is applied to small proportion of solutions. This function makes small, random changes to solutions, increasing diversity in the wider population.

2.3.1 Crossover Function

This GA used the EAX crossover proposed by Nagata and Kobayashi (1999). Because the solutions are represented as permutations rather than binary arrays, they require a special crossover function. A description and illustration of the EAX crossover can be found in Chapter 2. The pseudocode for the crossover can be found in Nagata and Kobayashi (1999).

2.3.2 Mutation Function

The mutation function is only applied to a small proportion of the newly generated offspring solutions. This is because, like the addition of rogue parents, the mutation functions primary

purpose is to diversify the population. Some of the mutations will help the algorithm overcome local minima and find better solutions, while others will worsen solutions. Initially, the intention of the authors was to replace the mutation function with the rogue parents. However, as will be shown in the following sections, both functions proved necessary to finding optimal solutions.

This mutation function selects a small, random segment of the solution and reinserts it into another portion of the solution. This segment may or may not be reversed before reinsertion. The pseudocode below shows exactly how the mutation function operates.

SUBROUTINE: Mutate route

Notation in primary routine: MUTATE(*r*)

Output: Route *r* with mutation

begin

maxLength := $\lfloor (|r| - 1) / 5 \rfloor$

length := random integer between 2 and *maxLength*

start := random integer between 1 and $|r| - \text{length} - 2$

tempSeg := [*r*_{*start*}, *r*_{*start* + *length*}]

r := *r* − *tempSeg*

insertPt := random integer between $|r| - 2$

k := random number between 0 and 1

if *k* ≤ 0.5 **then do**

begin

reverse tempSeg

end;

insert *tempSeg* after *insertPt*

end;

2.4 Convergence

At the end of each generation, the algorithm checks for convergence. This GA uses the number of generations without an improvement to the best solution as the convergence criteria. Ideally, the convergence criteria should balance quality of solutions with time to convergence. The convergence criteria should be set to a value at which it is unlikely the best solution will

substantially improve if the algorithm were to continue running. A discussion of where this value should be set will be included in the following sections.

3. Description of Experiments

An extensive set of experiments was conducted to validate and explore the two unique features of the GA:

1. The automatic “cloning” of tournament winners into the next generation
2. The addition of “rogue parents” (or newly generated solutions) into the reproductive pool.

These experiments were conducted to determine the impact of these features on the quality of the solutions and the efficiency of the algorithm. The experiments were also used to determine the effects of the five GA parameters (population size, tournament size, rogue parents, mutation rate, and convergence criteria).

In this analysis, four different algorithmic structures were tested, based on all possible combinations of two different decisions. The first decision was whether or not to clone parents into the next generation. The second decision was how to incorporate rogue parents into the algorithm. The first possibility is simply generating new solutions each iteration; however this may be time consuming, particularly for large networks. An alternative method which may save time is generating a small pool of rogue parents *a priori* which can be pulled from whenever rogue parents are necessary. Table 3.1 below outlines the 4 algorithm structures. Algorithm 1 is the algorithm described by the pseudocode in the previous section.

Table 3.1 Algorithmic Structures for Experiments

Algorithm 1	With clones; New rogue parents each generation
Algorithm 2	With clones; Rogue parents from pool
Algorithm 3	Without clones; New rogue parents each generation
Algorithm 4	Without clones; Rouge parents from pool

The algorithmic structures were tested on the travelling salesman problem (TSP) for Sioux Falls using all possible combinations of the parameters shown in Table 3.2. For each of the 4 algorithmic structures and 120 parameter combinations, the GA was run 30 times. The solutions found using the GAs were compared to the known optimal solution to the problem which was found using a branch and bound method in GAMs.

Table 3.2 Parameters for Experiments

Population Size	100, 300
Tournament Size (as a proportion of population)	0.05,0.10
Rogue Parents (as a proportion of population)	0, 0.03, 0.05, 0.1, 0.2
Mutation Rate	0, 0.01, 0.10
Convergence Criteria (Number of generations without improvement)	10,40

4. Discussion of Results

Each of the algorithmic structures and parameters will be evaluated with regards to both the quality of solutions and the time to convergence. Finding quality solutions is prioritized over finding solutions quickly, however time to convergence cannot become so high that the GA becomes unusable.

4.1 Results of Algorithmic Structure Testing

Algorithm 1, which clones parents into the next generation and generates new rogue parents each iteration, converged on the known optimal solution significantly more frequently than the other algorithmic structures. Figures 3.1 (a-d) plots the time to convergence versus the proportion of runs converging on the optimal solution for each of the experiments run on the different algorithmic structures. Please note, some outliers took more than 100 seconds to converge and are not shown in these figures.

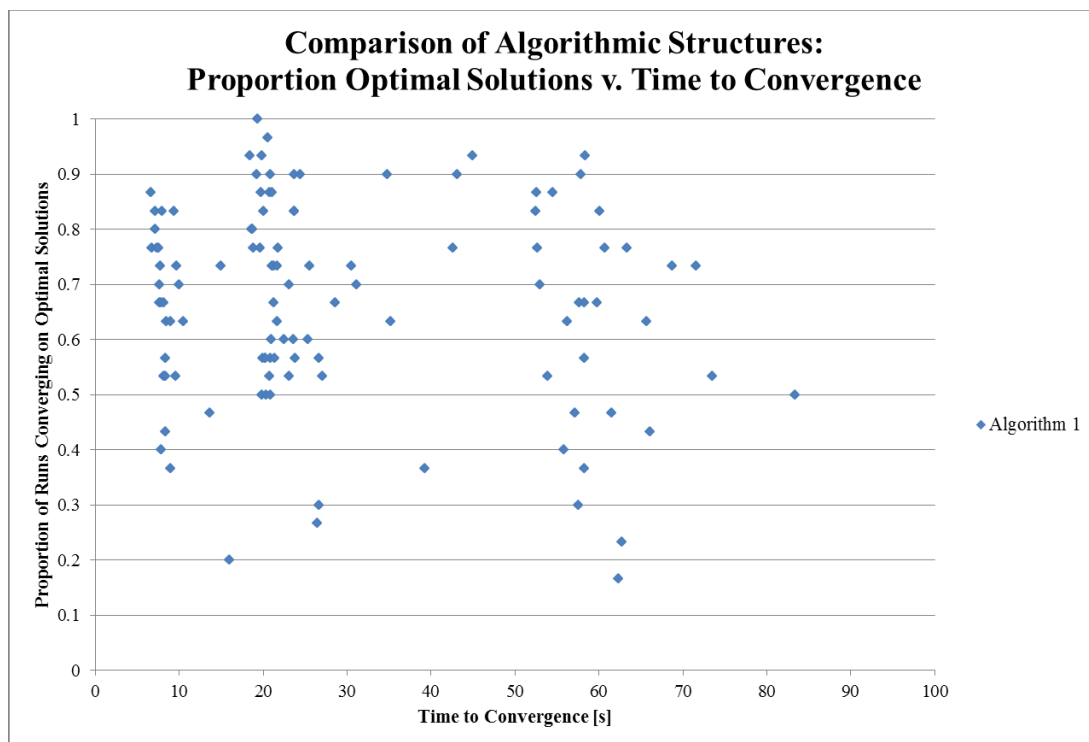


Figure 3.1 (a) Algorithm 1: Clones, New Rogue Parents

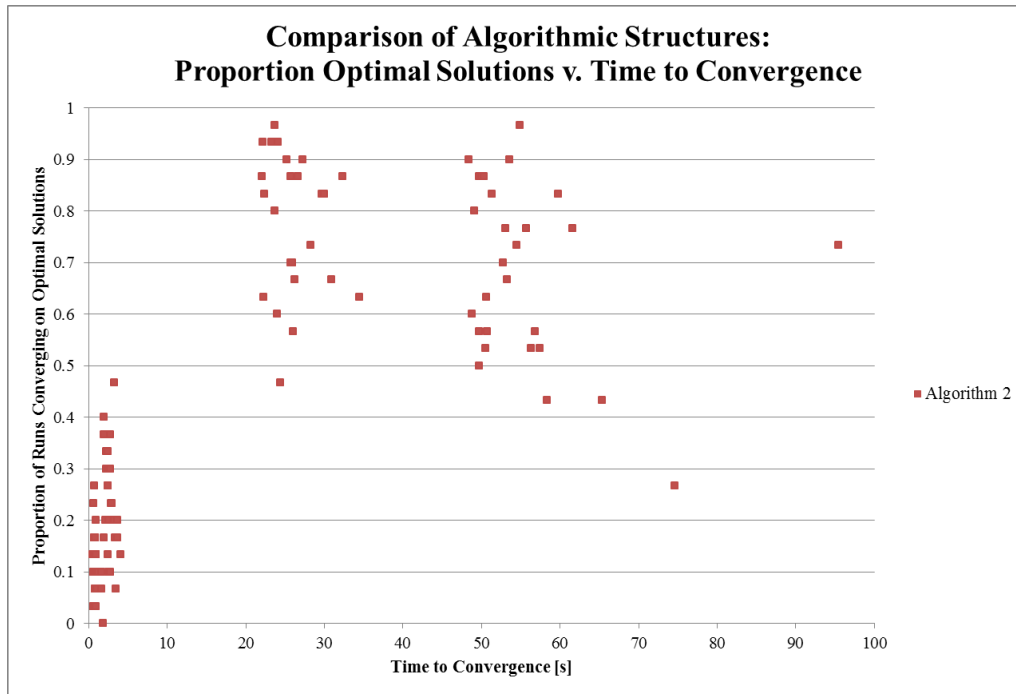


Figure 3.1 (b) Algorithm 2: Clones, Rogue Parents from Pool

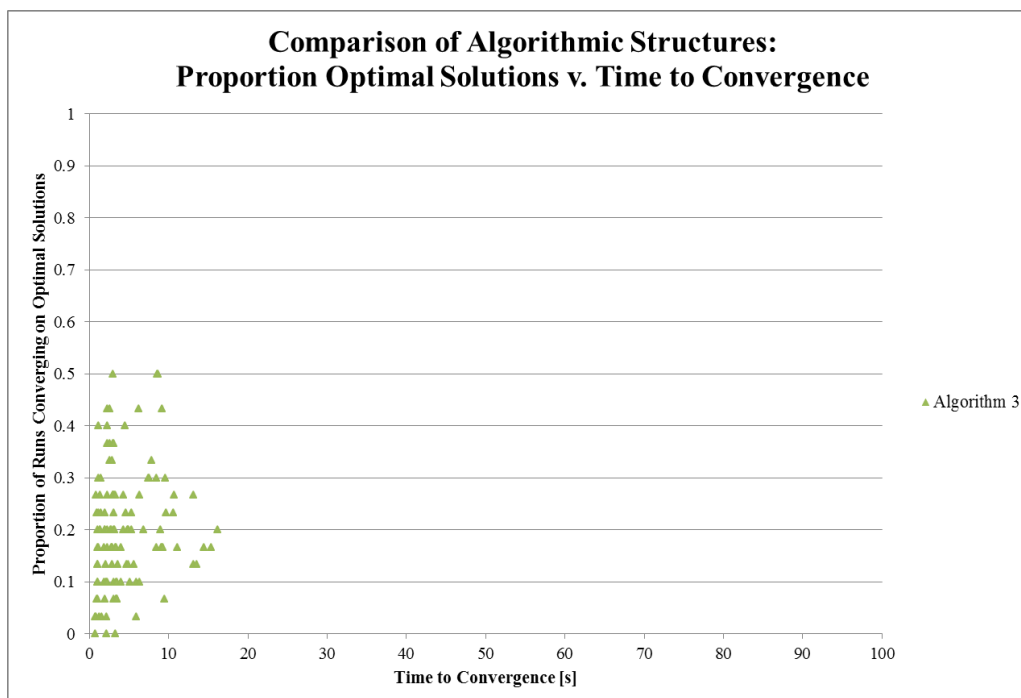


Figure 3.1(c) Algorithm 3: No Clones, New Rogue Parents

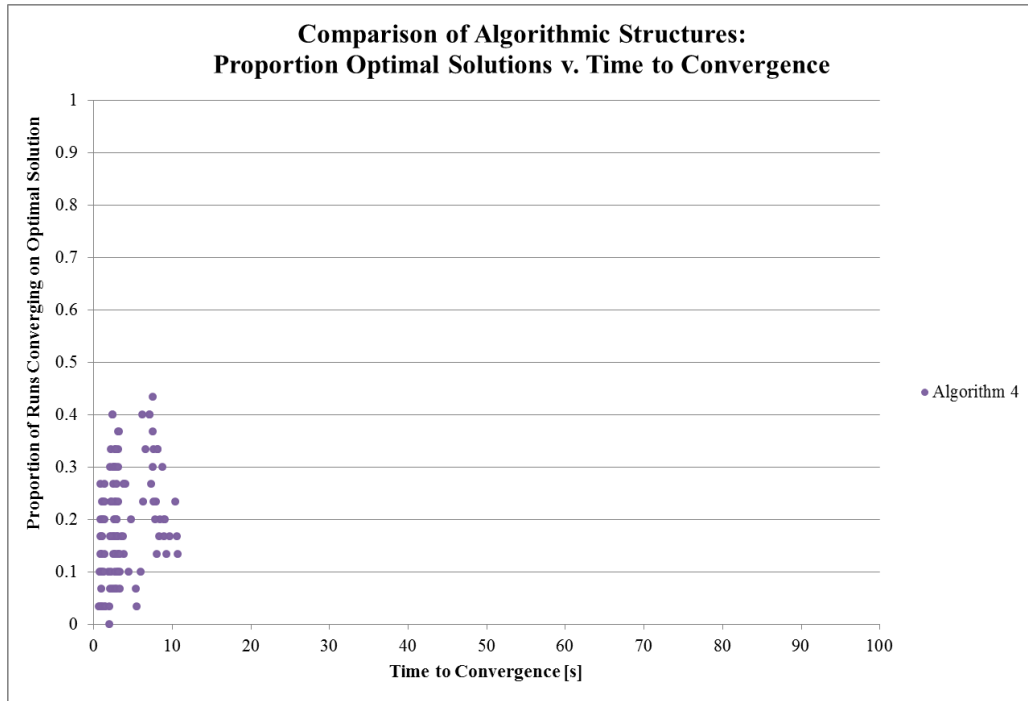


Figure 3.1(d) Algorithm 4: No clones, Rogue parents from pool

Figure 3.2 below compares the distribution of results for the different algorithmic structures using box plots.

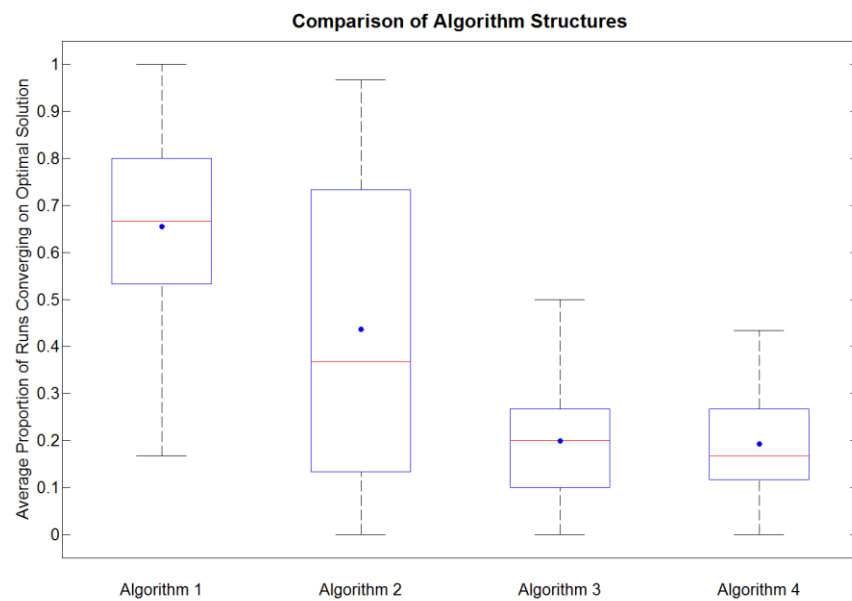


Figure 3.2 Comparisons of Algorithmic Structures by Quality of Solutions

Figures 3.1(a-d) and Figure 3.2 show that algorithms 3 and 4, those which did not clone tournament winners into the next generation were unable to converge on the optimal solution even 50% of the time. These algorithmic structures were immediately discarded. The differences in the quality of algorithmic structures 1 and 2 were less obvious. A t-test comparing algorithms 1 and 2 showed that algorithm 1 converged on the optimal solution significantly more often than algorithm 2 ($p < 0.001$). There was no statistically significant difference in time to convergence between algorithms 1 and 2.

To summarize, cloning the tournament winners into the next generation significantly increased the proportion of runs converging on the known optimal solution. Generating new rogue parents each generation also significantly increased the proportion of runs converging on the known optimal solution as compared to pulling clone parents from a pre-generated pool. There was not a significant difference in the time to convergence between the two treatments of rogue parents. Given these results, algorithm 1 will be used for all future testing.

4.2 Results of Parameter Testing

Calibrating the parameters to their actual optimal values would require the development of a secondary heuristic; a line of research which is outside the scope of this dissertation. However, the experiments conducted on the selected set of parameters for algorithm 1 provides insights on the effects of parameter values. While each of the five parameters was tested for their impact on the quality and time to convergence, this analysis will focus on the rogue parent and mutation rate parameters. Before going into a detailed discussion of these parameters, this section will provide a brief summary of the others.

The smaller population size (100) converged on the optimal solution significantly more often ($p<0.001$) in significantly less time ($p=0.04$). The smaller tournament size (0.05) converged on the optimal solution significantly more often ($p<0.001$) and had no significant influence on time. The higher convergence criteria (40 generations without improvement) converged on the optimal solution significantly more often ($p<0.001$) but required significantly more time ($p=0.04$). However, it is worth noting that while the GAs with a 40 generation convergence criteria did require significantly more time to converge, that does not mean the time was necessarily unreasonable. Table 3.3 below shows the combinations of parameters in which more than 90% of the runs converged on the optimal solution. (A table showing the full set of parameter combinations can be found in Appendix A.) For example, the parameter combination shown in the first row has a convergence criterion of 40 generations and returns the optimal solution in 100% of runs, yet each run only takes 19.33 seconds on average.

Table 3.3 Results for Best Parameter Combinations (Sioux Falls)

Pop Size	Tournament	Rogues	Mutation Rate	Gen	Runs Converging on Optimal [%]	Avg Time to Convergence [s]
100	0.05	0.05	0.10	40	100.00	19.33
100	0.05	0.10	0.01	40	96.67	20.67
100	0.05	0.10	0.00	40	93.33	18.47
300	0.05	0.03	0.01	10	93.33	19.73
100	0.05	0.00	0.10	40	93.33	45.98
300	0.05	0.03	0.10	40	93.33	58.81
100	0.05	0.05	0.00	40	90.00	19.06
100	0.05	0.05	0.01	40	90.00	21.07
100	0.05	0.03	0.10	40	90.00	22.67
100	0.05	0.20	0.10	40	90.00	24.98
300	0.05	0.00	0.01	10	90.00	35.88
100	0.05	0.00	0.01	40	90.00	43.20
300	0.05	0.03	0.01	40	90.00	56.37

Unlike the other parameters, both the rogue parent and mutation rate parameter had the option of being set to 0. This means that in some experiments no rogue parents were added and/or the mutation function was never applied. The five possible values for the rogue parent parameter were compared to each other using a t test. The results are shown in Table 3.4 below.

Table 3.4: Comparison of Rogue Parent Parameters: t-Test p values

		Value of Rogue Parent Parameter				
		0.00	0.03	0.05	0.10	0.20
Value of Rogue Parent Parameter	0.00		0.0002	0.0008	0.0037	0.2676
	0.03			0.1179	0.1410	0.0001
	0.05				0.8826	0.0005
	0.10					0.0236
	0.20					

GAs with rogue parent parameters set at 0.03, 0.05, or 0.10 converged on the optimal solution significantly more frequently than those set at 0 (no rogue parents) or 0.20. The interpretation of these results can be further colored by the boxplots of shown in Figure 3.3.

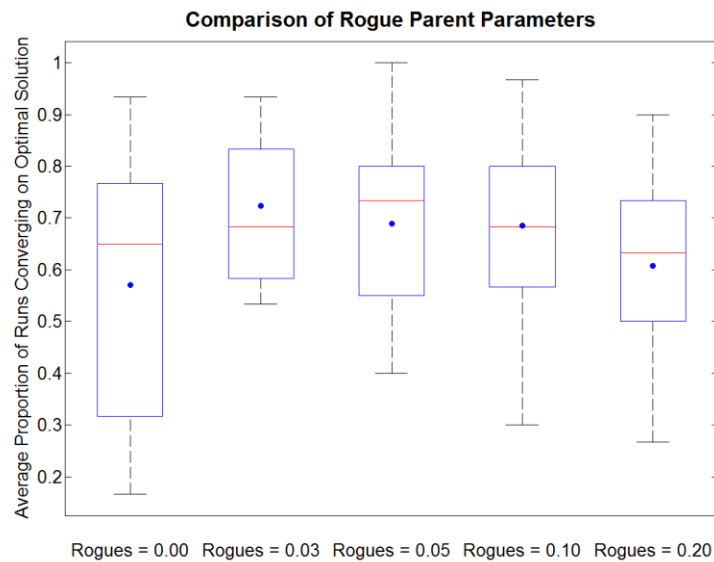


Figure 3.3 Comparison of Rogue Parent Parameter by Quality of Solution

Figure 3.3 shows that the distribution of the proportion of runs converging on optimal is much larger when rogue parents are not included. This suggests that not including the rogues made the GA much more sensitive to the parameters. Even when the rogue parent parameter was set to 0.2, which led to the GA converging on the optimal solution significantly less than the other parameters, the lower bound of the distribution is noticeably higher than when rogue parents are not included at all.

Additionally, GAs with rogue parent parameters of 0.05 or 0.10 converged significantly faster than those set at other values. These tests suggest that the introduction of rogue parents into the population is a useful way to improve both the quality and efficiency of the GA for small network TSPs.

The GAs with mutation rates of 0.10 converged on the known optimal solution significantly more often than those with mutation rates of 0.01 and 0. The distribution of the results can be seen in figure 3.4 below.

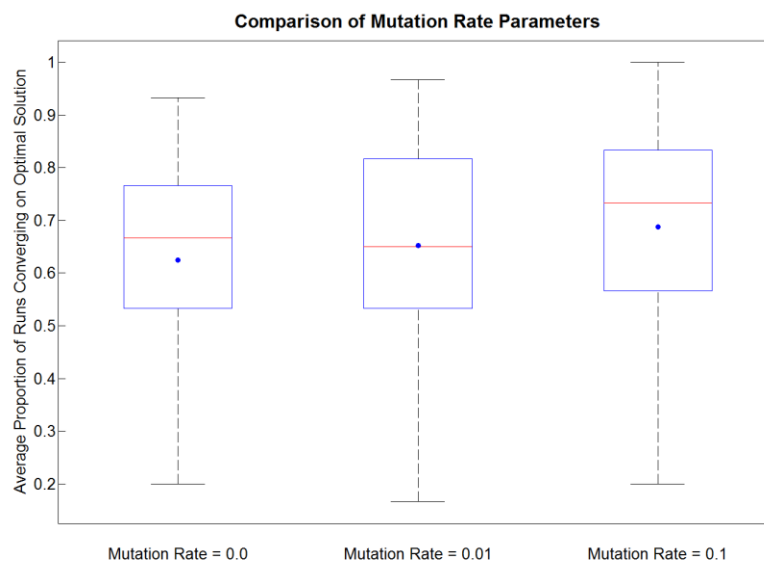


Figure 3.4 Comparison of Mutation Rate

This was a surprising result. The insertion of the rogue parents into the reproductive pool was initially intended to replace the mutation function. Not only do these results suggest it is necessary to keep the mutation function but the results also suggest maintaining a particularly high mutation rate. This may be due to the limited parameter values tested in this analysis but it is still unusual. The high mutation rate did lead to significantly higher time to convergence.

4.3 Larger Networks

After running this extensive set of experiments on the Sioux Falls network, a smaller set of experiments was conducted on the much larger Qatar national network (National TSPs, 2009).

The known optimal solution is shown in Figure 3.5 below.

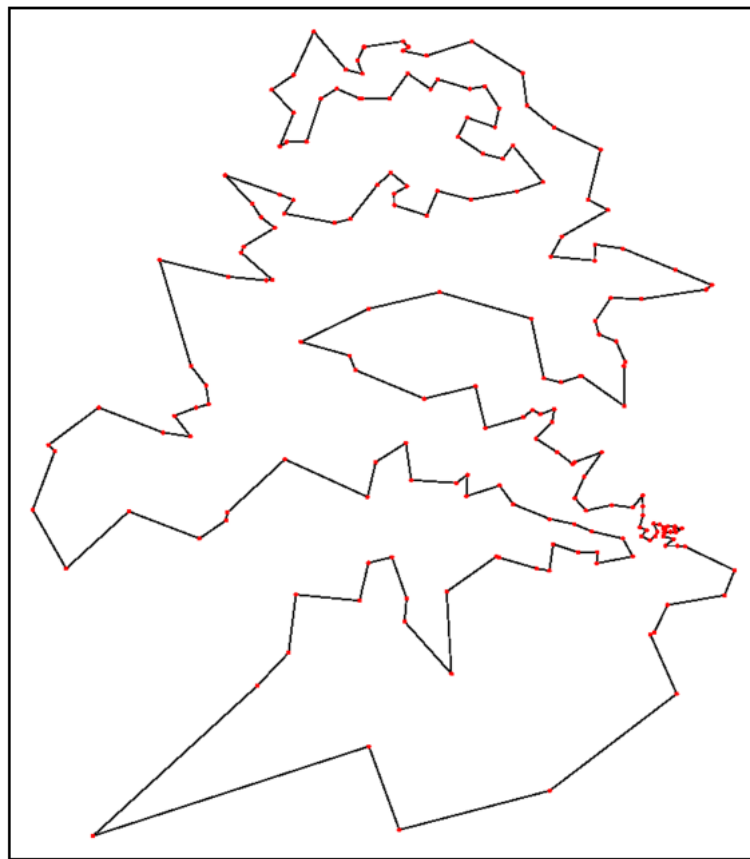


Figure 3.5 Optimal Solution to the Qatar National TSP

The Qatar national network is a fully connected network containing 194 nodes. Because the network was much larger, the GA took longer to converge, especially for poor parameter combinations. For this reason, the set of experiments on the Qatar TSP does not contain the complete set of parameter combinations. Table 3.5 below shows the results of some of these experiments. All of the experiments have a tournament size of 0.05. Due to space constraints, this parameter is not shown in Table 3.5.

Table 3.5: Results for Best Parameter Combinations (Qatar National)

Pop Size	Rogues	Mutation Rate	Gen	Runs within 1% of Optimal [%]	Runs within 5% of Optimal [%]	Avg Time to Convergence [s]
300	0.03	0.10	80	0.27	1.00	670.82
300	0.10	0.10	80	0.27	1.00	918.55
300	0.05	0.00	40	0.23	0.97	465.82
100	0.05	0.00	80	0.20	1.00	215.96
300	0.05	0.01	80	0.17	1.00	758.32
300	0.05	0.01	40	0.13	1.00	472.89
300	0.10	0.10	40	0.13	1.00	666.53
100	0.05	0.00	40	0.10	1.00	128.30
100	0.10	0.10	40	0.10	1.00	201.70
100	0.10	0.01	80	0.10	1.00	274.56
100	0.10	0.10	80	0.10	0.93	290.14
300	0.03	0.10	40	0.10	1.00	445.82
300	0.05	0.00	80	0.10	1.00	729.16
300	0.10	0.01	80	0.10	1.00	879.88

The Qatar experiments found a similar relationship between parameters and solution quality as the Sioux Falls experiments. The one exception is that GAs using the larger population parameter (300) were found to converge within 1% of optimal significantly more often. This change aligns with previous research that larger problems benefit from larger initial populations.

Though the GA takes longer to converge, it is capable of converging on near-optimal solutions for Qatar national TSP.

5. Conclusions

The genetic algorithm developed to solve the EqTSP presented in the previous chapter was tested and validated in this chapter. The proposed GA contained two unique features: cloning tournament winners into the next generation and the use of rogue parents. Cloning tournament winners was shown to significantly increase the number of runs converging on the optimal solution. The inclusion of rogue parents was also shown to significantly increase the number of runs converging on the optimal solution. However, including too many rogue parents can decrease the quality of the solutions. The recommended number of rogue parents to insert into the reproductive pool is between 0.05 and 0.10 of the population. These parameter values also caused the GA to converge significantly faster than the other rogue parent values.

This GA was tested on both the small Sioux Falls TSP and the larger Qatar National TSP. For the right combination of parameters, the GA was able to find the optimal solution for the Sioux Falls TSP in 30 out of 30 runs. While the GA was not able to find the exact optimal solution for the Qatar National TSP, it was able to find solutions within 1% of optimal for 8 out of 30 runs and 5% of optimal for 30 out of 30 runs for some parameter combinations. The best solution found by the GA on the Qatar national network was within 0.08% of optimal. These tests provide support for the use of this algorithm to solve the models presented in Chapter 2. As these models are applied to larger networks, it would be worthwhile to parallelize this algorithm. GAs are easy to parallelize, as there are many independently functioning pieces. Parallelizing this algorithm could drastically improve computational time.

Concluding Remarks and Future Work

The research presented in this paper proposed new methods for designed transit networks for equity and accessibility. The first chapter presents a tool which automates the measuring and mapping of transit accessibility using publically available data. The accessibility measure, the transit opportunity index (TOI), is a complex and comprehensive measure which includes spatial, temporal, and connectivity components. While the comprehensive nature of TOI made the measure appealing to researches, it was difficult for practitioners to implement, particularly those without expertise in geographic information systems (GIS). The automated tool discussed in Chapter 1 addresses this issue. By automating the calculation of TOI in Python, practitioners can leverage the TOI without a background in GIS. TOI is currently available to transit operators in the state of Connecticut through the t-HUB tool (t-HUB, 2015) and is being used by the Capital Region Council of Governments (CRCOG), Hartford's MPO, in their transit study.

Automating the TOI drastically reduced calculation and visualization time which opened up the possibility for more complex analyses. The chapter gives an example of conducted a time of day analysis on the Hartford system. Analyzing changes in transit accessibility during different time periods is an important component of helping vulnerable populations and conducting Title VI analyses so this could be a very powerful analytical tool.

While this tool was able to be applied to some medium-sized transit systems it has not yet been scaled up to a large scale, multimodal transit network. The author is currently working with a member of the Boston MPO to apply the TOI to the city of Boston. This will require a more nuanced look at transfers. In the current tool, transfers between different modes are treated the same as transfers between transfers between two vehicles of the same mode, though this is unlikely to be true. Transfer penalties should consider changes in mode. Future work should also consider the temporal aspects of transfers.

The second chapter develops a model which can be used to plan new transit networks for equity. While the tool presented in the first chapter is useful for examining existing systems, identifying gaps, and comparing alternative proposed network designs, the models proposed in the second chapter takes a more proactive approach. The first model, the EqTSP, was primarily used to test the feasibility of various inequity minimizing objective functions on the sequencing of stops. The results of testing on this model revealed that designing transit routes to minimize the coefficient of variation in the difference in travel times between modes led to a race to the bottom scenario. The model minimized variance by making the travel time between all origins and destinations equally bad. This is an undesirable scenario that would be rejected by planners, operators, and riders, alike. However, this race to the bottom scenario can be easily averted by calculating variance from an optimistic target value rather than the mean. These empirical results align with the proof of Dr. Boyles (2015). The testing of the model also suggested that a minimax objective function would not be able to provide a unique optimal solution and should therefore be rejected.

The final model proposed in Chapter 2, EqTNDP, incorporates equity directly into the stop sequencing and stop grouping components of the transit network design problem (TNDP). However, several unique decisions were made in the formulation of this TNDP that makes it both more flexible and more useful to practitioners than other TNDP's. First, the network used in the TNDP is the actual street network, not a simplified, theoretical graph. This means that rather than representing two-way streets as single, bi-directional arcs terminating, two-way streets are represented as two, uni-directional arcs. It also means that rather than treating all intersections as nodes, only locations that have been selected as possible stop locations are treated as nodes. The second difference in the proposed formulation is the cyclical structure of solutions. Given the structure of the network, cyclical solutions may results in routes that are loops, bi-directional

lines, or a combination of the two. This flexibility reflects the actual range of possibilities available to transit planners.

For the network used to test the final model, it generally provided minimal benefits to the vulnerable population to explicitly plan the entire system to minimize inequity. However, there are potential benefits to running this model. Running the model multiple times revealed which corridors and routes were most critical to serving certain populations or maintaining specific definitions of equity. While it may not be particularly beneficial or reasonable to plan the entire system according to one of these definitions, being able to identify specific corridors as critical to specific populations would be very helpful to transit planners. Future research will develop an algorithm for identifying critical corridors from the results of this model. Another important direction for future research is incorporating frequency setting into the model. As mentioned earlier in Chapter 1, the temporal aspects of accessibility are vitally important for improving the general accessibility of vulnerable populations.

The models solved in Chapter 2 were solved using a genetic algorithm which is discussed in detail in both Chapter 2 and Chapter 3. This genetic algorithm contains two unique features. The first is the cloning of tournament winners into the next generations. The second is the addition of rogue parents (newly generated solutions) into the reproductive pool. Both of these unique features were proven to significantly improve the number of times the GA returned optimal or near optimal solutions. The number of rogue parents inserted in the reproductive pool should not be too high or else the GA may converge on sub-optimal solution. The ideal number of rogue parents appears to be between 0.05 and 0.10 of the population. This GA was able to find optimal and near optimal solutions for both the Sioux Falls and Qatar National TSPs. Larger networks do substantially increase the time it takes for the GA to converge. Before the GA is applied to larger

networks, it should be parallelized. GAs are relatively easy to parallelize given that several functions must be applied to each member of the population independently.

In the next five to ten years, the work described in the Chapters 1, 2, and 3 will ideally be combined. The knowledge gained from open source geographic information can and should be harnessed by optimization models.

Appendices:

Appendix A: Full Set of Parameter Experiments (Sioux Falls).....	105
---	------------

Appendix A: Full Set of Parameter Experiments on Sioux Falls TSP

Pop Size	Tournament	Rogues	Mutation Rate	Gen	Runs Converging on Optimal [%]	Avg Time to Convergence [s]
100	0.05	0.05	0.10	40	100.00	19.33
100	0.05	0.10	0.01	40	96.67	20.67
100	0.05	0.10	0.00	40	93.33	18.47
300	0.05	0.03	0.01	10	93.33	19.73
100	0.05	0.00	0.10	40	93.33	45.98
300	0.05	0.03	0.10	40	93.33	58.81
100	0.05	0.05	0.00	40	90.00	19.06
100	0.05	0.05	0.01	40	90.00	21.07
100	0.05	0.03	0.10	40	90.00	22.67
100	0.05	0.20	0.10	40	90.00	24.98
300	0.05	0.00	0.01	10	90.00	35.88
100	0.05	0.00	0.01	40	90.00	43.20
300	0.05	0.03	0.01	40	90.00	56.37
100	0.05	0.05	0	10	86.67	6.68
300	0.05	0.1	0.1	10	86.67	19.72
100	0.05	0.1	0.1	40	86.67	20.74
300	0.05	0.03	0.1	10	86.67	21.09
300	0.05	0.1	0.1	40	86.67	52.60
300	0.05	0.05	0.01	40	86.67	54.51
100	0.05	0.1	0.1	10	83.33	7.16
100	0.05	0.03	0.1	10	83.33	8.00
100	0.05	0	0.01	10	83.33	9.38
300	0.05	0.03	0	10	83.33	20.05
100	0.05	0.03	0	40	83.33	23.75
100	0.05	0.03	0.01	40	83.33	23.75
300	0.05	0.05	0.1	40	83.33	52.44
300	0.05	0.03	0	40	83.33	60.11
300	0.05	0	0.01	40	83.33	444.36
100	0.05	0.03	0	10	80.00	7.15
100	0.05	0.2	0.01	40	80.00	18.62
100	0.05	0.2	0	40	80.00	18.78
100	0.05	0.05	0.01	10	76.67	6.77
100	0.05	0.1	0.01	10	76.67	7.36
100	0.05	0.05	0.1	10	76.67	7.51
300	0.05	0.1	0.01	10	76.67	18.80
300	0.05	0.05	0.01	10	76.67	19.68
300	0.05	0.05	0	10	76.67	21.78
100	0.05	0	0	40	76.67	42.63

300	0.05	0.1	0	40	76.67	52.69
300	0.05	0.2	0	40	76.67	60.68
300	0.05	0.05	0	40	76.67	63.33
300	0.05	0	0	40	76.67	148.45
100	0.1	0.1	0	10	73.33	7.75
100	0.05	0.2	0.1	10	73.33	9.70
100	0.05	0	0.1	10	73.33	14.98
300	0.05	0.05	0.1	10	73.33	21.04
100	0.1	0.1	0.1	40	73.33	21.24
300	0.05	0.2	0	10	73.33	21.66
100	0.1	0.05	0.1	40	73.33	25.54
100	0.1	0.2	0.1	40	73.33	30.51
300	0.05	0.2	0.01	40	73.33	68.69
300	0.05	0.2	0.1	40	73.33	71.58
300	0.05	0	0.1	40	73.33	762.52
100	0.05	0.03	0.01	10	70.00	7.65
100	0.05	0	0	10	70.00	10.01
100	0.1	0.05	0	40	70.00	23.14
300	0.05	0.2	0.1	10	70.00	31.08
300	0.05	0.1	0.01	40	70.00	52.95
100	0.05	0.1	0	10	66.67	7.67
100	0.1	0.1	0.01	10	66.67	7.84
100	0.05	0.2	0	10	66.67	8.11
300	0.05	0.1	0	10	66.67	21.25
100	0.1	0.03	0.1	40	66.67	28.60
300	0.1	0.05	0.1	40	66.67	57.63
300	0.1	0.03	0.01	40	66.67	58.29
300	0.05	0	0.1	10	66.67	59.76
100	0.1	0	0.1	40	66.67	354.50
100	0.05	0.2	0.01	10	63.33	8.51
100	0.1	0.1	0.1	10	63.33	8.95
100	0.1	0.03	0.1	10	63.33	10.48
300	0.1	0.03	0.01	10	63.33	21.65
300	0.05	0	0	10	63.33	35.19
300	0.1	0.03	0	40	63.33	56.21
300	0.1	0.2	0	40	63.33	65.66
100	0.1	0	0.01	40	63.33	114.02
300	0.1	0.1	0	10	60.00	21.01
100	0.1	0.2	0	40	60.00	22.45
100	0.1	0.2	0.01	40	60.00	23.62
100	0.1	0.03	0	40	60.00	25.32
100	0.1	0.05	0.01	10	56.67	8.41
300	0.1	0.05	0.1	10	56.67	19.96

300	0.1	0.03	0.1	10	56.67	20.20
300	0.1	0.05	0	10	56.67	20.27
300	0.1	0.03	0	10	56.67	20.81
300	0.1	0.1	0.01	10	56.67	21.33
300	0.1	0.1	0.1	10	56.67	23.81
100	0.1	0.03	0.01	40	56.67	26.65
300	0.1	0.03	0.1	40	56.67	58.23
100	0.1	0.03	0	10	53.33	8.17
100	0.1	0.03	0.01	10	53.33	8.41
100	0.1	0.2	0.01	10	53.33	9.63
100	0.1	0.1	0.01	40	53.33	20.78
100	0.1	0.05	0.01	40	53.33	23.08
300	0.05	0.2	0.01	10	53.33	27.06
300	0.1	0.05	0	40	53.33	53.93
300	0.1	0.2	0.1	40	53.33	73.48
300	0.1	0.05	0.01	10	50.00	19.89
100	0.1	0.1	0	40	50.00	20.30
100	0.1	0	0.1	10	50.00	20.91
100	0.1	0	0.01	10	50.00	83.32
100	0.1	0.2	0.1	10	46.67	13.64
300	0.1	0.1	0.01	40	46.67	57.12
300	0.1	0.1	0.1	40	46.67	61.50
100	0.1	0.05	0.1	10	43.33	8.39
300	0.1	0.2	0.01	40	43.33	66.04
100	0.1	0.05	0	10	40.00	7.88
300	0.1	0.05	0.01	40	40.00	55.83
100	0.1	0.2	0	10	36.67	8.93
300	0.1	0.2	0.1	10	36.67	39.24
100	0.1	0	0	40	36.67	58.27
300	0.1	0	0.1	40	36.67	5512.16
300	0.1	0.2	0.01	10	30.00	26.63
300	0.1	0.1	0	40	30.00	57.56
300	0.1	0.2	0	10	26.67	26.46
300	0.1	0	0.01	40	26.67	3420.00
300	0.1	0	0	10	23.33	62.70
100	0.1	0	0	10	20.00	16.02
300	0.1	0	0	40	20.00	200.86
300	0.1	0	0.1	10	20.00	1534.67
300	0.1	0	0.01	10	16.67	62.32

References:

- Adjo A. Amekudzi, Mshadoni K. Smith, Stefanie R. Brodie, Jamie M. Fischer, and Catherine L. Ross (2012) Impact of Environmental Justice on Transportation: Applying Environmental Justice Maturation Model to Benchmark Progress. *Transportation Research Record* 2320: 1-9.
- Ahuja, Magnanti, and Orlin (1993) *Network Flows*. Prentice Hall, Englewood Cliffs, New Jersey. pp 147-148.
- Alam, Bhuiyan Monwar (2009). Transit Accessibility to Jobs and Employment Prospects of Welfare Recipients Without Cars: A Study of Broward County, Florida, Using Geographic Information Systems and an Econometric Model. *Transportation Research Record*, Vol. 2110, pp 78-86.
- Baaj, M. Hadi and Mahmassani, Hani S. Hybrid Route Generation Heuristic Algorithm for the Design of Transit Networks. *Transportation Research Part C* 3(1), 1995, pp.31-50
- Beltran, Borja, Stefano Carrese, Ernesto Ciripriani, and Marco Petrelli. (2009) Transit network design with allocation of green vehicles: A genetic algorithm approach. *Transportation Research Part* Vol 17(5): pp.475-483.
- Bertolaccini, Kelly and Lownes, Nicholas E. (2014) Effects of scale and boundary selection in assessing equity of transit supply distribution. *Transportation Research Record: Journal of the Transportation Research Board* 2350: pp 58-64.
- Bowerman, Robert, Hall, Brent, and Calamai, Paul (1995). A Multi-Objective Optimization Approach to Urban School Bus Routing: Formulation and Solution Methods. *Transportation Part A: Policy and Practice*. Vol. 29, Issue 2, March pp. 107-123.
- Boyles, Stephen D (2015) Equity and Network-Level Maintenance Scheduling. *EURO Journal on Transportation and Logistics*. 4:175-193. doi: 10.1007/s13676-015-0075-7
- Blum and Roli (2003) Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*. 35(3): 268-308.
- Blumenberg, Evelyn and Paul Ong (2001) Cars, Buses, and Jobs: Welfare Participants and Employment Access in Los Angeles. *Transportation Research Record*, Vol 1756, pp 22-31.
- Blumenberg, Evelyn and Kimiko Shiki (2003) How Welfare Recipients Travel on Public Transit, and Their Accessibility to Employment Outside Large Urban Centers. *Transportation Quarterly*. Vol. 57(2), pp 25-37.
- Chien, Steven, Yang, Z. and Hou, E. (2001). Genetic Algorithm Approach for Transit Route Planning and Design. *Journal of Transportation Engineering*. Vol 127(3): pp. 200-207

Chakroborty, Partha and Dwivedi, Tathagat (2002) Optimal Route Network Design for Transit Systems Using Genetic Algorithms. *Engineering Optimization*. Vol. 34(1): 83-100.

Chakroborty, Partha (2003) Genetic Algorithms for Optimal Urban Transit Network Design. *Computer-Aided Civil and Infrastructure Engineering*. Vol. 18: 184-200.

President William Clinton (1994) Executive Order 12898

Connecticut State Data Center (2015) Connecticut Census Data: American Community Survey 5-year estimates (2008-2012). Website Accessed: November 29, 2015.

http://ctsdc.uconn.edu/connecticut_census_data/#2008_2012_ACS_5

CTTransit (2015) CTTransit GTFS. Website Accessed: November 27, 2015.

<http://www.cttransit.com/about/developers/gtfsdata/Main.asp>

Currie (2004) Gap analysis of public transport needs: measuring spatial distribution of public transport needs and identifying gaps in the quality of public transport provision. *Transportation Research Record* 1895, 137-146.

Dijkstra, E.W. (1959) A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* Vol 1: 269-271.

Deka, Devajyoti (2004) Social and Environmental Justice Issues in Urban Transportation. *The Geography of Urban Transportation*, 3rd Edition. Hanson, Susan and Guiliano, Genevieve. Guilford Press, New York. pp 332-355.

Delbosc and Currie (2011) Using Lorenz curves to assess public transport equity. *Journal of Transport Geography* 19: 1252-1259.

Desaulniers, Guy and Mark Hickman (2007). *Public Transit. Handbook in OR and MS*. Vol. 14 Chapter 2: 69-127.

Jennifer Duthie and S. Travis Waller. (2008) Incorporating environmental justice measures into equilibrium-based network design. *Transportation Research Record* 2089: 58-65.

Eliasson, Jonas and Lars-Goran Mattsson. (2006) Equity effects of congestion pricing: Quantitative methodology and a case study for Stockholm. *Transportation Research: Part A*. 40: 602-620.

Wei Fan and Randy B. Machemehl. (2006) Optimal Transit Route Network Design Problem with Variable Transit Demand: Genetic Algorithm Approach. *Journal of Transportation Engineering*, Vol. 132 (1): 40-51.

Wei Fan and Randy B. Machemehl. (2011) Bi-Level Optimization Model for Public Transportation Network Redesign Problem: Accounting for Equity Issues. *Transportation Research Record*.2263: 151-162.

Farahani, Reza, Miandoabchi, Elnaz, Szeto, W.Y., and Rashidi, Hannaneh. A review of urban transportation network design problems. *European Journal of Operations Research*. 229(1), 2013. pp.281-302

Farber, Steven, Melinda Z. Morang, and Michael J. Widener (2014) Temporal Variability in transit-based accessibility to supermarkets. *Applied Geography*, Vol. 53, pp 149-159.

Federal Highway Administration (FHWA) (2006). FHWA University Course on Bicycle and Pedestrian Transportation. Lesson 8: Pedestrian Characteristics. Publication Number: FHWA-HRT-05-099. Website:
<https://www.fhwa.dot.gov/publications/research/safety/pedbike/05085/chapt8.cfm>

Federal Transit Administration. (2012) Title VI Requirements and Guidelines for Federal Transit Administration Recipients. FTA C 4702.1 B.
http://www.fta.dot.gov/documents/FTA_Title_VI_FINAL.pdf

Erin M. Ferguson, Jennifer Duthie, Avinash Unnikrishnan, S. Travis Waller (2012) Incorporating equity into the frequency-setting problem. *Transportation Research Part A* 46:190-199.

Google (2015). Google Developers: Static Transit Overview. Website Accessed: November 28, 2015. <https://developers.google.com/transit/gtfs/?hl=en>

Grava, Sigurd. *Urban transportation systems: choices for communities*. McGraw-Hill, New York, 2003

Guiliano, Genevieve (2005). Low Income, Public Transit, and Mobility. *Transportation Research Record*, Vol. 1927, pp 63-70.

Hanson, Susan and Guiliano, Genevieve. *The Geography of Urban Transportation*. Guilford Press, New York, 2004.

Haupte, Randy L. and Haupte, S.E. (2004) The Binary Genetic Algorithm. In: *Practical Genetic Algorithm*, 2nd edn. Wiley, Hoboken, NJ, pp 27-50

Jansuwan, Sarawut, Keith M. Christensen, and Anthony Chen (2013) Assessing the Transportation Needs of Low-Mobility Individuals: Case Study of a Small Urban Community in Utah. *Journal of Urban Planning and Development*, Vol 139(2), pp 104-114.

Joshi, Nilesh N. and James H. Lambert (2007) Equity Metrics With Risk, Performance, and Cost Objectives for the Prioritization of Transportation Projects. *IEEE Transactions on Engineering Management*, Vol 54 (3): pp 539-547.

- Kain, John F. (1968) Housing segregation, negro employment, and metropolitan decentralization. *Quarterly Journal of Economics*, Vol. 82(2), pp. 175-197.
- Kenyon, Susan. Transport and Social Exclusion: access to higher education in the UK policy context. In *Journal of Transport Geography*. 19. 2011. pp.763-771
- Khisty, C. Jotin (1996) Operationalizing concepts of equity for public project investments. *Transportation Research Record*. 1559: pp 94-99.
- Knox, Paul L. and Pinch, Steven. *Urban Social Geography: an Introduction*. Pearson Prentice Hall, New York, 2010.
- Kruskal, J. B. (1956) On the shortest spanning subtree of a graph and the travelling salesman problem. *Proceedings of the American Mathematical Society*, Vol 7, pp 48-50.
- Liu, Yangwen and Cinzia Cirillo (2015) Measuring transit service impacts on vehicle ownership and use. *Public Transport*, Vol 7(2), pp 203-222.
- Leguizamon, Guillermo, Chistian Blum, Enrique Alba (2007) Evolutionary Computation. In: Teofilo F. Gonzalez (eds) *Handbook of Approximation Algorithms and Metaheuristics*. Chapman & Hall, Boca Raton, FL, USA. Pp 24-1 – 24-14.
- Ting L. Lei, yali Chen, and Konstadinos G. Goulias. (2012) Opportunity-Based Dynamic Transit Accessibility in Southern California: Measurement, Findings, and Comparison with Automobile Accessibility. *Transportation Research Record* 2276: 26-37.
- Nicholas Lownes, Nicholas Hart, Jeffrey Osleeb, Timothy Becker, Curtis Denton, Kelly Bertolaccini, Sha Mamun, Eric Jackson. (2013) The Connecticut Title VI Workshop: Building a Coordinated Strategy, CTI Report No. CTI-1-2013, Storrs, CT.
- Mamun, Lownes, Osleeb, Bertolaccini (2013) A method to define public transit opportunity space. *Journal of Transport Geography* 28: 144-154.
- Mattson, Jeremy (2011) Transportation, Distance, and Health Care Utilization for Older Adults in Rural and Small Urban Areas. *Transportation Research Record*, Vol. 2265, pp. 192-199.
- Mauttone, Anttone (2009) A route set construction algorithm for the transit network design problem. *Computer & Operations Research*. Vol 36(8):pp 2440-2449.
- Metropolitan Atlanta Rapid Transit Authority (MARTA) (2015). MARTA developer resources. Website accessed: November 27, 2015. <http://www.itsmarta.com/developers/data-sources/general-transit-feed-specification-gtfs.aspx>
- C.E Miller, A.W. Tucker and R.A. Zemlin (1960) Integer Programming Formulations and Traveling Salesman Problems. *Journal of Association for Computing Machinery* 7: 326-329.

Nagata, Yuichi and Kobayashi, Shigenobu (1999). Analysis of edge assembly crossover for the travelling salesman problem. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics v3: p III-628-III-633

Nagata, Yuichi (2007). Edge Assembly Crossover for the Capacitated Vehicle Routing Problem. Lecture Notes in Computer Science. Vol 4446 LNCS: pp142-153.

National Traveling Salesman Problems (2009). Qatar. Website Accessed: November 12, 2015. <http://www.math.uwaterloo.ca/tsp/world/countries.html>

Nazem, Mohsen, Martin Trepanier, and Catherine Morency. (2013) Integrated intervening opportunities model for public transit trip generation-distribution. *Transportation Research Record*, Vol 2350, pp. 47-57.

Nassir, Neema, Alireza Khani, Sang Lee, Hyunsoo Noh, and Mark Hickman (2011) Transit Stop-Level Origin-Destination Estimation Through Use of Transit Schedule and Automated Data Collection System. *Transportation Research Record*, Vol. 2263, pp 140-150.

Nayeem, Muhammad Ali, Rahman, Md. Khaledur, Rahman, M. Sohel. (2014) Transit network design by genetic algorithm with elitism. *Transportation Research Part C* Vol 46: pp 30-45.

Newman, Peter and Kenworthy, Jeffrey. *Sustainability and cities: Overcoming automobile dependence*. Island Press. Washington, D.C. 1999.

Pattnaik, S.B. and S. Mohan (1998) Urban Bus Transit Route Network Design Using Genetic Algorithm. *Journal of Transportation Engineering* Vol 124 (4): pp 368-375

Potvin, Jean-Yves (1996) Genetic Algorithms for the traveling salesman problem. *Annals of Operations Research* 63:339-370.

Rotger, Gabriel Pons and Thomas Sick Nielsen (2015). Effects of Job Accessibility by Public Transport System: Natural Experimental Evidence from the Copenhagen Metro. *European Journal of Transport and Infrastructure Research*, Vol 15(4), pp. 419-441.

Rego and Glover (2002) Local Search and Metaheuristics. In: Gutin and Punnen (Eds) *The Traveling Salesman Problem and Its Variations*. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp 309-369.

Sandoval, J.S. Onesimo, Robert Cervero, and John Landis (2011) The transition from welfare-to-work: How cars and human capital facilitate employment for welfare recipients. *Applied Geography*, Vol. 31(1), pp 352-362.

Samuel, Preethy S., Krim K. Lacey, Chesley Giertz, Karen L. Hobden, and Barbara W. Leroy (2013) Benefits and Quality of Life Outcomes From Transportation Vouchers by Adults with Disabilities. *Journal of Policy and Practice in Intellectual Disabilities*, Vol 10 (4), pp 277-288.

Syed, Gerber, and Sharp. Traveling Towards Disease: Transportation Barriers to Health Care Access. In *Journal of Community Health*. 38. 13 March 2013. pp 976-993.

Transportation Network Test Problems (2013). Website accessed: November 24, 2015.
<http://www.bgu.ac.il/~bargera/tntp/>

University of Connecticut, Transportation Logistics and Parking Services (UConn) (2015). Transportation Services. Website Accessed: November 23, 2015.
<http://transpo.uconn.edu/#home>

University of Connecticut, Transportation Logistics and Parking Services (UConn) (2015). Parking Services. Website Accessed: November 23, 2015. <http://park.uconn.edu/wp-content/uploads/sites/112/2015/09/uconnparkingmap.pdf>

Walker, Keane, and Burke. Disparities and access to healthy food in the United States: A review of food deserts literature. In *Health & Place*. 16. April 2010. pp.876-884,

Windham Region Transit District (WRTD) (2015). Website accessed: November 23, 2015.
http://www.wrttd.net/downloads/CB_Schedule.pdf.

Wong, James (2013) Leveraging the general transit feed specification for efficient transit analysis. *Transportation Research Record*, Vol 2338, pp 11-19.

Z.X. Wu, W.H.K. Lam and Hai-Jun Huang (2008). Equity and Efficiency Analysis of Pricing Strategies in a Bimodal Network with Heterogeneous User Groups. In *Transportation Research Record: Journal of the Transportation Research Board*, No.2089, Transportation Research Board of the National Academies, Washington, D.C., pp 43-50.

Yi, Chang (2006). Impact of Public Transit on Employment Status: Disaggregate Analysis of Houston, Texas. *Transportation Research Record*, Vol. 1986, pp 167-144.

Yen, Jin Y. (1971) Finding the K Shortest Loopless Paths in a Network. *Management Science*. Vol. 17(11): pp.712-716.

Zhang, Ming, Qing Shen, and Joseph Sussman (1988). Job Accessibility in the San Juan metropolitan region: Implications for rail transit benefit analysis. *Transportation Research Record*, Vol. 1618, pp. 22-31.