Evaluating the Sustainability Impacts of Intelligent Carpooling Systems for SOV Commuters in the Atlanta Region

March 2022

A Research Report from the National Center for Sustainable Transportation

Diyi Liu, Georgia Institute of Technology

Angshuman Guin, Georgia Institute of Technology

National Center for Sustainable Transportation

Georgia Tech School of Civil and Environmental Engineering
College of Engineering

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. NCST-GT-RR-22-11	**2. Government Accession No.** N/A	**3. Recipient's Catalog No.** N/A
4. Title and Subtitle Evaluating the Sustainability Impacts of Intelligent Carpooling Systems for SOV Commuters in the Atlanta Region		**5. Report Date** March 2022 **6. Performing Organization Code** N/A
7. Author(s) Diyi Liu Angshuman Guin, Ph.D., https://orcid.org/0000-0001-6949-5126		**8. Performing Organization Report No.** N/A
9. Performing Organization Name and Address Georgia Institute of Technology School of Civil and Environmental Engineering 790 Atlantic Dr, Atlanta, GA 30332		**10. Work Unit No.** N/A **11. Contract or Grant No.** USDOT Grant 69A3551747114
12. Sponsoring Agency Name and Address U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology 1200 New Jersey Avenue, SE, Washington, DC 20590		**13. Type of Report and Period Covered** Final Report (January 2020 – December 2021) **14. Sponsoring Agency Code** USDOT OST-R

15. Supplementary Notes

DOI: https://doi.org/10.7922/G2348HPT
Dataset DOI: https://doi.org/10.5281/zenodo.6354825

16. Abstract

Community-based carpooling has the potential to alleviate traffic congestion and reduce the transportation carbon footprint. Once technology, communication, demographic, and economic barriers are overcome, community-based carpooling can be fully exploited. One of the major barriers to implementation is the difficulty of optimizing carpool formation in large systems. This study utilizes two different methods to solve the carpooling optimization problem: 1) bipartite algorithm and 2) integer linear programming. The bipartite method determines the maximum number of carpooling pairs given acceptable reroute costs and travel delays. The linear programming method defines the most optimal performance that minimizes the most vehicular travel mileage. These two methods are carefully compared to evaluate the carpooling potentials among single-occupancy vehicles based on the output of activity-based model's (ARC ABM) home-to-work single-occupancy vehicle (SOV) trips that can be paired together towards designated regional employment centers. The experiment showed that under strict assumptions, an upper bound of around 13.6% of such trips could carpool together. The results are promising in terms of higher-than-anticipated carpool match rates and the predicted decrease in total vehicle mileage. Moreover, the framework is flexible enough with the potential to act as a simulation testbed, to optimize vehicular operations, and to match potential carpool partners in real-time.

17. Key Words Carpooling, Demographics, Sustainability, Traffic, Modeling	**18. Distribution Statement** No restrictions.

19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 49	**22. Price** N/A

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

About the National Center for Sustainable Transportation

The National Center for Sustainable Transportation is a consortium of leading universities committed to advancing an environmentally sustainable transportation system through cutting-edge research, direct policy engagement, and education of our future leaders. Consortium members include: University of California, Davis; University of California, Riverside; University of Southern California; California State University, Long Beach; Georgia Institute of Technology; and University of Vermont. More information can be found at: ncst.ucdavis.edu.

Disclaimer

Acknowledgments

This study was funded, partially or entirely, by a grant from the National Center for Sustainable Transportation (NCST), supported by the U.S. Department of Transportation (USDOT) through the University Transportation Centers program. The authors would like to thank the NCST and the USDOT for their support of university-based research in transportation, and especially for the funding provided in support of this project. The authors would also like to thank their colleagues for their guidance and support.

NCST

Evaluating the Sustainability Impacts of Intelligent Carpooling Systems for SOV Commuters in the Atlanta Region

A National Center for Sustainable Transportation Research Report

March 2022

Diyi Liu, School of Civil and Environmental Engineering, Georgia Institute of Technology

Angshuman Guin, PhD, School of Civil and Environmental Engineering, Georgia Institute of Technology

[page intentionally left blank]

TABLE OF CONTENTS

List of Tables

List of Figures

List of Symbols and Abbreviations

ARC	Atlanta Regional Commission
ABM	Activity-based (Travel Demand) Model
TAZ	Traffic Analysis Zone
SOV	Single-Occupancy Vehicle
LP	Linear Programming
ILP	Integer Linear Programming
SN	Shareability Network

Evaluating the Sustainability Impacts of Intelligent Carpooling Systems for SOV Commuters in the Atlanta Region

EXECUTIVE SUMMARY

Community-based carpooling has more potential to help alleviate traffic congestion and reduce energy use during peak hours than ride-hailing services, such as Uber or Lyft, because community-based carpooling avoids deadheading operations. However, community-based carpooling is not fully exploited due to communication, demographic, and economic barriers. This thesis proposes a top-down computation framework to estimate the potential market-share of community-based carpooling, given the outputs of activity-based travel demand models. Given disaggregate records of commute trips, the framework tries to estimate a reasonable percentage/number of trips among commuters in single-occupancy vehicles (SOV) that can carpool together, considering spatiotemporal constraints of their trips.

The framework consists of two major procedures: (1) trip clustering; and (2) trip optimization. The framework tackles the problems associated with using large amounts of data (for example, the Atlanta travel demand model predicts more than 19 million vehicle trips per day) by following “split-apply-combine” procedures. A number of tricks and technologies (e.g., pre-computing, databases, concurrency, etc.) are employed to make the mass computing tasks solvable on a consumer grade desktop computer in a reasonable time.

Two different methods are established to solve the carpooling optimization problem. One method is based on the bipartite algorithm, while the other uses integer linear programming. The linear programming method estimates both the *systemic optimal performance* in terms of saving the most vehicular travel mileage, while the bipartite-based algorithm estimates one *Pareto optimal performance* of such system that pairs the greatest number of carpool members (i.e., maximum number of travelers that can use the system) given acceptable (defined by the user) reroute cost and travel delays. The performance of these two methods are carefully compared.

A set of experiments are run to evaluate the carpooling potentials among single-occupancy vehicles based on the output of activity-based model’s (ARC ABM) home-to-work single-occupancy vehicle (SOV) trips that can be paired together towards designated regional employment centers. The experiments show that under strict assumptions, an upper bound of around 13.6% of such trips can be carpooled together. The distribution of these trips over space, time, and travel network are thoroughly discussed. The results are promising in terms of finding carpooling and decreasing total vehicle mileage. Moreover, the framework is flexible enough with the potential to act as a simulation testbed, to optimize vehicular operations, and to match potential carpool partners in real-time.

Chapter 1. Introduction

1.1 Research Goal

The research is inspired by the following question: Given reasonable spatiotemporal limitations such as additional travel time for passenger pickup, what percentage of commute SOV commute trips can carpool together during the morning commute period? In addition, the research team is inspired by the following potential benefits:

1. Benefits in reducing traffic system's congestion burden during peak hours when applying such a carpool system
2. Sustainability benefits (e.g., decreasing the carbon footprint)
3. Redundancy and reliability of community-based carpooling system generated by a system-wide optimization algorithm
4. Attractiveness of such system to its users due to relief from driving

The research team developed an analytical framework to assess the first two issues outlined above. The analytical framework consists of a set of different methods that are developed to address the different aspects of the problem as well as address computational efficiency. By analyzing the trip outputs generated from the activity-based travel demand model (ABM), the framework can find a specific optimal carpool assignment plan, which can provide the upper-bound potential of the carpooling system.

In this case study, commute trip information is generated by Atlanta Regional Commission's (ARC's) ABM, one of the most advanced travel demand models in the country (Davidson, et al., 2010). The ABM uses the Coordinated Travel – Regional Activity Modeling Platform (CT-RAMP) as operationalized in the Citilabs® Cube model (Davidson, et al., 2010). CT-RAMP first generates a regional population of synthetic households using the PopSynIII model. Each synthetic household is assigned demographic characteristics (e.g., number of residents, household income, vehicle ownership, etc.) and is populated with synthetic persons (specified by such factors as gender, age group, employment status, etc.) to match Census and other demographic characteristics (Davidson, et al., 2010). The ABM uses these household and person parameters to predict each household's and individual's travel activities in the form of shared and individual tours and trips. For each trip, the output trip table contains detailed information such as departure time (in one-half-hour time bins), departure coordinates, trip purpose, travel mode, and specific members of the household sharing the trip. The Georgia Tech team has also implemented a new path retention feature (Zhao, 2021) for ABM that allows researchers to output the trip paths (the series of links traversed by each trip from origin to destination) created during the Frank-Wolfe traffic assignment algorithms, which is very useful in detailed spatial and temporal travel demand analysis.

1.2 Literature Review

1.2.1 Overview

This chapter presents literature review relevant to the modeling of the carpooling/ridesharing/dial-a-ride problem as well as issues in carpooling practices. This study is focused on a Daily Carpooling Problem (DCPP). According to Calvo, et al. (2004), DCPP is a specific case of routing problem with pickup, deliverables, and time window. Calvo, et al. (2004) viewed this as a special case of a Dial-a-Ride problem (DARP), and heuristics were applied in their study due to the complexity of the problem. In this project, the problem is also treated as DCPP and both heuristics and precise integer linear programming are applied to obtain a good solution. The project emphasizes on integrating the carpooling problem to the existing agent-based traffic demand models, thus providing a testbed for analyzing other alternative carpooling/vanpooling scenarios in future research.

The project focuses on community-based/home-based carpooling, which is typically less common than the work-based/employment-based case where carpooling happens among employees in one company. In practice, the work-based carpooling problem can usually narrow down the problem size to a few hundred or a few thousand trips, given that the number of employees of one employer that can carpool together to one location (e.g., hospitals or schools or factories) are limited. This project removes the employer-based limitation and looks at pairing workers from different companies who share approximately similar origins and destinations. This assumption is future-oriented considering involvement of autonomous vehicle, internet-of-things, etc. In other words, this model focuses on upper-bound capability of the carpooling system considering only system efficiency without any information friction.

There are two different ways of perceiving the carpooling problem. One method, named the Traffic Network Model (TNM), views the carpooling problem as clustering trips sharing similar trajectory in spatiotemporal dimensions. The other method is to build a graph named Shareability Network, in which nodes represent traveler with or without role assigned, and links represent their likelihood/utilities/cost of carpooling. Ideas from both methods are borrowed and implemented to create the analysis framework proposed in this study. The traffic network model is used for clustering trips and narrowing down searching space of carpools, while shareability networks are used to model proposed carpool schemes between travelers. In other words, this framework generates both carpool assignments with role and operation rules concurrently.

1.2.2 Traffic Network Model

Problems like carpooling can be viewed as clustering trips sharing similar portions of spatiotemporal trajectories. To do this, a certain trip can be described as a sequence of spatiotemporal recordings. If all trips are known, algorithms can be devised to match demand with supply in the travel system. Cruz, et al. (2015) proposed a method of clustering trips into groups by their trip trajectories based on the Optics algorithm. A more complex TNM integrates temporal attributes into the graph, by either setting dynamic link weights or expanding the graph along the time axis to form a directed acyclic graph (DAG). For example, Jamal, et al.

(2017) implemented the two versions of graphs to integrate temporal dependencies and integrate transit operations into the carpooling problem. Dijkstra family algorithms are used in either of the two networks to find the shortest paths. Carpool-matching models based on TNM focus on finding matches for individual trips, which emphasize a detailed operation problem with a small data sample size. The algorithms like Dijkstra used in these problems are computationally intensive and it suffers computational issues as the problem size grows.

1.2.3 Shareability Networks

The other type of graph model in carpool modeling is the shareability network model (SNM), a notion implemented by Santi P, et al. (2014). Unlike traffic network models, a SNM denotes each passenger/entity as a node/vertex, and each candidate carpooling match as a link that connects two points. The weight/distance of each link reflects the likelihood of two corresponding parties denoted by the link's two end nodes. The objective is to find the maximum number of links without two links sharing the connection between one pair of nodes (i.e., conflicts that two persons perform two roles). There are three major types of models to model the problem: bipartite model, monopartite model, and integer programming model.

Several previous efforts have applied these algorithms in carpool matching by using graph-matching algorithms. For instance, Zhang, et al. (2019) modeled the preference-matching problem in ridesharing as a monopartite problem. They assume people can be grouped into several different social categories and each group has some preferences to interact/carpool between groups. Different scenarios are assumed for the interaction behavior within or between groups, and the algorithm provides results for trip matching in both number and quality. The claim is that preference-based matching provides closer matches than efficiency-based algorithms. Qi, et al. (2016) developed a graph similar to monopartite graph but modeled the problem as that of finding complete subgraphs for multiple parties pairing.

It is obvious that the optimization problem of shareability networks can be generalized as (integer) linear programming. Alonso-Mora, et al. (2017) designed a new real time algorithm for routing and ridesharing to reach near optimal outcomes. Hsieh, et al. (2018) formulated the carpooling problem as an integer programming problem and develop variants of Derivative Evolution algorithm to solve it. They resolved the carpooling problem by decomposing the analysis into two procedures: the route-planning phase and carpooler-matching phase.

1.2.4 Adopted Methodology

In this NCST project, two methods are proposed in formulating the carpooling problem: (1) a bipartite model formulation, and (2) a pure integer linear programming formulation. Both methods are able to solve the optimal problem but with different objective functions. The first one, the bipartite model, leads to a solution with maximum number of carpooling pairs. The second method, the linear programming formulation, can have flexible constraints, and the most basic target is to minimizing total vehicular travel time over the system, which guarantees a global systemic optimal value.

One advantage of the proposed method is the concurrent generation of both the pairing plan and the specific routing plan. Using the outputs from the ARC's activity-based travel demand model, analysis starts with the individual commute trips, with routes represented as shortest path between the traveler's OD pair. In the travel demand model outputs, each path is recorded separately, regardless of whether the trip is defined as a carpool trip by the ABM (the model assigns one-half of a vehicle to each independent route, see Zhao (2021)). However, the separate paths give us a means to thoroughly compare and evaluate the impacts.

Chapter 2. Analysis Framework

Starting with the population and their travel data (e.g., ARC ABM outputs) as inputs, a computational framework is designed to estimate the number of carpool-able trips, as shown in Figure 1. This framework consists of two key modules: (1) trip clustering module; (2) trip optimization module.

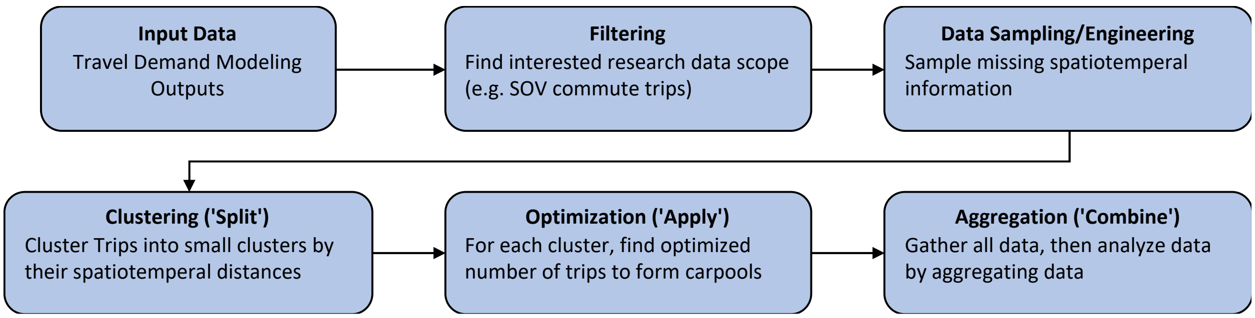

Figure 1. Proposed Carpooling Analysis Framework

The proposed framework is simple and interpretable since it aligns with the well-known idea of the “split-apply-combine” approach in data science/data engineering community. The “split” step refers to splitting a problem with big data size into many smaller groups with smaller data size, but with the same problem formulation. Since the efficiency of computation problems is typically very sensitive to data size, it is possible to gain speed and efficiency by splitting the problems into smaller problems with small data size and solving the smaller problems in parallel instead of solving a large problem at once in a serialized fashion. The “apply” step refers to applying a function to solve the problem for each individual group. The “combine” step is to gather outputs of every separate group and combining the outputs into a single format as the solution to the whole problem.

The first step of the proposed method is to filter out useless trips and cluster useful trips into many smaller groups. Only trips that fall into the same group are considered for pairing. The second step is to find all possible carpool-able combinations among travelers. The last step is to gather the solutions for each individual group, thus forming the solution to the whole problem.

2.1 Preparation Work (Data Filtering and Data Engineering)

2.1.1 Study Scope

In the current phase, the research focuses on estimating the number within the most appropriate population to study. The focus is on the drive-alone commuters (i.e., single-occupancy vehicles). This study further limits the destination of the trips to the employment centers identified by ARC. Trips are further limited by trip purpose to include commuting trips only (from home to work). These work commute trips typically are the major contributor to congestion during the morning peak period. An example of selecting data of interest is shown in Table 2.

2.1.2 Regenerate Precise Trip Information

Although ARC ABM provides individual trip outputs, there is a gap in precision between ABM ARC's travel demand modeling trip outputs and the demand for carpooling operation research. Model temporal resolution, spatial resolution, and path assignment all suffer from precision issues. This section discusses these limitations and presents methods for addressing them.

Temporal Resolution: ABM trip outputs trip departure times in 30-minute bins. The 30-minute time-bin is too wide for precise analysis of carpooling. Under normal circumstances, it is highly unlikely that a person will wait for 30 minutes to start a 15-45 minute commute trip via carpool. To provide a better representation of a realistic demand, a spline curve is fit to the departure time for all trips of interest generated by the ABM. Then, trips are sampled to generate a departure time with one-minute resolution, using the results in the fitted spline curve for the 30-minute time bin into which the trips fall.

Spatial Resolution: ABM trip outputs do not provide precise origin/destination locations for each trip. All trips are defined as starting at the centroid of a traffic analysis zone (TAZ) and arriving at the centroid of another TAZ. Hence, the origin and destination of the centroid coordinates are not the same as the longitude/latitude coordinates for the actual trip start location and trip end location (e.g., home and work). For a more realistic distribution of the start and end locations, a set of 100 randomly sampled locations within each traffic analysis zone is used and a random sampling process is employed to associate a coordinate location to a traveler's origin/destination location within the ABM designated TAZ.

Path Assignment: The path retention data provided by the ARC ABM only gives trip outputs between TAZ centroids. Carpooling trips sharing similar origins will be clustered into one TAZ centroid and the reroute cost for carpooling might be underestimated. To build a platform capable of analyzing realistic costs of carpooling, paths must be generated for the complete end-to-end OD trips. The study developed a new two-stage method for estimating reasonable traveling paths with minimal computational costs. The methodology is then further improved by dynamically generating carpool plans. The methods used to generate data are discussed in detail in the methodology section.

2.2 Trip Clustering Module

After establishing the scope of the study and refining the inputs, the next step is to find the maximum number of carpooling trips with a set of reasonable constraints. However, since such optimization problem is NP-complete, trips must first be clustered into smaller groups to enhance computational efficiency. Trips are clustered based on their vicinity in spatiotemporal dimensions. Only trips falling into the same cluster are considered for carpool combinations. Trips in two separate clusters are not considered for carpooled combinations due to the spatial gap in locations. For this framework, trips are clustered based on their proximity in origin location and departure time. This clustering assumption is justified for this problem since the study focuses on community-based carpooling, meaning that trips are carpool-able only if they share similar origin locations. The problem of whether a carpool is viable is further analyzed in the trip optimization step.

2.3 Trip Optimization Module

Once the target travelers/trips are clustered based on their spatiotemporal proximity, the system then only needs to optimize results within each cluster. After finding optimal results within each trip cluster, the solutions for all sub-problems can be aggregated to get the complete solution. Two algorithms are proposed to answer the optimization problem. One algorithm uses a bipartite algorithm plus heuristics rules. The other algorithm formulates a linear programming optimization problem and uses a LP solver to obtain the optimal results.

Assuming two travelers A and B are being considered for pairing in a carpool trip, there are four potential operation possibilities (this study only considers the first three options: 1, 2a, and 2b):

1. Both travelers drive alone from home to work (default operation).
2. If the travelers decide to carpool together, usually the driver picks up and drops off the passenger at the passenger's original and destination location respectively.
    a. First scenario: A is the driver and B is the passenger
    b. Alternative scenario: A is the passenger and B is the driver
3. Travelers A and B both drive to a midway point (e.g., park and ride) and carpool together

Chapter 3. Methodology

3.1 Data Processing

3.1.1 Resampling Departure/Arrival Time

ABM trip outputs provide the trip departure times in 30-minute bins. However, carpool pairing will require estimation of departure time by minute for a better representation of a realistic scenario. Instead of assuming that the trips are uniformly distributed over each 30-minute time bin, a spline curve fitting technique is used to assign higher resolution (1-minute) start times to the trips. Hence, more trips will depart earlier in the half hour bin if the trip generation in the previous half-hour is greater than the trip generation in the half-hour that follows. The following are the steps for generating the start times:

1. A histogram is plotted for the number of trip departures per 30-minute bin over all 48 time-bins during the analysis day
2. A spline curve is fitted to the histogram to obtain a departure distribution
3. For each trip in a 30-minute bin, a start time is assigned by sampling a minute out of the 30 minute bin as the individual trip's departure time, such that the departure time follows the probability distribution proportional to the slice of the spline curve within the particular 30-minute time bin.

Figure 2 shows an example of a trip departure histogram for the 48 modeled 30-minute bins and the spline curve fitted to the histogram.

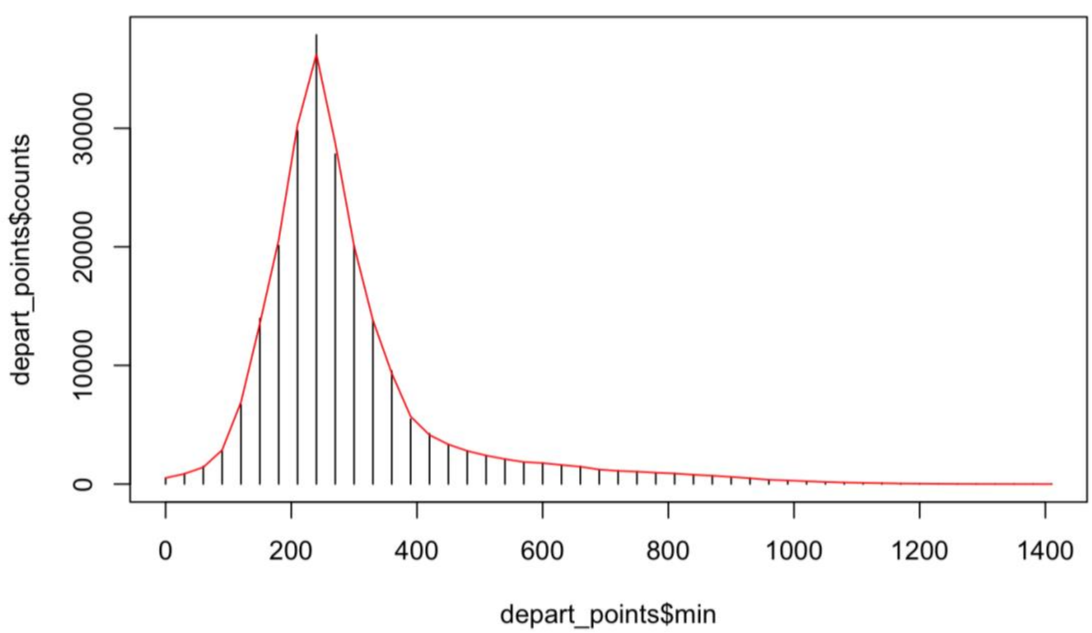

Figure 2. Histogram of Departure Time Over 30-minute Time Bins with its Spline Curved

3.1.2 Resampling Origin/Destination Coordinates

The ARC ABM travel demand model provides the departure and arrival TAZ numbers for each trip. The precise coordinate location of the trip (e.g., longitude/latitude) is not provided. Each trip is assumed to begin at the centroid and access the network through a centroid connector link (a hypothetical link that adds access distance and travel time to the on-network trip). Trips arriving at a destination traverse a similar centroid connector to the arrival TAZ centroid. Balancing the tradeoffs between computation complexity and precision, the following sampling plan is employed:

1. A set of 100 geometry coordinates/points within each TAZ parcel is generated by random sampling within the TAZ parcel extents. Each of the 100 locations are indexed from 1-100.
2. For each trip assigned to a TAZ centroid as the origin/destination, a randomly generated integer (1-100) is used to assign the origin or destination to the corresponding location within the TAZ, and the sub-TAZ location coordinates are assigned.

3.2 Shortest-path Estimation

3.2.1 Two-stage Shortest Path Estimation

The ARC ABM's path retention outputs provide paths between TAZ centroids for each trip (Zhao, 2021). Each TAZ centroid has at least one connector to the traffic network. In ABM, all trips departing from a same TAZ starts at the same location and usually uses the same connector to the nearest network. This simplification somewhat undermines the ability to analyze carpool operations. To make the carpool problem more realistic, all trip trajectories are re-computed using RoadwaySim, a shortest path calculation tool developed at Georgia Tech for the ARC ABM network using a Dijkstra-family algorithm.

Finding shortest path is computationally expensive using Dijkstra-family algorithm, which is especially true considering that the system is estimating millions of trips. A two-stage process is therefore used for finding the shortest path between each pair of origin and destination coordinates.

In the first stage, the system enumerates all possible TAZ origin-destination (OD) combinations. For each pair of TAZ centroids, the shortest path is estimated using RoadwaySim (creating the shortest path between any possible pair of TAZs). This pre-computed data is stored in a database for fast query efficiency.

After the first step, the shortest path between TAZ centroids is identified but the shortest path between precise OD coordinates is still unknown. The second step uses the pre-computed shortest path between TAZ centroids to approximate (rapidly) the shortest path between any pair of coordinates inside the corresponding pair of TAZs. The procedure steps of this approximation algorithm are listed in the following table:

Table 1. Two-Step Shortest Path Algorithm Procedures

Stage	Description
1.	For all combination of origins and destinations, compute the shortest link paths between the pair and store the results in a database for fast query.
2.	Given any pair of origin & destination coordinates: 1) Find the origin/destination TAZs to which the coordinates belong. 2) Obtain the shortest paths between origin/destination TAZ centroids from the database. 3) Temporarily set the weights of all links in the shortest paths to a very small positive number. 4) Rerun the Dijkstra's algorithm with the low weights helping to speed-up the computation. 5) Reset the original weights of all travel links in preparation for the next run. The result generated by step 4 is treated as the updated shortest path (which closely approximates the real shortest path).

Because Dijkstra's algorithm searches a priority queue, by setting costs of pre-computed shortest paths to zero, we prioritize the costs of the trips in the search frontier, making it much faster to find the destination node in the network. While the speed gain is easy to understand, it is not straightforward to show the optimality for the two-stage procedure. Because first stage paths include the shortest paths between the corresponding TAZs, the two-stage algorithm will generate sub-optimal shortest paths. This proposition is discussed in the Appendix. In practice, the algorithm generates high quality shortest paths most of the time.

3.2.2 Dynamic Shortest-path Estimation (CarpoolSim)

The same shortest path computation method can be extended to estimate travel paths for different trip segments of the carpooling trips. For example, assume that person A and person B are considering carpooling for the morning commute. The first scheme is the do-nothing scenario, where each person drives alone to work as they did before. In the second scheme, person A is assigned the role of driver with person B as passenger. A would depart from home, then pick-up B from B's home and drop off B at B's destination before A arrives at his/her destination. An alternative variation of the second scheme is to flip the roles and assign B as the driver with A as the passenger. In this way, the three possible operation schemes considered by the carpooling system are enumerated.

Figure 3 demonstrates these three basic carpooling scenarios. The values 1, 1', 2, and 2' represent person A's origin and destination and person B's origin and destination, respectively. The Georgia Tech team developed a computing framework called CarpoolSim with the capability to analyze the shortest paths for complex traveling cases like carpooling. CarpoolSim is run once for every potential carpool pair. The time/distance cost of each trip segment is computed and used as inputs to the optimization model.

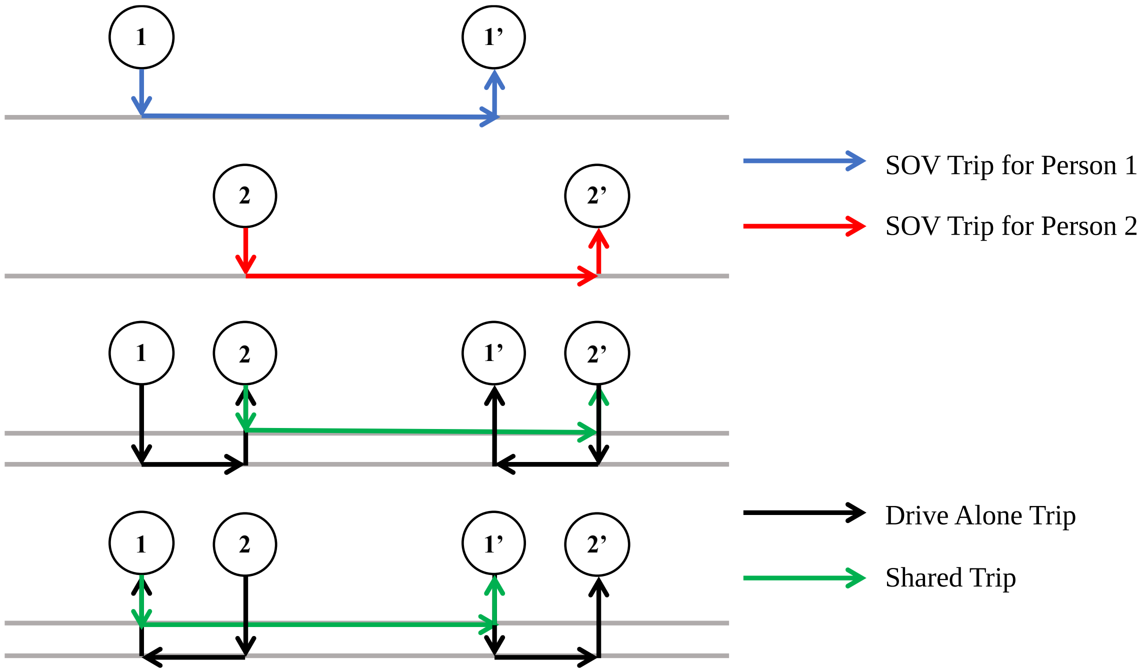

Figure 3. Three Basic Carpooling Assignments

In Figure 4 below, the triangular dots represent person A's origin and destination coordinates and the round dots represent the origin and destination coordinates of person B. Red represent the origins and the blue represent destinations. The thick blue portion of the line shows the carpooled portion of the trip, while the black arrows show the driver's whole trip. Even when two trips start with the same coordinates, the traveling schemes result in very different paths. Driver and passenger roles are very important in carpool assignment.

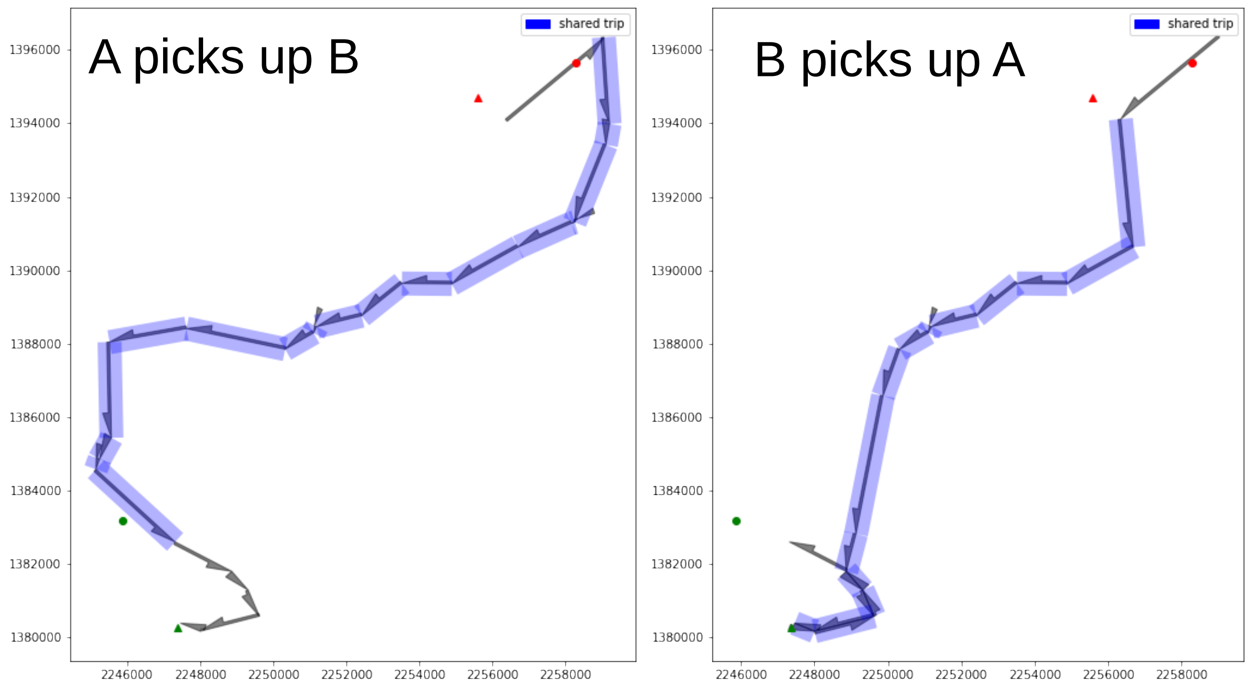

Figure 4. Visualization of the Carpool Paths Generated by CarpoolSim

3.3 Trip Spatiotemporal Clustering

The number of carpooling pairwise assignments grows as a polynomial function (i.e., $O(n^2)$), as the number of trips (n) grows. Hence, it is important to cluster trips into smaller groups to enhance computational efficiency. The concept here is to only cluster trips that share similar depart times and origin-destination locations. Trips are first clustered based on location information. For each location-grouped cluster, trips are then clustered based on temporal data (i.e., difference in departure time). The spatiotemporal clustering is formulated as the joint intersection of the two separate schemes.

In the programmatic implementation, trips are first clustered by their origin TAZs. Within each group, trips are clustered using the DBSCAN method, by setting search radius parameters as a maximum tolerated departure time difference (e.g., 5-min), and the minimum neighboring number parameter (n) equals two (two neighbors within a cluster). In this way, we generate a number of clusters, and only trips within the same cluster can be considered for potentially carpooling together. In other words, if two trips are in two different clusters, it is assumed that the trips are not likely to be carpooled together. In the implementation, many trips are clustered into an isolated cluster that contains only the one trip; hence, the optimization process is not needed for that trip. The next phase of this research will add another layer of complexity by relaxing the grouping constraints, allowing carpool pickups and carpool drop-offs at any TAZ along the path, as long as the deviation does not add too much additional travel time to the original trip (the threshold can set to any desired impedance).

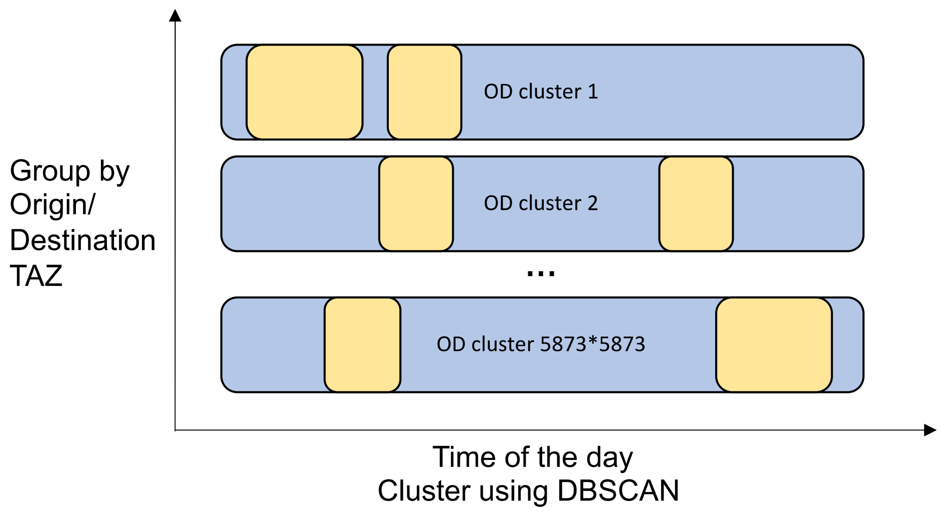

Figure 5. Demonstration of a Basic Trip Clustering Method

Location clustering over origin TAZs is reasonable given that this study is considering community-based carpooling, and some communities may be contained within a TAZ. However, the major disadvantage is that this clustering method ignores cases where a driver crosses a TAZ boundary to pick up a passenger in a neighboring TAZ. Future work (scheduled for fall 2022) will assess potential carpool joins across TAZs that fall within a set of time and distance tolerances. However, geographic proximity does not always lead to efficient routing. In reality, one might need to reroute for more than a mile just to pick up a passenger in an adjacent TAZ. Additional shortest path routes will also need to be developed for local roads to support adjacent TAZ origin pairing. To limit the scope of the problem, this preliminary study only clusters trips by departure TAZs.

After clustering the trips into groups, the next step is to solve the carpool optimization problem within each trip cluster. A feasibility matrix is used to store measurements between proposed trip pairs. Two methods are developed, one empirical method and one formal integer linear programming method. They will be discussed in section 3.5.

3.4 Feasibility Matrix

Given a trip cluster with m travelers, there are $m \times (m-1)$ number of possible carpooling assignments considering all combinations with roles. Notice that there are also m original SOV travel plans. We can use a single feasibility matrix with size $m \times m$ to denote all these potential travel plans. This feasibility matrix is used for each cluster. Only the trips within the same cluster can be considered as carpool trips. Trip pairs between cluster groups will never be considered to form a carpool.

Assume *F* denotes the feasibility matrix. Given m travelers in the system, the feasibility matrix is a $\langle x \cdot x \rangle$ square matrix, where the i^{th} row denotes the assignments where traveler *i* is the driver, and the j^{th} column denotes the assignments where traveler *j* is the passenger. The matrix entry *(i, j)* denotes a carpool relationship where the driver *i* is carpooling with the passenger *j*. The diagonal entries *(i, i)* are used to denote the original drive alone trips. For any entry, the value 1 denotes two trips that are carpool-able and the value 0 denote the opposite case. Analysts can employ multiple filters to identify as many infeasible carpooling plans as possible. One simple condition is that only when the departure time difference is less than 15 minutes, can the two SOV trips be considered to form a carpool trip. The 15 minutes threshold is an assumption for this study and should be verified in future commuter preference studies. Other potential filters that are not considered in this study might include demographic characteristics, personal preferences, etc. For example, person A_i might want to form a carpool only if they are driving their own vehicle. Then, we can set all entries along the i^{th} column to zero (except its own diagonal value *(i, i)*), eliminating the assignments where the traveler forms a carpool as a passenger.

To further tighten up the feasibility matrix and eliminate spurious or borderline, low-likelihood carpool trips, additional filters are employed. In addition to clustering trips into groups, the elimination of spurious "1"s in the feasibility matrix helps shrink the search space of the problem, reduce the likelihood of testing inefficient matches, and reduce computation times.

Similar to the feasibility matrix, there are matrices that store other parameters such as total vehicular travel time, absolute difference of travel departure time, etc. Additional filtering criteria are used to filter out carpooling assignments that are unlikely to happen. The following filters are employed:

- The difference in original departure time is less than a threshold (e.g., five minutes)
- For the proposed carpool trip, the driver's extra cost (expanded reroute traveling time) is less than a threshold (e.g., 15 minutes)

After generating a feasibility matrix that considers multiple conditions, the matrix can be used as an input to both the bipartite method and linear programming method. The matrices storing other values (e.g., vehicular travel time) are also used as input. These matrices and the feasibility matrix together are all called shareability matrix, which means they provide a measure of the utility/cost/possibility between proposed carpooling assignments with role.

3.5 Two Methods for Optimal Solutions

3.5.1 Bipartite Method

To solve the problem efficiently, the researcher proposed an empirical method borrowing bipartite algorithm plus heuristic decision rules to find the optimal solution (Figure 6). The bipartite method can be reduced to a maximum-flow problem. The problem's objective is: 'given the feasibility matrix, find the maximum number of traveling carpool pairs.' The proposed algorithm is shown as follows:

Step 1. Assume each person can be both driver and passenger at the same time, and based on the feasibility matrix, generate a bipartite pairing problem to find the maximum number of trip pairs. Notice that the assumption (i.e., one can be both driver and passenger at the same time) is untrue, but the conflicts caused by this assumption will be solved later using heuristics.

Step 2. To satisfy the bipartite algorithm assumption, one person cannot play two roles at the same time in two separate carpool trips. When there is a conflict, the directed graph can either form a singly linked chain or a loop (i.e., singly linked chain with both ends connected), as shown in Figure 7 as an example. In other words, it cannot form any structure with one node having three connections to other nodes. In this way, we can greedily solve conflicts from the start of the chain and get the optimal assignments of finding the maximum number of carpooling pairs.

Bipartite method schematics are shown in Figure 6, Figure 7, and Figure 8. Figure 6 shows an example of a feasibility matrix for a carpool network (as a bipartite problem), and the conversion of this bipartite method to a Max-flow problem. The paths of the Max-flow problem solution can be used to construct a graph that connects travelers, as shown in the right plot in Figure 7.

By observing this Max-flow problem, it has two optimal solutions shown by Figure 7. One only contains chains and the other only contains loops. Figure 8 gives the heuristics rule to break up conflicts related to the fact that one cannot become driver and passenger at the same time. Essentially, the rule is to de-select links in turn until there are no further conflicts. Because the graph structure is limited to singly linked chain or loop, the rule ensures the retention of the largest number of carpool trips.

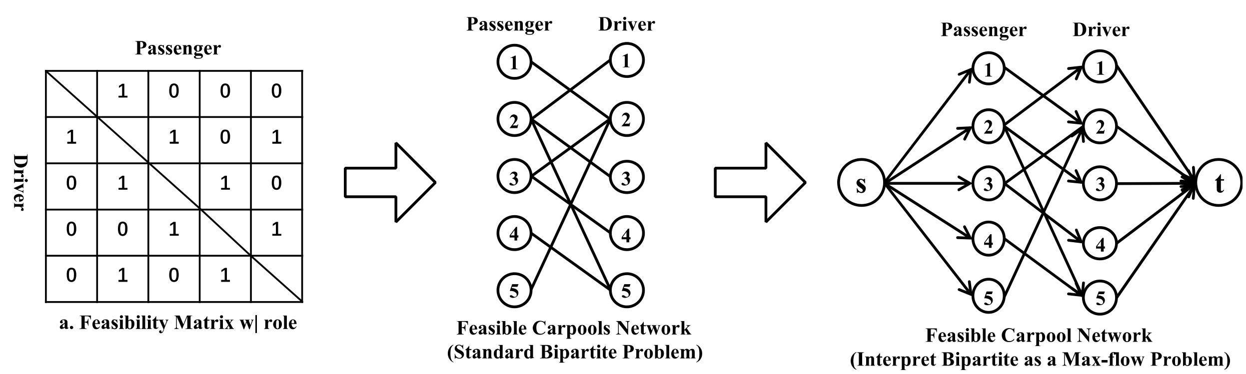

Figure 6. Use Bipartite Method to Represent the Feasibility Matrix

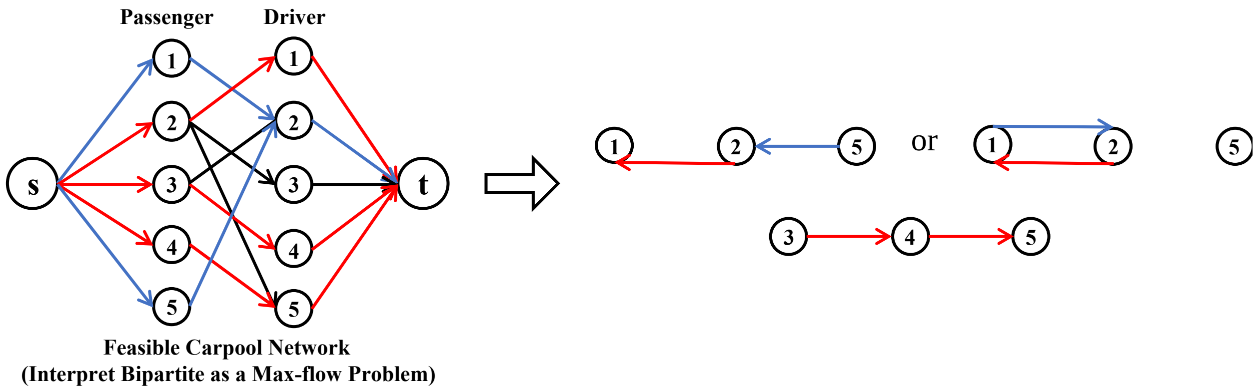

Figure 7. Two Optimal Outputs by Bipartite Algorithm

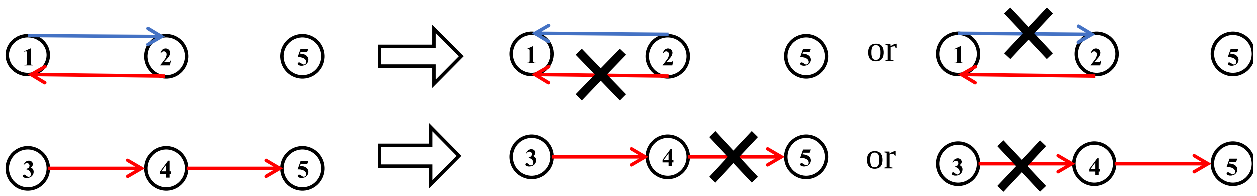

Figure 8. Heuristics to Break Up the Role Assignments Conflicts

Proposition 2. In the carpooling assignment problem, the bipartite algorithm only outputs singly linked chains or simple loops with both ends connected. One can index the links as [1, 2, …, n]. By preserving odd or even indexed links, and discarding the other links, one can guarantee that one of the assignments finds the optimal solution in terms of pairing maximum number of travelers together.

Proposition 2 is easy to understand in the max-flow context: where link costs all equal 1, any traveler node can at most be identified as a passenger once and as a driver once. Otherwise, the system would violate the max flow constraints of 1 between node source/target node and any traveler node.

3.5.2 Linear Programming Formulation

The bipartite algorithm only tries to find the maximum number of carpool trips. However, to further study this optimization problem, we need to set more flexible objective functions. For example, one might want to minimize total vehicular travel time in the system. Alternatively, one might want to maximize total utility for the system. Moreover, the constraints need to be flexible. For example, the user could constrain the utility costs of carpooling to be no more than 10 percent higher than for people driving alone. The bipartite algorithm, although efficient, only aims at finding the maximum number of carpools. Hence, a linear programming solution will provide more flexibility and can solve the problem using a linear programming solver.

For linear programming, the most basic objective function is to minimize the total vehicular travel time. A linear programming formulation is generated as follows:

Objective function:

Minimize: $\sum_{ij} x_{i,j} \times c_{i,j}$ (3.1)

where: $\forall\ (i, j)$ denotes all valid carpool plans between the driver i and passenger j. For $i = j$, it denotes the case of driving alone. $c_{i,j}$ denotes the travel cost for the traveling case denoted by (i, j).

Subject to:

Decision variables:

$x_{i,j} \in \{0, 1\}$ (3.2)

Role assignment constraints:

$\sum_j x_{i,j} \leq 1, \forall\ i \in \{1, \ldots, M\}$ (3.3)

$\sum_i x_{i,j} \leq 1, \forall\ j \in \{1, \ldots, N\}$ (3.4)

$\left(\sum_i x_{i,K} + \sum_j x_{K,j}\right) - x_{K,K} = 1, \forall\ K$, such that $x_{K,K}$ exists (3.5)

In the experiments, the cost $c_{i,j}$ is simply defined as the vehicular travel distance for the trip. Equations 3.2-3.5 describe the constraints of the integer linear programming problem. Equation 3.2 defines the decision variables for each feasible carpool pair between driver i and passenger j. Equation 3.3 denotes that for any driver (indexed from 1 to M), he/she can only join in at most one carpooled trip as a driver during the commute trip. Similarly, Equation 3.4 refers to the fact that any passenger (indexed from 1 to N) can only participate in at most one carpooled trip. Furthermore, in our study, since those originally SOV commuters can choose between the role of driver and passenger, another constraint is that one person cannot be the driver and passenger at the same time. This constraint is reflected in equation 3.5. In other words, if a person is already assigned to as a driver in a carpool assignment, then this person cannot be driver or passenger in other traveling plans. Combining all the components from equations 3.1 to 3.5, the integer linear programming problem is formulated to minimize the total cost. In the experiments, the cost $c_{i,j}$ is simply defined as the vehicular traveling distance for the trip. Under this configuration, the linear programming problem solution provides the optimal results by minimizing total vehicular mileage of the whole system.

Both the bipartite algorithm and the linear programming algorithm are implemented and tested in Chapter 4 and the results generated by these two methods are compared.

Chapter 4. Experiment

Two experiments are undertaken using trip outputs of Atlanta Regional Commission's activity-based model (ABM). The trip outputs are generated from the ABM15 model version of the ABM for the calendar year 2020 scenario. As described in the last chapter, not all trips are considered for carpooling. Instead, only single-occupant-vehicle commute trips towards designated employment centers are considered in this experiment. The large "employment centers" are designated TAZs identified by the Atlanta Regional Commission (ARC, 2017). The name and location of these centers can be seen in Figure 9. Use of a limited number of high-demand destinations makes the research results easier to interpret and aligns with the community carpooling study topic. To demonstrate the efficiency of the algorithms, all of the computation steps are run on a computer with 32-GB memory and a 2.4 GHz 8-Core processor.

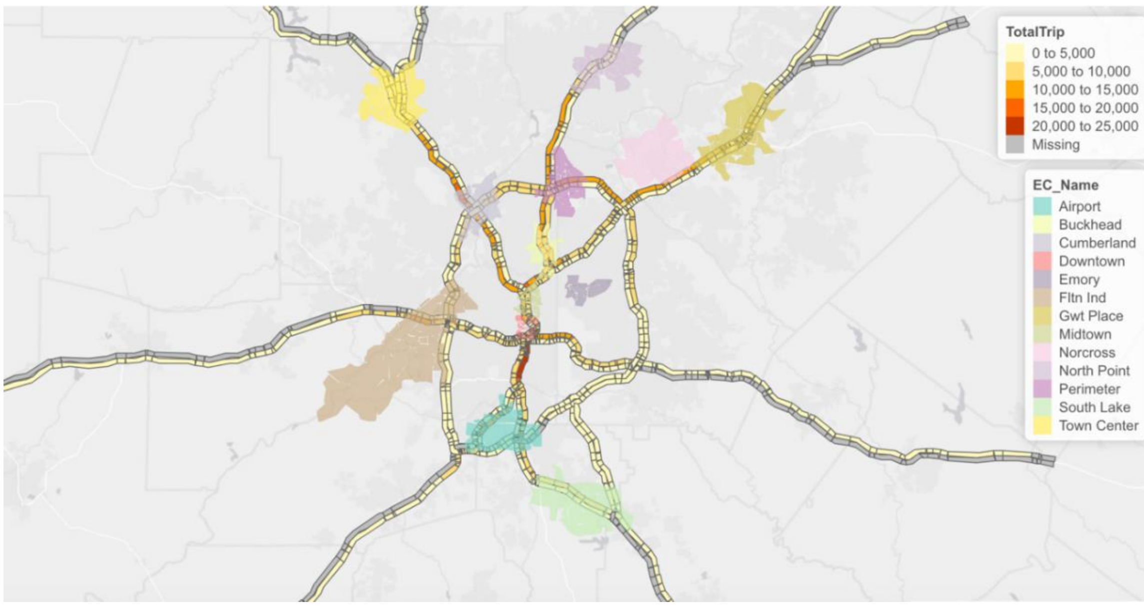

Figure 9. Employment Centers Identified by ARC

4.1 Experiment Settings

Only a small percentage of the entire set of trips is fed into the final step of the analysis framework. The filtering criteria described in the previous sections help eliminated the trips with extremely low likelihood for carpooling. Table 2 provides a summary of the percentage of trips eliminated in each filtering step. More than 19 million vehicle trips are output by the ABM. Multiple filters are applied in series to remove trips that are outside of the project scope. Of the 19 million total vehicle trips, the number of trips with an employment center destination shrinks the trip total to 5,329,225 (27.7% of total trips). Among those trips, 2,293,265 (11.9% of total trips) are SOV trips. Among these SOV trips, 1,147,379 trips (5.96% of total trips) are for work (commuting trips). Among these work-related trips, the number of trips that are from work to home (instead of home to work) is 432,043 (2.3% of total trips). In summary, 432,043 trips start from home, ends at workplace, and the destination workplace is in a TAZ corresponding to an employment center. Since the focus is on inbound travel towards these designated employment centers, the number goes down further to 385,880. These remaining 385,880 trips are used as the input of the analysis framework.

Table 2. Counts and Percentage of Trips in Each Screening Step

Trip Conditions	Trips	Percent Retained from Previous Step	Overall Percent Retained
Total daily trips	19,235,737		100%
Trips involving ECs	5,329,225	27.7%	27.70%
SOV trips	2,293,265	43.0%	11.92%
Work-related trips	1,147,379	50.0%	5.96%
Home-to-work trip	432,043	37.7%	2.25%
Destination TAZ is an EC	385,880	89.3%	2.0%

4.2 Disaggregate-level Results Analysis

Using the same settings, the problem is run using both the bipartite algorithm (with heuristics) and the linear programming algorithm. To compare the performance of these two algorithms, all steps before optimization use the same data set. Both algorithms are able to provide a solution with approximate aggregate numbers. To compare the difference between the linear programming outputs and the bipartite method outputs, one method is to compare the quality of the paired data (i.e., the statistical traits across the individual carpool plans).

The first step is to look into some trip clusters samples and visualize pairing results over the shareability network. Figure 10 shows the traveling cluster between traffic analysis zone (TAZ) #1557 and TAZ # 1583. Within the scope of interest, there are 60 trips between TAZ 1557 and TAZ 1583. As shown in the following graph, each node denotes a travel demand, and each link denotes a feasible carpooling plan.

The shareability network is constructed in a way that trip A can pick up trip B if trip A departs within a 15 minutes window ahead of the departure time of trip B. The x-axis of the graph denotes the trip's depart time while the y-axis is provided for visualization (in this example, a sinusoid function is used as input to the data index). The time difference of 15-minutes is used with the assumption that the proposed carpooling system is not a real time dispatch application, but is rather the result of negotiations between parties before the commute day. Notice that the shareability network is such that even when a time difference is set to 5 minutes in Figure 10 (b)-(c), there are still many red lines (i.e., valid carpool trips).

The bipartite method picks the assignment plans indexed either as odd or even. By contrast, the linear programming algorithm makes assignments that maximize travel time savings.

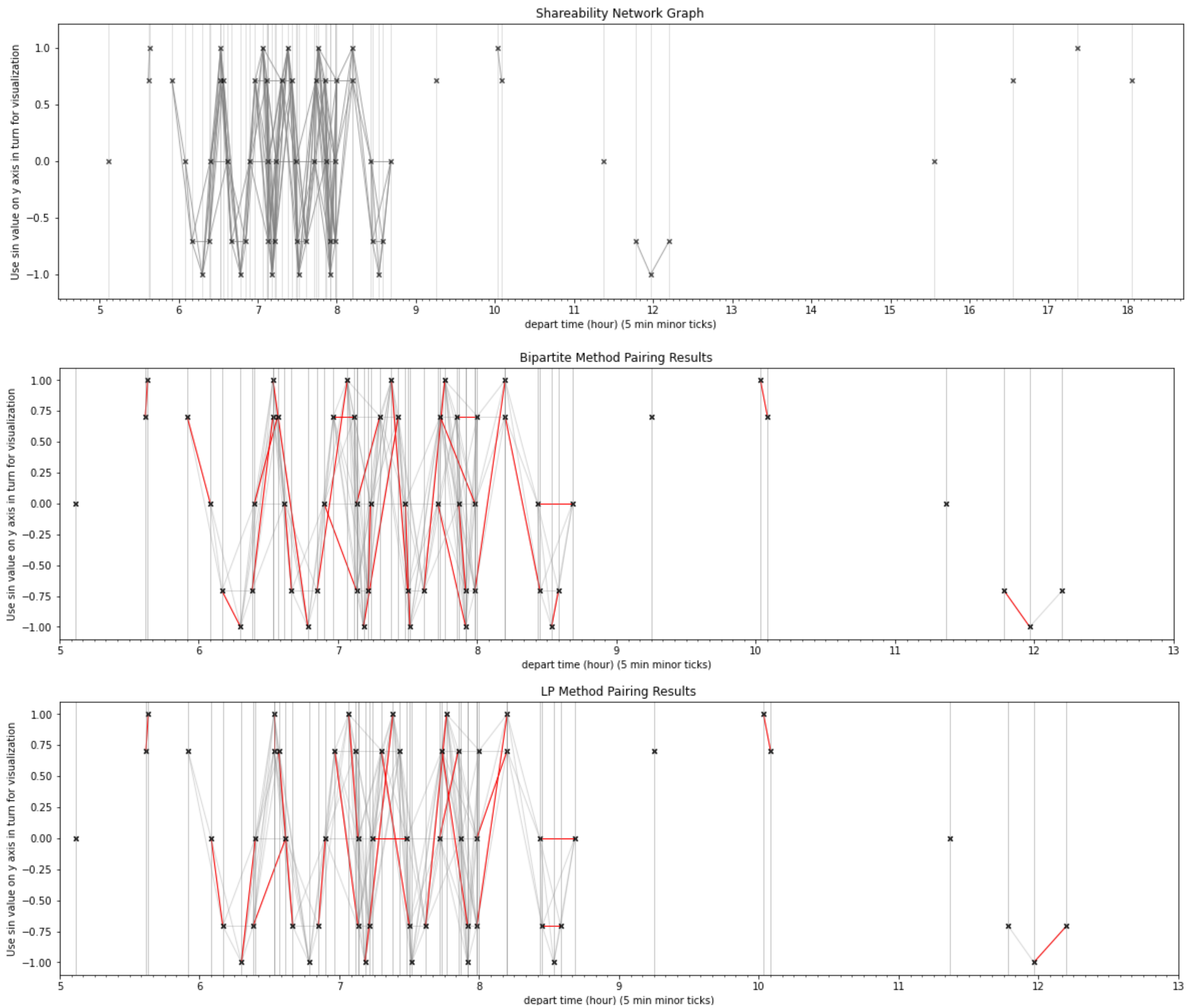

Figure 10. Shareability Network and the Comparison of Bipartite Results and Linear Programming Results

The travel time for carpool passengers remains unchanged in the system since only the drivers need to reroute to pick up and drop off passengers. Hence, only the travel changes for these carpool drivers are discussed. For the purposes of the discussions that follow, all SOV case before assigning carpools are identified as the “before” case and the scenario after running the carpool assignment are defined as the “after” case. In this way, two sets of figures are plotted as shown in Figure 11: 1) before-after comparison of travel time for carpooling drivers; 2) reroute time for carpool drivers. The two graphs are logically related but emphasize on different contexts. The travel time plots focus on the overall performance of the carpooling system. The reroute time emphasizes the extra costs to the carpooling drivers. Notice that the before-case travel time histograms (i.e., the blue histograms in the plots on the left) are not exactly the same for the different algorithms, because different algorithm may assign different SOV trips to form different carpools.

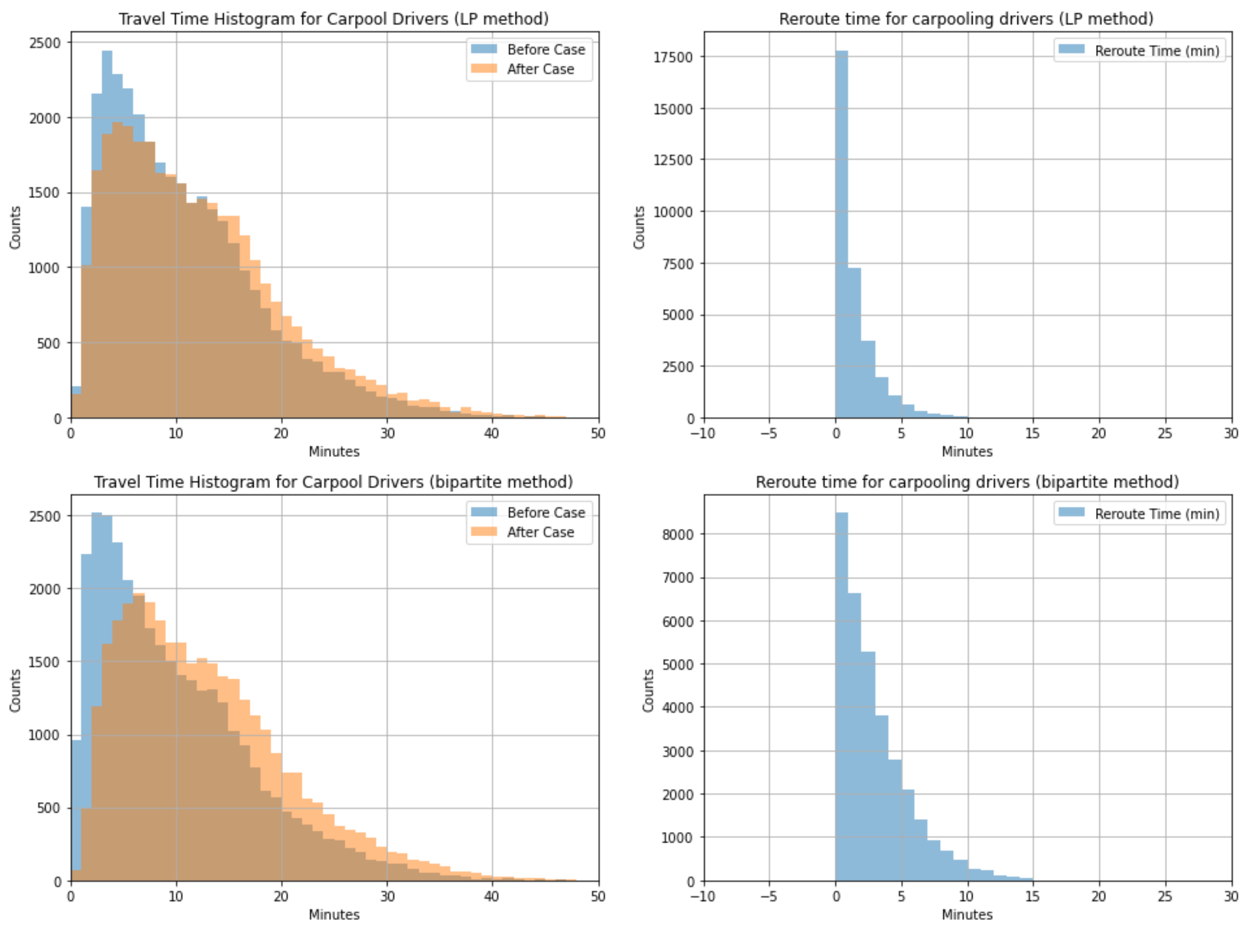

Figure 11. Change of Traveling Time among After-case Carpooling Drivers

While the overall distributions are not very different in shape, the reroute time distribution clearly shows that the linear programming method provides better solutions with much shorter reroute time (i.e., extra travel time), as the distribution tails are much shorter. This finding is *not* obvious as the linear programming method only guarantees the systemic minimum vehicular traveling time with no constraints on individual cases. Although the bipartite method is able to assign more carpools (as it is the objective function of the method), the efficiency of each assigned trip is worse than that of linear programming. The heuristics rules in the bipartite-method algorithm fail to sort out the best solution when there is a conflict. Moreover, one high quality carpool pair may outweigh five low carpool pairs.

The inflation ratio for travel time is derived from the concept of inflation rate in economics, but applied to travel time. A graph showing the inflation rate versus original travel time is presented in Figure 12. This further confirms the findings that linear programming gives more efficient carpool assignments (i.e., higher quality). For each graph, the top and the right marginal plots correspond to the marginal distribution of data points along the dimension. A contour plot is estimated as a coarse estimate of the joint distribution. The main finding is that the modes (i.e., the highest peak of contour plot) of the two methods do not completely overlap, with LP's peak very close to the zero inflation ratio, while the bipartite-method algorithm's peak is around 0.1-0.2. This further shows that the linear programming method is friendlier to carpool drivers.

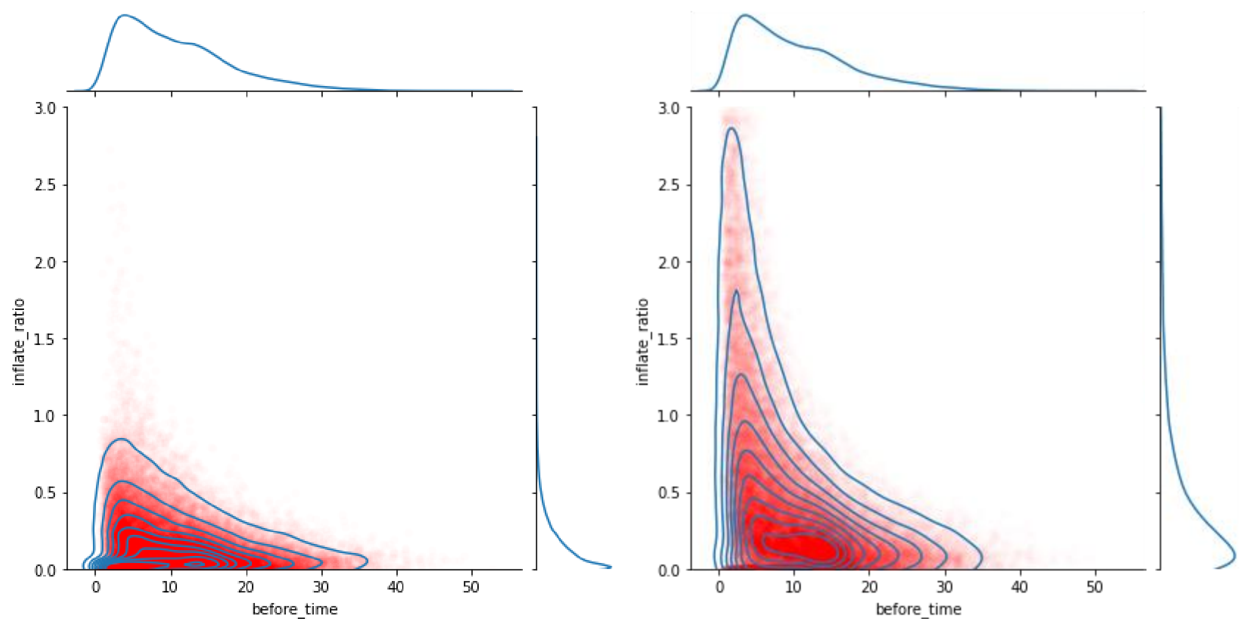

Figure 12. Inflate Ratio vs. Travel Time among Carpooling Drivers (LP vs. Bipartite Method)

4.3 Analysis of Aggregate-level Results

Figure 13 compares the paired data. The two diagrams show the data in linear scale, with the left diagram showing the counts for the linear programming method, and the right diagram showing the counts for the bipartite method. Each plot's x-axis shows whether the trip is within the same TAZ number or not. The blue bar represents the paired carpool trips while the orange bar represents the remaining SOV trips. The bipartite algorithm finds more local pairs traveling within the same TAZ. The linear programming algorithm finds a smaller number of trips because the gain in mileage/gas savings for short-range trips is very small, or even negative considering

TAZ access costs. On the contrary, the bipartite method prefers to pair as many pairs as possible as long as the reroute cost is within a reasonable threshold.

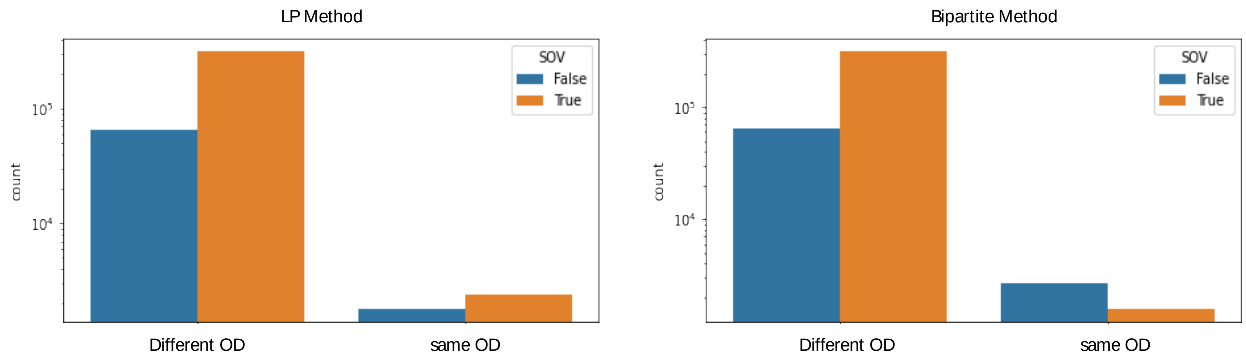

Figure 13. Comparison of Two Algorithm's Carpool Pairing Ratio for Trips between TAZs and Within TAZs

Instead of discussing trip details for individual trips, the overall impact on the system can be discussed over space and time. Based on the discussion in the last paragraph, one potential concern is that many carpool assignments are locally paired, which could be caused either by a bias towards short-range trips for the travel demand model, or by the optimization algorithms. All carpool trips in Figure 13 are paired only when the OD pairs are the same. The plot for the linear programming algorithm showed that: 1) the short-range trips with same origin/destination TAZs only represent a very small portion of trips; 2) short range trips are less likely to be paired than long distance trips. However, the same plot for the bipartite algorithm shows that it is more likely to pair short range trips, assuming that a reroute travel time of five minutes is acceptable, which in reality would not be desirable for the carpooling system to deliver high impact assignments.

One simple method to explore the trip's distribution is by aggregating trips by their destination TAZ or employment center, as shown in the bubble plot Figure 14. In Figure 14, the original number of SOV trips attracted and the percentage of carpooled results are plotted, with the size of the bubble corresponding to the number of carpool trips. This plot indicates that while the Midtown employment center attracts the greatest number of trips, only less than 10 percent of the trips are carpool-able. By contrast, although Emory has the second smallest number of trip attractions, it has the second largest number of paired trips. The Airport destination contributes the largest percentage and number of carpool-able commute trips.

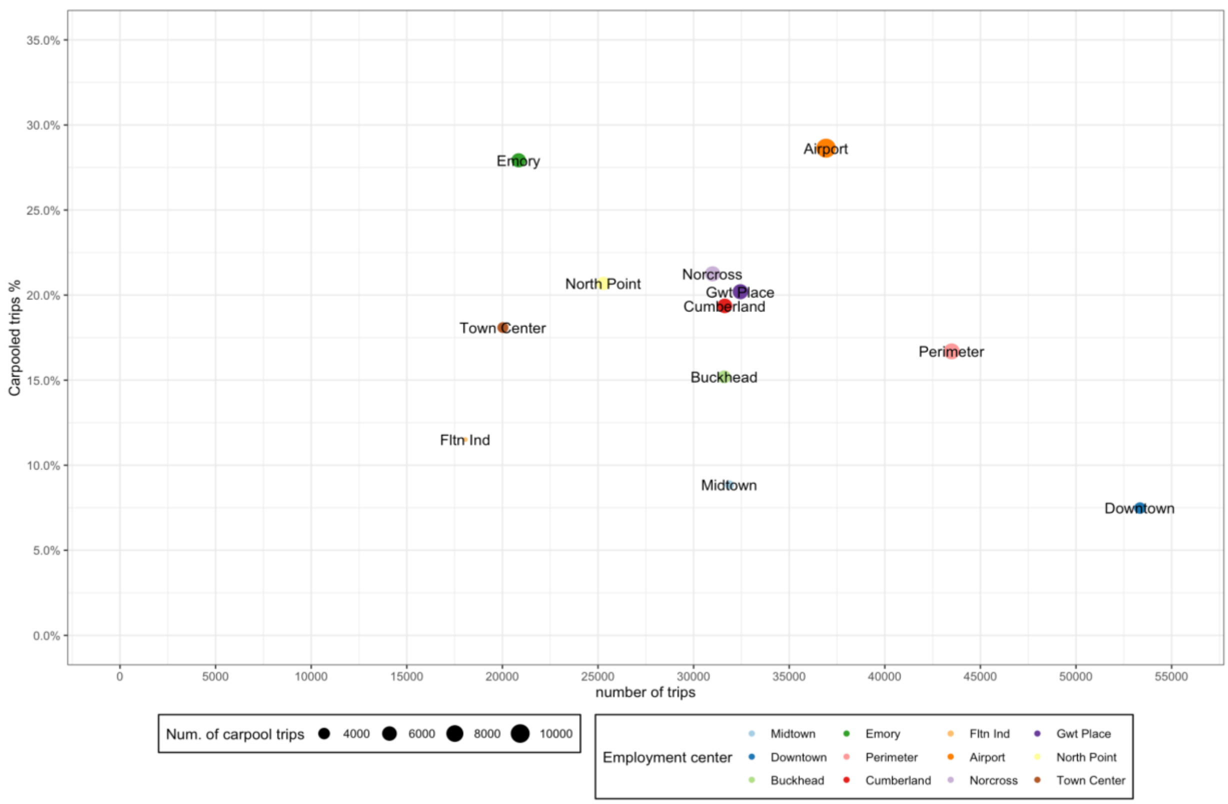

Figure 14. Bubble Plot for Each Employment Center Comparing SOV Trips and SOV Trips Assigned to Form Carpools

Figure 15 shows the counts of TAZs on the y-axis vs. the number of attracted trips for each TAZs. Each subplot represents the case for an employment center. This plot can show the heterogeneity of destination distribution over destination TAZs. Downtown and Midtown have many small TAZs that attract a small number of trips, given the assumption that destinations of the carpoolers must be to the same TAZs. The airport and Emory employment centers have several cases with very high number (over 5,000) of attracted trips, which is intuitive given their high demand.

Figure 16 shows bubble plots for carpooled percentage vs. total number of SOV trips (before consider carpooling). Each bubble corresponds to the trips to a specific destination TAZ. As the number of trips to a destination increases, the likelihood of identifying carpool pairs increases in a non-linear fashion. This is expected because the experiment only considers pairing trips for people with the same destination TAZ. There are three big outliers in the bubble plot, one is a TAZ at Emory University, and two others are TAZs in Airport. This is reasonable because the Emory University campus and hospital and airport terminals are large employment attractors.

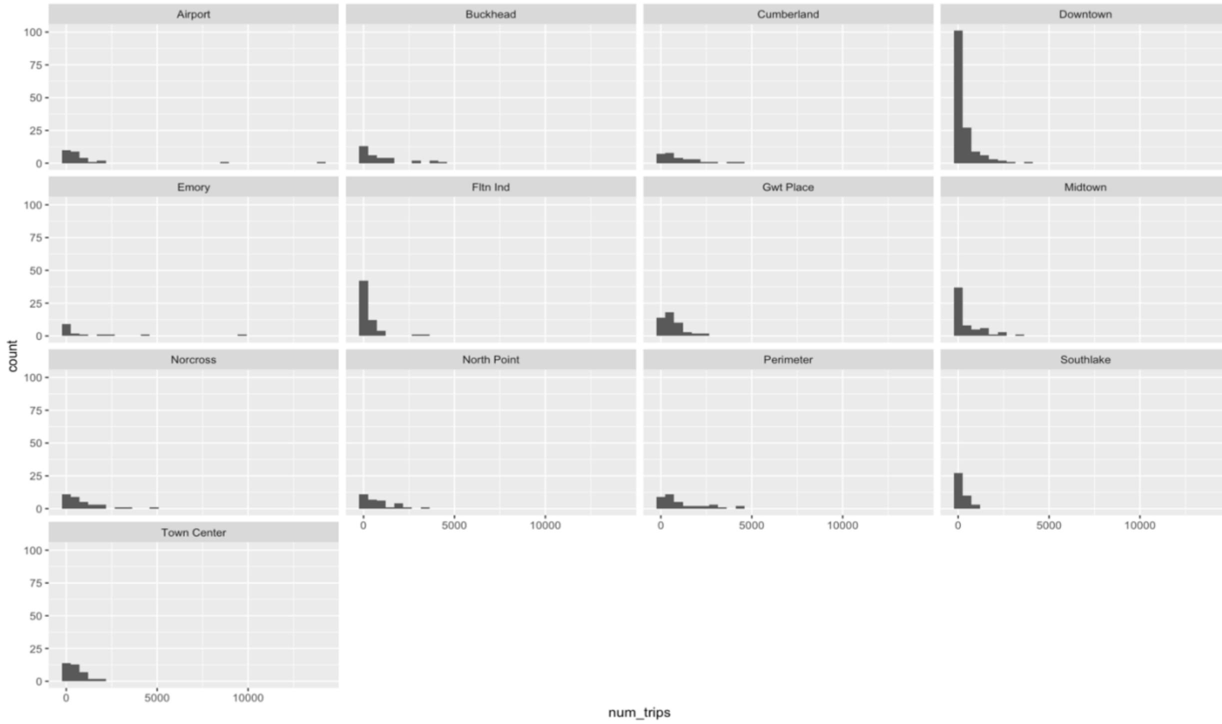

Figure 15. The Counts of TAZs vs. the Number of Attracted SOV Trips Grouped by Employment Centers

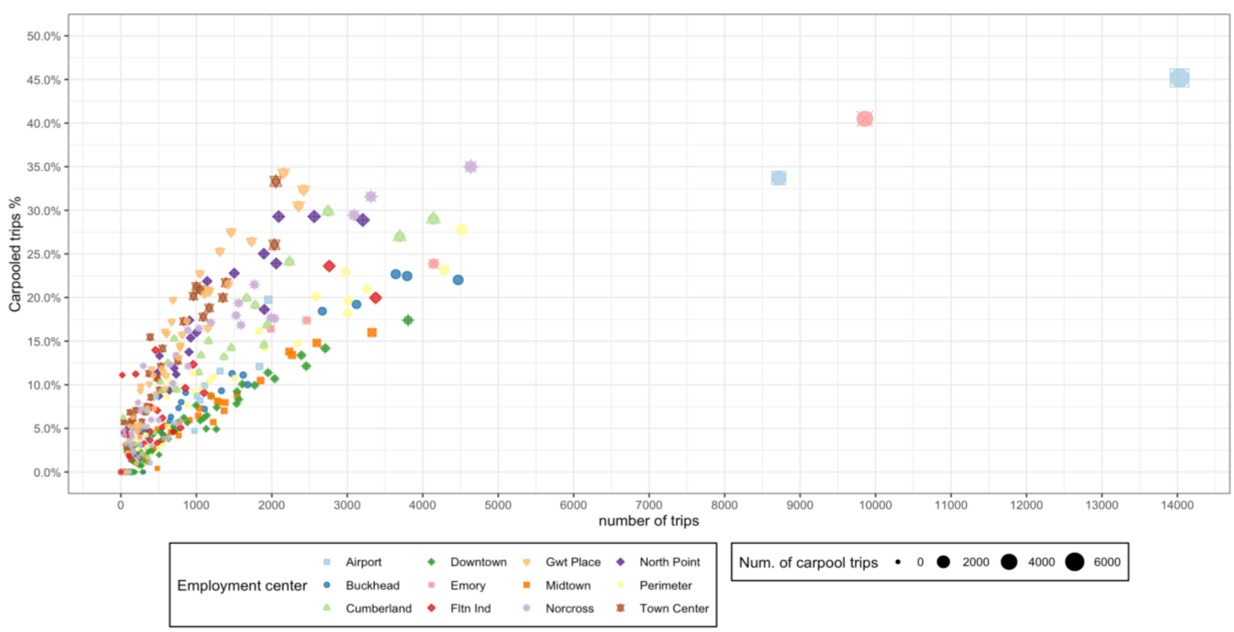

Figure 16. Bubble Plot for each Destination TAZ Comparing SOV Trips and SOV Trips Assigned to form Carpools

4.4 Network-level Results

The spatial distribution of trips is studied by analyzing the carpooling assignment's distribution on traffic networks and the impact on alleviating the pressure on the region's highway system.

As shown in Figure 17, we plot the absolute traffic volume distribution in the before scenario. Given the nature of the trips being studied, most trips are for inbound traffic, and many outbound links are not even used, as they are plotted in grey color and marked with a "Missing" legend. The Northern part of the city (i.e., I-85 inbound, I-75 inbound, US-19 inbound, and the northeastern corner of I-285 towards the west) has a large volume of loaded traffic (over 10,000 trips). A small northbound segment towards the city's downtown has over 20,000 trips loaded.

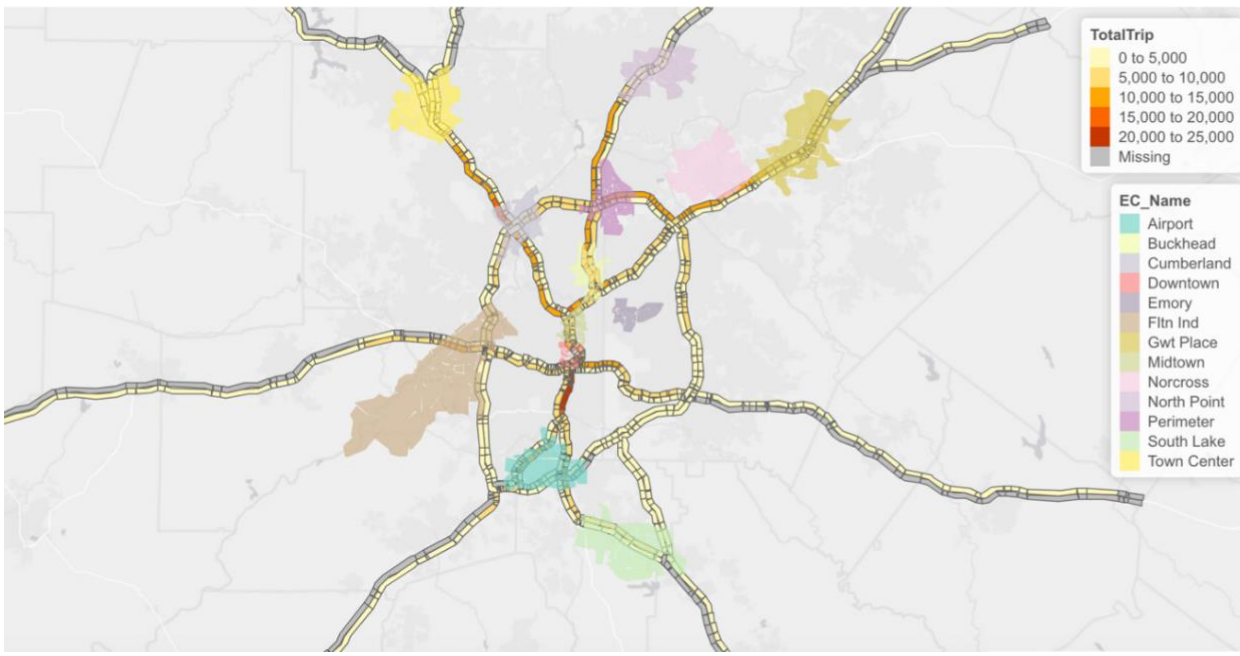

Figure 17. Traffic Volumes for all Before-case SOV Commute Trips along Skeleton Expressway Network

Among these SOV trips, the distribution of the before case SOV trips paired in the after case shows a slightly different pattern, as shown in Figure 18. US19 inbound traffic has the largest carpool travel counts, and it is higher than that of I-75 and I-85 inbound. The reasons for this are unclear (further analysis is required). It could be that the sparser origins of I-75 and I-85 travel makes the possibility of pairing trips lower (since the pairing is dependent on origin locations). Other employment or demographic reasons might be in play for the SOV trips happening in the northern region of Atlanta. For example, people living in Buckhead (north suburbs) have a higher income and may be more likely to drive alone, in the original output from the ARC ABM travel demand model, providing more SOV trip inputs that could be used to form carpools.

Figure 18 plots the volume counts of carpooling trips over the skeleton network. The pattern shows that most carpool trips happen for inbound traffic along US-19 near I-285 (in the north of Atlanta) with more than 2,000 carpool trips. There are over 1,500 carpool trips in the southern region of downtown Atlanta along I-75/I-85 northbound. While the inbound traffic from the north along I-75, I-85, and US-19 is similar (from 10,000 to 15,000 carpool trips), there are more carpooled trips on US-19 (Figure 17). This shows that SOV trips along US-19 have more potential to form carpools.

Figure 19 plots the percentage of assigned SOV trips over the skeleton network. The pattern of this distribution is different from the other two diagrams. It shows that inbound traffic along I-285 has the highest ratio of carpooled together (over 30 percent). This reveals that trips toward the Airport region (i.e., tiles in green) have the highest potential for being paired. Given that the TAZ of the airport terminal has a high concentration of inbound traffic, it is expected that it would be easier to pair trips to this destination and form carpools. This effect will be further studied in future research when the algorithm is upgraded to allow drop-offs at any TAZ along the commute path and to adjacent TAZ destinations.

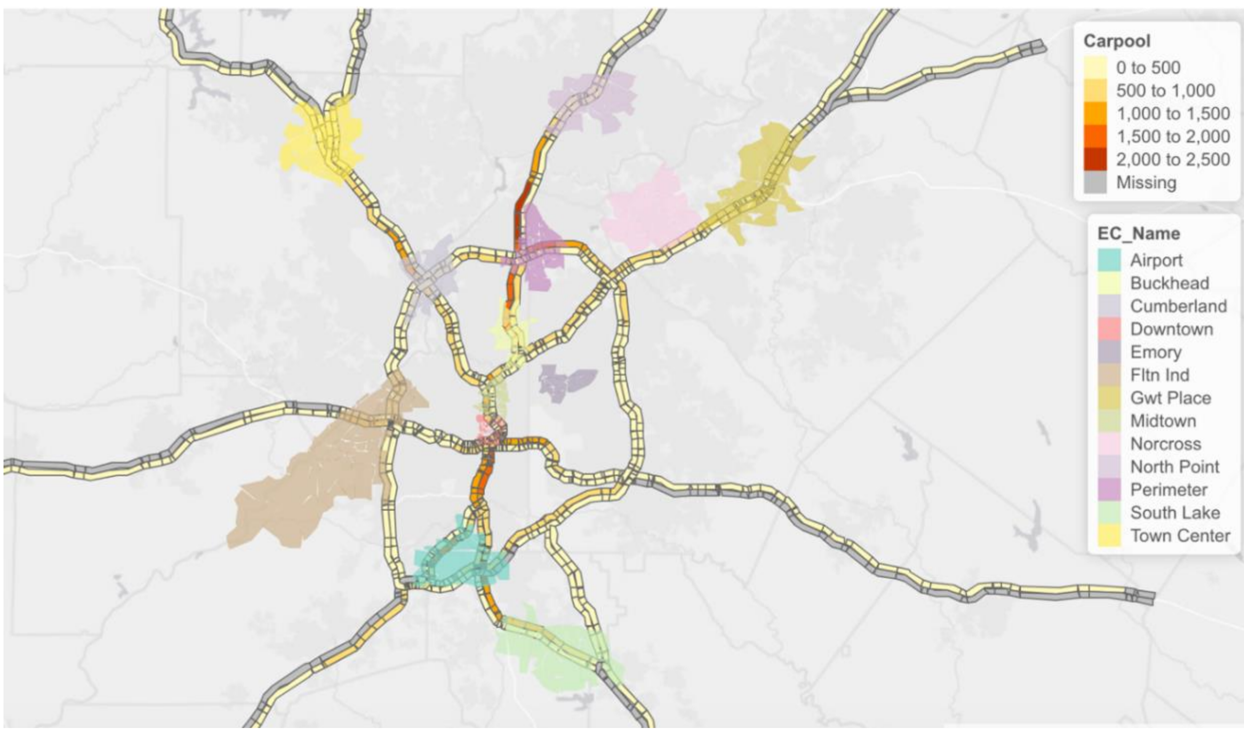

Figure 18. Traffic Volume Distribution of Carpooled SOV Commute Trips along the Skeleton Expressway Network

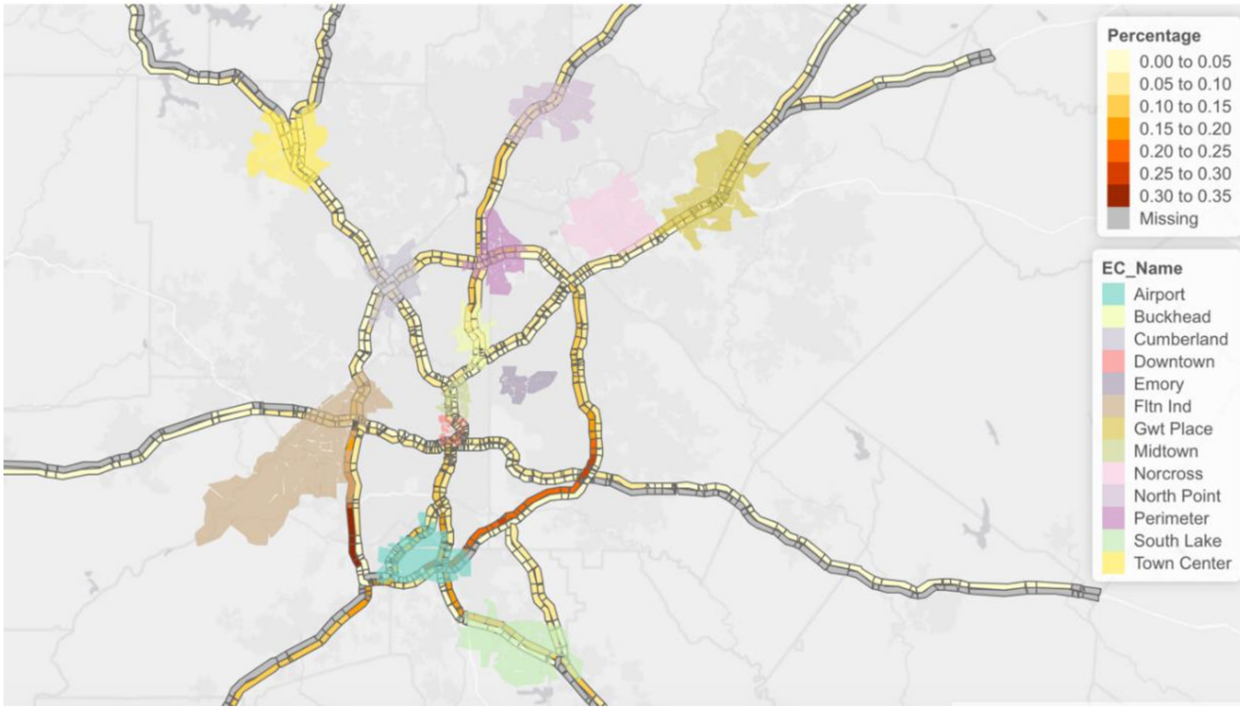

Figure 19. Percentage Distribution of Carpooled SOV Trips along the Skeleton Expressway Network

In conclusion, the experiment results show that community-based carpooling provide sufficient opportunities for decreasing the traffic load on the network during peak hours. The analysis results can be helpful in identifying parts of the network that can benefit from deployment of HOT lanes. The results of this analysis can also be further processed to quantify environmental benefit impacts of community-based carpooling.

The framework is suitable not only for use with ARC's ABM travel demand model for Metro Atlanta, but also for all activity-based model's trip outputs with similar level of detail for any region. The framework can also be used to assess other research goals. For example, by modifying the algorithm slightly, transit agencies can also use the model to find the optimal community-based vanpool schedule.

Conclusions

This research developed an analysis framework to estimate community-based carpooling trips among SOV travelers, given a set of reasonable spatiotemporal constraints. A set of experiments are run based on the trip outputs of the activity-based travel demand model employed by ARC. The results are promising in the sense that even under strict constraints, the model finds that around 13.5% of SOV commute trips in the morning peak period, from home to major employment centers, sharing the same departure TAZ departure and arrival TAZ, and sharing a similar departure time window could be paired to form carpools. The model outputs are further analyzed over space, time, and travel network. Two different optimization algorithms are proposed, one quickly finds the optimal solution in terms of minimizing the number of carpooling pairs while the other finds the systemic optimal value in terms of minimizing total vehicular hours. Analysis shows that the bipartite-based algorithm finds the greatest number of carpools, but fails to generate high-quality pairs for short-range trips. The integer linear programming solutions generates highest-quality carpool pairs designed to reduce total VMT and therefore energy use and emissions.

A second phase of research will extend the analysis framework in the following aspects: 1) analyze vanpool/transit operations in smaller areas; 2) implement a simulation platform to simulate stochastic popup travel demands thus to simulate different assignment algorithms and evaluate its performances; 3) improve clustering methods in the framework and their associated algorithms.

References

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E. and Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences, 114(3), pp.462-467. doi: 10.1073/pnas.1611675114.

ARC, Atlanta Regional Commission (2017). https://cdn.atlantaregional.org/wp-content/uploads/5-regionalsnapshot-employmentcenters-oct2017-web.pdf

Calvo RW, de Luigi F, Haastrup P, Maniezzo V. A distributed geographic information system for the daily car pooling problem. Computers & Operations Research. 2004 Nov 1;31(13):2263-78.

Cruz, M.O., Macedo, H. and Guimaraes, A., 2015, November. Grouping similar trajectories for carpooling purposes. In 2015 Brazilian Conference on Intelligent Systems (BRACIS) (pp. 234-239). IEEE. doi: 10.1109/BRACIS.2015.36.

Davidson, W., Vovsha, P., Freedman, J. and Donnelly, R., 2010, September. CT-RAMP family of activity-based models. In Proceedings of the 33rd Australasian Transport Research Forum (ATRF) (Vol. 29). Website link: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.455.2553&rep=rep1&type=pdf

Hsieh, F.S. and Zhan, F.M., 2018, July. A Discrete Differential Evolution Algorithm for Carpooling. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)(Vol. 1, pp. 577-582). IEEE. doi: 10.1109/COMPSAC.2018.00088.

Jamal, J., Montemanni, R., Huber, D., Derboni, M. and Rizzoli, A.E., 2017. A multi-modal and multi-objective journey planner for integrating carpooling and public transport. Journal of Traffic and Logistics Engineering Vol, 5(2). doi: 10.18178/jtle.5.2.68-72.

Qi, X., Wang, L. and Wang, X., 2016, July. Optimization of carpooling based on complete subgraphs. In 2016 35th Chinese Control Conference (CCC) (pp. 9294-9299). IEEE.

Santi P, Resta G, Szell M, Sobolevsky S, Strogatz SH, Ratti C. Quantifying the benefits of vehicle pooling with shareability networks. Proceedings of the National Academy of Sciences. 2014 Sep 16;111(37):13290-4.

Zhang, H. and Zhao, J., 2018. Mobility sharing as a preference matching problem. IEEE Transactions on Intelligent Transportation Systems. doi: 10.1109/TITS.2018.2868366.

Zhao, Y. (2021). Distributive Justice Impact Assessment Using Activity-Based Modeling With Path Retention. Dissertation. Georgia Institute of Technology, School of CEE. May 2021.

Data Summary

Products of Research

In this study, the primary data used are the SOV trips from the ABM15 travel demand model that was employed by the Atlanta Regional Commission (ARC) until 2020 for all planning and conformity work. The ABM2015 model has since been replaced by ABM2020. Future work should employ the outputs from the ABM2020 model and latest suite of planning model scenarios (e.g., the regional transportation plan (RTP) scenario, Transportation Improvement Program Amendment 1 (TPA1), etc.). The same basic principles apply, but the latest version of the model (ABM2020) has integrated a variety of model improvements associated with tour generation, mode choice, trips distribution, and route assignment.

Among the metadata used to support the simulation, traffic network data and the Traffic Analysis Zones (TAZ) data corresponding to ABM15 are the most important. Moreover, the 13 employment centers identified by the ARC in 2017 were also used to filter trips by their destinations and to investigate carpool effects on those centers. All aforementioned data sets combined form the data input for the research.

Before the carpool matching process, shortest network paths between any origin and destination TAZs are precomputed. The shortest paths information is then hosted in a local database for rapid querying. Although the shortest paths are automatically computed using Dijkstra's shortest path algorithm, this dataset can be openly shared (although future users are advised to derive these shortest paths for the latest ABM2020 version of the model).

The carpool matching process generates the matching plan between SOV trips that are recorded in a table. The carpool matching results are tracked just by expanding the original data table with new columns describing the role of each person, the partner for the person used to form the carpool trip, the travel time before and after considering carpool matching, etc.

Data Format and Content

ABM Trip Table (Data Set 1): A subset of ABM15 simulation trip outputs containing home to work SOV trips. A brief description of each table column is as follows. One can refer to the following website (https://atlantaregional.org/transportation-mobility/modeling/modeling/) for more details about the model and the table. Note that some columns unrelated to this study are dropped in the data repository. Original data files can be downloaded from ARC's website.

Field	Description	Values
hh_id	Household ID	1,743,333 distinct values from 1 to 1,928,783
person_id	Person ID	3,854,398 distinct values from 1 to 5,176,115
person_num	Person Number	1 to 15
tour_id	Tour ID	9 distinct values from 0 to 22
stop_id	Stop ID	-1 to 3
inbound	Is Inbound Trip	0 and 1
tour_purpose	Tour Purpose	atwork_business; atwork_eat; atwork_maint; eatout; escort_kids; escort_no kids; othdisr; othmaint; school_drive; school_predrive; shopping; social; university; work_bluecollar; work_health; work _retailandfood; work_services; work_whitecollar
orig_purpose	Origin Purpose	atwork; eatout; escort; Home; othdiscr; othmaint; school; shopping; social; university; work
dest_purpose	Destination Purpose	atwork; eatout; escort; Home; othdiscr; othmaint; school; shopping; social; university; work
orig_taz	Origin TAZ	5846 distinct values from 1 to 5873
orig_walk_segment	Origin walk market segment	0
dest_taz	Destination TAZ	5841 distinct values from 1 to 5873
dest_walk_segment	Destination walk market segment	0
parking_taz	Parking TAZ used	0; 107 distinct values from 656 to 776
depart_period	Departure time period	1 to 48 (1→3:00am, 48→2:30am)
trip_mode	Trip Mode	1 to 15
tour_mode	Tour Mode	1 to 15
tour_category	Tour Category	AT_WORK; INDIVIDUAL_NON_MANDATORY; MANDATORY

Traffic Analysis Zones (TAZ) Shapefiles (Data Set 2): This geographic shapefile contains information for all traffic analysis zones (TAZs) for the ABM2015 model run.

Employment Center Shapefiles (Data Set 3): This is a subset of TAZs representing major employment centers.

Shortest Paths (Data Set 4): This data set contains the pre-calculated results of shortest travel path runs for the travel time between each TAZ centroids. The data, recorded using JSON style data structure in native Python data structure (i.e., tuple, list, and dict.), is saved in files ended with ".pkl" or ".pickle." For each origin and destination TAZ, the shortest path on the traffic network, the travel time, and the mileages are recorded as key value pairs.

('1', '2', 1.6642285714285716, ['1', '80483', '2'])

The notation above shows that the shortest travel time from TAZ origin 1 to TAZ destination 2 is 1.66 minutes, and the trip traverses the following node progression: node 1 to node 80483 to node 2. If a trip traverses 18 nodes, the final sequence in square brackets contains the 18-node progression from origin node to destination node.

Analytical Outputs (Data Set 5): The analysis outputs are stored in a single ".csv" file. The data table contained in the file has new columns that denote the status of a person. The "SOV" column denotes whether the person drives alone. If not, the person must join a carpooling trip as either a passenger or driver. The "as_passenger" column denotes whether the person becomes a passenger during a carpool trip. Otherwise, the person is a driver in a carpooled or SOV trip. A third column "partner_idx" denotes the row index that identifies the partner during the carpooled trip. The index is generated by the "trip_id" column of the trip input data.

	newmin	before_time	before_dist	after_time	after_dist	SOV	as_passenger	partner_idx
1011	21	38.85	38.69	38.85	38.69	True	False	1011
1015	26	45.35	44.12	45.35	44.12	True	False	1015
2695	32	37.27	37.31	37.27	37.31	True	False	2695
2716	58	42.60	45.16	42.60	45.16	True	False	2716
5765	80	51.54	53.15	51.54	53.15	True	False	5765

Data Access and Sharing

The data is available for public access at https://doi.org/10.5281/zenodo.6354825.

Reuse and Redistribution

The ABM model data are the property of the Atlanta Regional Commission and were used by permission. Use of ARC models, meta-data supporting ABM modeling (i.e., TAZs of Atlanta, employment centers, etc.), and ABM model outputs requires execution of a data use agreement with the ARC. Instruction for downloading ARC model outputs can be found on the ARC website. Carpool matching results and shortest travel path data can be freely shared and users should refer to this NCST grant and report as the source.

Appendix A. Problem Definition & Notation

Let's assume all trips of interest are denoted as a set $G = \{g_1, g_2, \ldots, g_n\}$, where n is the total number of trips of interest for analysis. A specific $g_i \in G$ stands for a specific trip demand. A trip demand may contain a party of one or several people sharing completely same trip demand. In this case study, the trip demand always corresponds to a drive-alone commuter. For any trip demand g_i, it contains the following spatiotemporal information o_i, d_i, l_i, a_i, which corresponds to the origin, destination, departure (leaving) time and arrival time of the trip demand g_i.

Two parameters, Δ and Γ, controls the lower-bound of carpooling quality and they are used as controlling thresholds to filter out infeasible trips. $\Delta \geq \max_i\{\delta_d, \delta_p\}, \forall$ i stands for the maximal tolerable modification in departure time or pick up time. In the carpooling settings, either the passenger needs to meet driver at pick up location δ_d minutes earlier, or the driver needs to depart δ_p minutes earlier to get to the pickup location in time. This kind of schedule modification is possible because the pairing plan is assigned before the traveling day. Notice that in the settings $\delta_d \geq 0, \delta_p \geq 0$, meaning that δ_p and δ_d are nonnegative values. The analysis assumes that setting Δ to between 5 and 15 minutes is reasonable as this will not significantly change the daily commute routines. Otherwise, the two individual trips do not form a good quality carpooling trip. In other words, either the driver departs earlier or the passenger meets the driver at the pick up point earlier. When Δ is a small number (e.g., 30 seconds), given the system's situation at a given time, the problem reduces to a ride-sharing optimization problem given the spatiotemporal ride-sharing contexts.

The other parameter Γ, denotes the maximal extra reroute time for the driver compared to his or her driving-alone case. Since the driver's reroute time is always larger than the passenger, only trips with the driver's reroute time smaller than a threshold Γ are considered feasible carpools. Γ controls the maximal extra tolerable reroute time for the driver.

For the analyses, k denotes the maximal number of trip demands served in a ride-sharing vehicle. When $k \geq 3$, the optimization problem becomes hard to solve. Also, previous studies found that $k \geq 3$ doesn't contribute too much to system performance. In this study, we only discuss the case of $k = 2$, i.e., in the carpooling settings, a driver either drives alone or carpools with just one other commuter during the trip.

Appendix B. Two Carpooling Plans

For this study, a carpooling assignment means assigning multiple trips each of which uses one vehicle to form two-person carpools. In carpool formulation, unlike the taxicab service or ride hailing, the vehicle belongs to the commuter/owner. Because the system only considers carpooling among two travel demands ($k = 2$), there are only two carpool scenarios between trip i and j, that is either i acts as the driver and j acts as the passenger or vice versa. In the carpooling contexts, assuming i and j forms a carpool trip, the trip sequence can only be either $[o_i \rightarrow o_j \rightarrow d_j \rightarrow d_i]$ or $[o_j \rightarrow o_i \rightarrow d_i \rightarrow d_j]$.

B.1 Trip Filtering

Before enumerating all feasible carpooling combinations, the number of trip combinations needs to be narrowed down, filtering isolated trips and clustering trips into small groups based on their departure/arrival time and location. Based on the problem settings and research need, different clustering plans can be devised. The key is to filter out infeasible carpooling combinations, but not all infeasible combinations.

B.1.1 Trip Filtering with Approximate Destinations Constraints (Naïve Scenario)

Simply grouping trips by their origin and destination serves as a starting point. Since the study assumes a community-based strategy, carpools are only formed when they start in the same origin transportation analysis zone. Moreover, these trips must drop off passengers at the same location (transportation analysis zone) in this simplified case. In this way, OD pairs can be considered separately, and carpooling assignment can be made in parallel.

Within each OD pair, run DBSCAN(*T;Δ',n = 2*) over the departure time. *Δ'* is the maximum difference in departure time between two original SOV trips. $\Delta' = |l_d - l_p| \approx \Delta$ travel time for the driver picks up the passenger is small in community-based carpooling (Figure 19).

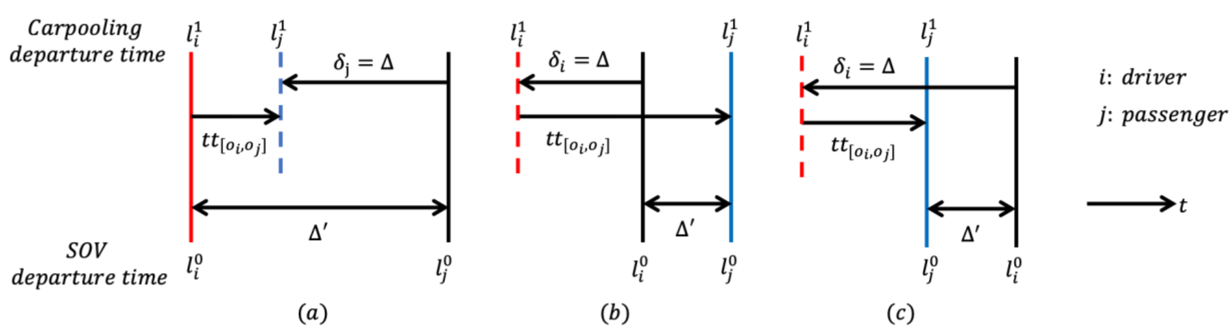

Figure 20. In Pre-planned Community-based Carpooling Problem, Δ' is a Good Approximation for Filtering Trips when Pickup Time is Relatively Small

In Figure 20, assume i denotes the driver and j denotes the passenger. Case (a) represents the case that the driver i picks up passenger j by starting at the original scheduled (SOV) departure time, whereas the passenger needs to meet earlier than their SOV schedule. Case (b) and (c) stands for the cases where driver i leaves earlier than the original SOV case to accommodate the passenger j. The terms l_i^0 and l_j^0 stand for the original SOV depart time for driver and passenger respectively. Similarly, l_i^1 and l_j^1 denotes the carpooling depart time for driver and passenger respectively. There are only three possible cases considering i needs to pick up j. Notice that when $tt[O_i, O_j]$ is small compared to Δ, and there is $\Delta' \approx \Delta$. In this way, the filtering step on the original SOV departure time using $DBSCAN(T; \Delta', n = 2)$ is reasonable.

A relatively large value (e.g., Δ'=15 min) instead of small number such as $\Delta = 30$ second was used in this study because the carpooling assignment is not a real time dispatch but pre-planned ahead of the day of travel. Using the Δ overlap of departure times, a travel group can be further split into smaller groups. We use the DBSCAN to quickly sort out the clusters plus the potential feasible trips. In summary, the naive case can very quickly narrow down the searching scope in the expense of not considering integrating carpools with the case of dropping of passengers along the driver's travel path.

B.1.2 Trip Filtering without Approximate Destinations Constraints (Complete Scenario)

Similar to the naive case, trip pairing performed in the complete scenario is limited to the same departure TAZ to reflect the community-based strategy. However, in this case a driver can drop off passengers along their travel if they are not sacrificing too much of their commute time. In the complete scenario, the destinations among carpoolers can be to different TAZs.

To query distance relationships between TAZs, a map of adjacent TAZs is established. For any trip g_i, given the shortest path between its origin & destination, all of the TAZs a trip passes are denoted as a sequence $Z^i = [Z_1, \dots, Z_k]$. Within this sequence of TAZs, if there is a feasible trip with OD among the sequence of TAZs, then a carpooling trip is considered. Interestingly, the DBSCAN can still be applied to cluster trips since they still share similar origins. Before applying DBSCAN, we need to cluster trips by destination along the shortest traveling paths. DBSCAN is the last filtering step to maximize its impact on clustering trips into smaller groups.

Appendix C. Generate Precise Shareability Network with Role

The last section only filters out the isolated trips and cluster carpool trips into groups. In the settings, carpooling trips can only happen within the same travel group but not between different travel groups. To find the optimal carpooling assignment, a shareability network is constructed for each group. A feasibility matrix is then used to store the information for the shareability network.

A feasibility matrix $F = [f_{ij}], \forall i, j \in \{1, \dots, n\}$ can be constructed to store all feasible carpool assignments among this trip cluster. For any f_{ij}, a trip can be considered as a carpool if and only if:

$$|l_i - l_j| \leq \Delta,$$

$$c_{ij} = [tt(O_i, O_j) + tt(O_j, D_j) + tt(D_j, D_i) - tt(O_i, D_i)] \leq \Gamma$$

In the above formula, $c_{i,j}$ is the extra delay for the driver *i*. Only trips sharing similar departure time and short reroute time can be considered as a feasible carpooling pair. To more precisely filter the trips, an optional filter based on the ratio between the driver's before-after traveling time can be added in addition to the filters based on differentiation as shown below:

$$[\ tt(O_i, O_j) + tt(O_j, D_j) + tt(D_j, D_i)]\ /\ tt(O_i, D_i) \leq \alpha$$

Appendix D. Proposition 1

Proposition 1. Let's call the cyclic links circulating a TAZ, the TAZ's minimum circulating loop (MCL). Assume the shortest path between TAZ's MCL is contained in the shortest path between two TAZ centroids, then the two-step procedure gives the sub-optimal paths between two coordinates bounded by a constant.

Proof.

As one can see in Figure 21, almost every TAZ centroid is surrounded by major arterial roads (lines in purple), where the green links are the connectors from TAZ centroids to nearest one or few neighboring networks. In this way, the concept of minimum circulating loop holds for ABM network.

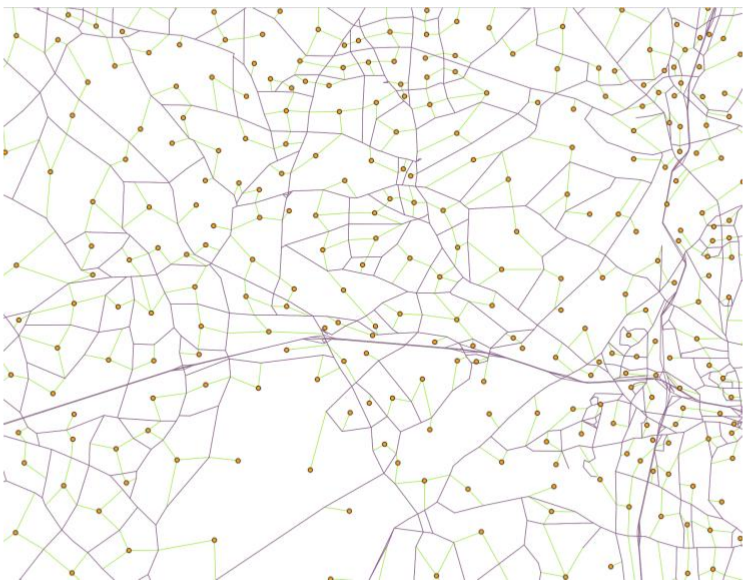

Figure 21. A Zoom-in Part of ABM Network with TAZ Centroids and its Connectors

As shown in Figure 22, consider the shortest path from TAZ centroid A and TAZ centroid B. The shortest paths contain three segments: two connector links connecting centroid to network nodes, and the shortest paths between the two network nodes connects to the centroids. The latter shortest connecting path segment is also a very good candidate of the shortest paths between the MCLs of TAZ A and TAZ B. Assume this is true, then during stage two computations, we can just route the trip from their coordinates to the nearest node then to the two end nodes of the shortest path between two MCLs. In this way, we reconstruct a good approximation of shortest path, which is sub-optimal except adding some reroute time no worse than the perimeters of the two MCLs.

This proof may not be ideal, but it gives an upper-bound of the estimation's quality compared to optimality. In practice, it performs very well and achieves high computation speed. A manually constructed worst case is shown as follows. The estimated path is not the best option, as can be determined by observing that taking the link connecting TAZs on top is a better choice. However, this rarely happens in the real case given the geometries of the network.

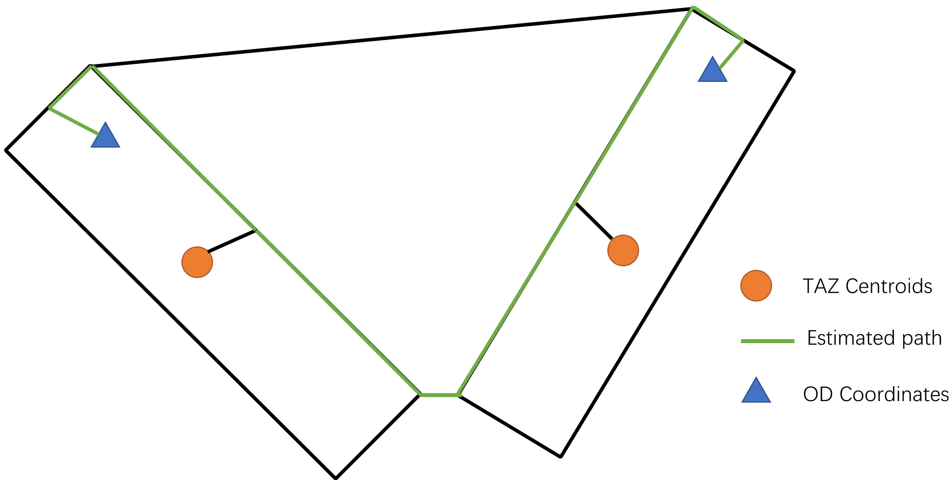

Figure 22. A Counter Example of Finding Non-optimal Shortest Paths