Research Report

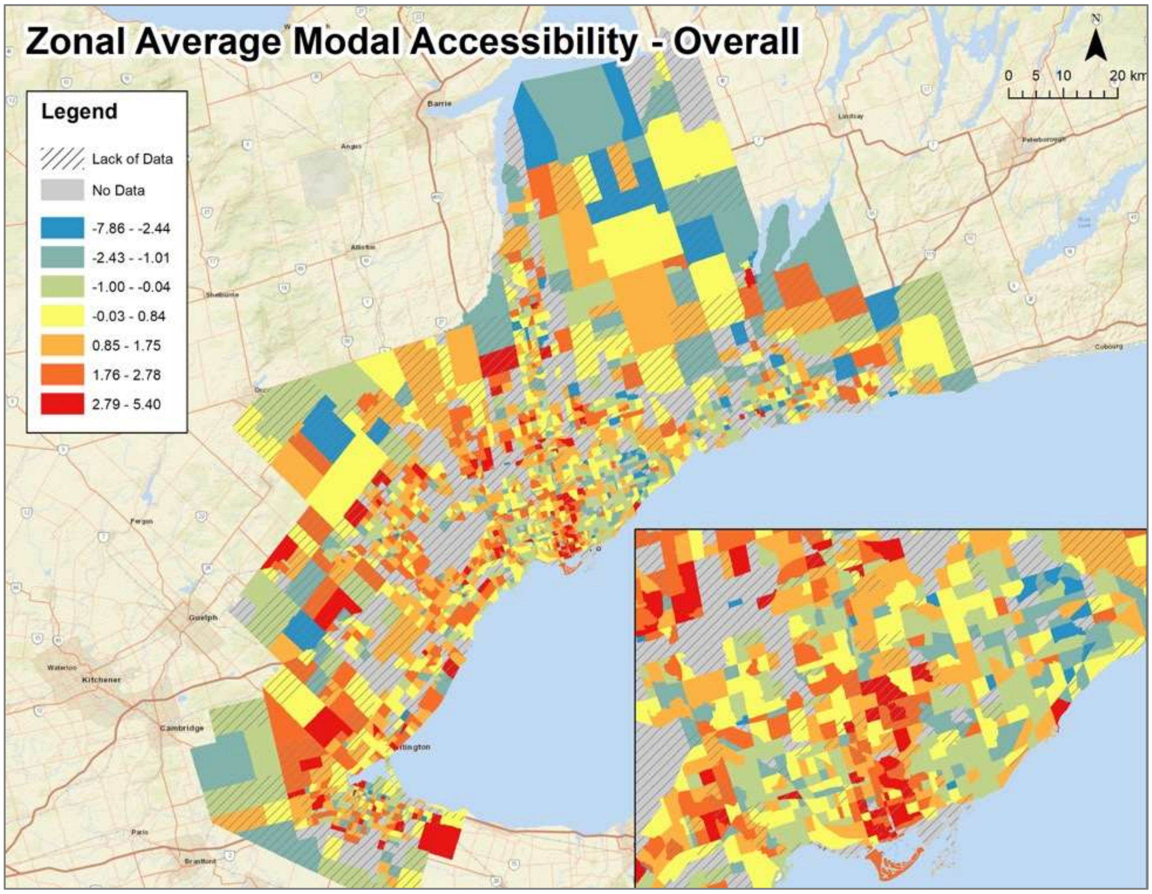
# Transportation Accessibility Advice

Final Report

Christopher Higgins, Amber DeJohn, Steven Farber, Matthew Palm, James Vaughan, Michael Widener, Luna Xi, Eric J. Miller

January 2020

# TRANSPORTATION ACCESSIBILITY ADVICE




# FINAL REPORT

Submitted to the City of Toronto Planning Department
By

Christopher Higgins
Amber DeJohn, Steven Farber, Matthew Palm, James Vaughan, Michael Widener, Luna Xi
Eric J. Miller

January 2020

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING
Transportation Research Institute

UTTRI

# TABLE OF CONTENTS

**Page No.**



Table of Contents 1
List of Figures 2
List of Tables 2
Acknowledgements 2
Executive Summary 3

**LIST OF FIGURES** **Page No.**



**LIST OF TABLES** **Page No.**



**ACKNOWLEDGEMENTS**

The authors would like to thank James Vaughan for, as always, his very valuable help in supporting the analyses presented in this report. The support of the Travel Modelling Group in providing access to its computer server system is also gratefully acknowledged.

# EXECUTIVE SUMMARY

This purpose of this study is to provide advice to the City of Toronto concerning the current best practice and state of the art in transportation accessibility measurement, and the use of such measures in transportation planning modelling and analysis, leading to recommendations for implementing a practical and robust accessibility measurement procedure. Accessibility is defined as the potential for reaching spatially distributed opportunities while considering the difficulty involved in traveling to them. It is an important indicator of a number of aspects of a city's urban spatial structure. Several types of accessibility measures exit, including *infrastructure-based*, *person-based*, *place-based*, and *utility-based*. This project focuses on place-based accessibility measures.

The first phase of the project involved a review of the literature to identify the current state of the art in accessibility measurement as embodied in available software for computing accessibilities. Emerging from this review, seven primary tools were identified for further preliminary evaluation. These were: R/Python packages; ArcGIS Network Analyst; Emme 4; Open Trip Planner and Open Street Maps; Conveyal; TransCAD; and Google Maps. These seven tools were then evaluated and assigned scores on a 0-2 scale across a range of criteria. Based on this evaluation, four approaches were selected for more detailed analysis and evaluation: ArcGIS Pro 2.4.2, Conveyal Analysis, Emme 4.4, and OpenTripPlanner (OTP) for R and Python.

Recommendations for developing an accessibility measurement tool are as follows:
1. A multi-stream procedure is required to address the full range of accessibility analysis and planning needs of the City of Toronto.
2. Among the multi-stream procedures evaluated in this study, it is recommended that both R-OTP and ArcGIS Pro be deleted from further consideration as the primary basis for the City's accessibility tool.
3. A hybrid Emme – Python-OTP solution is proposed as the multi-stream procedure for City of Toronto adoption.
4. While not sufficient as the City's primary accessibility measurement tool, the City should conduct its own assessment of Conveyal Analysis to determine its cost-effectiveness for specific purposes within the overall accessibility measurement toolbox that is to be developed.

The recommended hybrid workflow consists of:
1. All network coding and editing is done within Emme (exactly as currently occurs).
2. For any scenario of interest, GTAModel (or, if sufficient for the given scenario, simply Emme road and transit assignments) can be run to generate equilibrated, congested auto and transit travel times/speeds.
3. A GTFS file containing the transit network and its equilibrated speeds is exported.
4. The GTFS transit network file and the OSM-format road network file are imported into Python-OTP.
5. All alternative mode-based (e.g. walking, cycling, walking and transit, cycling and transit, etc.) accessibility calculations are undertaken with Python-OTP. Points of Interest (POI) and other GIS-based data required to undertake the accessibility calculations can be imported as needed.

6. If auto-based accessibility analyses are required at this lower level of spatial aggregation, a road network file with congested link speeds is exported from Emme and edited to include local streets using a GIS program. The combined network is exported in an OTP-readable OSM format.

The first three tasks would be implemented within XTMF/GTAModel. Tasks 4 and 5 would be implemented within Python-OTP. This workflow needs to be fully automated so that results from a GTAModel/Emme run are automatically translated into the GTFS and OSM files and imported into OTP. Further, it should be possible to request as a GTAModel run option "standard" accessibility calculations to be automatically generated as part of the post-run processing. In addition, the Python-OTP accessibility calculator should be able to run in stand-alone mode to provide dynamic, real-time accessibility calculations for an analyst for a given scenario that had been previously processed by GTAModel/Emme. This solution requires the development of procedures (in some combination of C# and Python) to:

- Export a GTFS transit network file from Emme that is readable by OTP. We experimented a bit with this but were not able to get the procedure fully operational within the time constraints of the project across all multi-stream tools. We do not see any fundamental obstacle, however, to developing such a working procedure.
- Similarly, export an OSM-formatted file from Emme of the congested road network that is readable by OTP for use in automobile-based accessibility analysis in that software, if required.
- Generate spatial coordinates, stored as CSVs, to be called by Python-OTP script for use as origins and/or destinations in accessibility calculations.
- Compute the desired accessibilities. These might include a variety of gravity-based measures, isochrones, etc.
- Provide a user-interface for the analyst to:
    - Dynamically request custom accessibility calculations.
    - Display and view maps, graphs, etc. of the calculated accessibilities. A "dashboard" design would be the desired solution for this capability. Note that ArcGIS Pro, Conveyal Analysis and various open-source GIS and visual display tools all might provide methods for the display of accessibilities, both spatially and otherwise.

# 1. Introduction & Study Objectives

This purpose of this study is to provide advice to the City of Toronto concerning the current best practice and state of the art in transportation accessibility measurement, and the use of such measures in transportation planning modelling and analysis. Specifically, as stated in the September 20, 2019 *Request for Quote* (RFQ) received from the City, the objectives of the study are to:

1. "Identify and evaluate possible approaches for the City of Toronto to use transportation accessibility modelling for various analytical purposes, including but not limited to Business Case Analysis. The recommended approach(es) should reflect the current state of practice and value for money." (RFQ, page 1)
2. "Recommend one (1) tool for implementation and two (2) alternatives. The basis for the recommendation will be clearly stated and include the findings of the research into the state of practice and available tools." (RFQ, page 3).

Section 2 of this report presents a brief discussion of key issues and options in transportation accessibility measurement that provide the foundation for the approach taken in this study for addressing the project objectives. Section 3 briefly summarizes a literature review of accessibility measurement current practice.[1] In addition to this literature review, a stakeholder meeting was held on December 12, 2019 to inform interested departments within the City of the project and to solicit their views on their needs and applications for accessibility measures.

Based on these preliminary investigations, the study team developed an overall conceptual approach for addressing the City's accessibility measurement needs and selected of a set of measurement tools for more detailed review for possible implementation within this framework, as described in Section 3. Section 4 describes the set of tests undertaken to evaluate and compare the short-listed methods and tools, as well as presents and discusses the results of these tests. Section 5 concludes the report with a summary of key findings and recommendations to the City for next steps in adopting an improved "tool box" of accessibility measurement tools for use in their planning and analysis work.

# 2. Issues & Options in Accessibility Measurement

Accessibility, which we define here as the potential for reaching spatially distributed opportunities while considering the difficulty involved in traveling to them (Páez, Scott, & Morency, 2012), is an important indicator of a number of aspects of a city's urban spatial structure. There are several types of accessibility measures in the literature, including *infrastructure-based*, *person-based*, *place-based*, and *utility-based* (Geurs & Van Wee, 2004). This project focuses on place-based accessibility, which, in its simplest form, can be operationalized as:

$$A_i = \sum_j O_j f(t_{ij})$$

[1] The full literature review is contained in project Report 1.

where the accessibility $A$ of origin $i$ is the sum of all opportunities $O_j$ available at destinations $j$ weighted by some function of the travel time $t_{ij}$ between $i$ and $j$. In this sense, accessibility is a measure of spatial interaction that captures three elements: the characteristics of a city's underlying transportation network infrastructure, the distribution of opportunities contained within different land uses reachable within this network, and the dynamics of flows on the network reflected in the measure of impedance.

Although researchers and practitioners have been utilizing measures of accessibility for several decades, there are number of issues and considerations that should be taken into account when performing practical accessibility analysis (Miller, 2019). In terms of best practice, there is a rich history of theoretical development and a large body of academic and grey literature on applications to be considered. On the practical side, users are presented with a growing suite of software options, each of which entails a different workflow and relative strengths and weaknesses. Despite this progress, three fundamental challenges persist in applied work: the design and operationalization of underlying transportation networks, areal configurations of origins and destinations, and specification of the impedance function. In response, this project provides guidance on major aspects of applied accessibility analysis and software options that is sensitive to these challenges in the City of Toronto.

# 3. Summary of Literature Review Findings

The first phase of the project involved a review of the literature (both academic and professional) to identify the current state of the art in accessibility measurement as embodied in available software for computing accessibilities. Emerging from this review, seven primary tools were identified for further preliminary evaluation. These are:

1. R/Python packages.
2. ArcGIS Network Analyst.
3. Emme 4.
4. Open Trip Planner and Open Street Maps.
5. Conveyal.
6. TransCAD.
7. Google Maps.

These seven tools were then evaluated and assigned scores on a 0-2 scale for each of the following criteria:

- *Perception factors.* Can the tool account for differences among travellers in their perceptions of times, distances, etc.?
- *Generalized costs.* Can the tool handle different definitions of the "effort" of travel, such as travel time, travel cost, distance, and/or combinations thereof?
- *Impedance functions.* Can the tool use user-defined impedance functions?
- *Competition.* Can the tool account for competition among people for participation in activities at destination locations (e.g., competition among workers for jobs in employment accessibilities)?
- *Flexibility.* How flexible is the tool to use? How readily does it adapt to new or extended applications?

- *Temporal variation.* Does the tool handle variations in accessibility over the course of the day?
- *Spatial refinements.* Can the tool work at different levels of spatial resolution? What is the computational efficiency of the tool as spatial resolution is increased?
- *GTAModel compatibility.* How easily can the tool interface with GTAModel, which might be used to generate the travel times, etc. used in the accessibility calculations? Can GTAModel and the accessibility tool be part of an integrated planning analysis workflow?

Table 1 reproduces the summary of the scores from the *Literature Review* report. More detailed explanations of the assessments underlying these scores are provided in this report. Based on this review three high-performing software tools that compute travel times for accessibility modelling were identified: EMME, ArcGIS and Open Trip Planner/Open Streets Map. These tools can be combined with R and/or Python scripts to create various "multi-stream" accessibility measurement workflows. We have also identified a "single-stream"[2] sketch planning tool, Conveyal, that can compute the most common metrics and travel times independently, although this greater independence means its outputs are less likely to match findings from a GTAModel-based analysis. These four tools were then examined in much greater detail in the second phase of the project in order to assess their performance with respect to computation times, interoperability with GTAModel, usefulness of tool outputs, among other factors. These tools, the tests undertaken and the results of these tests are all described in detail in the next section.

**Table 1. Evaluation Scores for Accessibility Tools**

| | Perception factors | Generalised costs | Impedance functions | Competition | Flexibility | Temporal Variation | Spatial Refinements | GTAModel Compatibility | Score |
|---|---|---|---|---|---|---|---|---|---|
| R/Python packages | 1 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 14 |
| ArcGIS Network Analyst | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 13 |
| Emme 4 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 13 |
| Open Trip Planner & Open Street Maps | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 13 |
| Conveyal | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 13 |
| TransCAD | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 0 | 12 |
| Google Maps | 1 | 1 | 1 | 1 | 0 | 2 | 2 | 1 | 9 |

[2] So labelled because it is a self-contained tool that does not require development of a multi-software workflow.

# 4. Empirical Evaluation Results

## 4.1 Introduction

As discussed in the previous section, this study has identified a number of promising software tools for calculating accessibility within a given study area. In addition to the selection criteria already discussed, through discussions with city staff and stakeholders it became apparent that a key additional consideration is the ability to perform accessibility analyses using public transit as the travel mode. This criterion assisted the team in narrowing down the list of tools for further analysis to ArcGIS Pro 2.4.2, Conveyal Analysis, Emme 4.4, and OpenTripPlanner (OTP) for R and Python.

This is because while many routing tools exist, some tools work with only select modes. For example, modelling the shortest path between an origin and destination for walking or driving involves minimizing travel time or distance across a network of edges and junctions using a solver algorithm (see Bast et al. (2016) for an in-depth review of different solver algorithms). ArcGIS uses Dijkstra's shortest path and OTP uses contraction hierarchies to solve such street network-based analyses. But because public transit networks tend to be schedule-based, different types of solvers are required.

In particular, transit networks can be represented and modelled in a GIS environment through General Transit Feed Specification (GTFS) files, which consist of a system of text files corresponding to transit trips, service calendars, route geometry, and other attributes. While the GTFS system can be cumbersome, it appears to have become a defacto standard for many transit agencies around the world to share their network information, and, as such, it is currently the foundation of many of the available tools for analyzing transit accessibility. For such a transit workflow, ArcGIS uses a custom Public Transit Evaluator and OTP currently uses an algorithm called A*, although future revisions of OTP will use a newer $R^5$ solver (currently in use by Conveyal Analysis) based on Microsoft's RAPTOR algorithm. Other routing tools such as pgRouting for PostGIS, OSRM and Pandana for Python, network extensions for QGIS, and the GraphHopper service appear promising, but do not offer any public transit routing solutions and are not considered further here.

It is important to note however that GTFS-based networks inherently represent an idealized version of the transit network according to its stated service schedules. Such optimal networks may not reflect the reality of traffic congestion, service delays, or the temporal cadence of frequency-based transit services. The increasing availability of on-vehicle tracking presents an interesting new avenue for examining the observed performance of the transit network. For example, Wessel et al. (2017) used TTC tracking data to create a retrospective routable transit network for OTP by replacing scheduled stop times with observed vehicle performance in GTFS files. This early work offers an indicator of where future research and tools in this area can and will likely develop. Nevertheless, in terms of evaluating solutions for accessibility calculation, the development of GTFS-based routing tools is more mature and implemented in a variety of software packages.

The tools selected for analysis can be further divided into two broad families based on their ability to work with GTFS transit networks and schedules. In the first family, which we refer to as a “single-stream” analysis tool, is Conveyal Analysis. This software constitutes an entire multi-modal scenario planning platform that enables the analyst to not only model place-based accessibility from existing GTFS and street networks, but to alter the transit network and run comparative analyses. In contrast, we refer to ArcGIS, Emme, and OTP as “multi-stream” tools that can fit into a larger transit accessibility workflow. In their native form, these tools do not explicitly have the ability to edit GTFS files and additional work is required of the analyst to calculate accessibility scores. In this sense, such tools can fit into a workflow involving other software packages for GTFS editing and additional programming of accessibility calculations. These considerations help inform our conclusions below. Section 4.2 discusses the multi-stream tools and their assessment. Section 4.3 then similarly assesses Conveyal, the only single-stream tool under consideration.

## 4.2 Runtime Analysis (Multi-stream Tools)

### *4.2.1 Study Design*

To evaluate the strengths and weaknesses of ArcGIS Pro, Emme, and OTP using R (R-OTP) and Python (Python-OTP), a series of timed scenarios were run. These scenarios involve:

1. Creating a combined street and transit network for analysis.
The first step is to gather the required input files to assemble routable multi-modal networks. This includes street networks and GTFS transit schedules.

2. Calculating origin-destination (OD) matrices for trips.
Origins and destinations are generated as random points within the boundary of the City of Toronto using ArcGIS Pro, consisting of:

- 100 points
- 1,000 points
- 10,000 points
- 100,000 points

In addition, a single input point was manually created at the approximate centre of Toronto City Hall. The latitude-longitude pair of each point was then calculated and all features exported to comma separated values (.csv) format for use by each tool.

From these points, the analyses are structured around “local” and “regional” scenarios. In the local scenario, each tool is used to estimate an OD matrix from the single City Hall point to all other points. This smaller workflow mimics work that would be done to evaluate accessibility from a single point in the city to external opportunities. In the regional scenario, a full OD matrix is estimated from all origins to all destinations to simulate the more computationally intensive workflow associated with estimating accessibility across the entire City of Toronto or Greater Toronto Area.

3. Calculating accessibility for each origin place.
The final step is to utilize the returned OD matrices to calculate place-based accessibility for the origin points. Given the multi-stream nature of these tools, additional work is required to calculate the accessibility scores. This involves joining the opportunities available at the

destinations to the computed origin-destination cost matrix through a common ID field, multiplying the destination opportunities by the travel time involved in reaching them from the origins, and then summing these weighted values over the origins. To facilitate this calculation, the "opportunities" available at each destination point are set equal to 1. In this case, we utilize a cumulative accessibility measure where the accessibility for a given origin point is the sum of all points reachable within 45 minutes. As noted in the literature review, such cumulative measures are easily understood, but tend to be an arbitrary simplification of actual travel behaviour compared to the more continuous weighting functions of gravity specifications. Nevertheless, it allows the project team to implement a simple accessibility use case in its code across the packages. Implementing other impedance functions is not expected to meaningfully affect compute time. This work is done in ArcGIS Pro for that tool, in R for the R-OTP and Emme 4 tool workflows, and in Python for the Python-OTP workflow.

#### *4.2.2 Hardware Configuration*

The computer selected for this runtime analysis is a server hosted by the University of Toronto's Transportation Research Institute. This server is configured with an Intel Xeon E7-8890 v3 CPU with 18 cores/36 threads running at a base/turbo frequency of 2.5GHz/3.3GHz with 64GB of memory. For the individual software packages, software versions are listed below:

- ArcGIS Pro version 2.4.2
- Emme version 4.4
- OTP uses:
  - Version 1.4.0 of the OTP Java runtime
  - Java 8 Runtime Environment (64-bit)
- R-OTP uses:
  - R version 3.5.2
  - RStudio Desktop 1.2.5033
- Python-OTP uses:
  - Python 2.7 with built-in IDLE for code
  - Command-line interface shell
  - Jython 2.7.1 for linking Python with Java

#### *4.2.3 Analysis Configuration*

To enable the analysis of transit networks in the study region, the team collected the most recent GTFS files for:

- Toronto Transit Commission.
- GO Transit.
- GO Transit UP Express.
- Brampton Transit.
- Burlington Transit.
- Durham Region Transit.
- Mississauga MiWay.
- York Region Transit.

Note that Oakville Transit is missing from our list of GTFS packages as it was omitted by Conveyal Analysis in their workshop. To keep the study consistent, Oakville is also not included here. Furthermore, our work focuses on trip origins and destinations within the City of Toronto, so this should not present an issue.

Because GTFS transit networks are schedule-based, a trip start time and date must be specified to facilitate the estimation of the OD matrix. For this, a start time/date of Monday December 30, 2019 8:00:00 AM was used for the ArcGIS Pro and OTP implementations. While service reductions over the year end period mean this day and time do not reflect a typical day of transit service, it is within all of the downloaded GTFS package calendars and enables the team to test the routing performance of the individual tools. For Emme, a custom GTFS-based transit network can be created within the program itself based on original GTFS packages. Such custom networks can utilize factors such as traffic conditions and are more general in terms of temporal dependence. For the Emme workflow, a typical morning-peak period (6 a.m. to 9 a.m.) in 2016 was assumed based on the available data (e.g., travel demand). Because Emme can only perform a one-hr assignment to get an OD travel time matrix, a peak-hour factor was applied to the peak-period demand to get the one-hour demand. In this sense, the inputs and output from Emme 4 are not directly comparable with the ArcGIS and OTP tools.

#### *4.2.4 Tool Overview and Configuration*

Although we have taken care to make the workflows comparable across tools, the idiosyncrasies of each tool offer different parameters that can customize the analysis but necessarily impact results. This section offers an overview of the individual tools and explains the configurations of each of their inputs for the simulated accessibility analysis.

##### *4.2.4.1 ArcGIS Pro*

ArcGIS Pro is the latest version of ESRI's ArcGIS software suite for spatial analysis and mapping. Compared to previous iterations of their ArcMap software, ArcGIS Pro is a 64-bit program with an enhanced set of tools and Python backend for programming. A screenshot of the ArcGIS Pro interface is shown in **Figure 1**. The 2.4 version of ArcGIS Pro offers an updated Network Analyst workflow that stores the network and associated input-output layers in-memory, which can speed up a network analysis. However, the core Network Analyst extensions are natively single-threaded, which greatly limits the performance of the software in a large-scale accessibility analysis workflow.

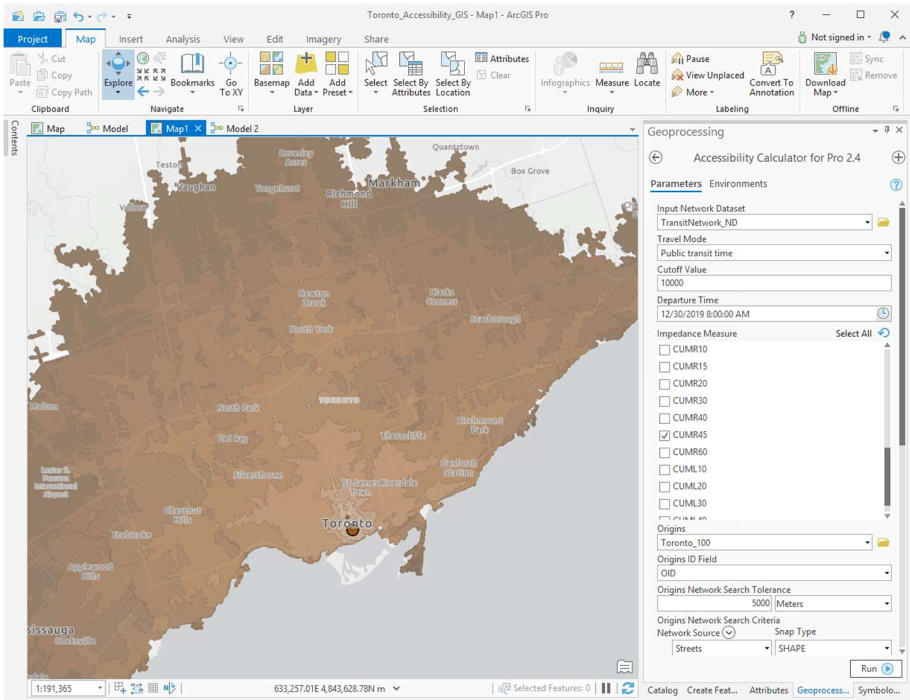


**Figure 1. ArcGIS Pro Interface**

However, ArcGIS Pro's Python backend does accept custom multi-process workflows. To that end, a custom multi-processor enabled Python script was used for this analysis based on the toolbox published in Higgins (2019). This core toolbox greatly simplifies the steps required to compute place-based accessibility within that environment. Specifically, the tool combines the steps involved in adding network locations, setting up an OD cost matrix layer and parameters, calculating the OD matrix, joining destination attributes, calculating accessibility, and summing the access scores over the input features. This custom toolbox was altered for the present research to include a multi-processing batch feature that segments input points into a number of smaller files that can be spread over worker functions. Each worker function loads its own copy of the network as a layer and solves the OD matrix. Timer functions for the relevant steps were also added to facilitate the analysis below.

Because the OTP-based solutions below utilize OpenStreetMap data for their underlying street networks, OSM was also used as the underlying network for the ArcGIS scenario to enable comparability across networks. However, ArcGIS cannot natively load OSM extract .pbf files. Instead, the open-source "OSMnx" python tool from Boeing (2017) was first used to collect a street network for Toronto from OSM and export it as a shapefile for ArcGIS. The OSMnx tool allows users to specify different types of networks: a walking network for example does not include highway links as these are not pedestrian traversable. As this work is primarily interested in multimodal walking and transit trips, the walking network option was specified.

*4.2.4.2 Emme 4.4*

Emme 4.4 is the latest version of Emme, a multimodal transportation planning software that supports the City's travel demand modelling. Emme itself is a product of INRO Consultants Inc., and a TMG Toolbox is developed by the Travel Modelling Group (TMG) at the University of

Toronto to facilitate advanced analyses of travel demand forecasting models implemented in Emme. A screenshot of the Emme 4.4 interface is shown in **Figure 2**.

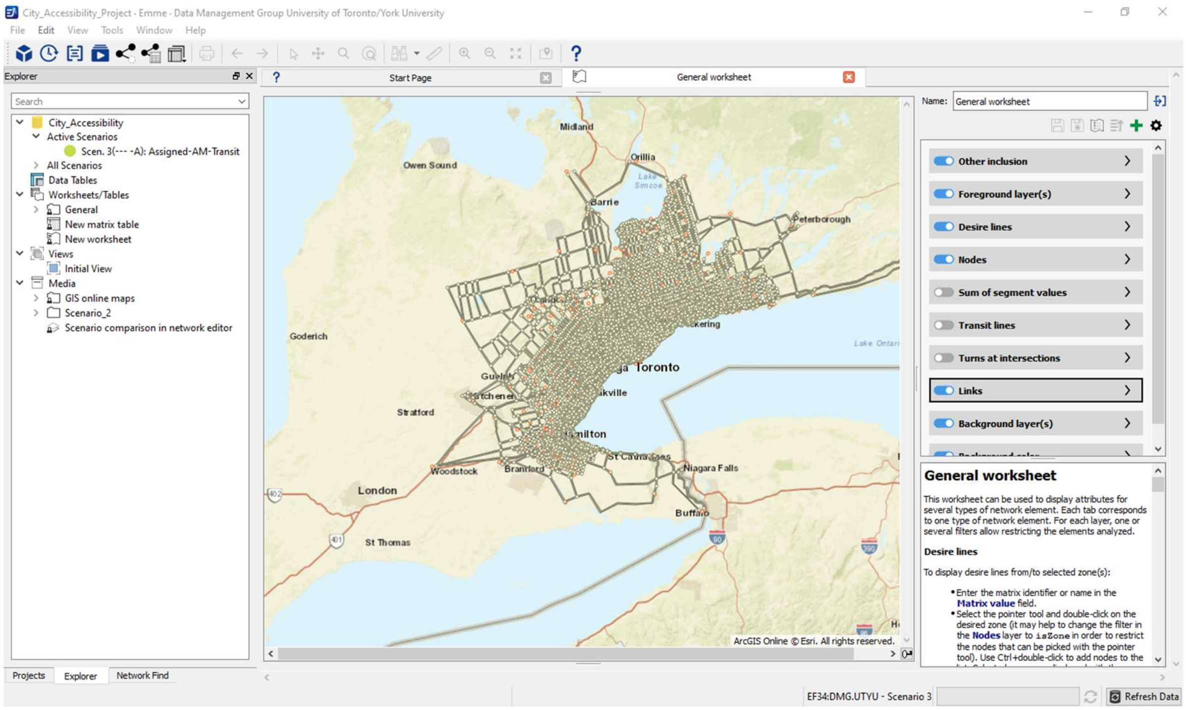



**Figure 2. Emme Interface**

Both auto and transit assignments need to be performed in Emme to estimate auto and transit OD travel times. An existing GTAModel Emme network that contains both road and transit information was used instead of importing the GTFS and OSM files directly due to a restrictions on the number of links supported by the current license. The transit network was previously built from the 2016 GTFS files, which should generally be similar to the current 2019 GTFS files used by ArcGIS or OTP-based tools. On the other hand, the road network only consists of major streets across the region, which is a sparser network as compared to the OSM networks above.

A "shortest path" tool that is newly available in Emme 4.4 was utilized to generate the auto OD matrices, and an XTMF model system built by TMG was used to generate the transit OD matrices. One limitation of this procedure is zero travel time between nearby points for auto in the local and regional scenarios. The points are imported into Emme as regular nodes and connected to the road network through zone centroids. Since the travel time on the centroid connectors is zero, it is possible that nearby points are connected to the same centroid which results in zero travel time between the points.

Another limitation of this comparison concerns Emme's ability to estimate OD matrices for transit in the local scenario (e.g., 1 x 100 OD matrix). The Level-of-Services (LOS) for transit can only be extracted at the centroid level in Emme, therefore, the points are assigned as new centroids in the network and a transit assignment is performed to obtain the travel time between the centroids/points. Due to the nature of the algorithm used in the transit assignment, only full OD matrices can be generated at the moment (e.g., 100 x 100 OD matrix). Essentially, this means a transit OD matrix cannot be computed for the local scenario. For illustration, the computational times for the local scenario below are estimated based on the regional scenario using the following equation:

$$Computational\ time\ for\ 1xN = Max\left(1, \left\lceil \frac{N-1}{RemainingCentroids - 1} \right\rceil\right) * TimeFor1Iteration$$

Where *RemainingCentroids* is the number of remaining centroid spaces under the Emme license (e.g., 1358 spaces remaining in the GTAModel network with a license size of 15), and *TimeFor1Iteration* is the computational time for 100 x 100 OD matrix with only 1 iteration. Taken together, these program limitations mean the Emme results are not strictly comparable with the ArcGIS and OTP outputs.

*4.2.4.3 General OTP, R-OTP, and Python-OTP*

OTP is an open-source multimodal trip planner that was originally developed for TriMet in Portland, OR. Since its initial development from 2009 to 2011, the tool has continued to be developed and implemented by teams from across the world. The OTP tools utilize OpenStreetMap (OSM) for their underlying street networks and employ a Java runtime environment for their routing backend. This Java environment presents a point-and-click interface for point-to-point route analysis, however, the environment can be utilized for larger analyses through additional tools such as R and Python.

In terms of setting up the tool for the runtime analysis, the team first collected a metropolitan-level OSM extract for Toronto using the free extract service from interline.io[3]. This service allows the analyst to download a .pbf OSM file, which OTP looks for when building its network graph. Alongside the relevant GTFS files, an OTP "server" can now be created using R or Python. There is flexibility as to whether the graph building is performed on the machine's disk or on a computer's local area network (or web) server—the choice is at the discretion of the user. Graphing on a networked server creates an OTP graph that any computers on a network can access, enabling shared use of a single OTP instance. Graphing on disk creates a local OTP server and allows for greater flexibility for batch analysis, so tests for this study were performed on a graph saved to the disk.

Support for utilizing OTP within R is afforded by the *opentripplanner* package. This package allows the analyst to build and set up an OTP network graph using OSM and GTFS data and connects it to R as a local server. From this, http requests are made to the OTP Java server and it returns trip itineraries for the given modal options. The R-OTP analysis does have a multi-processing option, where multiple threads are used to make requests to the OTP server and return the trip itinerary. A screenshot of the R-OTP workflow implemented in an R Notebook in R-Studio is shown in **Figure 3**. In addition to OD matrix calculation, R-OTP can be used to create isochrones as simple features in R.

[3] https://www.interline.io/osm/extracts/

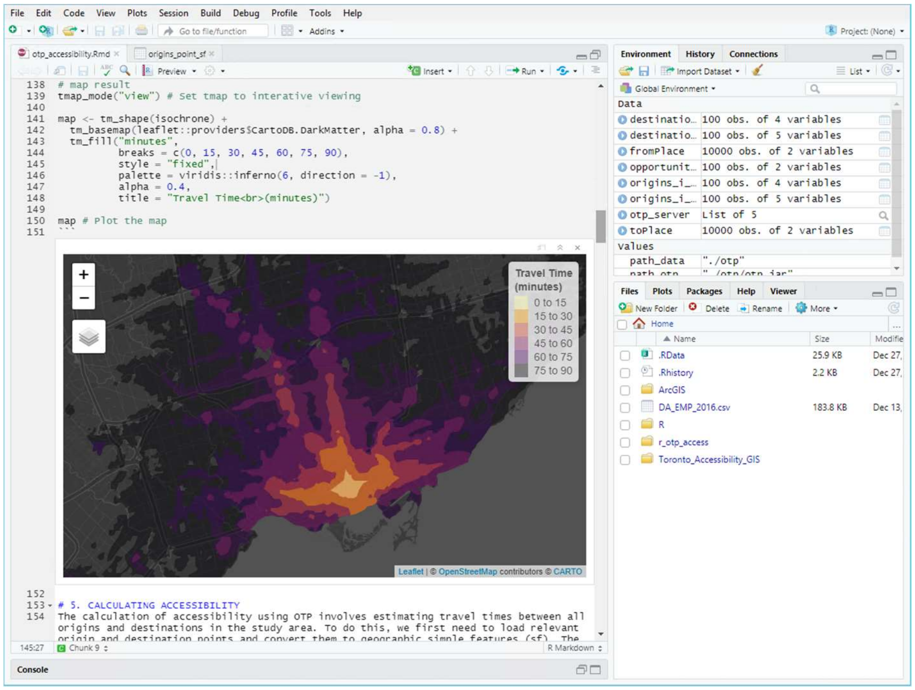



**Figure 3. R-OTP Interface**

For Python, the implementation is less streamlined than R-OTP. Commands are used to call an OTP Java executable to build an OTP graph referencing relevant GTFS and OSM files and Java. Once built, a Python script can be used to load the graph, parameterize the router, and obtain origin-destination information. The Python script uses OTP's scripting API library. To run the script, Jython is required to allow Python to run Java—this is captured in a command which calls Java 8, the OTP Java executable, a Jython Java executable, the Java main class for standalone OTP analysis, and finally, the Python script. The Python script, after loading the graph, establishes rules for routing (e.g. maximum walking distance), and loops over the origin(s) and destination(s). Python-OTP does allow for multi-processing in a similar way to R-OPT, where multiple threads can call a single OTP graph for routing. Like R-OTP, Python can also be used to create isochrones from OTP, which are saved as GeoJSON files.

In any implementation of OTP, the analyst has the ability to customize the routing algorithm in a number of ways. For example, trip modes can include many combinations of walking, cycling, transit, and driving. Walk reluctance parameters can also be set to account for the maximum distance individuals are expected to walk when using transit, which is unique to this tool relative to the others. OTP also has the option of returning several trip itineraries between a given origin and destination pair on a set time and date and can account for flexibility in a trip's start time. For example, if a departure time is set at 8:00 AM, but the next train does not come until 8:30 AM, OTP can be set to not start the individual's trip until the total walking and in-vehicle travel time is minimized. On the other hand, a hard departure time can be set, where the trip begins at 8:00 AM and includes waiting time for the train (or it may route the individual on a different path that minimizes trip time). This can present some conceptual questions to the analyst in terms of accessibility analysis, as in the latter case, the first returned itinerary may not minimize overall walking and in-vehicle travel time, but it does minimize the total trip time. In such a scenario, the computed accessibility measures are highly sensitive to the set departure time. In the present case, ArcGIS and Emme work according to hard departure times with minimized trip times, so we set OTP to work in this way as well.

#### *4.2.5 Results, Local Scenario Analysis*

Results for the local analysis scenario are presented in Table 2. For ArcGIS and Emme, there are small penalties associated with calculating the location of input points on the network. Within Emme and ArcGIS, these are one-time costs if the same input points are re-used. Comparable network location time costs are opaque in the OTP tool and cannot be separated out from the larger solve time. They also do not carry over to future analyses and must be re-calculated each time.

For calculating the OD matrix, increasing the number of destinations dramatically reduces the performance of the R-OTP package. The Python-OTP and Emme perform much better. In ArcGIS Pro, the multi-processing script is based on batching origins; with only a single origin in the local scenario, this workflow is limited to a single thread.

Finally, computational time involved in calculating accessibility for the local scenario is comparatively less for all tools using R (R-OTP and Emme) or Python (Python-OTP), typically requiring about a second or less. In comparison, the ArcGIS Pro process of joining destination attributes and calculating accessibility using the Field Calculator takes much longer than its R or Python counterparts. The custom tool relies on the core Add Join, Field Calculator, and Summarize Attributes tools in ArcGIS Pro to join destination opportunities to the OD matrix, calculate accessibility, and summarize over the origins. Finally, for Emme, the larger 1 x 100,000 analysis exceeds the licence available at the university, and as such was not run. Note that point-based calculations within Emme are only available within screen command line operations and so are not a feasible proposition for any automated procedure. Thus, all Emme-based calculations must occur at the traffic zone level of spatial aggregation.

**Table 2. Local Analysis Results**

| Activity | Scenario | ArcGIS Pro | R-OTP | Python-OTP | Emme (estimated) |
|---|---|---|---|---|---|
| Make/Build Network | All | 11m 26s | 8m 19s | 7m 24s | 1m 25s |
| Calculate OD Matrix Travel Time | 1 x 100 | 26s | 20s | 17s | 2m 23s |
| | 1 x 1,000 | 35s | 2m 49s | 18s | 4m 46s |
| | 1 x 10,000 | 2m 33s | 22m 37s | 22s | 28m 36s |
| | 1 x 100,000 | 18m 20s | 7h 12m 15s | 57s | not run |
| Join, Calculate, and Summarize Accessibility | 1 x 100 | 19s | <1s | <1s | <1s |
| | 1 x 1,000 | 19s | <1s | <1s | <1s |
| | 1 x 10,000 | 35s | <1s | <1s | 2.6s |
| | 1 x 100,000 | 28m 24s | 1.5s | <1s | not run |

#### *4.2.6 Results, Regional Analysis*

Results for the full n x n regional analysis matrices are presented in **Table**3. In line with the increase in OD size, each of the tools now exhibit greater computational time. Interestingly, the computational time for the $100^2$ scenario is greater than that for the 1 x 10,000 scenario above, which we attribute to extra time involved in finding the shortest street network and transit schedule paths between points in a much more geographically varied sample. However, this type of analysis magnifies some of the shortcomings associated with each workflow. First, as above, the R-OTP implementation exhibited the worst performance and was not able finish the $1{,}000^2$ scenario within 24 hours. The batching of input origins enables the ArcGIS Pro workflow to perform well in the regional analysis, and the Python-OTP and Emme tools show high performance for large accessibility calculations. Where ArcGIS Pro falls short is in the calculation of accessibility. The existing script relies on joining tables and using the Field Calculator to calculate accessibility. This process does not scale well in the project team's custom code, but could be changed with further development time. In comparison, the accessibility calculations for Python-OTP and Emme are significantly faster.

**Table 3. Regional Analysis Results**

| Activity | Scenario | ArcGIS Pro | R-OTP | Python-OTP | Emme |
|---|---|---|---|---|---|
| Make/Build Network | All | 11m 26s | 8m 19s | 7m 24s | 1m 25s |
| Calculate OD Matrix Travel Time | $100^2$ | 2m 48s | 25m 30s | 43s | 5m 5s |
| | $1{,}000^2$ | 7m 58s | >24h | 9m 10s | 18m 33s |
| | $10{,}000^2$ | 1h 40m 18s | *not run* | 9h 12m 26s | 13h+ |
| | $100{,}000^2$ | *not run* | *not run* | *not run* | not run |
| Join, Calculate, and Summarize Accessibility | $100^2$ | 22s | 1s | <1s | <1s |
| | $1{,}000^2$ | 6m 1s | *DNF* | <1s | 1.5s |
| | $10{,}000^2$ | *DNF* | *not run* | *1s* | 6m 58s |
| | $100{,}000^2$ | *not run* | *not run* | *not run* | not run |

To compare each tool's routing results, we have performed a scatterplot comparison of travel times within the output OD matrix for each tool for the 1,000 x 1,000 scenario. Figure 4 plots the results for Python-OTP and Emme, showing a difference in estimated transit travel times for the points across the City of Toronto with an $R^2$ of 0.84. In particular, Python-OTP returns a mean transit travel time of 77.78 minutes with a standard deviation of 30.93 while Emme's travel times have a mean and standard deviation of 92.5 and 36.34 minutes respectively. Fundamentally, these differences are no doubt in part due to Emme's congested transit network, which arguably offers more realistic travel times for surface transit, and its sparse street network that alters the location and "last mile" connection between different origins and destinations.

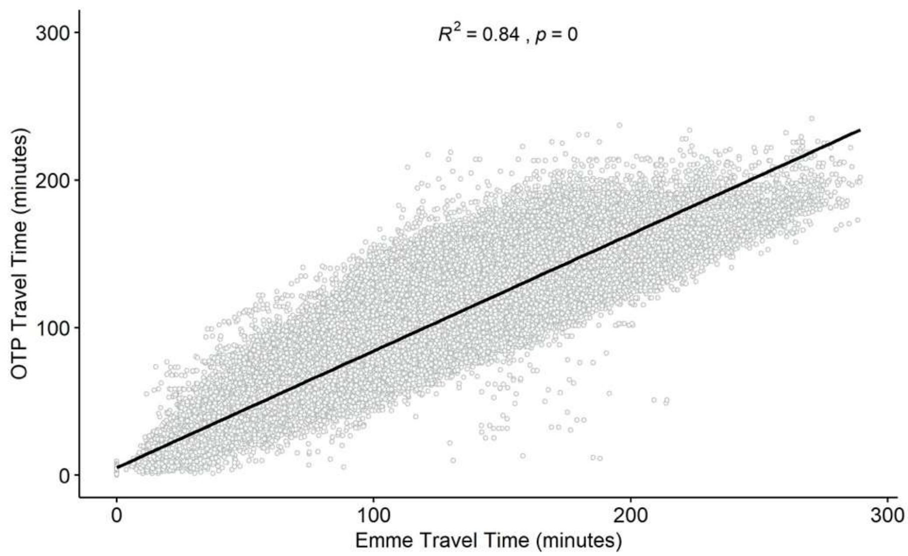



*Figure 4. Scatterplot of OTP vs. Emme Travel Times*

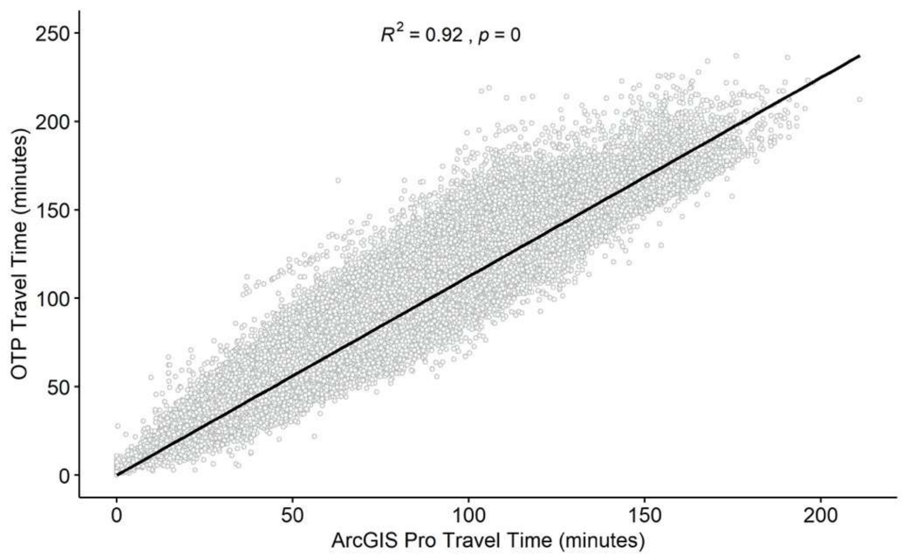



*Figure 5. Scatterplot of OTP vs. ArcGIS Pro Travel Times*

In comparison, the travel times generated across ArcGIS Pro and Python-OTP are more consistent (Figure 5) with an $R^2$ of 0.92. While built on a similar underlying OSM network with the same GTFS files, ArcGIS calculates some trip durations of up to 211 minutes versus OTP's maximum of under 241 minutes. Mean travel time from ArcGIS is 69.55 minutes with a standard deviation of 26.58. These factors suggest an overall bias towards shorter travel times for the same trips in ArcGIS Pro. Reasons for these differences are less clear but could be attributed to each ArcGIS Network Analyst's street and transit routing algorithms. For comparison, the $R^2$ between ArcGIS Pro and Emme is 0.88. Such results can have a significant effect on accessibility analysis and warrant further investigation in the academic community.

#### *4.2.7 Multi-stream Tool Conclusions*

Based on the results of the local and regional analyses, we now draw some conclusions about the relative strengths and weaknesses of each tool for the given accessibility workflows. Tests revealed that Emme was the most time-efficient for smaller tasks, while ArcGIS Pro's Python multiprocessing allows network analyses to scale in terms of calculating the OD matrix, but falls short in the calculation and summation of accessibility from the matrix. While it does not scale as well as ArcGIS Pro for larger OD analyses, Python-OTP performed well overall considering its feature mix that combines the strengths of ArcGIS Pro in terms of OSM network density and Emme in terms of computational time.

*4.2.7.1 ArcGIS Pro 2.4.2*

| | |
|---|---|
| Strengths | • Accessibility calculation is straightforward using the *Accessibility Calculator* custom toolbox produced by Higgins (2019)<br>• Ability to edit networks and engage in local quality control of source network features<br>• Mapping of results is built-in to the software<br>• Greater degree of support versus open-source solutions<br>• Point-and-click alongside Python coding enables a flexible work environment with reduced usability challenges versus the OTP solutions<br>• Several steps are run-once, such as calculating network locations of input points. This can save processing time versus the OTP solutions<br>• Ability to customize many trip options through the properties of the network dataset, such as access and travel mode |
| Weaknesses | • Commercial software requires licensing<br>• Less flexible transit routing scenarios compared to OTP. With a travel time cut-off of 60 minutes for example, ArcGIS Pro will allow trip makers to walk 60 minutes, which does not likely reflect how people behave.<br>• Public transit routing algorithm is not open-source, so it is not clear how it is being implemented versus other transit routing algorithms such as A* (OTP) or $R^5$ (Conveyal).<br>• For scenario-based sketch planning, the underlying network must be rebuilt to incorporate changes to GTFS routes or schedules.<br>• Ease of calculation is a result of a custom user package produced by Higgins (2019)<br>• Significant bottlenecks in performance associated with large datasets on both ends of an accessibility analysis, including estimating the OD matrix and calculating accessibility, lead to second-slowest run time. Part of this is because ArcGIS Pro’s network solver is only natively single-threaded, however there is potential for this to be sped up through custom coding of a multi-process OD cost matrix solver. For accessibility calculation, alternative methods to table joins and the field calculator exist such as utilizing python dictionaries and search cursors, but more custom coding is required. |

*4.2.7.2 Emme 4.4*

| | |
|---|---|
| Strengths | • Although the Emme road/transit network could take weeks to be built completely, it can be easily and quickly used for any scenario afterwards<br>• Supports the mode-specific travel time calculations for OD matrices<br>• Customizable auto and transit assignments (e.g., demand, convergence setting)<br>• The networks can be modified for future scenario tests (e.g., expanding the highway, adding a transit line)<br>• Local and full OD Matrix analyses for auto are based on the same network and assignment results. Once the assignment is done, both local and full OD matrix analyses can be performed quickly.<br>• Greater degree of support versus open-source solutions |
| Weaknesses | • It requires a certain amount of preparation (e.g., generating demand, building network, writing Emme scripts, building XTMF model system). However, most of the preparation works is a one-time job, which does not need to be re-done every time.<br>• To calculate the accessibility, it requires the integration of multiple tools (e.g., Emme and its python-based scripts, R, XTMF). The Emme license needs to be purchased but R and XTMF are free open-source.<br>• Building the Emme network from OSM or GTFS files is difficult and making a complete Emme road/transit network from scratch could take weeks<br>• The existing Emme road network (i.e., GTAModel network) is sparser than the OSM network<br>• Possible zero travel time for auto between nearby points due to the sparse network<br>• The number of points that can be tested is restricted by the Emme license size (e.g., a max size of 43,124 nodes under the Emme license size of 15)<br>• Local analysis is currently unavailable for transit, only full OD matrix can be generated for transit at the moment<br>• Additional programming required to calculate accessibility from the OD matrix (R used in this case); Python could have been used |

*4.7.2.3 OTP 1.4.0, R-OTP, and Python-OTP*

| *General OTP* | *Common to all implementations of OTP* |
|---|---|
| Strengths | • Free and open-source<br>• Highly customizable input analysis parameters, including the specification of more realistic routing based on factors such as walk reluctance and maximum walking distance to transit<br>• Routing engine is contraction hierarchies for fast on-street network routing<br>• Multi-processing and parallel computing support depending on analyst code<br>• Ability to set up shared routing server on network or local graph on disk |
| Weaknesses | • Must use OSM (or OSM-format) streets, which involves a bulky download and reduced ability to locally edit the network or engage in quality control<br>• For scenario-based sketch planning, the underlying OTP graph must be rebuilt to incorporate changes to GTFS routes or schedules.<br>• Limited technical support available, OTP documentation could be considered generally poor or overly technical<br>• The existing transit routing algorithm is relatively slow. Simple network routing of walking or driving uses the contraction hierarchies algorithm and is faster than transit routing, which uses the A* algorithm for schedule-based routing. Updates to OTP with a new $R^5$ transit routing engine appear promising and are to be released in early 2020. This engine already powers Conveyal Analysis, the results for which we will discuss further below.<br>• As transit services as are schedule-based and OTP takes many different factors into account, it may be that the first returned trip itinerary by OTP is not the minimum travel time. Often, to get the fastest route, multiple itineraries can be returned for a given OD pair (typically three) and the minimum selected from them. This results in an increase in compute time: for the 1 to 10,000 R-OTP compute scenario for example, changing the number of itineraries returned from 1 to 3 increases the run time from 22m 37s to 26m 52s.<br>• No GTFS editing capabilities<br>• Compared to ArcGIS Pro and Emme, pre-processing steps such as calculating network locations of input points are run for every analysis<br>• Additional code required to calculate accessibility<br>• Additional packages or software required for mapping results |

| *R-OTP* | *For R-OTP in addition to the general comments above* |
|---|---|
| Strengths | • Multiprocessing support is native to the package that sets up an initial OTP server in Java and makes calls to it for routing between any given latitude-longitude pair, with the calls distributed over different R processes<br>• Code and results easy to store together in an R Notebook using R Studio<br>• In addition to OD matrix calculation, R package offers ability to calculate and map isochrones |
| Weaknesses | • Code-based interface can present usability challenges<br>• Very slow OD matrix calculation versus Python-OTP and Emme<br>• Multiprocessing does not appear to scale well at the high end. With all cores specified, the UTTRI system never utilizes above 70% compute capacity, suggesting there is a bottleneck associated with repeated calls to the single OTP server in the R-OTP package implementation |

| *Python-OTP* | *For Python-OTP in addition to the general comments above* |
|---|---|
| Strengths | • Python scripting is most commonly used for OTP, and most well-documented by users<br>• Multiprocessing of OTP using Python appears to scale better than implementation in R-OTP and will utilize all of the computer's compute resources<br>• Ability to run batch requests on disk instead of on local server allows for significantly faster processing at the loss of more detailed information |
| Weaknesses | • Python and command line-based interface can present usability challenges<br>• Commonly used information (e.g. Isochrones) are not easily obtained unless using local server for routing requests |

### 4.3 Single-stream Tool: Conveyal Analysis

Conveyal is a company that has grown out of a core in OTP and has developed a single-stream tool for building and testing the accessibility impacts of different travel scenarios, including transit, walking, driving, and cycling. At the core of Conveyal Analysis is a web-based geographic information system for visualizing, editing, and analyzing data. Behind this is a routing engine that uses OpenStreetMap and GTFS data to build a network graph for a region, the $R^5$ routing algorithm for solving routing problems, and a server infrastructure for parallelizing workflows. As a commercial product, Conveyal Analysis is not free to use. To evaluate the potential of this option, the team welcomed a representative from Conveyal to demonstrate the Analysis software platform on December 2, 2019. A screenshot of the Conveyal Analysis interface is displayed in **Figure 6**. The implementation of Conveyal Analysis that we tested does not natively handle transportation factors like traffic congestion or the demand for transit – it is more akin to a very powerful custom implementation of OTP.

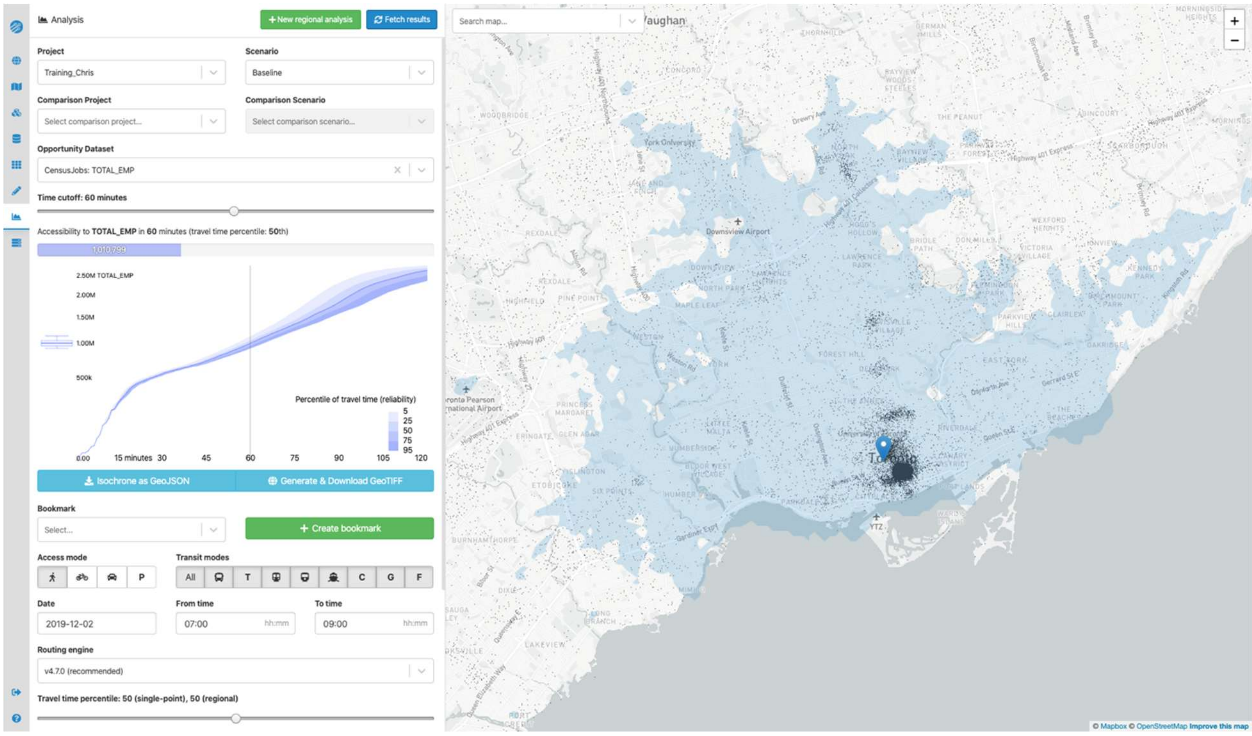



**Figure 6. Conveyal Analysis Interface**

Conveyal Analysis is based on analysis scenarios. First, the analyst can utilize the existing GTFS network to establish baseline travel network and analyze cumulative accessibility to some destinations from an origin location. This single-origin workflow is similar to the local analysis scenario for the multi-stream tools above. At the backend of such an analysis with Conveyal is a highly scalable routing workflow: to facilitate computation, Conveyal dedicates up to 250 individual instances of the network object on servers run by Amazon Web Services. Compared to the multi-stream workflows above, which benefitted from an 18-core/36-thread server, the parallelization over 250 individual servers provides a foundation for rapid scenario testing.

In particular, one notable benefit of this computational power can be seen in the estimation of accessibility in the local analysis scenario. The analyst first selects typical criteria, such as the access modes to transit (e.g. walking, cycling, driving for park-and-ride), transit modes available for the trip, the origin point location, and a dataset of geographically distributed opportunities. In addition, while the analyst still specifies a maximum travel time and trip start time/date to align with the GTFS schedules, Conveyal Analysis estimates of local accessibility are probabilistic in nature in the sense that multiple start times are simulated for every minute within one hour of the selected start time. Moreover, trip travel times between 0 and 120 minutes are also automatically computed. The result is a stacked percentile chart highlighting accessibility (**Figure 7**). This graphic shows cumulative access to the number of jobs in traffic zones from the Transportation Tomorrow Survey over the 0-120 minute travel time window and a simulation of transit schedule randomness that highlights accessibility between the 5$^{th}$ percentile and 95$^{th}$ percentile of trip scenarios. The 5$^{th}$ percentile is viewed as a "very lucky" scheduling optimum, where total travel time is minimized over the transit schedules, while the 95$^{th}$ percentile represents a "most likely" travel time scenario. These accessibility analyses can also be visualized by isochrones that change as the analyst moves the travel time and trip percentile sliders in the Conveyal Analysis interface.

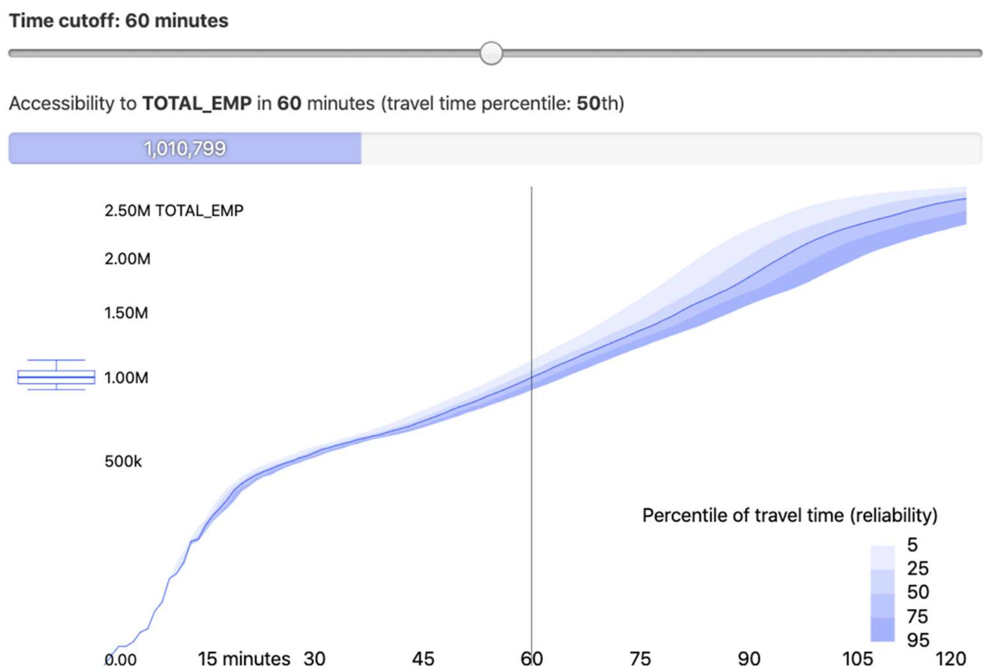



**Figure 7. Stacked Percentile Chart of Accessibility by Travel Time**

From this baseline, the analyst can then perform edits on the underlying transit network. Options available are to adjust the trip pattern of a transit service, its dwell time and travel speed, conversion of a transit service from schedule-based to frequency-based, removing stops or trips, rerouting the service, and other custom modifications. Once these modifications are set, the transit network can again be used to run comparison cumulative accessibility scenarios within the local analysis workflow.

Once the local analysis scenarios are set, the analyst can then perform a regional analysis. As in the multi-stream workflows above, this involves the computation of accessibility from all origins to all destinations. However, in the case of Conveyal Analysis, the origins and destinations utilize a raster-based data model rather than a point-based vector data model employed in our tests of ArcGIS Pro, Emme, and OTP. In particular, Conveyal Analysis converts the entire study area into a regular grid of 250m x 250m cells. In the demonstration for Toronto, the analysis region consisted of 747,065 individual cells (**Figure 8**). Destination opportunities are also rasterized upon upload to the same grid (for use in both the local and regional analysis scenarios). Next, the transit network, travel options, travel time, and travel reliability percentile set after the local analysis are used to estimate cumulative accessibility for all raster cells in the region.

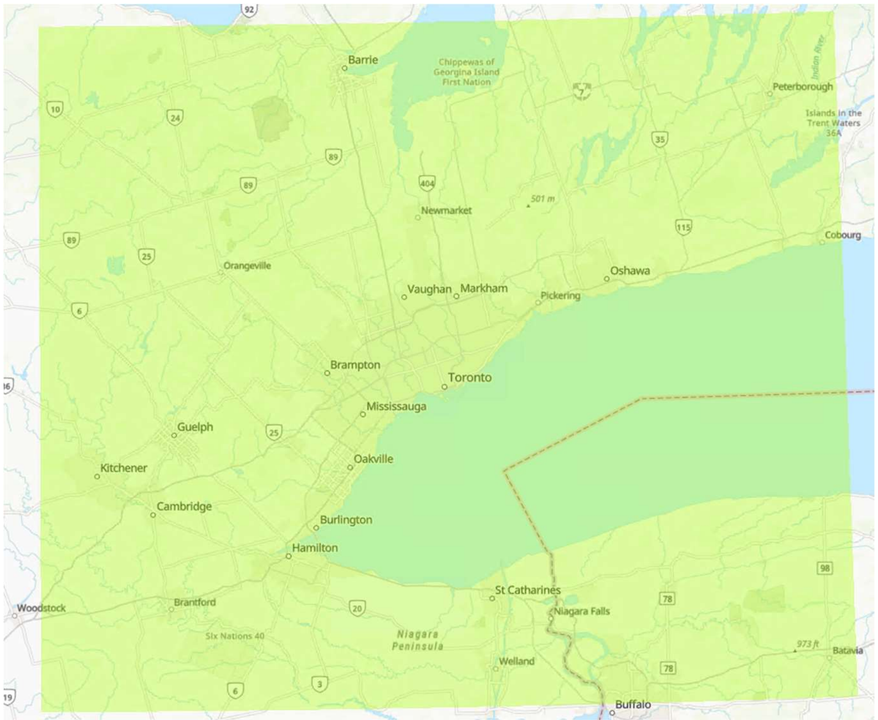



**Figure 8. Conveyal Analysis Region**

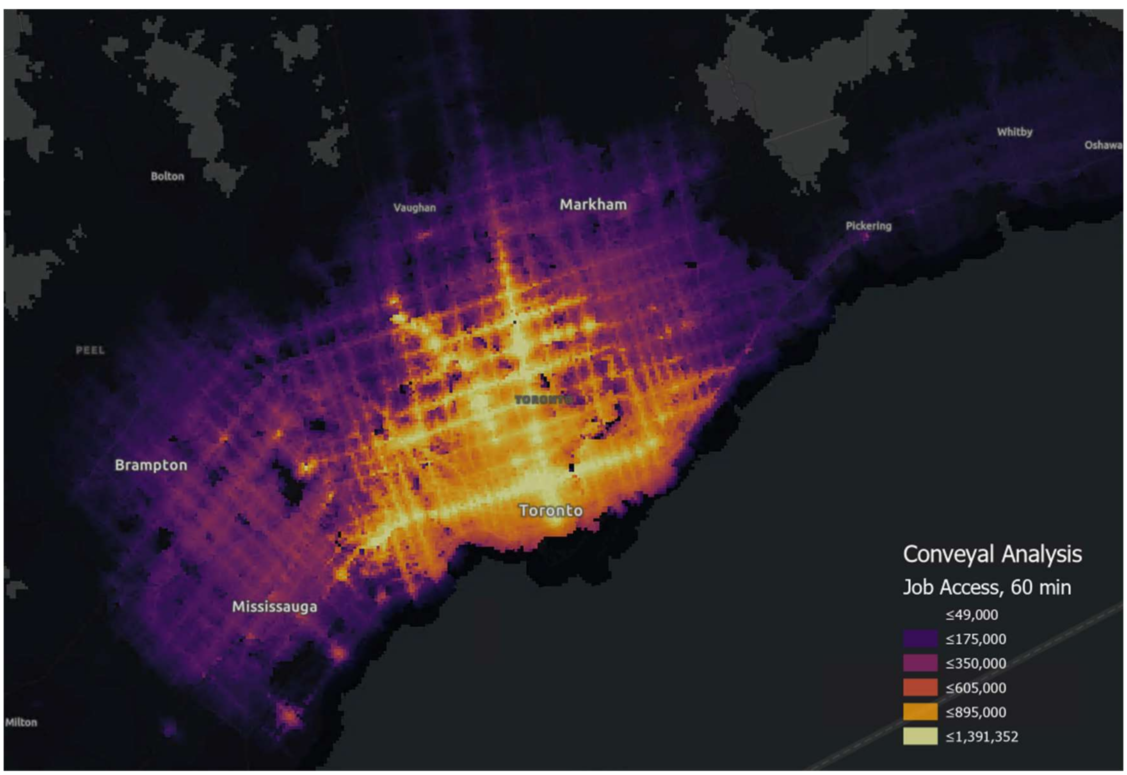



**Figure 9. Conveyal Analysis Demonstration Regional Analysis**

The output from the regional analysis is the underlying raster with cell values representing the cumulative accessibility score for each cell centroid. Compute times for the regional analysis may vary based on the type of transit customizations employed, but for a test run using the existing transit network, an analysis of accessibility at the 50$^{th}$ percentile to jobs reachable within 60 minutes at Dissemination Areas for the Hamilton, Oshawa, and Toronto CMAs took approximately 6 minutes. In effect, this is a 747,065 x 747,065 OD matrix calculation, parallelized over the 250 Conveyal servers. **Figure 9** displays the output from this analysis, which was exported as a raster from Conveyal Analysis and mapped in ArcGIS Pro to test this type of workflow. Notice that results from Oakville are missing due to those GTFS files being omitted from our demonstration data.

As a commercial enterprise Conveyal offers a number of other additional services and customizations that could be pursued by the client. For example, Conveyal has indicated that customizations that have been implemented for clients include linking delay factors to roads to better model congested travel times and road network performance (either through zone-based delay polygons or linking measures of delay to OSM link IDs) and reading “retrospective” GTFS files created through the methods outlined in Wessel et al. (2017) to model the observed performance of transit vehicles rather than stated schedule optimums. At present, there does not seem to be any method for accounting for transit vehicle capacities or crowding. Conveyal has also noted to the project team that alternative resolutions of the underlying raster grid framework could be pursued to achieve a finer level of spatial granularity than the current 250m by 250m cells. However, exploring the implementation of such opportunities is beyond the scope of the present project. To summarize, our opinions of the strengths and weaknesses of Conveyal Analysis are as follows:

| | |
|---|---|
| Strengths | • Natively parallelized workflow leads to dramatic time savings for local and regional analysis<br>• Similarly, the $R^5$ routing engine contributes to computational time savings for schedule-based routing compared to the A* algorithm still used in OTP<br>• Available computing power enables the calculation of sliding time windows and trip probabilities for local analyses. This would take multiple runs of the multi-stream tools to replicate.<br>• Built-in GTFS editing for scenario building<br>• Customizable trip inputs such as access mode<br>• Point-and-click interface means no coding required<br>• Ability to export analysis results for mapping in other applications<br>• Other options demonstrated to the project team include Monte Carlo simulations to fill in the trip times of frequency-based transit schedules, the incorporation of fares, the potential to incorporate link-level travel times for street-based modes such as cars and buses in traffic, and a suite of stakeholder engagement tools<br>• Additional vendor customizations possible based on needs of the client |

| | |
|---|---|
| Weaknesses | • Cumulative accessibility is the only accessibility option. Other accessibility functions that use a continuous decline function and are more consistent with travel behaviour are not options in the software<br>• No ability to natively output an OD matrix for a set of inputs (apparently this can be customized)<br>• Trip parameter customizations appear more limited to the analyst than base OTP, there does not appear to be a way to set factors like walk reluctance or maximum walking distance in the base interface<br>• Raster-based regional analysis alters data from the original vector-based zones or points, which may not be desired behaviour<br>• Similarly, raster-based data model can lead to errors in the analysis. Because all zones join the network from the 250m x 250m cell centroid, there is location error of up to 125m. This can lead to cells "snapping" to the network at incorrect locations, and may not be suitable for micro-scale analyses.<br>• The relatively limited GIS aspects of Conveyal Analysis mean additional packages or software may be required for mapping results |

# 5. RECOMMENDATIONS

## 5.1 Primary Recommendations

The findings of the analysis described in the previous section lead to five primary recommendations.

**1. A multi-stream procedure is required to address the full range of accessibility analysis and planning needs of the City of Toronto**. While an interesting product (as discussed further below), the single-stream Conveyal Analysis tool is not sufficiently broad or complete in its capabilities to be suitable as the only accessibility measurement procedure in the City's toolbox.

2. Among the multi-stream procedures evaluated in this study, **it is recommended that R-OTP be deleted from further consideration as the primary basis for the City's accessibility tool**. Python-OTP dominates R-OTP in terms of computation time. While R is a very flexible and universally used language, so is Python. Further, Python is much more compatible with GTAModel than is R.

3. Similarly, **it is recommended that ArcGIS Pro be deleted from further consideration as the primary basis for the City's accessibility measurement tool.** While not without its strengths, ArcGIS Pro is dominated by the other procedures from a computational perspective, at least when considering the software in its native form. While development time could be used to speed up and customize accessibility workflows using the built-in Python environment, the software is generally not as flexible in terms of transit routing customization and its strengths are largely duplicated by the other procedures considered. If the City has ArcGIS Pro licences for other purposes, it may prove to be useful for some purposes with the City's overall planning functions – notably to exploit its very strong visualization and mapping capabilities, but we

cannot recommend that it should constitute the foundation for the multi-stream, general-purpose accessibility measurement tool at this time.

**4. A hybrid Emme – Python-OTP solution is proposed as the multi-stream procedure for City of Toronto adoption.** The proposed solution and the rationale underlying this recommendation is discussed in detail in Section 5.2

5. While not sufficient as the City's primary accessibility measurement tool, Conveyal Analysis has many promising capabilities and features for City use as a sketch planning tool for a number of possible applications. It may also be viewed as a possible extension of the recommended multi-stream tool. Significant trade-offs, however, exist between the potential utility of this tool and its cost to the City as a commercial product that are beyond the scope of this study to resolve. Thus, **it is recommended that the City conduct its own assessment of Conveyal Analysis to determine its cost-effectiveness for their purposes**. Section 5.3 provides some additional guidance concerning issues and opportunities that the City should consider in this assessment.

## 5.2 A Hybrid Emme – Python-OTP Multi-Stream Procedure

After evaluating an Emme solution and a Python-OTP solution as possible "either-or" options (as presented in Section 4), it became clear to the study team that the optimal, practical solution is to develop a hybrid workflow that:

- Exploits the best features of both software systems.
- Minimizes (indeed, eliminates) redundancies in staff workload with respect to network coding.
- Maximizes compatibility/consistency in network calculations of travel times, etc. between both systems.
- Is fully automated with respect to:
    - Flow of information between the two systems.
    - GTAModel generation of inputs for and (routine) calls to the accessibility calculator.
- Is cost-effective in that OTP and Python are both free, open-source software and the City already owns the Emme licences needed.

As discussed in Section 4, Emme's strengths include:

- It is core software for all GTAModel network modelling. Any accessibility calculations should be maximally compatible with the travel times and associated travel demand forecasts that are generated by GTAModel and its Emme-based road and transit assignments.
- Although a commercial product, City Planning already owns Emme licences and so no additional cost is involved in the use of Emme within an accessibility calculation workflow.
- City Planning staff are fully trained in the use of Emme and have decades of experience in its use.
- Both base and future year road and transit networks are coded in Emme as part of City Planning's normal workflow. Thus, no additional work is required to create Emme-based

networks for accessibility calculations. Emme has powerful network editing and display capabilities to support network coding and debugging.

Emme's primary weakness with respect to accessibility calculations is its inherently zone-based structure, which makes spatially-disaggregate, point-to-point calculations cumbersome at best. The transit network, developed from GTFS files, is spatially disaggregate, but Emme's street network is comparatively sparse. Input origin and destination points must be coded as centroids with connector links to the road and transit network. While this can be done, it is an "awkward" use of the software, and license size restrictions ultimately limit the number of points that can be included in the calculations.

Python-OTP's strengths and weaknesses to a fair extent are the converse of Emme's. OTP is designed to do point-to-point calculations and so is very well-suited to compute accessibilities at disaggregated spatial resolutions, especially for transit-based measures. Its transit network is GTFS-based (as is Emme's) and its transit assignment procedure is comparable to Emme's in many ways. Moreover, the software is very flexible, and could be set up as a shared server for shared use and programmed into a larger overall multi-stream accessibility analysis workflow within the city.

Two major issues, however, exist in an OTP-based solution. First, OTP's road assignment procedure is not directly suitable for computing road-based accessibility measures since it cannot compute congested travel times. Given the level of roadway congestion within the City and GTHA as a whole, it is very important that congestion effects be included in any road-based accessibility measure. Roadway congestion also affects surface transit speeds and travel times. These effects are endogenously accounted for in the GTAModel Emme implementation, but they are not computable within OTP.

Second, construction of road and transit networks as stand-alone products (i.e., independent of the existing Emme networks) would require considerable effort on the part of City staff that would be an almost entirely redundant effort relative to the effort that is already being invested (and will continue to be invested) in Emme network updating. In particular, it is not even clear how future road and transit networks could be constructed for OTP given that neither GTFS nor OSM files exist for future year networks – which, presumably would be a major, if not the primary, application for accessibility calculations (i.e., as opposed to base year analyses). The consistency/compatibility of such independently constructed networks with the GTAModel-based Emme networks (and hence the consistency of the accessibility calculations with GTAModel results) is far from clear and could represent a major challenge to ensure. Moreover, the use of OSM networks within OTP raises issues of data quality, as the street files are open and editable to the larger community and thus may include geographic or topological errors and changes over time.

Given these observations, the proposed solution is to create a hybrid workflow, as illustrated in Figure 10, in which:

1. All network coding and editing is done within Emme (exactly as currently occurs).

2. For any scenario of interest, GTAModel (or, if sufficient for the given scenario, simply Emme road and transit assignments) can be run to generate equilibrated, congested auto and transit travel times/speeds.
3. A GTFS file containing the transit network and its equilibrated speeds is exported.
4. The GTFS transit network file and the OSM-format road network file are imported into Python-OTP.
5. All alternative mode-based (e.g. walking, cycling, walking and transit, cycling and transit, etc.) accessibility calculations are undertaken with Python-OTP. Points of Interest (POI) and other GIS-based data required to undertake the accessibility calculations can be imported as needed.
6. If auto-based accessibility analyses are required at this lower level of spatial aggregation, a road network file with congested link speeds is exported from Emme and edited to include local streets using a GIS program. The combined network is exported in an OTP-readable OSM format.

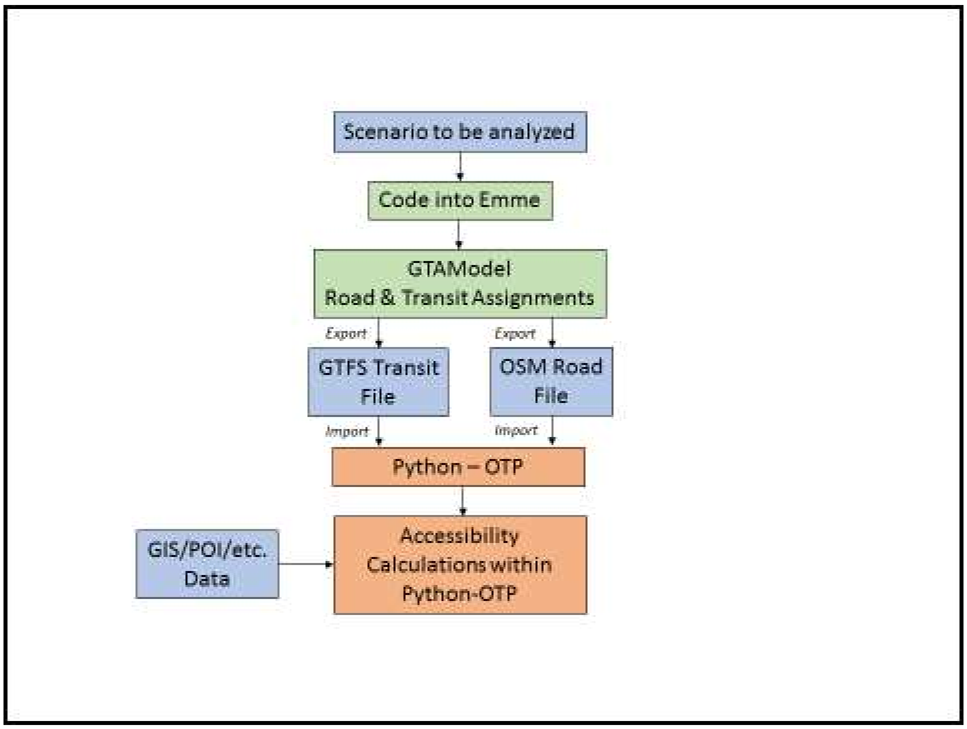



**Figure 10. Hybrid Emme – Python-OTP Accessibility Calculation Work Flow**

The first three tasks would be implemented within XTMF/GTAModel. Tasks 4 and 5 would be implemented within Python-OTP. This workflow needs to be fully automated[4] so that results from a GTAModel/Emme run are automatically translated into the GTFS and OSM files and imported into OTP. Further, it should be possible to request as a GTAModel run option "standard" accessibility calculations to be automatically generated as part of the post-run processing. In addition, the Python-OTP accessibility calculator should be able to run in stand-alone mode to provide dynamic, real-time accessibility calculations for an analyst for a given scenario that had been previously processed by GTAModel/Emme.

[4] "Manual" editing of road and transit networks in Emme aside.

This solution requires the development of procedures (in some combination of C# and Python) to:

- Export a GTFS transit network file from Emme that is readable by OTP. We experimented a bit with this but were not able to get the procedure fully operational within the time constraints of the project across all multi-stream tools. We do not see any fundamental obstacle, however, to developing such a working procedure.
- Similarly, export an OSM-formatted file from Emme of the congested road network that is readable by OTP for use in automobile-based accessibility analysis in that software, if required.
- Generate spatial coordinates, stored as CSVs, to be called by Python-OTP script for use as origins and/or destinations in accessibility calculations.
- Compute the desired accessibilities. These might include a variety of gravity-based measures, isochrones,[5] etc.
- Provide a user-interface for the analyst to:
    - Dynamically request custom accessibility calculations.
    - Display and view maps, graphs, etc. of the calculated accessibilities. A "dashboard" design would be the desired solution for this capability. Note that ArcGIS Pro, Conveyal Analysis and various open-source GIS and visual display tools all might provide methods for the display of accessibilities, both spatially and otherwise.

Another important task in developing the proposed accessibility calculator is to resolve the inconsistency in road network resolutions between Emme and OSM/OTP. Emme maintains an abstracted, simplified network that excludes most local/minor streets. OSM is an "all-streets" network, which is much better suited for disaggregated accessibility calculations. If the proposed interfacing of Emme with OTP is to occur, then a process for populating a more disaggregated OSM road network with Emme road network assignment results needs to be developed. Experimentation will be required to devise the best algorithm for doing this, but the study team is confident that a robust algorithm can, indeed, be developed.

In addition, it is important to note that OSM is a crowd-sourced, open-source product which anyone in the world can edit and modify at any time. This really is unacceptable for City usage from a quality control (as well as temporal consistency) standpoint. Also, as already noted, evolving a base year network to reflect future year conditions must also be possible.[6] The proposed process for addressing these issues is as follows. For alternative-mode only accessibility scenarios:

1. Take the current OSM network "off-line" and edit to remove unnecessary detail (if any) or alternatively, start with known disaggregate network files.
2. Export the combined network to OSM format for use in OTP (or Conveyal).
3. Export GTFS network from Emme.
4. Import network and GTFS into Python-OTP for accessibility analysis.

[5] In the case of isochrones, as discussed further in Section 5.3, the City may want to adopt Conveyal Analysis as a ready-made, OTP-based solution for this application.

[6] Note that similar issues do not exist for the transit network. Base GTFS are "truth" and future year files will be generated by Emme and simply need to be imported into OTP.

To calculate spatially-disaggregate accessibility for automobile-based modes, further work would be required:

1. Edit OSM network or start with known disaggregate network files.
2. Reconcile Emme and OSM links and node coding. This is a non-trivial issue that needs further study. Most likely, the OSM network node labelling will need to be converted to Emme (TMG Network Coding Standard) labels.
3. Develop a Python-based procedure for updating the network given an exported network (with congested link speeds) from Emme. Again, a non-trivial issue that needs further study. The procedure needs to be able to handle network additions and (possibly) deletions.
4. Export the combined network to OSM format for use in OTP-based automobile-based accessibility analysis.

Finally, the discussion throughout this report has focussed on auto- and transit-based accessibility calculations. It may well be the case that the City will also want to compute walk and bicycle accessibilities as well. While not explicitly investigated in this study the OTP solution definitely supports walk accessibility calculations and we are confident that bicycle accessibilities could be computed as well. The later might require the coding of bicycle network attributes within the OSM network (which is supported by OSM) that are not currently contained within the Emme network.

### 5.3 Further Assessment of Conveyal Analysis

Conveyal Analysis presents an interesting new solution for transit-based accessibility analysis. Based on our testing, the software platform enables straightforward "on-the-fly" editing of transit networks and very rapid calculation of local and regional accessibility scenarios. That said, because Conveyal Analysis is essentially built on top of an enhanced version of OTP, it is limited by many of the same issues, such as the properties of the underlying OSM street network.

Although Conveyal Analysis was studied by the project team as a "single-stream" tool, it could be further adopted in a "multi-stream" way. This could include using the congested transit travel times in a GTFS export from Emme and a consistent underlying street network in OSM format detailed above in Section 5.2. Within this framework, Conveyal Analysis would become a user-friendly tool for very rapid accessibility computation, particularly when compared against the computational requirements of the other tools reviewed. Nevertheless, it is unclear in this scenario how the use of an Emme-derived GTFS network would limit Conveyal Analysis' built-in GTFS editing tools. Furthermore, the same weaknesses of Conveyal Analysis as a platform relative to the other tools still apply, including a lack of gravity-based accessibility measures and a raster-based data model.

Given the commercial nature of Conveyal Analysis, and their openness to customizable workflows, **we recommend that the City determine through further discussion with Conveyal whether the costs associated with using Conveyal Analysis are worth the benefits of having it in their toolbox.** These costs will include:

- Software license purchase.
- On-going annual maintenance charges.

- Staff time for initial training and actual use of the tool for planning analysis purposes.
- Cost of any potential customizations beyond the standard product

## References

Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T. et al. (2016). Route planning in transportation networks. In *Algorithm Engineering* (pp. 19-80). Springer, Cham.

Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, *65*, 126-139.

Geurs, K. T., & Van Wee, B. (2004). Accessibility evaluation of land-use and transport strategies: review and research directions. *Journal of Transport Geography, 12*(2), 127-140.

Higgins, C. D. (2019). Accessibility Toolbox for R and ArcGIS. *Transport Findings*.

Miller, E.J. (2019). *Measuring Accessibility: Methods & Issues*, International Transport Forum Discussion Paper, Paris: OECD.

Páez, A., Scott, D. M., & Morency, C. (2012). Measuring accessibility: positive and normative implementations of various accessibility indicators. *Journal of Transport Geography*, 141-153.

Wessel, N., Allen, J., & Farber, S. (2017). Constructing a routable retrospective transit timetable from a real-time vehicle location feed and GTFS. *Journal of Transport Geography*, 62, 92-97.