Comparison between MATSim & EMME: Developing a Dynamic, Activity-based Microsimulation Transit Assignment Model for Toronto

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science

Department of Civil Engineering
University of Toronto

Peter Kucirek, B.A.Sc.
University of Toronto
© Copyright by Peter Kucirek, 2012
Comparison between MATSim & EMME: Developing a Dynamic, Activity-based Microsimulation Transit Assignment Model for Toronto

Peter Kucirek

Master of Applied Science

Department of Civil Engineering, University of Toronto

2012

Abstract

Public transit is becoming an increasing important field of study to combat global issues such as traffic congestion and climate change. Accurate simulation of public transit is therefore likewise vital, as it is an important tool for understanding potential impacts of public transit policies. The research presented in this thesis describes the implementation of a multimodal, dynamic, agent-based supply-side simulation model of public transit implemented in the open-source platform MATSim for the city of Toronto. Transit schedule data was converted from Google Transit Feed Specification (GTFS) and map-matched to a region-wide road network to obtain a congestion-based multimodal assignment for transit. Volume-based results from the assignment showed under-prediction of subway volumes and slight over-prediction of bus volumes, but were generally comparable with static EMME/3 assignment for the same data. Travel time analysis indicated that further calibration of network specification is needed.
Dedication

To Vanessa: Nothing here could I have done without you by my side. I love you.
Acknowledgements

I would like to express my gratitude to all of the people who helped along the way. Without their support, the work that went into this thesis would not be possible.

First of all, let me thank Dr. Eric Miller for his supervision of this thesis and his financial support. He has always believed in my abilities since my undergraduate degree, and I am indebted to him for the opportunities he has provided me in this Master’s thesis.

Thank you to Prof. Amer Shalaby for reviewing my work.

Thank you to the MATSim team in Berlin, and in Zurich: Andreas Nueman, Marcel Rieser, Sergio Ordonez, and Prof. Kay Axhausen. They have been extremely welcoming and helpful in understanding the world of MATSim.

A special thanks to Marcel Rieser and James Vaughan for extended debugging support. Coding is a messy affair, and their willingness to dive into the mud is much appreciated.

Thanks to the Data Management Group (DMG) at the University for the use of their Transportation Tomorrow Survey (TTS) data. The TTS is an indispensable tool for transportation research!

Thank you to Prof. Habib, whose keen interest in using MATSim for his research has challenged me to improve my own understanding.

Thank to all of my colleagues over the years: Keith, Mohammed, James, Michael, Jared, Gurbani, Franco, Bryce, and all of those students working in the ITS lab.

Finally, thank you to my family; my fiancée Vanessa, my parents, and my siblings. I am grateful for their love and support.
Table of Contents

List of Tables .......................................................................................................................... vii
List of Figures ............................................................................................................................ viii
Glossary of Terms ..................................................................................................................... ix
1 Background ........................................................................................................................... 1
   1.1 Motivation ......................................................................................................................... 2
   1.2 Paper Outline ................................................................................................................... 2
2 Study Area ............................................................................................................................. 4
3 Literature Review .................................................................................................................. 5
   3.1 Overview of Transit Assignment ...................................................................................... 5
   3.2 Static, Frequency-Based Approach .................................................................................. 6
   3.3 Dynamic Schedule-Based Approach ............................................................................... 7
4 EMME Background .............................................................................................................. 10
   4.1 Overview of Transit Trip Assignment in EMME .............................................................. 10
   4.2 GTAModel Standard Transit Assignment ....................................................................... 11
5 MATSim Background ........................................................................................................... 13
   5.1 Overview ......................................................................................................................... 13
   5.2 Using MATSim for Trip Assignment .............................................................................. 14
6 Public Transit in MATSim .................................................................................................... 17
   6.1 File Structures ................................................................................................................. 17
   6.2 Transit Routing .............................................................................................................. 19
   6.3 Transit Assignment / Simulation .................................................................................... 23
   6.4 Commentary .................................................................................................................... 23
7 Building the Network Model .............................................................................................. 25
   7.1 Data Sources .................................................................................................................. 25
   7.2 Methodology: Base Network Conversion ...................................................................... 27
   7.3 Transit Schedule Conversion ......................................................................................... 30
8 Building the MATSim Population ....................................................................................... 35
   8.1 Data Sources .................................................................................................................. 35
   8.2 Methodology .................................................................................................................. 35
   8.3 Results of Demand Data Conversion ............................................................................ 37
9 Transit Assignment Parameters & Scenarios ................................................................... 39
   9.1 EMME/3 Transit Assignment Parameters ..................................................................... 39
   9.2 MATSim Transit Assignment Parameters ..................................................................... 39
   9.3 Scenarios ....................................................................................................................... 41
10 Transit Assignment Results ............................................................................................... 43
10.1 Methodology.................................................................................................................43
10.2 Data Sources ..................................................................................................................43
10.3 Computational Performance ..........................................................................................44
10.4 Departures & Arrivals by Time of Day..........................................................................46
10.5 Route Boardings ..............................................................................................................46
10.6 Travel Time Components ...............................................................................................49
10.7 Subway Boardings and Alightings ..................................................................................51
10.8 Unfinished / Stuck Riders, and All-Walk Trips ...............................................................53
10.9 Analysis of Non-Assigned and Infeasible Transit Trips ..................................................57
11 Discussion ..........................................................................................................................58
11.1 Data Restrictions & Validation ......................................................................................58
11.2 Accuracy of the Transit Router .......................................................................................58
11.3 Accuracy of the Microsimulator / Mobsim ......................................................................59
12 Conclusions & Future Work .............................................................................................61
12.1 Conclusions ....................................................................................................................61
12.2 Future Work ....................................................................................................................61
13 Works Cited..........................................................................................................................63
# List of Tables

Table 8-1: List and description of Google Transit Feed Specification (GTFS) files ........................................... 26  
Table 8-2: Mapping of base link attributes .................................................................................................. 27  
Table 8-3: Performance attributes of Excl. ROW vehicles ........................................................................... 28  
Table 8-4: MATSim vehicle types ................................................................................................................... 32  
Table 10-1: EMME transit assignment parameters .......................................................................................... 39  
Table 10-2: Abridged list of MATSim transit parameters ................................................................................. 40  
Table 10-3: Scenario cross-combination .......................................................................................................... 41  
Table 11-1: Transit rider event sequence ....................................................................................................... 45  
Table 11-2: Router network size comparison .................................................................................................. 44
**List of Figures**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Map of rapid transport services in the GTHA (Metrolinx, 2008)</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Map of TTC services (TTC, 2006)</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Standard flowchart of a MATSim simulation</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>The standard MATSim flowchart re-cast as a trip-assignment model</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>Evolution of agents’ scores in MATSim (Gao, 2009)</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>MATSim supply-side file structures</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>Illustration of MATSim’s transit router network generation procedure (Rieser, 2010)</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>Illustration of advantages of disaggregate assignment</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>Coding of dedicated streetcar ROW in EMME</td>
<td>29</td>
</tr>
<tr>
<td>10</td>
<td>Flowchart of transit schedule creation procedure</td>
<td>31</td>
</tr>
<tr>
<td>11</td>
<td>Illustration of the shared-stop problem</td>
<td>33</td>
</tr>
<tr>
<td>12</td>
<td>Mapping of zone centroids to activity coordinates to links</td>
<td>36</td>
</tr>
<tr>
<td>13</td>
<td>Transit rider population comparison</td>
<td>37</td>
</tr>
<tr>
<td>14</td>
<td>Histogram of transit rider arrivals, departures and en-route (Scenario = CG-RaSched)</td>
<td>46</td>
</tr>
<tr>
<td>15</td>
<td>Total all-day boardings on transit, for different MATSim boarding penalty scenarios</td>
<td>47</td>
</tr>
<tr>
<td>16</td>
<td>Total morning boarding periods by mode (TTC lines only)</td>
<td>48</td>
</tr>
<tr>
<td>17</td>
<td>Average morning trip component times</td>
<td>49</td>
</tr>
<tr>
<td>18</td>
<td>Average midday trip component times</td>
<td>50</td>
</tr>
<tr>
<td>19</td>
<td>Average afternoon trip component times</td>
<td>50</td>
</tr>
<tr>
<td>20</td>
<td>Average total transit travel time, all assignment periods</td>
<td>51</td>
</tr>
<tr>
<td>21</td>
<td>Bloor-Danforth subway line boardings and alightings, OBS vs. simulated MATSim</td>
<td>52</td>
</tr>
<tr>
<td>22</td>
<td>Yonge-University-Spadina subway line boardings and alightings, OBS vs. simulated MATSim</td>
<td>52</td>
</tr>
<tr>
<td>23</td>
<td>Sheppard subway line boardings and alightings, OBS vs. simulated MATSim</td>
<td>52</td>
</tr>
<tr>
<td>24</td>
<td>Number of stuck agents for selected scenarios</td>
<td>54</td>
</tr>
<tr>
<td>25</td>
<td>Morning observed TTC Station boardings</td>
<td>54</td>
</tr>
<tr>
<td>26</td>
<td>Morning MATSim simulated station boardings</td>
<td>55</td>
</tr>
<tr>
<td>27</td>
<td>Morning EMME simulated station boardings</td>
<td>55</td>
</tr>
<tr>
<td>28</td>
<td>Number of all-walk trips for boarding penalty scenarios</td>
<td>56</td>
</tr>
<tr>
<td>29</td>
<td>Number of all-walk trips for other scenarios</td>
<td>57</td>
</tr>
</tbody>
</table>
Glossary of Terms

- **GTHA**: Greater Toronto and Hamilton Area
- **TTC**: Toronto Transit Commission – the largest transit operator in the GTHA.
- **TTS**: Transportation Tomorrow Survey.
- **GUI**: Graphical User Interface
- **WGS84**: World Geographic Survey 84 – a standard cartographic projection based on decimal longitude/latitude.
- **NAD83 UTM17N**: North American Datum 83, Universal Transverse Mercator Zone 17N – the standard cartographic projection used in the GTHA, based on coordinates in meters.
- **Transit Route**
  - In EMME: An ordered sequence of nodes, with associated travel time characteristics
  - In MATSim: An ordered list of Transit Stops, and a specified network route
- **Transit Line**
  - In EMME: No specific meaning, can be used interchangeably with transit route
  - In MATSim: A collection of Transit Routes representing a named service; usually (but not necessarily) sharing similar link- and stop-sequences.
- **Transit Route Stop / Transit Stop**: Represents an access/egress point for transit agents
- **NCS11**: Network Coding Standard 2011
- **ROW**: Right-of-way – e.g. a dedicated lane for streetcars
- **IVTT**: In-Vehicle Travel Time
- **FIFO**: First-in-first-out
1 Background

In October of 2011, the United Nations estimated that the Earth’s population had reached seven billion (United Nations, 2011); at least half of which lived in cities. As we enter this age of urban living, urban mass transit is playing an increasingly important role in facilitating the movement of people. Since transit has a higher efficiency per lane-meter of road space – not to mention high-capacity underground transit – as cities become denser and more populous, transit becomes the only way to increase travel capacity.

It is important, therefore, for public officials to be able to make informed decisions about urban public transit policy. To do so, they must be able to make predictions about the impacts of a potential policy. Whether it be the construction of a new subway tunnel, or whether it be adopting a new fare system to encourage transit ridership, assessing the impacts of these policies requires study and research.

Computer simulation of travel behaviour is a major tool available to policy makers in studying policy. However, despite the increased attention paid to public transit and computer simulation, many widely-used simulation platforms, EMME/3 in particular, still rely on static, headway-based models which do not accurately reflect the complexity of modern transit networks. Such models do not have sufficient fidelity to be sensitive to many policy measures, such as fare pricing, and in any case were designed primarily for an era of lower computation power.

With increasing computational power, dynamic, multi-agent platforms are becoming feasible for use in transit simulation.

One such platform, the open-source MATSim, has recently been extended into the realm of public transit simulation. MATSim is an activity-based model designed for simulating large urban populations at the planning level, and has already been implemented in Toronto (Gao, 2009; Hao, 2009). The work presented in this paper extends the existing Toronto MATSim model to a full multi-modal including simulation of public transit. The model is then compared against the existing static EMME morning peak period transit model, in order to validate its results.
1.1 Motivation

The primary motivation for the research presented in this paper is a desire to implement a transit simulation model more detailed than the current state of practice, with a focus on the city of Toronto. Such a model would allow for not only more detailed analyses of infrastructure plans, but also more nuanced analyses of transit policy.

One specific area of interest is the modelling of transit fare policy (e.g., prices and fare schemes) on the route choice of passengers. The operational model used for large-scale planning in the region is limiting further study in this area, as it requires convoluted abstractions in order to mimic what should be explicit behaviour.

Finally, there is a strong shift in the field of transportation planning from trip-based models to activity-based models, as these offer more explanatory power and sensitivity to subtle policy. However, as the demand-side of the equation changes, so must the supply-side; indeed, as noted by Algers et al. (2005) without accurate, dynamic supply-side models, many of these advantages disappear.

1.2 Paper Outline

This paper begins with a review of literature relevant to the topic at hand, namely supply-side simulation of public transit, in Section 3. This review is focused primarily on explaining the algorithms and processes behind existing state-of-the-craft transit assignment platforms, including some up-and-coming developments in the field of dynamic transit simulation.

Section 4 presents some additional information about EMME/3, the standard platform for transit assignment in the study area. GTAModel, which is a specific implementation of a classic four-stage long-range transportation model, is presented as an example of the current state-of-the-craft in the study region.

Section 5 introduces the basic concepts and philosophies behind MATSim, the simulation platform which is the focus of this paper. This section also briefly dissects the MATSim assignment procedure in order to re-cast it as a trip assignment model.

Section 6 expands the details of MATSim’s algorithms and procedures for simulating public transit. File structures are described, as well as a detailed review of MATSim’s transit router algorithm.
Kucirek, 2012

Section 7 describes the process in which the supply-side transit network model was built for the TTC. It includes an overview of Google’s GTFS file format, which was used to create a transit schedule for MATSim.

Section 8 describes an improved procedure for building a MATSim population from a database of trip records.

Section 9 reviews assignment parameters relevant to transit assignment in both EMME and MATSim. This section also introduces the assignment scenarios used for validation.

Section 10 presents the results of the MATSim transit assignment runs, compared against observed data and compared against EMME/3.

Finally, Section Error! Reference source not found. presents the conclusions based on the results of simulation, and Section 12 discusses potential future work.
2 Study Area

The Greater Toronto-Hamilton Area, or GTHA, is located on the northwestern shores of Lake Ontario. It includes 6 different upper-tier regional municipalities and 25 lower-tier local municipalities for a total population of over 6.5 million people (Statistics Canada, 2011). It is the most populous metropolitan region in Canada, and the economic heart of Ontario.

Public transit in the region is provided by eight separate agencies, each with a defined geographic jurisdiction. All of them, save two, exclusively use buses operating in mixed traffic. The two operators that operate higher-order transit are: GO Transit, which operates commuter rail services focused radially on downtown Toronto as well as inter-municipal coach buses; and the Toronto Transit Commission (TTC), which operates underground subways and surface-level streetcar (tram) lines operating mostly in mixed-mode traffic. The TTC also provides considerable bus services within the city limits. Figure 2-1 shows a map of the region with higher-order transportation corridors, while Figure 2-2 shows a map of TTC services.

The GTHA was recently ranked as one of the worst cities in North America for congestion, with an average travel time of 82 minutes in 2008 (Toronto Board of Trade, 2010). With limited available
land to build road infrastructure on, effective investment in public transit and related policy by many as the best way to improve travel time performance.

Figure 2-2: Map of TTC services (TTC, 2006)

3 Literature Review

This section offers a brief review of literature relevant to supply-side simulation of public transit.

3.1 Overview of Transit Assignment

This review begins by first defining the transit assignment problem in general. There are many applications of transit simulation models, including driver staffing, logistics, planning for inter-urban services (e.g. passenger trains), and intra-urban services (e.g. metro systems). It is the last application that this thesis will focus on.

The transit assignment problem is described by (Florian & Spiess, 1989) as the determining of route choices for transit riders in order to assign flows to a transit network. In other words: finding a path for a passenger, which is then aggregated over all passengers to predict route volumes in a transit network. The main motivation behind casting the transit assignment problem this way is in the evaluation of transit infrastructure investments: to be able to predict potential ridership as well as
time savings. However, there are additional dimensions to transit assignment than simply route loads and travel times. Policy measures such as transit fares, real-time information provision, departure time choice (Wahba, 2008), and vehicle reliability all affect the performance of a specific transit policy or investment. Even perception of different modes can affect route choice: during analysis for this paper, the author encountered an actual recorded trip in which the passenger made three subway transfers in a ‘U’ simply to avoid using a much more direct bus!

Perception factors aside, which mainly affect higher-level decisions about whether to drive or not, simulating the basic route choice phenomenon of transit is complex. Unlike driving, a transit trip consists of several distinct trip components (Florian & Spiess, 1989), each which can have a nonnegative associated cost:

- Access to a transit stop from an origin
- Waiting for transit vehicles
- Boarding transit vehicles
- Paying fares
- Riding vehicles
- Alighting from vehicles
- Egress from a transit stop to a destination

By comparison, auto assignment is very simple as it contains at most two components: drive in-vehicle, and pay road tolls. Furthermore, transit riders are constrained by the transit schedule; they are not free to depart whenever they choose.

3.2 Static, Frequency-Based Approach

In practice, the first major operational solution to the general transit assignment problem was proposed by (Florian & Spiess, 1989). Their solution to the transit assignment problem is referred to as strategy-based assignment; where a strategy is defined as a set of rules a traveler uses to reach their destination. In the simple case, where a passenger has only one feasible path to his or her destination, this reduces to a single path based on least-cost. However, in a complex transit network, where multiple attractive lines are available to a destination, a strategy might be more complex and include rules for boarding certain lines depending on which arrives first.
Given a static transit network, however, there is no explicit knowledge of waiting times for vehicles; there is no ‘first’ vehicle arriving at the stop as all lines serve the stop simultaneously. As such, waiting times must be estimated based on expected or average times, for assumed distributions of passenger arrivals. (Florian & Spiess, 1989) show that the combined expected waiting time for a strategy at a link ‘a’ is given by:

$$W(\bar{A}) = \frac{\alpha}{\sum_{a \in \bar{A}} f_a}, \alpha > 0$$

where $f_a$ is the nonnegative frequency of each transit route serving the link, and $\alpha$ is a parameter. The case $\alpha=0.5$ gives an approximation of constant inter-arrival time (headway) and is considered the standard for public transit assignment using this method. In other words, the total expected waiting time is based on the weighted sum of line frequencies. Subsequent flows are assigned to links (routes) based on their waiting time relative to this weighted sum:

$$P(\bar{A}) = \frac{f_a}{\sum_{a \in \bar{A}} f_a}$$

This strategy-based approach is widely-used in practise, and is implemented by EMME/3.

### 3.3 Dynamic Schedule-Based Approach

This approach of using expected waiting times for vehicles has been criticized for its lack of complexity, its inability to deal with time-dependant analyses (Nuzzolo & Crisalli, 2009), its poor handling of low-frequency inter-urban services (Hofsäß, Friedrich, & Wekeck, 2001), and its inability to handle capacity constraints (Wahba, 2008). As such, there is considerable demand to develop the ‘next wave’ of transit assignment solutions, based on transit timetables or schedules. Indeed, many of the constraints of this method, namely low computational power and lack of sufficiently detailed data sources (Wang, 2009), have been overcome with the advent of internet search-engine-based trip planners.

However, before jumping ahead, it is useful to understand the current state-of-the-craft for schedule-based simulation. Several existing platforms are reviewed below:

**MADITUC**

*MADITUC* (Model for the Analysis of Disaggregate Itineraries on a Transit Network) is an early example of a dynamic approach to transit assignment. Its approach to solving the transit assignment
problem is to **not** solve it; rather, it uses observed route choice data to force agents to their chosen routes (Wang, 2009). **MADITUC** is used by the TTC’s planning department for strategic and operational analysis of its transit network.

**VISUM**

**VISUM** is commercial software designed for large-scale simulation of planning-level networks, and is used extensively throughout both North America and Europe (PTV Group). It can be characterized as being a *dynamic, aggregate* model: Demand provided for simulation can be defined for each OD pair either as total flow for specific time periods, or as a function of time-of-day.

**VISUM** supports *timetable-based* dynamic transit assignment, based on the branch-and-bound algorithm developed by (Hofsäß, Friedrich, & Wekeck, 2001). Their solution to the transit assignment problem involves explicit calculation of waiting times at stops for a known stop arrival time and route departure time. In order to improve calculation times, they employ heuristics to apply bounding parameters. Such parameters include minimum wait time, maximum wait time, and maximum number of transfers.

This implementation has been used successfully to support not only large-scale planning – such as in Switzerland (Rieser, 2010) – and also more fine-grained analysis of dynamic transit operations – such as in Vancouver (Fisher & Scherr, 2009).

MATSim provides natively conversion from VISUM transit schedule to MATSim transit schedule.

**MILATRAS**

**MILATRAS** (**MIcrosimulation Learning-based Approach to Transit Assignment**) was developed at the University of Toronto by (Wahba, 2008), and employs a disaggregate approach to simulating transit. It takes the traditional transit assignment problem and re-imagines it through the perspective of psychology and artificial intelligence. Agents have a complex mental model, which evolves over iterations through reinforcement learning techniques. In this respect, **MILATRAS** is very much a *demand-side* model; intended to test advanced ITS technologies (Wahba, 2008). The supply-side simulation within **MILATRAS** is sparsely described expect to note that it is based on existing GIS technologies already in use. It is nominally dynamic, except that input files allow only for static headways.
Although MILATRAS represents a very significant shift in the philosophy of transit assignment and simulation, it remains primarily a research tool. It currently lacks the breadth of tools and functions necessary to becoming an operational planning tool. Additionally, MILATRAS is highly parameterized and requires very extensive calibration using genetic algorithm techniques.

**MATSim**

As mentioned earlier, the MATSim platform is the primary focus of this paper. Its approach and methods are described in detail in Section 5.

**TRANSIMS**

TRANSIMS is advanced dynamic microsimulation software developed at the Los Alamos National Laboratory with support from the American federal government (US FHWA, 2009). Like MATSim, TRANSIMS supports fully disaggregate dynamic traffic and transit assignment. That there are many similarities between TRANSIMS and MATSim is no accident, as MATSim was originally developed as an open-source alternative to TRANSIMS before it became open-source (Rieser, 2010).

TRANSIM’s transit assignment procedure is nearly identical to that used by MATSim (see Section 5.2 and Section 6.2), especially its transit routing procedure. The key difference is in the microsimulator (the ‘mobsim’ in MATSim jargon), which is run after calculating routes for all agents. TRANSIMS’s microsimulator is based on dividing the road network into cells, allowing for complex lane-changing and traffic-following behaviour (Argonne National Laboratory, 2009). It even permits partial lanes for traffic signal turns.

The majority of the remaining differences between the two platforms are largely architectural: TRANSIMS is designed as a series of linked executables written in Python, while MATSim generally runs a single controller written in Java. MATSim uses hierarchical XML-format for data, whereas TRANSIMS data is largely tabular.

The TRANSIMS software is already being deployed in several U.S. cities, including Chicago, Portland, Buffalo, and Atlanta (Argonne National Laboratory, 2009; US FHWA, 2009; R.I.T.A., 2009; AECOM, 2008).
4 EMME Background

The acronym EMME stands for “Equilibre Multimodal, Multimodal Equilibrium.” The software was first developed in the 1970’s at the Centre for Research on Transportation (CRT) at the University of Montreal. Over 40 years, EMME has developed as a comprehensive package for simulating transportation used widely across Canada, and particularly in the region of study. INRO (the company which owns and maintains the EMME program) has recently released EMME/3 Modeller, which includes a user friendly graphical user interface (GUI) for network editing, as well as a suite of user-friendly tools for simulation, analysis, and more.

4.1 Overview of Transit Trip Assignment in EMME

As discussed in Section 3, standard public transit assignment in EMME is based on the optimal strategies approach developed by (Florian & Spiess, 1989). It should be noted that with the recent release of EMME 3.4.1 Modeller, INRO has added several extensions to the standard transit assignment, including some support for a version of schedule-based assignment. This paper will use the standard transit assignment tool in EMME, however, since it is the state-of-the-craft standard and therefore the best validated (INRO, 2011). Newer EMME assignment algorithms are therefore outside the scope of this paper.

Transit Data

EMME transit lines consist of two components: a header, and a route. The header contains information about the line: a unique id, mode, vehicle type, headway, and line speed. The headway parameter defines the level-of-service that a line provides; line speed by default defines the speed at which a line operates. The route represents the path a transit line services and is stored as a sequence of nodes which are turned into transit segments when processed. The route also defines which nodes are valid transit stops for passengers to board at or alight from. Finally, the route can specify a custom transit travel time function on each link, which by default is based on the link length and line speed. This function can be used to “copy over” link times from a traffic assignment to replicate the effects of congestion on both routing and simulation.

Trip Assignment

All trips in EMME begin or end at zone centroids; this includes transit trips. As such, the transit walk (or “auxiliary transit” in EMME jargon) mode uses actual links to derive walk times. During the transit assignment, EMME will calculate the optimal strategy from every zone to every other zone; but it will
only assign volume to those OD pairs with non-zero demand. Thus, the resulting travel time component matrices will always be the same for the same network (and same parameters) even while demand changes.

There are no capacity constraints in the EMME assignment, which makes sense since there is no way of resolving which passengers are unable to board during static simulation.

4.2 GTAModel Standard Transit Assignment

GTAModel is a traditional four-step transportation demand forecasting model developed for the GTHA (Miller E., 2001). On the supply side, GTAModel prescribes a trip assignment procedure and complementary network data structures built on the EMME/2 (and now EMME/3) platform, with a transit network representing morning peak hour service levels only.

Recently, a major overhaul of the transit assignment procedure was developed in order to properly simulate the effect of transit fares on route choice (Miller E., 2007). This so-called “version 3” assignment procedure creates additional layers of base network links – one for each agency – and connects them with virtual connectors calibrated to incur a time-value-of-money penalty equal to the fare paid by a passenger. As of writing, this procedure has yet to be fully embraced by those in the practise, and is only used for academic research. Therefore, only the standard “version 2” assignment procedure will be used in this thesis, as it represents the state-of-the-craft.

Transit Data Structures (Network Coding Standard)

The Network Coding Standard (NCS) specified by GTAModel imposes several additional conditions on top of the standard base and transit network data structures. For the base network, NCS specifies attribute locations for link free-flow speed, and for link per hour capacity. For the transit network, NCS makes several requirements:

- All transit lines are coded one-way (i.e. one line for each direction).
- Boarding penalty values are stored with each line.
- Link speeds for exclusive ROW lines are stored with each transit segment.
- Dwell time of 0.01 minutes for all nodes which represent stops
- Headways are in minutes.
- Speeds are all in km/hr.
During assignment, line boarding penalties are used to account for the perceived cost of certain mode. However, these boarding costs are not accumulated in the final transit travel times.

GTAModel also specifies default parameter values for assignment; these will be introduced in Section 9.1.

Finally, it should be noted that the NCS is currently in the process of being updated for 2011 (“NCS11”); however it is expected that this will not impact the transit network.
5 MATSim Background

MATSim, which is short for “Mult-Agent Transport SIMulation,” (MATSim website) is a Java-based open-source platform developed primarily at VSP TU Berlin and IVT ETH Zurich. It has been used fairly extensively in both cities, both for simulation of public transit and for simulation of private automobiles. Simulations outside of Europe include South Africa (Axhausen & Joubert, 2011), Singapore (Ordonez & Erath, 2011), and, of course, Toronto. The core code is modular in design, which allows for a great deal of flexibility – users can design their own routing algorithms, their own link-cost-calculators, their own schedule planning algorithms, etc. Perhaps the one drawback is that there are few graphical tools currently available for use, which makes using the software daunting for new users. However, this is beginning to change, with the senozon company commercializing its own GUI-based tools for MATSim (particularly its proprietary visualizer VIA) (senozon Website). The core code for MATSim remains free to anyone who wishes to use it.

5.1 Overview

MATSim is often described as a dynamic, agent-based, activity-based model. This contrasts sharply with the nature of EMME, which is conversely static, aggregate, and trip-based. Of course there are a number of trade-offs in using a more complex model, with data-intensity and computing times being the most obvious.

- **Static vs. Dynamic:** Public transit –indeed all simulation – in MATSim is done second-by-second. Time in MATSim is measured as a double floating-point value in seconds past midnight, meaning that for all intent and purposes MATSim is fully disaggregate with respect to time.

- **Trip Based vs. Activity Based:** Unlike traditional four-stage modelling platforms, MATSim assumes that all trips are derived from agents’ activity schedules. Therefore, demand data used by MATSim is not person-trips, but rather person-schedules.

- **Agent-Based (Disaggregate) vs. Aggregate:** MATSim is also fully disaggregated with respect to demand, in that it simulates a population of integer agents. Producing this data can be challenging, particularly when forecasting when a synthetic population must be created.

- **Stochastic vs. Deterministic:** MATSim combines elements of determinism and stochasticity. Shortest-path calculation as well as agent plan execution phases are deterministic, giving the same result for the same agent departing from the same location at the same time.
However, the agent re-planning is highly stochastic: replanning strategy modules (such as re-calculate route, or mutate travel time) are selected for each agent using a Monte Carlo draw against user-defined probabilities.

5.2 Using MATSim for Trip Assignment

Although MATSim provides extensive functionality for demand-side agent activity scheduling, the primary goal of this research was to use MATSim purely as a trip assignment model. The key task of a trip assignment model is to resolve the reciprocal relationship between demand and supply. That is, any choice of path depends on the supply-side behaviour of the network, but simultaneously the behaviour of the network depends on the aggregate path choices of the demand. In this way, any trip assignment simulation is effectively an iterative route-choice model.

![Flowchart of a MATSim simulation](MATSim website)

**Controller Events:**
1. Simulation Starts ("Startup")  
2. Iteration Starts  
3. Before Mobsim  
4. After Mobsim  
5. Scoring  
6. Iteration Ends  
7. Replanning  
8. Simulation Ends ("Shutdown")

**Figure 5-1:** Standard flowchart of a MATSim simulation (MATSim website)

Within MATSim, the route-choice problem is cast as a choice between different plans, where an agent’s plan contains information about its path choices through the network. Figure 5-1 shows the standard flowchart of a MATSim run, while Figure 5-2 shows how this standard framework is re-cast as a trip-assignment model. In the re-cast framework, pre-processing initializes a single least-cost path (plan) for each trip (leg) for each agent. Each agent’s plan is then executed using the mobility simulator ("Mobsim"), and scored using the same utility calculation used during routing. Travel conditions are measured and stored in memory. The replanning step which follows uses Monte-Carlo simulation to select which strategy an agent employs (i.e. what algorithm the replanning module calls; in which there are two choices: explore a new least-cost path (plan), or stochastically select from available plans. After re-planning, the new plans are executed in simulation as a new iteration; the run lasts for as much iteration as specified.
The important thing for readers to remember is that there is a strong separation between route/path choice and simulation/assignment; they are two separate models handled interdependently. It is also important to note that, unlike EMME, MATSim can be configured for reciprocal transit assignment; where feedback from the simulation phase can be used again during planning.

![Flowchart diagram](image)

Figure 5-2: The standard MATSim flowchart re-cast as a trip-assignment model

From this observation, it is important to note that although MATSim mimics the framework of a standard User Equilibrium (UE) simulation model; it does not perform a UE assignment. Although
Kucirek, 2012

(Gao, 2009) has shown that agent plans can reach a stable state (see Figure 5-3), there is no guarantee that this equilibrated state meets UE requirements.

On the other hand, the unpredictable nature of human behaviour, coupled with the constant changing nature of a transportation even during a single day makes such an equilibrium condition tentative at best (Mahmassani, 1997).
6 Public Transit in MATSim

Public transit simulation has been under development in MATSim in various forms since 2009 and, through the work of (Rieser, Adding Transit to an Agent-Based Transportation Simulation, 2010) has matured to being a stable component of the MATSim framework. This section describes the supply-side simulation of public transit in MATSim.

6.1 File Structures

There are three files which describe network supply in MATSim: The base network (colloquially, the “network”), the transit schedule (colloquially, the “schedule”), and the ‘population’ of vehicles (which includes vehicle types and attributes). Figure 6-1 shows a schematic of these files.

Base Network Format

The base network in both EMME and MATSim represents a region’s actual, physical infrastructure. Like many large-scale platforms, MATSim’s base network comprises a set of nodes and connecting links. Each link has a set of attributes, including a unique ID; this is relevant since MATSim permits
multiple links to connect the same nodes.

Transit Schedule

The transit schedule file consists of two main collections: a collection of transit stops and a collection of transit lines.

- **Transit Stops:** Transit stops are locations in the network where transit vehicles stop to pick-up/drop-off passengers; and typically correspond to physical infrastructure. In MATSim, agents can only board/alight transit vehicles at transit stops. Each stop, in addition to a coordinate location, is referenced to a link in the base network; this is necessary since MATSim only process actions at the end of links. Finally, stops can be configured to block their corresponding link. This behavior is correlated to reality: some stops are roadside and physically block traffic when a vehicle is parked there.

- **Transit Lines / Transit Routes:** In MATSim, a Transit Line is a named collection of Transit Routes. This is a useful convention, since in reality often several branches (usually one in each direction) operated under a common name. Each transit route consists of three components (apart from the route ID and mode): a stop profile, a link profile, and a list of departures.
  - **Stop Profile:** A list of transit stops, in the order the route services them. Each stop in the profile must reference a valid Transit Stop, as well as arrival and departure offsets. A stop’s arrival offset represents how much time it takes to arrive at the stop, relative to the first stop. The first stop cannot have an arrival offset. Departure offset is similar, except that it can be handled by MATSim in multiple ways (as set by the ‘await departure’ flag). The default behaviour (\texttt{awaitDeparture = false}) is that a transit vehicle departs either a) when no more passengers are looking to board / alight, or b) when the vehicle is full. If the flag is set to ‘true,’ then the vehicle leaves at its scheduled departure time regardless. All stops except for the final stop must have a departure offset.
  - **Link Profile:** A list of links that the transit route travels on, in the order that it travels on them. Each entry in the list must reference a valid link id, and each list must contain the links its stops are referenced to.
  - **Departures:** Each departure is a single ‘trip;’ a single vehicle serving the route starting at a given time. Together, the departures represent the level of service a route provides; analogous to headway. If a route has three departures per hour, then it has a headway
of 20 minutes, and vice-versa. In MATSim, each departure requires an id, a departure time, and a reference to a valid vehicle id (from the Vehicles file).

Vehicles

MATSim’s vehicles file contains both the definitions of vehicle types as well as a ‘population’ of actual vehicles. Both transit and private automobile vehicle types can be defined and used here; although for the purposes of this thesis only transit vehicles are defined. Each vehicle type has a unique id, length, width, and a passenger car equivalency factor (PCE). At the time of writing, it is unclear what effects the vehicle length and width have on simulation, however the PCE factor is critical as it allows transit vehicle to take up more or less space on links compared to regular traffic.

In addition, transit vehicles can be assigned various attributes affecting their behavior in simulation. Transit vehicle capacity is set in two parts, seated capacity and standing capacity, where total capacity is the sum of seated and standing, minus one person who is the driver. Attributes determining loading/unloading behavior can also be assigned: access and egress times per person, as well as a door operation mode. In serial operation mode, such as on a bus, passengers enter / exit the vehicle in sequence. In parallel operation mode, multiple passengers can enter and exit simultaneously.

Finally, each vehicle in the vehicle population itself requires a unique id and a corresponding vehicle type.

6.2 Transit Routing

In the jargon of MATSim, routing refers to the path choice of an agent’s plan; it is conceptually the same as the transit assignment problem solved in EMME.

Unlike other software platforms, MATSim does not implement a specific transit assignment algorithm per se; instead it builds on existing modular dynamic shortest-path algorithms already implemented in MATSim. Over the years, a number of different shortest-path algorithms have been coded into MATSim, including A* Euclidean, A* Landmarks, and Dijkstra. Although A* has been shown to run considerably faster within MATSim (Gao, 2009), complexities in the transit network combined with the pre-processing requirement of A* has precluded using this algorithm for routing transit (Rieser, Adding Transit to an Agent-Based Transportation Simulation, 2010). As such, only Dijkstra’s shortest-path algorithm has been configured for transit routing in MATSim.
In order to use the dynamic Dijkstra module in MATSim, the Transit Schedule is converted into a TransitRouterNetwork, as described in (Rieser, Adding Transit to an Agent-Based Transportation Simulation, 2010). In brief, each route is converted into a sequence of special links, where the transit stops are nodes connected by a single link. MATSim simplifies the links such that only a single link connects two stops, regardless of how many actual links the route travels on in the base network. Transit stops serviced by multiple routes have multiple nodes referencing them: one for each transit route servicing that stop. MATSim considers all stops within a parameterized distance to be feasible transfer points, and connects them with links whose length correspond to the Euclidean distance between these stops. Figure 6-2 illustrates this procedure.

It is important to note that, in MATSim, agents do not walk on network links. Walking in MATSim is considered a ‘teleported’ mode; the distance an agent walks from their origin coordinate and their transit stop of access is given by the Euclidean distance between these two points. MATSim uses a parameterized walk speed which accounts for the effects of nonlinear walk distances.
Calculating a shortest path on this network requires one additional step: creating access and egress links. These special transfer links are unique to each origin and destination coordinate pair, and connect these two points to their set of feasible transit stops. MATSim considers all transit stops within a specified radius (set by default as 1.0 km) to be feasible; if none are found within 1 km, MATSim extends its search radius by a specified parameter.

**Generalized Cost (Disutility) Calculation**

The generalized cost calculator is the ‘meat’ of the shortest-path algorithm, as it is where perception factors and calibration parameters get applied. The modular nature of MATSim allows users to very easily write their own calculator (known as a TransitRouterNetworkTravelTimeAndDisutility), although MATSim does provide a default calculator. The default equation used to calculate generalized cost for transit router links is given by:

\[
\text{Link Cost} = \begin{cases} 
-\text{walkTime} \cdot \beta_{\text{walk}} - \text{additionalTransferTime} \cdot \beta_{\text{wait}} - \beta_{\text{transfer}} & \text{if transferring} \\
-\text{linkTravelTime} \cdot \beta_{\text{ivtt}} - \text{linkDistance} \cdot \beta_{\text{pt.dist}} & \text{if travelling}
\end{cases}
\]

where:

- \(\text{walkTime}\), \(\text{additionalTransferTime}\), \(\text{linkTravelTime}\) are in seconds,
- \(\text{transferLinkTime} = \text{walkTime} + \text{waitTime}\), calculated as the joint time an agent spends on the transfer link,
- \(\text{linkDistance}\) is in meters,
- \(\beta_{\text{walk}}, \beta_{\text{wait}}, \beta_{\text{transfer}}, \beta_{\text{ivtt}}\) are in utils/second,
- \(\beta_{\text{pt.dist}}\) is in utils/meter, and
- \(\beta_{\text{walk}}, \beta_{\text{wait}}, \beta_{\text{transfer}}, \beta_{\text{ivtt}}, \beta_{\text{pt.dist}} \leq 0\).

There are two issues with the default cost calculator. First, its equation for disutility does not include a term for waiting time at a stop, instead using a parameter meant to represent the physical delay in exiting / entering transit vehicles (i.e. to prevent transfers from occurring instantaneously). It is not clear why no wait time is calculated explicitly, as wait times incur a heavy ‘psychic’ cost. Some studies suggest that every minute a passenger spends waiting for a transit vehicle feels like two.
The second major issue with the default calculator is that it is based entirely on the static transit schedule. In other words, it has no memory of the simulation results and delays incurred. Again, it is not clear as to why this is the case. This behavior is particularly critical for the Toronto case, where bus and streetcar routes have to share road space with car drivers, and often experience heavy congestion causing delays.

These two issues were sufficient to warrant writing a new calculator. Again, the modularity of MATSim made coding this calculator relatively straightforward: a special TransitDataCache object does most of the heavy work in storing the as-simulated travel times for transit vehicles by listening for events created by vehicles arriving and departing at stops. The cache is initialized with data from the transit schedule, so, at iteration 0, it behaves exactly like the old calculator.

Within the new UpgradedTransitNetworkTravelTimeAndDisutility calculator, the equation for generalized cost was updated to be:

$$\text{Link Cost} = \begin{cases} 
-\text{walkTime} \cdot \beta_{\text{walk}} - \text{waitTime} \cdot \beta_{\text{wait}} - \beta_{\text{transfer}} - \text{transferFare} & \text{if transferring} \\
-\text{linkTravelTime} \cdot \beta_{\text{vtt}} - \text{linkDistance} \cdot \beta_{\text{pt.dist}} - \text{inlineFare} & \text{if travelling}
\end{cases}$$

where:

- \(\text{walkTime}, \text{waitTime}, \text{linkTravelTime}\) are in seconds,
- \(\text{waitTime} = \text{waiting} + \text{additionalTransferTime}\)
- \(\text{linkDistance}\) is in meters,
- \(\beta_{\text{walk}}, \beta_{\text{wait}}, \beta_{\text{transfer}}, \beta_{\text{vtt}}\) are in utils/second,
- \(\beta_{\text{pt.dist}}\) is in utils/meter, and
- \(\beta_{\text{walk}}, \beta_{\text{wait}}, \beta_{\text{transfer}}, \beta_{\text{vtt}}, \beta_{\text{pt.dist}} \leq 0\).

This cost is similar to the cost used for Scoring, except that MATSim allows for different parameters to be used for routing/scoring.
6.3 Transit Assignment / Simulation

In general, the control structure used by MATSim to keep track of transit rider agents is outside of the scope of this paper – more details can be found in (Rieser, 2010). However, there are a few aspects of the actual dynamic simulation which impact the behaviour of the overall model.

Transit Line Speeds as Simulated

MATSim uses a queue-based approach to simulating vehicle congestion on links. Vehicles are served on a FIFO basis, after spending enough time on the link to ensure that they traverse it. Transit is no exception: the general definition of ‘vehicle’ is extended to include transit vehicles. The result of this is that, while buses and other ROW-sharing modes (e.g. streetcars) can experience congestion, there is no equivalent for ‘average line speed’ in MATSim. A bus, for example, traveling on an empty link is assumed by MATSim to travel at freeflow speed. Transit vehicles on their own ROW travel at the link’s freeflow speed except in the unlikely event that they encounter congestion.

Accounting for dwell times at stops is handled explicitly, with vehicles having access / egress time parameters. These parameters are per person, so a vehicle waiting at a stop with ten passengers waiting will wait at the stop longer than if there was only one passenger waiting.

Vehicle Capacity Constraints

The current implementation of MATSim has a strict cut-off procedure for vehicle capacity. Once a vehicle reaches full capacity (which includes the driver as one agent), it no longer allows passengers to board. Passengers left at their stop will simply wait until the next vehicle from their selected route arrives.

6.4 Commentary

The implementation of public transit in MATSim is considerably complex for a planning-level simulation platform, requiring detailed demand and supply data in order to work properly. However, this detail pays off as MATSim is able to dispense with the path-splitting approximation used by EMME to account for temporal and spatial aggregation.

For example, in a large traffic zone with two major routes running along its borders – a common occurrence since the borders of many traffic zones are defined by major roads – the most attractive line for each rider is dependent on the rider’s precise location of access (see Figure 6-3). If zonal aggregation is applied, and all riders within a zone must access the system from a single location,
this heterogeneity is lost. On the other hand, MATSim simulates this heterogeneity explicitly; with each agent only taking a single route. It is for this reason that the model population was expanded to represent a close-to-100% sample.

Furthermore, where the optimal strategies algorithm relies on relative headways to simulate the effects of waiting times at stops, MATSim can simulate waiting times explicitly; since the arrival time at a stop is known for both passengers and vehicles.
7 Building the Network Model

Building a working network model was the most laborious task of this research, since a full 24-hour transit schedule had never been previously built for the GTHA’s multimodal network. The static EMME model commonly used for analysis in the region is of the morning peak hour only; which is a problem for transit since level-of-service varies dynamically throughout the day. To further complicate matters, a fully multimodal network was required, allowing surface transit (e.g. buses and streetcars) to interact with vehicular traffic. This is an important aspect of transit simulation, since congestion is a major source of unreliability in transit service.

As a result, additional data on dynamic service schedules were required, and these data needed to be compatible with a low-resolution planning-level network. This turned out to be a monumental task, and only the TTC’s data could be incorporated into MATSim. The author is indebted to Yi (Ivy) Liang for her assistance in converting these data.

7.1 Data Sources

For this research, the base network used in MATSim was created from the 2006 planning-level GTHA network provided by the Travel Modelling Group (TMG). TMG is a new inter-jurisdictional initiative whose mandate is to facilitate transportation model building across its member municipalities (Miller & Hatzopoulou, 2008).

Transit schedule data was provided by each of the major operators for 2006, however no two of them agreed on a consistent format. As such, it was decided to focus exclusively on coding and simulating TTC routes only. The TTC serves 80% of all daily rides made by transit in the region, offering a large payoff for the labour invested.

Data for the TTC’s routes came in two parts: the first was a 2006 service summary listing service headways for different time periods; the second was the Google Transit Feed Specification (GTFS) files from March of 2012.

Google Transit Feed Specification (GTFS)

GTFS is a standard in representing transit data with a high level of fidelity which Google uses for publically-available transit-trip-planning using its Google Maps application. This application is dynamic with respect to time, and includes scheduled service for several user-defined time periods.
Kucirek, 2012

(such as weekdays and weekends). As such, it is practically ideal for creating MATSim transit schedules.

At the core of the specification are six files – with an additional seven that are optional. Table 7-1 lists currently-used files, as well as which files were publically available from the TTC. The TTC’s own GTFS files are updated regularly to correspond to current service levels; the files used in this research are from March of 2012.

The data most relevant to transit schedule creation is primarily stored in the stops.txt, routes.txt, trips.txt, stop_times.txt, and frequencies.txt files. The stops file matches very neatly with MATSim’s definitions of stops, except without link references; while the other four together define transit routes. Note that the frequencies.txt file is optional: if it is not specified, then the stop arrivals and departures specified in stop_times.txt are considered absolute and each trip in trips.txt is treated as an individual departure. Otherwise, the frequencies.txt file describes a route’s level of service.

<table>
<thead>
<tr>
<th>File</th>
<th>Required?</th>
<th>Description</th>
<th>Provided by the TTC?</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency.txt</td>
<td>Required</td>
<td>One or more transit agencies that provide the data in this feed.</td>
<td>Yes</td>
</tr>
<tr>
<td>stops.txt</td>
<td>Required</td>
<td>Individual locations where vehicles pick up or drop off passengers.</td>
<td>Yes</td>
</tr>
<tr>
<td>routes.txt</td>
<td>Required</td>
<td>Transit routes. A route is a group of trips that are displayed to riders as a single service.</td>
<td>Yes</td>
</tr>
<tr>
<td>trips.txt</td>
<td>Required</td>
<td>Trips for each route. A trip is a sequence of two or more stops that occurs at specific time.</td>
<td>Yes*</td>
</tr>
<tr>
<td>stop_times.txt</td>
<td>Required</td>
<td>Times that a vehicle arrives at and departs from individual stops for each trip.</td>
<td>Yes*</td>
</tr>
<tr>
<td>calendar.txt</td>
<td>Required</td>
<td>Dates for service IDs using a weekly schedule. Specify when service starts and ends, as well as days of the week where service is available.</td>
<td>Yes</td>
</tr>
<tr>
<td>calendar_dates.txt</td>
<td>Optional</td>
<td>Exceptions for the service IDs defined in the calendar.txt file. If calendar_dates.txt includes ALL dates of service, this file may be specified instead of calendar.txt.</td>
<td>Yes</td>
</tr>
<tr>
<td>fare_attributes.txt</td>
<td>Optional</td>
<td>Fare information for a transit organization's routes.</td>
<td>No</td>
</tr>
<tr>
<td>fare_rules.txt</td>
<td>Optional</td>
<td>Rules for applying fare information for a transit organization's routes.</td>
<td>No</td>
</tr>
<tr>
<td>shapes.txt</td>
<td>Optional</td>
<td>Rules for drawing lines on a map to represent a transit organization's routes.</td>
<td>Yes</td>
</tr>
<tr>
<td>frequencies.txt</td>
<td>Optional</td>
<td>Headway (time between trips) for routes with variable frequency of service.</td>
<td>No*</td>
</tr>
<tr>
<td>transfers.txt</td>
<td>Optional</td>
<td>Rules for making connections at transfer points between routes.</td>
<td>No</td>
</tr>
<tr>
<td>feed_info.txt</td>
<td>Optional</td>
<td>Additional information about the feed itself, including publisher, version, and expiration information.</td>
<td>No</td>
</tr>
</tbody>
</table>

* = File required pre-processing to generate

Table 7-1: List and description of Google Transit Feed Specification (GTFS) files
7.2 Methodology: Base Network Conversion

The desired output of the map-matching procedure was a multimodal network, allowing surface transit (e.g. buses and streetcars) to interact with vehicular traffic. Therefore, the EMME base network provided was used fully; with only a few minor edits to include side streets used by some bus lines.

In order to aid network conversion, a new EMME2MATSimConverter module was written; with the goal of reading in standard GTAModel-compatible EMME export files and converting them to a MATSim network. A simple GUI was built to allow users to perform various actions on the network, including coordinate projection\(^1\), turn-restriction implementation, mode filtering, and others.

Node/Link Attributes

In general, node and link attributes map very closely from EMME to MATSim, as shown in Table 7-2. Link capacity and link freeflow speed are not specified in EMME by default; fortunately they are required as part of the NCS11 for use in GTAModel. Units were converted as necessary, for example MATSim expects capacity in total vehicles per user-specified time period (nominally per hour), whereas NCS11 uses vehicles per hour per lane.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>EMME / GTAModel</th>
<th>MATSim</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>From node</td>
<td>→</td>
<td>From node</td>
</tr>
<tr>
<td>j</td>
<td>To node</td>
<td>To node</td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>(km)</td>
<td>→</td>
<td>Length (m)</td>
</tr>
<tr>
<td>modes</td>
<td>Single-char modes</td>
<td></td>
<td>Text modes</td>
</tr>
<tr>
<td>type</td>
<td>Geographic area</td>
<td>→</td>
<td>n/a</td>
</tr>
<tr>
<td>lanes</td>
<td></td>
<td></td>
<td>lanes</td>
</tr>
<tr>
<td>vdf</td>
<td>Volume-delay function</td>
<td>→</td>
<td>Type</td>
</tr>
<tr>
<td>ul1</td>
<td>Count station</td>
<td></td>
<td>n/a</td>
</tr>
<tr>
<td>ul2</td>
<td>Speed limit (km/hr)</td>
<td>→</td>
<td>Freeflow speed (m/s)</td>
</tr>
<tr>
<td>ul3</td>
<td>Capacity per hr per lane</td>
<td></td>
<td>Capacity per hour</td>
</tr>
</tbody>
</table>

Table 7-2: Mapping of base link attributes

Link Speeds on Transit-Exclusive Links

Freeflow speeds on links used exclusively by transit (e.g. subway links) required special handling, as not only were they not specified by the EMME model, but also because MATSim uses a common

\(^1\) Provided for use in map-matching with GTFS files, which use decimal longitude/latitude (WGS84) coordinates.
function for determining travel time on links based on the freeflow speed and link capacity. Capacity makes little sense with respect to exclusive transit links, since any real-life capacity constraints are reflected in the scheduled service level. As such, link capacities are set to 9999 with subway PCE set to 1.0 – essentially setting link capacity at 9999 subway trains per hour – so that no subway congestion is observed. Therefore, actual speeds on these links are equivalent to the freeflow speeds on these links.

The question is: what is an appropriate link speed? In MATSim, station dwell times can be modeled explicitly and dynamically, therefore link speeds had to only include the effects of acceleration / deceleration – which can be significant for trains. Basic Newtonian physics describes time-and-space relationships for an object moving under constant acceleration and deceleration rates $a$ and $b$, for maximum velocity $V_{\text{max}}$, with station spacing of $L$; as follows:

$$T(L) = \begin{cases} \frac{3.6L}{V_{\text{max}}} + \frac{V_{\text{max}}}{7.2} \left( \frac{1}{a} + \frac{1}{b} \right), & L \geq L^* \\ \sqrt{\frac{2(a + b)L}{ab}}, & L < L^* \end{cases}$$

Where:

- $T(L)$ is the travel time on the link in sec,
- $a$ and $b$ are in m/s$^2$,
- $V_{\text{max}}$ is in km/hr,
- and $L^*$ is the critical length in m, which is defined as the shortest length in which a vehicle can reach $V_{\text{max}}$.

Table 7-3 summarizes the values for $a$, $b$, and $V_{\text{max}}$ used for calculation of speeds on subway and streetcar links.

<table>
<thead>
<tr>
<th></th>
<th>Subway</th>
<th>Streetcar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration rate</td>
<td>0.85 m/s$^2$</td>
<td>1.47 m/s$^2$</td>
</tr>
<tr>
<td>Braking rate</td>
<td>1.3 m/s$^2$</td>
<td>1.6 m/s$^2$</td>
</tr>
<tr>
<td>Maximum velocity</td>
<td>80 km/hr</td>
<td>link max speed</td>
</tr>
</tbody>
</table>

Table 7-3: Performance attributes of Excl. ROW vehicles (Corley, 1996; TTC, 1998)

These speeds do not account for the interference of traffic signals on exclusive streetcar ROWs. Although MATSim provides facilities for simulating the effects of traffic signals, these were determined to be outside the scope of this work and were therefore omitted.
Treatment of Streetcar Dedicated ROW

Dedicated streetcar lanes are coded a virtual ‘separate layer’ of links in the EMME planning-level network (see Figure 7-1). Since the walk-access-egress mode in MATSim does not use links, the real location of links is important in ensuring accurate behavior around dedicated streetcar links. MATSim allows for multiple links to connect any node pair, so dedicated streetcar links were reconnected to nodes used by their corresponding road links. This ensured that streetcars travelling on these links are unimpeded by traffic congestion.

Turn Restrictions

MATSim does not allow node-based turn restrictions like EMME; instead a module of code from (Hao, 2009) was reprised to create the effects of turn restrictions using virtual one-way links. Some minor manual editing was required to ensure that the final network was compatible with multi-modal assignment.

Figure 7-1: Coding of dedicated streetcar ROW in EMME
Centroid Connectors

Centroid connectors were kept in the network to simulate the effects of ‘internal’ links (i.e. links smaller than the resolution of the network) on automobile traffic.

7.3 Transit Schedule Conversion

To create the MATSim transit schedule for the TTC, the GTFS2MATSimConverter module was used. This semi-automated procedure was developed by (Ordonez & Erath, 2011) to aid in the creation of a transit network for Singapore; although it was originally designed to aid in map-matching GTFS files with additional GPS data. During the map-matching process for Toronto, certain idiosyncrasies of the Toronto data necessitated considerable pre- and post-processing of the GTFS data for successful use with the GTFS Converter. The actual conversion process primarily involved creating the link profiles (sequence of links travelled upon) for a given stop profile contained in the GTFS data, and is summarized schematically in Figure 7-2.

Network Resolution

One important issue to note is that the TTC’s GTFS data are at a much higher resolution than the planning-level base network. Very often, a link will see two or three stop-pairs incident along the length of the link. Major inter-modal transfer points consist of several adjacent, but not overlapping stops representing individual bus bays. Ideally, the transit network should specify the planning network’s base resolution, such that no link sees more than two transit stops (one for each direction) incident along its length, with major transfer stations aggregated together. Unfortunately, due to time constraints, this in-congruency will have to be resolved in future work.
Vehicles

Transit vehicle type definitions were copied over from the EMME/3 database for consistency, since MATSim and EMME share common vehicle attributes. All vehicle widths were set to 1.0m, as it does not appear that MATSim uses vehicle dimensions in any way meaningful for trip assignment. Table 7-4 summarizes the vehicle types used in this paper.

The vehicle population used is synthetic, created such that each departure in the transit schedule is linked to a unique vehicle.
Table 7-4: MATSim vehicle types

<table>
<thead>
<tr>
<th>Id</th>
<th>Mode</th>
<th>Descr.</th>
<th>Capacity</th>
<th>Length (m)</th>
<th>Width (m)</th>
<th>Access/Egress Time (sec/person)</th>
<th>Door operation type</th>
<th>Base PCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Train</td>
<td>Train consist</td>
<td>1600</td>
<td>2000</td>
<td>279.7</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Subway</td>
<td>6-car subway consist</td>
<td>480</td>
<td>1200</td>
<td>138</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Subway</td>
<td>4-car subway consist</td>
<td>320</td>
<td>800</td>
<td>92</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Subway</td>
<td>SRT consist</td>
<td>120</td>
<td>320</td>
<td>52</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Streetcar</td>
<td>Tram consist</td>
<td>50</td>
<td>75</td>
<td>15</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Streetcar</td>
<td>Artic. tram consist</td>
<td>60</td>
<td>110</td>
<td>23</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Bus</td>
<td>Artic. bus consist</td>
<td>60</td>
<td>90</td>
<td>16</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>Bus</td>
<td>30' bus consist</td>
<td>40</td>
<td>60</td>
<td>9</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>Bus</td>
<td>40' bus consist</td>
<td>30</td>
<td>40</td>
<td>12</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>Bus</td>
<td>coach bus consist</td>
<td>45</td>
<td>60</td>
<td>13.7</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>Bus</td>
<td>bilevel coach bus consist</td>
<td>80</td>
<td>-</td>
<td>13.2</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Route Exclusions

25 routes from 2012 were deliberately excluded during conversion. Of these, one was excluded because it was not in service in 2006; and 24 were excluded because they were flagged as night-time service. The night buses (also known as ‘blue routes’) were excluded because they do not have defined transit stops, picking up and dropping off passengers on request. This requires special handling of these routes currently unsupported by MATSim.

Pre-Processing

Several pre-processing steps were required to use the GTFS Converter properly. The first major step was to split the original files by mode and exclude data for weekend and holiday service (which were also originally included in the data). This was required since the Converter hard-codes an assumption...
that each travel mode belongs to a separate agency. Fortunately, this step was relatively simple to code an algorithm for.

The next major step involved mapping transit stops to their nearest mode-filtered network node. This step was the first required to properly reference each transit stop to a network link, as MATSim requires. In the second step, special loop links of length 0 meters with the same to- and from-node were created and assigned to stops.

The reason for using loop links instead of referencing stops to the nearest link was due to inconsistent formatting in the GTFS data, where, occasionally, routes operating in opposite (or sometimes perpendicular) directions served the same stop. This dilemma is shown in Figure 7-3, where Routes 1 and 2 (not real TTC routes) share the same transit stop but operate perpendicular to each other.

![Figure 7-3: Illustration of the shared-stop problem](image-url)
The final step in pre-processing was to create the frequencies file. While technically not required for the GTFS Converter’s map-matching procedure, this was done in order to greatly speed up the process. Without this step, the Converter requires manual confirmation for each trip; where each trip is a single departure. With most routes having hundreds of departures during the weekday period, an each route requiring manual confirmation, processing a single line takes over an hour.

Running the GTFS Converter

Compared to pre-processing, running the GTFS Converter was a relatively simple, if laborious task. With the trips grouped by common stop sequence, the main task during conversion was ensuring that they travelled on the correct links and ensuring that all stops were linked to their correct loop link. Occasionally some routes had to be truncated or extended to match stops serviced in 2006.

Post-Processing

In the final steps of the process, a synthetic population of transit vehicles was created, based on the vehicle types defined earlier, with each vehicle being assigned to a single departure in a transit route. Before exporting, the transit schedule’s stop coordinate projection was converted back from WGS84 to NAD83 UTM17N. A validation module written by Marcel Rieser was used to ensure that the final schedule format was 100% compatible with the base network (to catch minor errors).
8 Building the MATSim Population

8.1 Data Sources

The primary source of travel behavior data used in this paper is the Transportation Tomorrow Survey (TTS). The TTS is a quinquennial telephone-survey, timed to coincide with national census, representing approximately a 5% sample of people in the Greater Golden Horseshoe (GGH) area in Southern Ontario. At the time of writing, the most recent year for which data was available was 2006.

TTS data obtained by consent for this research primarily consists of individual trip records. Each trip record is keyed to a specific household id and person id, and includes information about the trip’s start time, its zones of origin and destination, activities performed at both ends of the trip, and, for transit riders, a sequence of transit routes used in their trip.

8.2 Methodology

MATSim requires initial full-day activity plans for every individual agent in the simulation. A plan consists of one or more activity episodes, which sandwich travel episodes (referred to as ‘legs’). Each leg is assigned a mode; in this case only ‘car’ and ‘pt’ (public transit) modes were assigned to agents in the population.

Overview

To generate this population of agents with plans, the Converter algorithm from (Hao, 2009) was adapted and re-configured to handle more complex leg-mode behaviour. In brief, individual trips records (from the TTS) are loaded and linked to an individual, such that each agent is linked to a list of trips ordered by trip-start-time. Each trip has an activity at its origin and destination; these properties are used to build the agent’s activity schedule. Agents whose plans are internally inconsistent – that have only one trip, or have mismatching activities or zones at the adjacent ends of subsequent trips – are thrown out. Additionally, input file specification can be found in Appendix A.

Converting from Zones to Coordinates

MATSim requires each activity to be assigned to a specific coordinate, which is a much greater resolution than is provided by the dataset (due to privacy restrictions, the provided TTS individual trip records only included traffic zones of origin and destination). For trips by automobile, these are mapped to the nearest link; trips made by transit begin and end at transit stops.
The Converter program ‘picks’ a random coordinate within a certain radius around a zone’s centroid; as recommended and used by (Gao, 2009). The radius around each zone’s centroid is defined as 0.7 times the distance to the nearest centroid (see Figure 8-1). This has been shown to provide good coverage (Rieser, Nagel, Beuck, & Balmer, 2007).

In order to ensure consistency across households and activities, the Converter keeps track of assigned coordinates for a) Home activities of all agents within the same household; and b) work or School activities occurring in the same zone for a particular agent. Other activities have random coordinates generated for each subsequent episode, regardless of whether they occur in the same zone twice.

Household Expansion Factors

Each household is assigned a decimal expansion factor, or weight, which is typically used to expand the TTS population to better match census data. MATSim does not use agent weights; therefore each household must be replicated an integer number of times to expand the population. This replication is performed prior to coordinate generation, so household clones (and their respective persons) have varying actual coordinates. For a full population, this adds heterogeneity to the demand data.
Activity Types

MATSim requires that all activity types entered into the plans data must be pre-defined in the configuration file. The parameters assigned to each activity type (such as activity priority) are primarily used in calibrating the scoring function proposed and detailed in (Charypar & Nagel, 2005).

As this scoring function is used primarily for activity-based scheduling and demand-side replanning, these parameters are irrelevant to using MATSim as a trip-assignment model. Accordingly, the activity type parameters used in this paper have been copied over from (Gao, 2009), since they apply to the same data set.

8.3 Results of Demand Data Conversion

The results of the demand conversion are presented in Figure 8-2. A subset of transit trips were selected for use in the data conversion, with the goal of ensuring that all trips were able to be assigned to a TTC-only network. As such, only those trips which met the following criteria were used:

- Trip used the TTC exclusively
- Trip accessed to / egressed from transit by walking
- Both trip ends were in the GTHA

![Figure 8-2: Transit rider population comparison](image)

As the chart shows, the MATSim population is smaller than the full population of TTS transit riders from which it was generated; with a loss of 15%. This is a result of the Converter’s consistency.
checker: the Converter automatically detects inconsistent tours / trip-chains, of which theoretically there should be none, but exist in the data due to data filtering. As a result, this also rejects tours which include both an auto trip and a transit trip, since this would result in an inconsistent tour for both autos and transit\(^2\).

\(^2\) The trip records were converted separately, in order to expand the transit population to 100% but leave the auto population at 5%. The Converter process is fully capable of handling tours contains trips of any mode.
9 Transit Assignment Parameters & Scenarios

This section outlines assignment parameters and scenarios used for validation of MATSim and EMME transit assignment against observed data.

9.1 EMME/3 Transit Assignment Parameters

EMME’s standard transit assignment tool has been in use for many years, and updating and calibrating its assignment parameters remains the subject of ongoing work. Fairly standard parameter values have evolved over the years and are used in this thesis.

Table 9-1 lists and describes all of the parameters used by EMME during standard transit assignment, as well as the standard GTAModel values used for each parameter.

<table>
<thead>
<tr>
<th>Name</th>
<th>EMME id</th>
<th>EMME units</th>
<th>EMME value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk speed</td>
<td>ms03</td>
<td>km/hr</td>
<td>4.0</td>
</tr>
<tr>
<td>Wait time factor</td>
<td>ms04</td>
<td>unitless</td>
<td>0.5</td>
</tr>
<tr>
<td>Wait time weight</td>
<td>ms05</td>
<td>unitless</td>
<td>2.0</td>
</tr>
<tr>
<td>Walk time weight</td>
<td>ms06</td>
<td>unitless</td>
<td>2.0</td>
</tr>
<tr>
<td>Boarding time weight</td>
<td>ms07</td>
<td>unitless</td>
<td>1.0</td>
</tr>
<tr>
<td>Boarding time penalty</td>
<td>variable, by agency and mode:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boarding penalty, TTC</td>
<td>ut3</td>
<td>minutes</td>
<td>2.0</td>
</tr>
<tr>
<td>Boarding penalty, TTC</td>
<td>ut3</td>
<td>minutes</td>
<td>2.0</td>
</tr>
<tr>
<td>Boarding penalty, TTC</td>
<td>ut3</td>
<td>minutes</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 9-1: EMME transit assignment parameters

9.2 MATSim Transit Assignment Parameters

As an activity-based model, MATSim uses many more parameters than does EMME. Fortunately, most of these pertain to demand-side mutation & replanning of agent’s plans and thus are not relevant to using MATSim for transit trip assignment. Those parameters important to transit trips assignment are listed and described in Table 9-2; an expanded list can be found in Appendix B.
### MATSim transit router parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchRadius</td>
<td>Initial radius for determining feasible transit stops for access/egress</td>
<td>m</td>
<td>1000.0</td>
</tr>
<tr>
<td>extensionRadius</td>
<td>Extension radius for finding transit stops if none are found</td>
<td>m</td>
<td>500.0</td>
</tr>
<tr>
<td>maxBeelineConnectionDistance</td>
<td>Radius for selecting neighbouring stops as feasible transfers</td>
<td>m</td>
<td>100.0</td>
</tr>
<tr>
<td>additionalTransferTime</td>
<td>Buffer for agents transferring to account for the fact that no transfers are instantaneous</td>
<td>sec</td>
<td>0.0</td>
</tr>
<tr>
<td>walkSpeed</td>
<td>Straight-line walking speed</td>
<td>km/hr</td>
<td>3.0</td>
</tr>
<tr>
<td>beelineDistanceFactor</td>
<td>Factor to account for non-beeline walk paths</td>
<td>-</td>
<td>1.3</td>
</tr>
<tr>
<td>waiting</td>
<td>Waiting time perception 'cost'</td>
<td>utils/hr</td>
<td>0.0</td>
</tr>
<tr>
<td>travelingWalk</td>
<td>Walking time perception 'cost'</td>
<td>utils/hr</td>
<td>-6.0</td>
</tr>
<tr>
<td>utilityOfLineSwitch</td>
<td>Transfer penalty</td>
<td>utils</td>
<td>-1.0</td>
</tr>
<tr>
<td>travelingPt</td>
<td>In-vehicle travel time perception 'cost'</td>
<td>utils/sec</td>
<td>-6.0</td>
</tr>
</tbody>
</table>

### Mobsim parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowCapacityFactor</td>
<td>Scales capacity of links</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>storageCapacityFactor</td>
<td>Scales vehicle storage length on links</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>startTime</td>
<td>First second of the mobsim</td>
<td>time</td>
<td>-</td>
</tr>
<tr>
<td>endTime</td>
<td>Last second of the mobsim</td>
<td>time</td>
<td>-</td>
</tr>
</tbody>
</table>

### Strategy parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxAgentPlanMemorySize</td>
<td>The maximum number of plans an agent can have</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>SelectExpBeta probability</td>
<td>The probability an agent will use the SelectExpBeta module during replanning</td>
<td>-</td>
<td>0.9</td>
</tr>
<tr>
<td>ReRoute probability</td>
<td>The probability an agent will use the ReRoute module during replanning</td>
<td>-</td>
<td>0.1</td>
</tr>
</tbody>
</table>

### Custom parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCE factor</td>
<td>Passenger Car Equivalents factor. Used to scale down base PCE values for transit vehicles</td>
<td>-</td>
<td>1.0</td>
</tr>
<tr>
<td>Access time subway</td>
<td>Access and egress times per passenger to/from transit vehicles</td>
<td>sec/person</td>
<td>-</td>
</tr>
<tr>
<td>Access time streetcar</td>
<td></td>
<td>sec/person</td>
<td>-</td>
</tr>
<tr>
<td>Access time bus</td>
<td></td>
<td>sec/person</td>
<td>-</td>
</tr>
<tr>
<td>Egress time subway</td>
<td></td>
<td>sec/person</td>
<td>-</td>
</tr>
<tr>
<td>Egress time streetcar</td>
<td></td>
<td>sec/person</td>
<td>-</td>
</tr>
<tr>
<td>Egress time bus</td>
<td></td>
<td>sec/person</td>
<td>-</td>
</tr>
<tr>
<td>Vehicle Capacity Factor</td>
<td>Factor to scale transit vehicle capacities (number of agents a vehicle can carry)</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Demand type</td>
<td>Description of demand: Autos, transit, mixed, etc.</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Fixed Schedule</td>
<td>Flag for using a network with 'awaitDepartures=true' for all transit stops</td>
<td>boolean</td>
<td>TRUE</td>
</tr>
<tr>
<td>Congestion-based transit routing</td>
<td>Flag for using congestion-based routing</td>
<td>boolean</td>
<td>FALSE</td>
</tr>
<tr>
<td>Upgraded calculator</td>
<td>Flag for using the upgraded as-simulated calculator</td>
<td>boolean</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

Table 9-2: Abridged list of MATSim transit parameters
9.3 Scenarios

MATSim Scenarios

For MATSim, four different general scenario types were tested, based on the cross-combination of two dimensions: routing (as-scheduled vs. as-simulated) and congestion (no-congestion vs. congestion):

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Route-as-simulated</th>
<th>Route-as-scheduled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transit only (no-congestion)</td>
<td>NC-RaSim</td>
<td>NC-RaSched</td>
</tr>
<tr>
<td>Mixed traffic (congestion)</td>
<td>CG-RaSim</td>
<td>CG-RaSched</td>
</tr>
</tbody>
</table>

Table 9-3: Scenario cross-combination

For scenarios with as-simulated routing, a ‘pre-run’ is performed with no transit riders. The pre-run events file is then used to pre-load the TransitDataCache with simulated transit travel data. This lets transit riders select their paths based on MATSim performance characteristics. For runs which use as-scheduled routing, no pre-run data is loaded; yielding paths based on schedule travel data.

Scenarios which are labeled as ‘congested’ or ‘congestion-based’ use a mixed plans file which combines unassigned transit riders (100% sample size) with assigned drivers (5% sample size). Drivers’ plans have already been executed in a prior simulation, which lasted for 30 iterations. The purpose of these runs is to examine the effects of road congestion on not only transit travel times, but transit routing as well.

In addition to these two main dimensions of scenario design, two parameters were experimented with. The first is MATSim’s equivalent of boarding penalty (utilityOfLineSwitch) with some nominal estimates used. An additional boarding penalty scenario was tested, in which differential boarding penalties were assigned to each mode. The second parameter is a factor applied to the PCE of transit vehicles. Runs performed without congestion effects have a PCE factor of 0 applied; the runs with congestion test PCE factors of 4%, 5%, and 6%. PCE values less than 1.0 were selected in order to scale the ‘weight’ of transit vehicles with the capacity of the network, which was also set to 6% based on the work by (Gao, 2009). Higher PCE values would cause transit vehicles to create much more congestion than is realistic for a 5% sample size.

Vehicle capacities were set to 10 times their nominal value for the MATSim scenario. This was done to ensure that MATSim’s capacity constraints do not affect assignment behaviour, since such behaviour is outside the scope of this paper.
EMME Scenarios

For EMME, five scenarios were tested: the standard AM-peak model, courtesy of TMG; and four scenarios with transit lines exported from MATSim, one for each assignment period. This was done not only to ensure network parity with MATSim, but also because the standard EMME transit network is coded for the morning peak hour only.

The four assignment periods were chosen to correspond with the TTC’s service period definitions: Morning (Start – 08:59:59), Midday (09:00:00 – 14:59:59), Afternoon (15:00:00 – 18:59:59), and Evening (19:00:00 – 21:59:59).

Service attributes of lines exported from MATSim to EMME were processed as follows:

- **Service Headway:** The average headway during a service period was defined as the service period length, divided by the number of departures in the service period, converted into minutes.
- **Line Speed:** Transit line speed (which is used by EMME to calculate in-vehicle transit time) was converted by dividing the line length, the summation of all link lengths in its route, by the arrival offset of the last stop.
- **Transit Stops:** Transit stops were not mapped into EMME. EMME allows access and egress at nodes only.
- **Congestion Effects:** No congestion effects were modeled in EMME.
10 Transit Assignment Results

This section presents the results of the scenario runs, compared against observed data. Analyses performed were somewhat limited by data, as well as by EMME itself. For example, a dynamic analysis of route boardings cannot be performed in EMME. In general, analyses were performed to coincide with the assignment time periods used for EMME scenarios; and are generally aggregated over that time period.

The time periods selected for analysis correspond to those used in the EMME assignment; namely: Morning (Start – 08:59:59), Midday (09:00:00 – 14:59:59), Afternoon (15:00:00 – 18:59:59), and Evening (19:00:00 – 21:59:59). The morning peak period was selected as the primary period for analysis, since it is the period for which the most benchmark data are available.

10.1 Methodology

One of the main features of MATSim is that it allows for great flexibility in analyzing output data. The primary data file that MATSim exports for a run is the events file, which lists specific events in the chronological order they were recorded during simulation. Events files are very large (a Toronto run with approximately 500,000 agents produces 70 million events), so MATSim also provides a processing tool called an EventsManager. Customized analyses are performed by passing the manager a handler, which is coded to ‘do something’ for a selected set of event types. For example, a simple handler can ‘listen’ for all the events which a specified agent generates, and collect information about that agent’s simulation path. An example of an agent’s event-path is shown in Table 10-2.

Two main event handlers were written to analyze MATSim data. The first handler, AggregateBoardingsOverTimePeriodHandler, listened for agents entering vehicles, counting the number of boardings for each line. The second handler, AgentTripChainHandler, listened for various trip-related events, in order to re-assemble agents’ trips.

10.2 Data Sources

TTS data are the primary source of validation data, since each transit trip record provides an ordered sequence of transit routes the traveler used. These definitions readily map into MATSim and EMME lines.

The TTC also generously provided additional validation data. The most recent morning peak period (from the start of service until 9:00AM) boarding counts were provided for surface routes. See
Appendix C for a detailed breakdown of which count year was provided for which route. Detailed station boarding/alighting data from 2006 was also provided for subway routes.

10.3 Computational Performance

Computer performance has improved considerably since the last time Toronto was simulated in MATSim. Initially when (Gao, 2009) first ran her 50 iterations on a population of 134,519 agents, she reported a simulation time of up to 8 hours. (Hao, 2009) also reported long simulation times of up to 6 hours for the same population, with 50 iterations. By comparison, a trial run of automobiles only, with 118,935 agents took a little over 90 minutes to run 30 iterations on an Intel Core i7-2600 3.40 GHz, with 8 processors and 16 GB of RAM (64-bit).

Transit assignment runs considerably slower, with a single iteration taking as few as 35 minutes to as much as 6 hours for 436,107 agents; with a typical run taking about 2 hours. Fortunately, if capacity constraints are omitted, and boarding times are nullified, there is no need to perform multiple iterations since the performance of the network is not affected by passengers.

The slowest part of the assignment process is calculating the shortest paths for agent plans, which takes up the bulk of the computation time; loading data and executing the agent plans together typically take 5 minutes for the same population of agents. The reason for this is simple: the Transit Router Network automatically generated by MATSim is very large. As Table 10-1 shows, the transit router network contains seven times as many links as the base road network, with the vast majority of links being transfer links between transit stops. Early trial runs of MATSim with the maximumBeelineConnectionDistance parameter (the parameter which dictates the cut-off distance for link transfer) extended from 100m to 500m yielded networks which ran three times longer (6 hours for a single iteration).

<table>
<thead>
<tr>
<th>Transit Router Network</th>
<th>Base Network</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nodes</strong></td>
<td>34,174</td>
</tr>
<tr>
<td><strong>Non-Transfer Links</strong></td>
<td>33,173</td>
</tr>
<tr>
<td><strong>Transfer Links</strong></td>
<td>354,298</td>
</tr>
<tr>
<td><strong>Total Links</strong></td>
<td>387,471</td>
</tr>
<tr>
<td><strong>Nodes</strong></td>
<td>18,239</td>
</tr>
<tr>
<td><strong>Total Links</strong></td>
<td>53,238</td>
</tr>
</tbody>
</table>

Table 10-1: Router network size comparison
<table>
<thead>
<tr>
<th>Time</th>
<th>Activity Type and Attributes</th>
<th>Description</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00:00</td>
<td>ActivityEndEvent [type='H']</td>
<td>AT ACTIVITY</td>
<td></td>
</tr>
<tr>
<td>12:00:00</td>
<td>AgentDepartureEvent [mode='transit_walk',link='11837-7_LOOP']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:16:27</td>
<td>AgentArrivalEvent [mode='transit_walk',link='10831_LOOP']</td>
<td>Walk-access</td>
<td>0:16:27</td>
</tr>
<tr>
<td>12:16:27</td>
<td>ActivityStartEvent [type='pt interaction']</td>
<td>pt-interaction</td>
<td>0:00:01</td>
</tr>
<tr>
<td>12:16:28</td>
<td>ActivityEndEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:16:45</td>
<td>PersonEntersVehicleEvent [veh='VehType9_43113']</td>
<td>In-vehicle</td>
<td>0:01:49</td>
</tr>
<tr>
<td>12:18:34</td>
<td>PersonLeavesVehicleEvent [veh='VehType9_43113']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:18:34</td>
<td>AgentArrivalEvent [mode='pt',link='10819_LOOP']</td>
<td>Alighting</td>
<td>0:00:00</td>
</tr>
<tr>
<td>12:18:35</td>
<td>ActivityStartEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:18:35</td>
<td>ActivityEndEvent [type='pt interaction']</td>
<td>pt-interaction</td>
<td>0:00:01</td>
</tr>
<tr>
<td>12:21:11</td>
<td>PersonEntersVehicleEvent [veh='VehType9_10811']</td>
<td>In-vehicle</td>
<td>0:04:48</td>
</tr>
<tr>
<td>12:25:59</td>
<td>PersonLeavesVehicleEvent [veh='VehType9_10811']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:25:59</td>
<td>AgentArrivalEvent [mode='pt',link='10660_LOOP']</td>
<td>Alighting</td>
<td>0:00:00</td>
</tr>
<tr>
<td>12:26:00</td>
<td>ActivityStartEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:26:00</td>
<td>ActivityEndEvent [type='pt interaction']</td>
<td>pt-interaction</td>
<td>0:00:01</td>
</tr>
<tr>
<td>12:28:00</td>
<td>AgentDepartureEvent [mode='transit_walk',link='10660_LOOP']</td>
<td>Walk-egress</td>
<td>0:10:35</td>
</tr>
<tr>
<td>20:00:00</td>
<td>ActivityStartEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:00:00</td>
<td>ActivityEndEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:00:00</td>
<td>AgentDepartureEvent [mode='transit_walk',link='11435_LOOP']</td>
<td>Walk-access</td>
<td>0:08:03</td>
</tr>
<tr>
<td>20:08:03</td>
<td>AgentArrivalEvent [mode='transit_walk',link='11732_LOOP']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:08:03</td>
<td>ActivityStartEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:08:04</td>
<td>ActivityEndEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:08:04</td>
<td>AgentDepartureEvent [mode='pt',link='11732_LOOP']</td>
<td>Boarding/Waiting</td>
<td>0:05:38</td>
</tr>
<tr>
<td>20:13:42</td>
<td>PersonEntersVehicleEvent [veh='VehType9_19475']</td>
<td>In-vehicle</td>
<td>0:01:39</td>
</tr>
<tr>
<td>20:15:21</td>
<td>PersonLeavesVehicleEvent [veh='VehType9_19475']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:15:21</td>
<td>AgentArrivalEvent [mode='pt',link='11787_LOOP']</td>
<td>Alighting</td>
<td>0:00:00</td>
</tr>
<tr>
<td>20:15:21</td>
<td>ActivityStartEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:15:22</td>
<td>ActivityEndEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:15:22</td>
<td>AgentDepartureEvent [mode='transit_walk',link='11787_LOOP']</td>
<td>Walk-access</td>
<td>0:01:57</td>
</tr>
<tr>
<td>20:17:19</td>
<td>AgentArrivalEvent [mode='transit_walk',link='97067_LOOP']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:17:19</td>
<td>ActivityStartEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:17:20</td>
<td>ActivityEndEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:17:20</td>
<td>AgentDepartureEvent [mode='pt',link='97067_LOOP']</td>
<td>Boarding/Waiting</td>
<td>0:00:25</td>
</tr>
<tr>
<td>20:17:45</td>
<td>PersonEntersVehicleEvent [veh='VehType4_422']</td>
<td>In-vehicle</td>
<td>0:04:38</td>
</tr>
<tr>
<td>20:22:23</td>
<td>PersonLeavesVehicleEvent [veh='VehType4_422']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:22:23</td>
<td>AgentArrivalEvent [mode='pt',link='97070_LOOP']</td>
<td>Alighting</td>
<td>0:00:00</td>
</tr>
<tr>
<td>20:22:23</td>
<td>ActivityStartEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:22:24</td>
<td>ActivityEndEvent [type='pt interaction']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:22:24</td>
<td>AgentDepartureEvent [mode='transit_walk',link='97070_LOOP']</td>
<td>Walk-egress</td>
<td>0:19:42</td>
</tr>
<tr>
<td>20:42:06</td>
<td>AgentArrivalEvent [mode='transit_walk',link='11837-7_LOOP']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:42:06</td>
<td>ActivityStartEvent [type='H']</td>
<td>AT ACTIVITY</td>
<td></td>
</tr>
</tbody>
</table>

Table 10-2: Transit rider event sequence
10.4 Departures & Arrivals by Time of Day

A histogram of agents’ departures, arrivals, and en-routes for each trip mode type (car, transit, and transit walk) is a default output of MATSim runs. Figure 10-1 shows the histogram from a single MATSim run (CG-RaSched). Transit departures for all scenarios were the same; en-routes and arrivals were fairly similar. Departure times spike at regular intervals, reflecting an inherent bias in the data whereby survey respondents generally round their trip departure times to convenient time bins (such as 5 or 10 minutes).

It is also important to note that, towards the tail of the chart, a number of agents are not arriving at their destination. As will be shown later, the majority of these agents are stuck waiting for transit vehicles that they were not able to board.

![Figure 10-1: Histogram of transit rider arrivals, departures and en-route (Scenario = CG-RaSched)](image)

10.5 Route Boardings

Aggregate route boardings were the primary data used to validating transit assignment results; as they are analogous to link volumes for validating traffic assignment.
Effects of Boarding Penalty on Daily Boardings

Figure 10-2 shows the total daily boardings for different boarding penalty scenarios. The baseline for comparison is the TTS, since the TTC data was only provided for the morning peak period. As the graph shows, MATSim without any boarding penalty over-predicts all-day boardings by a considerable margin. The remaining scenarios are ordered with respect to decreasing boarding penalty value as entered into the Config file. As it turns out, this is an important distinction, as the boarding penalty value is not converted during routing. This is unlike the other parameters used during generalized cost calculation (such as wait time and walk time preferences), which are converted from utils/hr (in the config) to utils/sec (used in cost calculation). For example, a wait time factor of 2.0 enters the generalized cost equation as 2.0 / 3600 << 1.0!

The second-to-last scenario, by contrast uses a boarding penalty of 1.0 minutes, correctly converted so that it scales within the cost equation; which produces much more reasonable results. The best scenario, with differential boarding penalties for bus, subway, and streetcar, still over-predicts boardings by 8% compared with the full TTS data set (pictured), and as much as 23% compared against the restricted TTS data set assigned in the model.

Morning Boardings by Mode

Different transit modes in the Toronto context are often perceived by travelers differently; beyond simply having different performance characteristics. Therefore, it is useful to look at boardings,
grouped together by mode. Morning boardings by mode for selected scenarios, as well as reference data, are shown in Figure 10-3.

![Figure 10-3: Total morning boarding periods by mode (TTC lines only)](image)

The first observation to note is that there is a significant discrepancy between the full, unrestricted TTS data and the restricted TTS data with respect to subway boardings. This reflects the fact that the unrestricted TTS boarding counts include travelers accessing / egressing transit by a mode other than walking (e.g., by another transit operator, or by car). This is significant since the population used for simulation in both platforms was similarly restricted.

It is also interesting that the restricted TTS data matches so well with the TTC boarding counts, given that those boarding counts are completely unrestricted.

The MATSim scenario runs in general largely over-estimate boarding counts for bus routes, over-estimate for streetcar routes, and under-estimate for subway routes; while over-estimating total boardings overall. Furthermore, the introduction of a differential boarding penalty (NC-RaSim Diff.BP) does not appear to have much effect in shifting boardings from surface modes to subway. This is despite there being no boarding penalty for subways. Based on the results of overall boardings, it was decided to use the differential boarding penalty for subsequent scenarios.

Congested runs with different PCE factors were omitted from the chart because their boarding counts were nearly identical. This demonstrates the independence between simulation and routing.
EMME scenarios, both standard and otherwise, generally perform very well with respect to boardings; although both tend to under-predict subways as well.

### 10.6 Travel Time Components

An analysis of travel time components is useful in showing how different parameters are effecting simulated travel times. Unfortunately, no time-dependant data were available, so MATSim times are compared against those simulated in EMME. Figure 10-4 shows average trip component times for the morning peak period. Average component times were used in order to show the effects of large-scale parameter changes.

![Figure 10-4: Average morning trip component times](image)

The major difference between EMME and MATSim is in the in-vehicle-travel-time (IVTT). This is easily explained by the fact that EMME’s transit lines are configured to run at their average line speed, whereas in MATSim vehicles (particularly buses) travel at the link speed. The effects of congestion on the average IVTT can be clearly seen by comparing the first two no-congestion runs to the latter four congested runs. It is also clear that the PCE values selected have minimal impact on simulation times.
Figure 10-5: Average midday trip component times

The other important observation from this chart is that the MATSim’s routing network appears to be less sensitive to walk times. This is a key difference, since EMME uses walk-on-links while MATSim uses beeline-walk. MATSim average waiting times appear to be slightly higher than EMME’s. This observation is consistent across all three assignment periods.

Figure 10-6: Average afternoon trip component times
The final travel time chart is Figure 10-7, which shows the average total transit travel time for all time periods. As the chart shows, although their average component values vary, the average total travel time is fairly consistent comparing EMME to MATSim. Additionally, the effects of congestion on total travel time can be seen; with higher average times for congested scenarios in the morning and afternoon peaks.

Based on the results from boardings as well as the results from travel time analysis, subsequent analyses are presented using data from the final scenario (CG-RaSched PCE=0.06).

10.7 Subway Boardings and Alightings

Subways are the backbone of the Toronto transit network, therefore ensuring that they are being simulated correctly is important for model validation. Subway boarding and alighting data were available in both space (stations) and time (counts over time periods of 15 minutes) for all stations and lines. Two aggregations of this data are presented: total boardings and alightings by time of day (aggregation over space), and station boardings and alightings by assignment period (aggregation over time).

Line Boardings and Alightings by Time of Day

Figure 10-8 and Figure 10-9 show the boardings and alightings aggregated over all stations, for the Bloor-Danforth line (BLR) and the Yonge-University-Spadina line (YUS) respectively. These data are compared against observed counts provided by the TTC.
Figure 10-8: Bloor-Danforth subway line boardings and alightings, OBS vs. simulated MATSim

Figure 10-9: Yonge-University-Spadina subway line boardings and alightings, OBS vs. simulated MATSim

Figure 10-10: Sheppard subway line boardings and alightings, OBS vs. simulated MATSim
As the figures show, the data from MATSim generally fits the morning peak period well for the two major subway lines (BLR and YUS), although a little lower than the observed values. Activity during the middle of the day as well as the afternoon peak is considerably lower, although this appears to be linked to the distribution of demand shown in Figure 10-1. The Sheppard subway line is massively under-predicted except for the afternoon peak where it is still poorly used.

Station Boardings / Alightings by Assignment Period

Station boardings and alightings are included in order to look at the geographic distribution of subway activity. Charts were prepared for all four assignment periods, to allow for EMME results to be included in the comparison. Figure 10-12, Figure 10-13, and Figure 10-14 show morning period boardings from the TTC observed counts, MATSim scenario CG-RaSched, and EMMe + MATSim transit; the remaining charts can be found in Appendix D. Data point size is scaled relative to station activity (i.e., it is not absolute) since the goal is to illustrate distribution. Absolute activity is already recorded in Figure 10-3.

From these charts, it can be seen that the general distribution of agents boarding and alighting to/from subway in MATSim fits the observed data well. Notable exceptions to this are major transfer stations, such as Union, Kipling, and Finch; all of which are under-predicting boardings considerably. This is expected, since the TTC is the only operator being simulated, and park-and-ride trips are not being simulated. This is also true of EMME, although the EMME results show a more even distribution of subway boardings.

These general trends carry throughout the day.

10.8 Unfinished / Stuck Riders, and All-Walk Trips

Tested scenarios included a simulation end time, to prevent the mobsim from running indefinitely if agents became stuck. Stuck agents are identified during the analysis process as those with unfinished trips. Figure 10-11 shows which part of their trip agents are stuck in when the simulation
Figure 10-11: Number of stuck agents for selected scenarios

Figure 10-12: Morning observed TTC Station boardings
Figure 10-13: Morning MATSim simulated station boardings

Figure 10-14: Morning EMME simulated station boardings
Kucirek, 2012

terminated. As the chart shows, the majority of agents are stuck waiting for vehicles which will not
arrive until the subsequent day. This is expected behaviour in a sense, as these agents illustrate that
some departure time mutation and/or choice is necessary for simulating dynamic transit.

Agents stuck in-vehicles are problematic, in that no vehicles should be operating by the time the
simulation ends. Stuck transit vehicles indicate that either MATSim has removed them from the
simulation, or that an error exists in the network formulation.

In addition to stuck riders, an analysis of all-walk trips is also illustrative of where the simulation is
not functioning as desired. In theory, all of the trips being assigned to the network should be
assigned to at least one transit route, however some all-walk trips are expected. These can result
from coordinates which are very close together for intra-zonal trips. Figure 10-15 and Figure 10-16
show the number of all-walk trips in selected scenarios.

Figure 10-15 is particularly illustrative of the effects of different boarding penalties on the
simulation. High boarding penalty values clearly show transit trips being shifted to all-walk trips
which avoid incurring such high costs.

![Figure 10-15: Number of all-walk trips for boarding penalty scenarios](image)
10.9 Analysis of Non-Assigned and Infeasible Transit Trips

Often in travel demand modelling of transit, it is desirable to know information about trips not taken, as well as trips taken. Mode choice models, for example, often include travel times made by each mode as significant explanatory variables. With the transit mode, it is also important to know when and where transit trips are not possible (particularly in the North American context), as the distribution of transit services varies widely in time and space.

As it turns out, this kind of data is very easily extracted from MATSim, both in post-processing and during a run, depending on the setup. For example, if the population is set up such that each agent has two plans, one for each mode, then MATSim automatically scores each plan during Scoring. This can also be accessed post-run, by simply loading up the output events file along with schedule data.

Within MATSim, infeasible trips are fairly easily identified. By default, MATSim will not be able to find paths for trips with origins or destinations further than a specified radius. The default router also cannot find paths if the network is not well-connected. This can occur if the maximum connection parameter is too small.

With respect to time, MATSim’s behaviour is determined by its `TransitRouterNetworkTravelTimeAndDisutility` calculator; specifically by how it deals with agents arriving at a stop past the last scheduled departure. The default behaviour is that MATSim returns the first available departure the next day. The upgraded calculator used in this paper was configured.
Kucirek, 2012

to return two days (48 hours * 3600 sec / hr). In either case, ‘infeasible’ transit trips with respect to time have very high waiting times.

11 Discussion

This section discusses the findings above in greater detail, providing additional context and analyses to support final conclusions.

11.1 Data Restrictions & Validation

It is important to keep in mind the particulars of the demand data used for this simulation, particularly that only walk-access-transit is being represented. Given a smaller subset of data, it is expected that volume across all platforms should be lower than expected. Furthermore, it is generally expected that this under-prediction occur with respect to subways, since subway stations are major multimodal transfer points to modes not represented in this model (e.g. local transit, drive-access-subway). The relative size of the subset of assignment versus the set of unrestricted trips is illustrated in Figure 8-2.

11.2 Accuracy of the Transit Router

The boardings chart by mode (Figure 10-3) is illustrative of the performance of the transit router; which in the case of MATSim, is a separate step from simulation. The boardings from the two EMME runs are indicative of “routing as-scheduled, walk-on-links” for two different transit network scenarios: the first being the standard GTAModel transit network, and the second network derived from the MATSim transit schedule. These two runs demonstrate how low subway boardings are a product of this specific combination of demand and network data. Compared against both restricted and unrestricted boardings from the TTS, both EMME runs under-predict subway boardings.

The MATSim scenarios, on the other hand, explore “routing-as-scheduled” versus “routing-as-simulated” for “walk-in-straight-lines.” To fairly compare with EMME’s transit router as-scheduled, one must also look at the MATSim scenarios routed as-scheduled. Two such scenarios are displayed on the chart: one without any boarding penalties (NC-RaSched BASE), and one with mode-variable boarding penalties equivalent to those used in the EMME assignment (CG-RaSched). The latter of these two can be considered the ‘best’ run, and, indeed compares favourably against the EMME scenarios and against the observed data.
In general, the MATSim runs tend to over-predict boardings on surface routes (buses and streetcars). Indeed, much of the total over-prediction of boardings can be attributed to the over-loading of surface routes, since subway boardings are generally constant across all scenarios.

This over-loading appears to be attributable to two main factors: The first is that the simulated speeds of surface routes are higher than they should be (which will be discussed later). As such, passengers are electing to board surface routes to make short connections instead of walking, resulting in additional boardings being recorded. This has been confirmed through spot-checks of the event-paths of a handful of random passengers.

The second factor influencing the over-prediction of surface routes is less clear, however. The effects of faster simulated surface route speeds can only be attributed to those scenarios routed “as-simulated,” where the results of the simulation are used for routing. Yet scenarios routed as-scheduled also show over-prediction of bus routes. This could be attributed to the specific geography of transit stops, and how it interacts with the router’s handling of access/egress/transfer links. This is only one possible explanation, and is yet to be validated by evidence.

11.3 Accuracy of the Microsimulator / Mobsim

The travel time charts (Figure 10-5 to Figure 10-7) were included to illustrate the time-based results of the transit assignment; specifically, the results of the mobility simulation (mobsim). It should be noted that there are no corresponding ‘ground truth’ data available; rather, comparisons are made against the EMME model. This comparison is further complicated by the fact that EMME’s travel times are computed as-scheduled, whereas MATSim’s travel times are based on the simulation.

This turns out to be a key distinction, particularly in the uncongested scenarios, as surface routes travel unimpeded at link freeflow speeds, with instantaneous acceleration / deceleration. This is further compounded by zero second dwell times at stops and stations. Passenger loading / unloading time is a major source of delay for transit vehicles which is accounted for in the transit schedule. Clearly, omitting stop delay during simulation results in considerably higher speeds for all modes in all scenarios. Higher transit speeds are most evident during visualization, but can their effects can also be seen in the travel time charts: MATSim average simulated in-vehicle travel times are lower than EMME’s in all cases.
It was thought that introducing congestion would rectify this situation by slowing down surface routes. However the results from congested scenarios suggest that, although congestion is decreasing line speeds as expected, it is not decreasing line speeds enough. In other words, the observed effects have the correction direction but not the correct magnitude. The reason for this was discovered during visualization: most minor roads with bus routes are seeing relatively low levels of automobile usage. With such a small sample of drivers (approximately five percent), there is simply not enough spatial and temporal heterogeneity to induce realistic traffic on non-major roads. Therefore, most bus routes remain largely unimpeded throughout most of the day.

The other trip component times are simulated in MATSim to be much higher than those used by EMME.

Since the transit-walk mode is handled by MATSim in a straight line, it could be that higher walk times are a result of insufficiently-calibrated walk speed parameters. Transit stop geometry should also affect walk times, although there does not appear to be a specific issue with the transit stop data. One possible culprit is that, in this implementation, it is a common occurrence that several transit stops are mapped to one loop link. This might result in inaccurate behaviour: agents will pick the nearest stop to their destination, but alight from their vehicle at a link much further away. It is unclear how MATSim handles this, but it could be that in such instances the passenger incurs an additional walk episode moving from the link of egress to the stop of egress.

With respect to average total waiting times, a proper analysis is problematic given that EMME calculates expected wait times instead of actual wait times for transit users. It appears that average wait time as reported by EMME is quite low, especially in the afternoon peak with an average time of 3 minutes. Without additional ground truth data to compare against, it is difficult to properly validate waiting times.


12 Conclusions & Future Work

12.1 Conclusions

This thesis presents an implementation of the MATSim transit model for Toronto, and it introduces several new techniques using MATSim transit. It demonstrates that MATSim can be configured to include the effects of congestion in transit routing. This technique can further be extended with passenger boarding / alightings delays to produce an iterative transit assignment, although this was not used in this thesis. It also demonstrates techniques for map-matching transit schedule data from high-resolution transit stops to a low-resolution planning-level network. Finally, this thesis demonstrates that MATSim can assign traffic and transit separately as well as simultaneously.

From the results presented above, it is clear that this specific implementation of the MATSim transit still requires considerable work in validation and optimization before it can be considered a stable, well-tested model. The biggest issue with respect to model accuracy is high simulated line speeds, particularly on surface routes. This results from a deliberate decision to use zero dwell times at stops, as well as low density of automobile traffic on side-streets where many buses run. Additional complications arise from the mismatch in spatial resolution between the transit schedule data and the planning-level network used.

However, it is also clear that these issues stem from this specific implementation of the MATSim model, not from the platform itself as MATSim has been used to produce large transit simulation models for other large regions. The suggestions for future work laid out in this section chart a path to increasing the accuracy of the MATSim Toronto model to a viable level.

12.2 Future Work

Calibrating and Optimizing the Router

It is clear that the way in which MATSim creates its transit router network from Toronto data is in need of some heavy optimizing. Too many transfer links are being created as a result of very high spatial resolution of stop locations. This is impacting both computation speed and model accuracy. Therefore, more research is needed to establish an appropriate network resolution, both for transit stops and for the base network itself. Such research would respect computational constraints while at the same time maximizing model accuracy.
There is also some work to be done calibrating walk / transfer parameters used by the router. The sensitivity of the router to different walk speeds – as well different beeline distance factors – needs to be established.

Calibrating the Microsimulator

In order to improve the accuracy of the microsimulator, transit line speeds need to be lowered to closer match scheduled speeds. First, calibrated boarding and alighting delays at stops need to be implemented. Data on dwell times at stops per passenger may need to be obtained for all vehicle types simulated in order to do so. This will introduce a significant element of delay at the cost of introducing a reciprocal relationship between supply and demand. Second, it is desirable to expand the automobile sample size to a full population. Doing so will introduce additional spatial heterogeneity and hopefully create more realistic traffic conditions on non-major streets. The drawback is that this would increase computation time considerably.

Additionally, it would be useful to prepare a base network and transit schedule which produces as-scheduled transit line speeds. Such a network could be used to quickly test router parameters, and performance against EMME. However, a simplified network would not be multimodal and thus completely insensitive to traffic congestion.

Finally, the impacts of capacity constraints need to be investigated more thoroughly. Leaving passengers waiting at stops when they miss their connection is best-suited for high-frequency lines or for stops in areas with limited transit coverage; however this might not make sense for areas with very high density of transit coverage. In such areas, passengers are able to choose another route either serving the same stop or within a short walk.

Extending the Transit Network

There is still a strong need to extend the simulated transit schedule to include other operators, such as GO Transit, to allow for a fully integrated 24-hour dynamic transit model for the region. Schedule data exists for all operators, usually in wildly different formats; fortunately, many operators in the region are migrating data to the GTFS format to allow their customers to plan trips online using Google.
Stochasticity of Simulation

It is unclear how stochasticity affects the results of microsimulation models such as MATSim. Further research is required in order to understand just how stable the final state of MATSim is.

Additional Simulation Functionality

**Transit Fares:** Work on using MATSim to model the effect of transit fares on route choice originally part of the scope of this paper, but was omitted due to time constraints. Nevertheless, MATSim remains an exceedingly flexible platform for implementing such work. In particular, the updated travel time calculator presented in this paper readily extends to including the disutility of fares – fares are simply an additional term to the cost equation. MATSim even allows for heterogeneous perception factors for fares; although such work would require extensive calibration & parameter estimation.

**Mixed Mode Trips:** Park and ride trips, also known as auto-access-transit trips, play a significant role in the GTHA’s transit network; indeed, such trips are the core of GO Transit’s ridership. However, MATSim at the moment does not provide a fully functioning and tested implementation of these trips. For MATSim to be used to model transit in the GTHA more extensively, it needs to implement this functionality.

13 Works Cited


Kucirek, 2012


Cover photo credit: FLICKR user togirl666