

LARGE-SCALE MULTIMODAL TRANSPORT MODELLING

Part II: Implementation and Validation

Quintin van Heerden^{a,b} and Johan W. Joubert^b

^a Transport and Freight Logistics, Built Environment, Council for Scientific and Industrial Research, Meiring Naudé Road, PO Box 395, Pretoria, 0001, South Africa.
Tel: 012 841 3377, Fax: 012 841 4044, Email: qvheerden@csir.co.za

^b Center of Transport Development, Industrial and Systems Engineering, University of Pretoria, Private Bag X20, Hatfield, 0028, South Africa.

ABSTRACT:

Recent developments in agent-based transport simulation provide promising results. However, the agent-based approach is frequently criticized for its apparent dependence on vast amounts of, mostly unattainable, data. In this second paper of a two-part series, we demonstrate the use of the Multi-Agent Transport Simulation (MATSim) with the multi-modal synthetic population generated in the first part of the series. The paper addresses the co-evolutionary approach with which agents attempt to maximize their individual utilities using both time and cost elements. The agent-based model is run for the Nelson Mandela Bay Metropolitan area using multiple modes that include commercial vehicles. We validate the simulation results against regional traffic counts, as well as other spatio-temporal anchors. The co-evolutionary nature of agents' decision-making is shown as the simulation progresses, and we analyse how agents' choices evolve over time as the simulation moves towards a steady-state. The paper ends with a few examples of how the disaggregate modeling can be used to answer policy and infrastructure decisions.

1 INTRODUCTION

In the first of this two-part paper series we showed how the necessary data is prepared to run a Multi-Agent Transport Simulation (MATSim) model. The first requirement is a synthetic population. In this paper we use both private individuals and commercial vehicles. The private individuals span all age groups, employed and unemployed, and consider the individuals' complete activity chains across a 24-hour period. That is, we do not model only the morning or afternoon peak, and activity types include home, work, education, shopping, leisure and other activities.

Providing a network is the second requirement. The network indicates the detail connectivity as well as the capacity in terms of the number of lanes and the carrying capacity per lane per hour.

The contribution in this paper is to describe the simulation process once the data is prepared. To that extent the paper is structured as follows. In the next section we provide details on configuring the simulation. That entails addressing specific components in the configuration file, elaborating on activity types, and describing how the different simulation modules interact together to provide the final results. In Section 3 we provide a number of standard and novel analysis to assist in validating the model.

2 EXECUTING MATSim

All input files are provided in the Extensible Markup Language (XML), namely the population and network files. The iterative simulation is also configured using an XML configuration file.

2.1 Configuration

The configuration file, or config file for short, is made up of multiple modules, each dealing with specific elements of the simulation, and having a number of parameters that the user can set. Some modules are quite generic, such as the coordinate reference system used, a single random seed and the location of the various input files.

Other modules deal with the execution of the simulation, such as the simulation horizon, for example 24 hours, and also the number of iterations that will be executed during the simulation run. In this implementation we opted for 200 iterations, an arbitrary choice based only on our experience. One of the modules has to do with the mobility simulation used. The descriptions of the different options are available on the MATSim website (Matsim Development Team, 2012), suffice to say that we use the Java Deterministic Event Driven Queue Based Simulation, or *JDEQSim* for short. The implementation is based on a discrete event simulation model while the traffic simulation for streets is based on a first-in-first-out queue model. The simulation has deadlock prevention mechanisms and gaps that are generated at the front of the queue when vehicles leave must first propagate backwards. This results in a more realistic traffic model.

2.2 Refining activity types

At the end of the mobility simulation each agent evaluates the executed plan during a process called '*scoring*'. Each activity is scored using a function similar to that discussed by Charypar and Nagel (2005). In the configuration file there are a number of parameters that can be set, and these need to be set for each activity type that occur in the agents' plans.

The shape of the logarithmic function for each activity type is determined by two key parameters: the *minimum duration* indicating at which point an agent starts earning utility, and the *typical duration* that an agent is aiming for. The overall simulation achieves a steady state when the marginal utilities of the different activities in an agent's plan become similar. That is, the slope of the utility functions of the different activities becomes similar. The scoring function is crucial in the success of the simulation results. To that extent we spent quite some effort on ensuring we understand the different activity types in a lot of detail. For example, we firstly split the single home (h) activity into three different types: those that occur at the start of an activity chain; those that occur at the end of a day, and home activities that occur throughout the day in the middle of activity chains. The reason is that although they occur at the same location, they are distinct in nature and duration.

Furthermore, we analysed the duration distributions of the different activity types. If, based on our expert opinion, the durations were reasonably similar, we retained the original activity type. In this regard, consider for example the activity that deals with attending primary or secondary education, e_1 . Figure 1a shows the duration distribution of all e_1 activity occurrences in the population. The dotted vertical lines represent the deciles, i.e. every 10% of the distribution. The mean duration is 6 hours and 10 minutes and is used as

typical duration. The same approach was followed for activities dealing with dropping kids at school, e_3 , and activities classified as *other*, o .

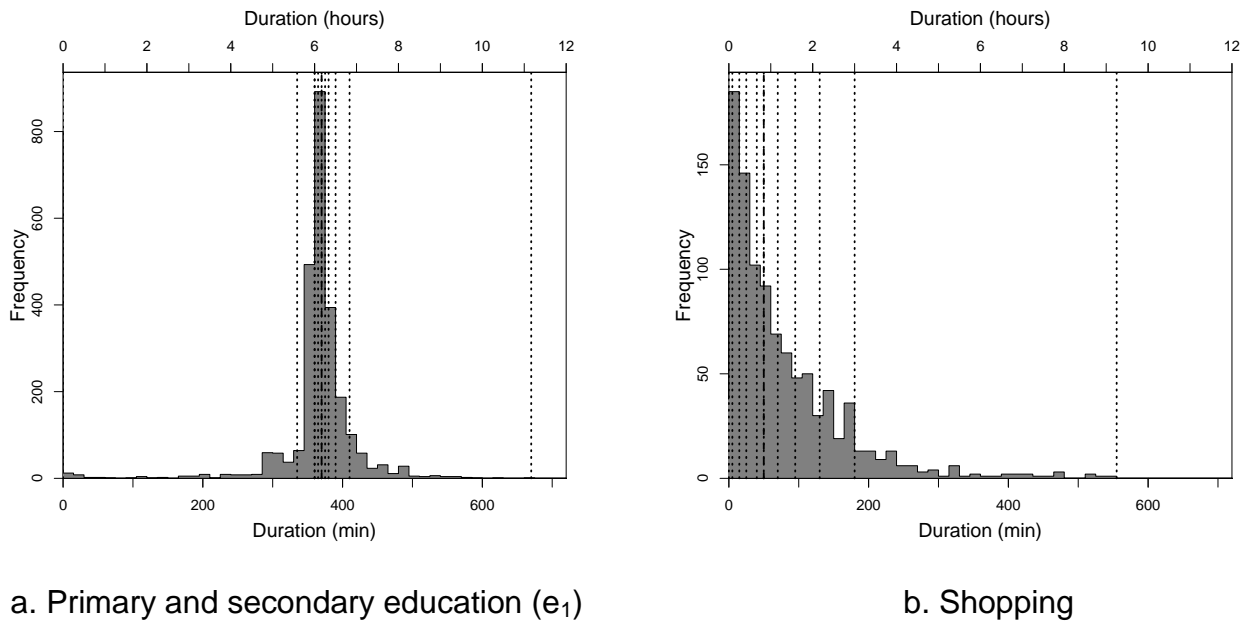


Figure 1 - Activity duration distributions indicating the spread and the deciles.

Now consider Figure 1b where we show the duration distribution of shopping activities (s). The distribution is much wider. Instead of using a single typical duration, we break up the activity type into quantiles, that is, five equal proportions. All shopping activities that fall within the first quantile is reclassified as s_1 , and those in the second quantile, s_2 , etc. The typical duration for an activity type is then the median duration within that quantile. The same approach was followed for the three home activity types, work, tertiary education, leisure, and the major and minor activities of commercial vehicles.

In the configuration file's `planCalcScore` module the typical activity durations are specified, each with its own `activityType` and `activityTypicalDuration` parameter value. The scoring function of the simulation will then ensure that agents aim for activity durations similar (not necessarily equal) to those reported in the original travel survey from which the travel demand was derived.

When agents score their executed plans, the utility value of the plan is updated and agents are ready to pick a plan for the next iteration of the simulation.

2.3 Replanning

In reality people make changes to their travel patterns in an attempt to improve the expected outcome. Similarly, agents are allowed to replan. The replanning strategies are also set in the configuration file in a module called `strategy`, and specifically the parameters indicating the choice of strategy and the probability of being applied. In this implementation we've opted for a basic set of three strategies.

For the next iteration, the majority of agents (70%) will choose an existing plan from their memory. The strategy, called `ChangeExpBeta`, uses the score (utility value) of each plan to determine the likelihood of being chosen. Plans that are expected to score well have a higher probability of being chosen.

A small proportion (15%) of agents are allowed to reroute an existing plan, a strategy referred to in the configuration file as **ReRoute**. That means that they pick a plan from memory, and alter the route between two consecutive activities. The adapted plan is then added to the agent's memory. The configuration file allows for the maximum number of plans any agent is allowed to keep, to be set. If the added plan sees the maximum number of allowed plans will be exceeded, the plan in the agent's memory with the lowest score is deleted before the newly adapted plan is added. This added plan would then be used during the next iteration.

The third strategy is called **TimeAllocationMutator** and is also chosen with 15% probability. Although sounding very complex, it actually just means that an agent can adapt the timing of an activity. The strategy randomly changes the activity end times. The extent of the temporal shifts can also be configured. When an agent is chosen to employ this strategy, a single activity's timing is changed, and the adapted plan is added to memory, again dropping the current worst plan if the number-of-plans-in-memory limit is exceeded.

The strategies in our implementation are configured so that rerouting and activity-timing adaptations are only allowed for the first 150 iterations, after which the simulation is run for another 50 iterations with only the **ChangeExpBeta** strategy. This allows for the simulation to settle into a relaxed state.

Once each agent has selected a plan for the next iteration, the current iteration ends, and the new one starts. The sequence of execution, scoring and replanning is managed in MATSim by an object called the *CONTROLLER*. A number of controller-specific parameters can be set in the configuration file. One of these is the frequency that events are written to file.

As we use an event-driven mobility simulation, every movement of an agent and any change in the state of the simulation are recorded in the form of events. For example, when an agent leaves home to travel to work, a series of events are captured. At first the home activity is ended with an **activityEndEvent**, followed by **AgentDepartureEvent** and **AgentWait2LinkEvent** if the agent travels by private car. For the journey, every time an agent moves from one link in the network to the next a series of **LinkLeaveEvent** and **LinkEnterEvents** are generated. Once the agent gets to the work location, an **AgentArrivalEvent** and **ActivityStartEvent** are generated. Each event type has its own attributes, but usually these include the time the event occurs, and the agent responsible for the event. Different event types then have specific attributes, for example an **ActivityStartEvent** would be required to indicate the type of activity that started.

An agent with a simple home-work-home activity chain, travelling by car can therefore easily be responsible for nearly a thousand events throughout a day. To save space, the configuration file allows one to indicate the number of iterations that should pass before events are written to file. One benefit of the detailed events file is that it can be analysed over and over without having to rerun the simulation, as one can parse the events file from the final iteration, and perform the necessary analysis.

3 SIMULATION RESULTS

Over the course of the simulation run, agents iteratively update their plans in a co-evolutionary manner. That is, they adapt their plans based on their experience, which in turn, is the result of many other agents' actions as well. If we take the best plan of each agent, based on the utility value of the plan, and average over the entire population, we can plot the utility over the simulation run. In Figure 2 that is indeed what we show.

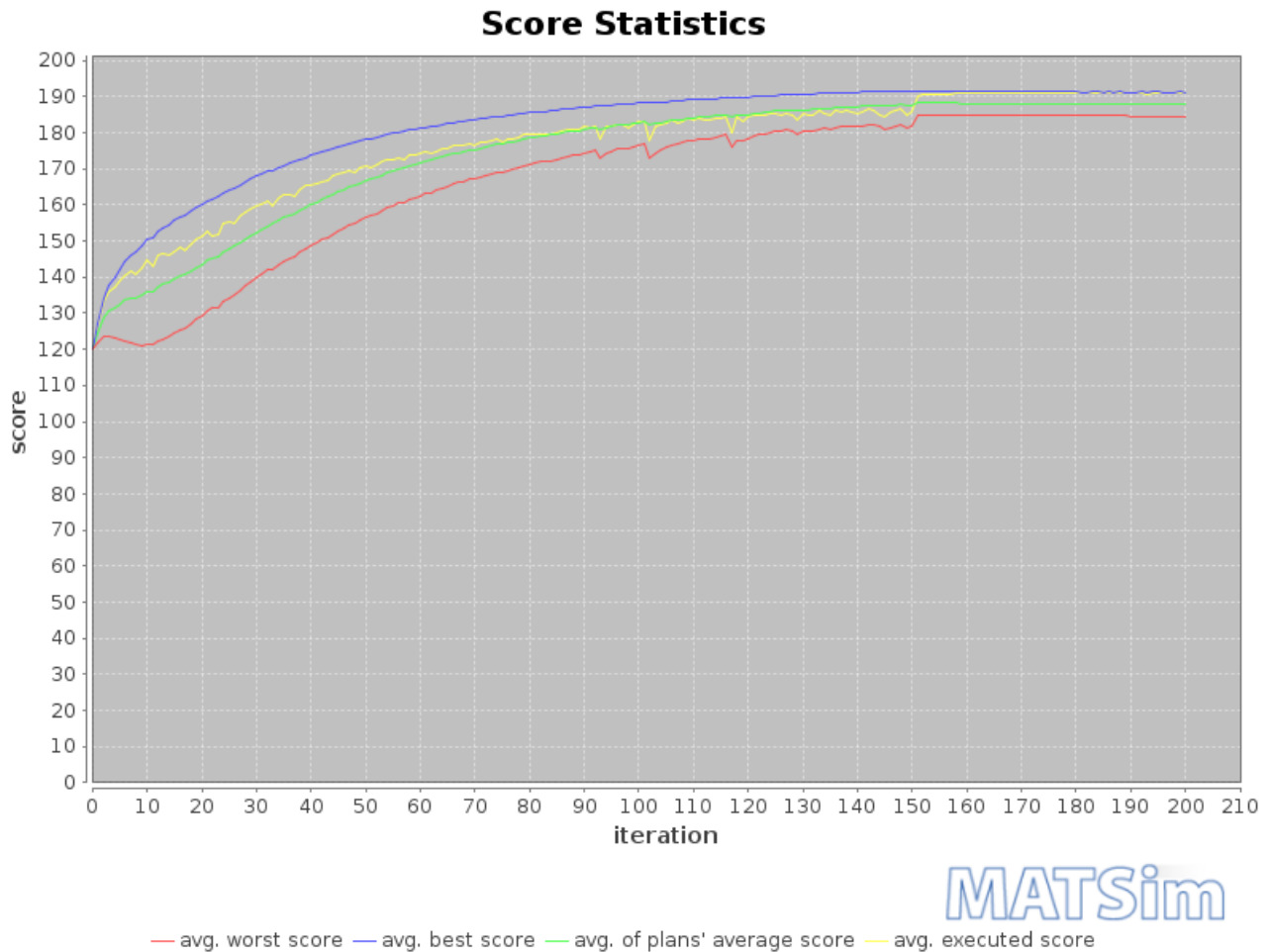


Figure 2 - Overall improvement of agents' plans over the iterative simulation.

The blue line shows the average over the population's best plans. The yellow line shows the average over the population's executed plans; the green their average plans; and the red their worst plans. What should be noted is how the average executed plans' score improves steadily from approximately 120 units at the start of the simulation until it reaches a relaxed state with a score of just over 190 units.

There is a slight jump at iteration 150 when the **ReRoute** and **TimeAllocationMutator** strategies are terminated in favour of only the **ChangeExpBeta** strategy for plan selection.

In this section we focus on mainly two analyses to validate the model. The first focuses on a spatiotemporal comparison of the simulated vehicle counts against average traffic counts. The second is concerned with commercial vehicles in particular and evaluates the vehicle kilometers travelled.

3.1 Traffic counts

The best regional traffic counts we could get were the 2011 data published by PBS-SA, the appointed consultants for the Eastern Cape region. Unfortunately many of the counting stations have only been recorded for a 24 – 72 hour period.

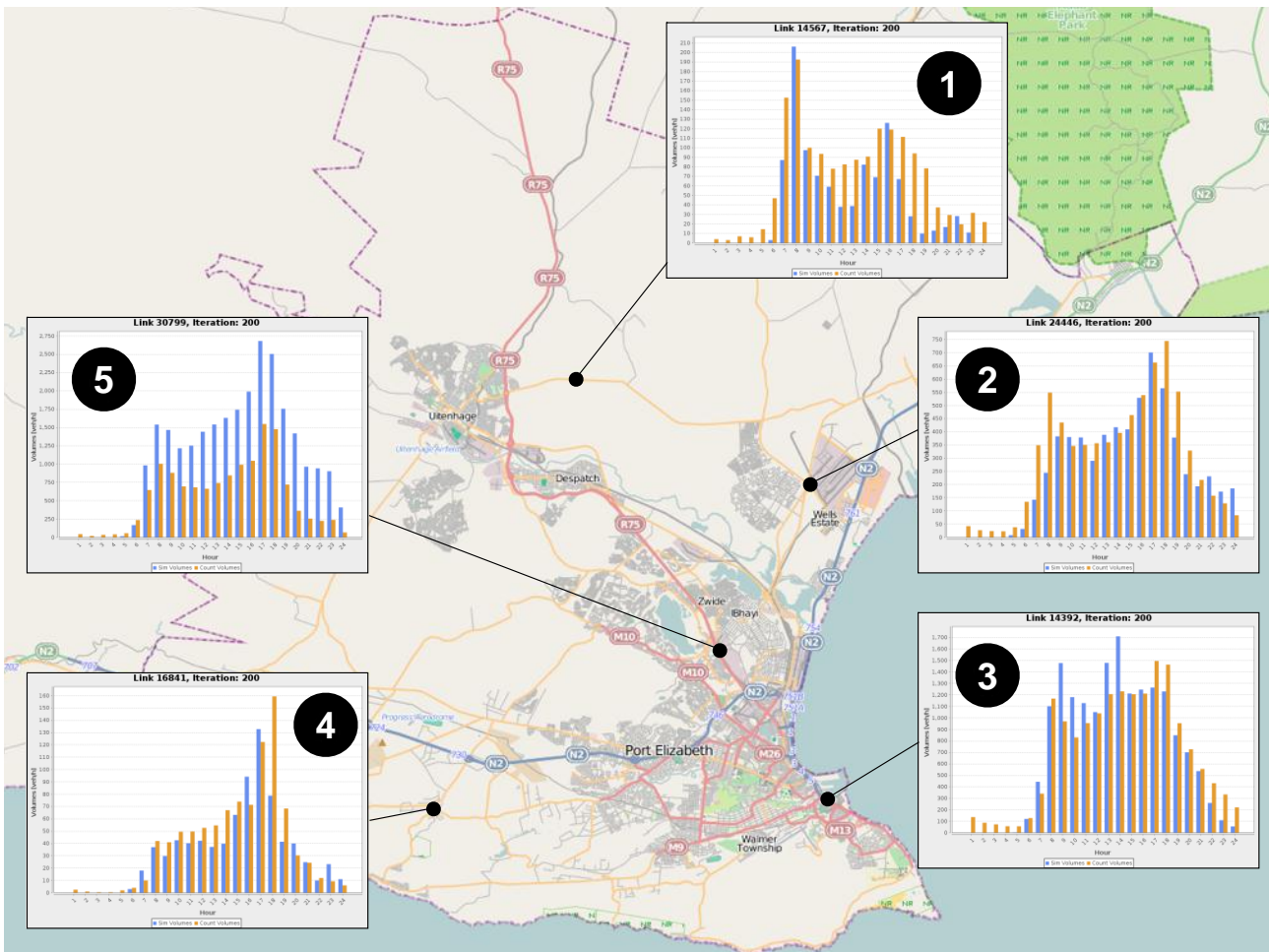


Figure 3 - Selection of counting station comparisons.

Figure 3 shows five of these counting station comparisons. Each orange bar indicates the number of vehicles that were counted on that link for that specific hour. Each blue bar indicates the number of vehicles simulated over that link during that hour. From Figure 3 we see that we can compare the simulation over an entire 24-hour period, and that the shapes of the simulated vehicle counts corresponds quite well with the actual observed vehicle counts throughout the day.

The fifth counting station shows a consistent over-estimation of the traffic from 07:00 in the mornings until 20:00 in the evening. The state-of-practice is to ‘calibrate’ the model; to force it to fit to the counts. In an agent-based setting we believe count comparisons is a valuable tool to improve data quality. Instead of manipulating the model to return the counts, we aim to understand why the differences are observed. In the case of station 5, one might proceed as follows.

When converting the network from *OpenStreetMap*, conversion defaults are used. In the case of station 5 in Figure 3, the counting station is associated with link 30799, which in reality is the northbound section of the R75 just south of the intersection with the M14. In

OpenStreetMap, the link is classified as a *Primary Road* and tagged as having two lanes per direction, which is indeed the case in reality. During the conversion, *Primary Roads* have a default capacity of 1500 vehicles per hour per lane, and a free speed of 80km/h. Although the speed corresponds with the actual set speed limit on the road, the default capacity may be an over-estimation of actual capacity, and may warrant a change.

Over estimation on one link may also hint at a connectivity issue of the network as a whole. That is, a road that acts as a viable alternative to the R75 may indeed be absent, or under-capacitated, in the network used for the simulation.

More likely, though, is the variance that exists in real traffic counts. In that regard, consider Figure 4 that shows the traffic count distribution at a single counting station. All the observations in the distribution are taken for a specific hour (here 15:00 – 16:00), and using only those days that are considered ‘*normal*’. That is, public holidays and abnormal days are excluded.

Wednesday, 15:00

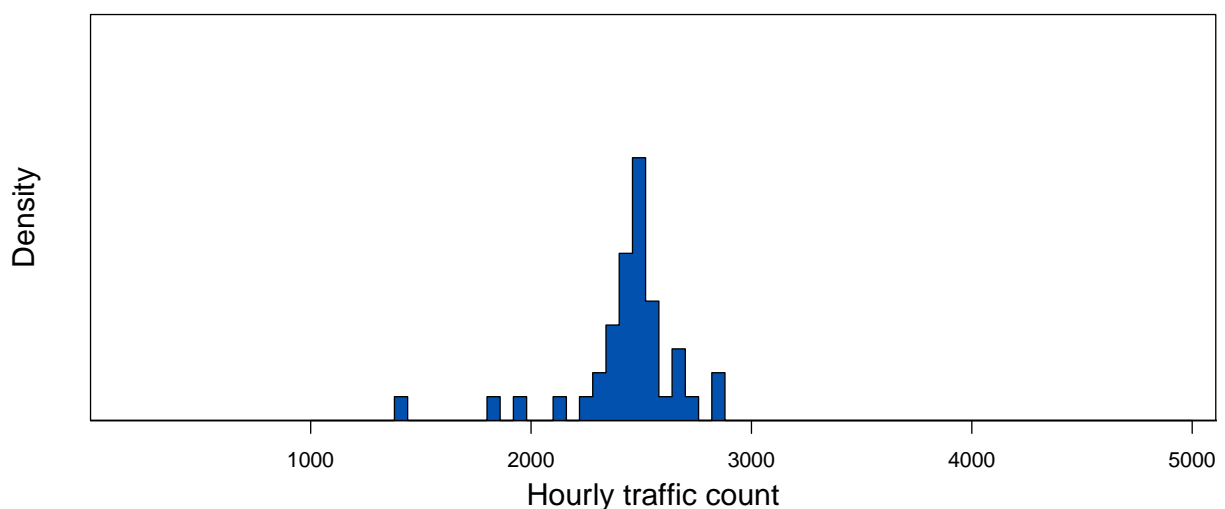


Figure 4 - The distribution of observed traffic counts for a typical Wednesday between 15:00 and 16:00 showing the wide spread.

It may just be that the short period over which the traffic counts were taken were indeed on the lower portion of the distribution, while traffic simulated represented typical traffic on the higher end of the distribution. That is indeed the challenge with models: we may never know.

Although traffic count comparisons may provide intuitive and visual hints to where network data can be refined, it is not absolute pointers to building better models. Even worse, we should not blindly calibrate our models to fit those counts.

3.2 Vehicle kilometers travelled

The total distance that vehicles travel, also referred to as the *Vehicle Kilometers Travelled (VKT)*, is a proxy to measure the impact of commercial vehicle activity on the roads and environment. We utilize this measurement to validate our commercial vehicle synthetic population. Van Heerden and Joubert (2012) show that the activity chain characteristics of intra- and inter-provincial commercial vehicles differ and should be addressed separately.

Accordingly, we determined the VKT for both intra- and inter-provincial commercial vehicle activity chains anticipating a difference in VKT.

Recall that we generated activity chains from GPS logs (described in Part I of this two-part series). GPS signals are infrequent and consecutive signals are not necessarily consistent, sometimes exceeding 5 minutes in duration. For this reason accurate map matching is not possible and the VKT must be estimated. The estimated *observed* VKT was the result of an A*-landmark routing algorithm that determines the shortest path between successive activities (Hart et al., 1968). We compared this with the VKT from our synthetic commercial vehicle population. We determined the VKT for different road types, based on the road types on *OpenStreetMap*. The results of inter-provincial vehicles are shown in Figure 5. We only show inter-provincial vehicles since the results of intra-provincial vehicles are similar.

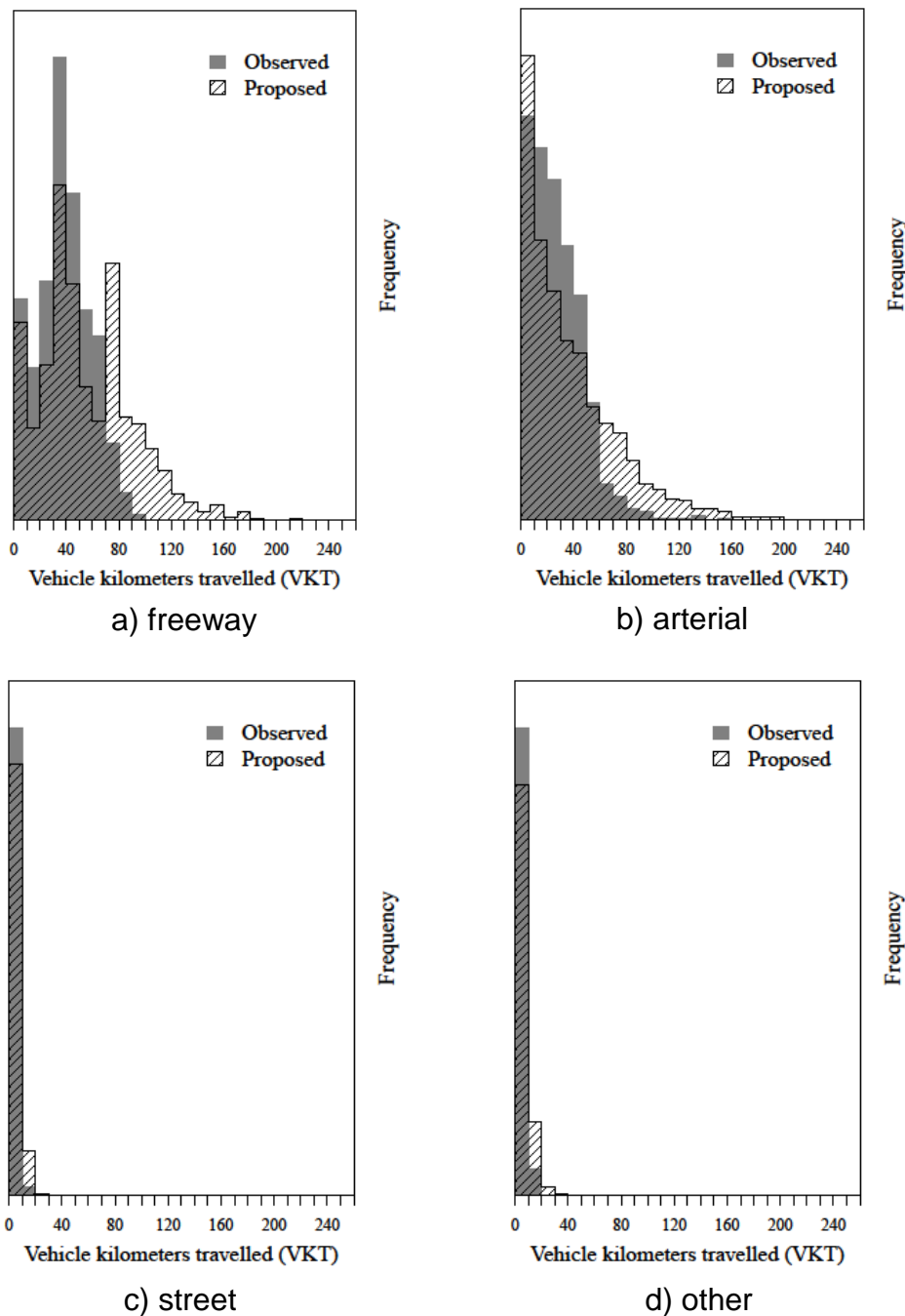


Figure 5 - Comparison of Vehicle Kilometers Travelled (VKT) for different road types.

We managed to obtain a favourable comparison between VKT from observed data and the synthetic population for the different road types. We do, however, slightly under- or over-estimate the VKT in some cases. This could be due to a sampling bias when the synthetic population was generated. It could also be the result of a small sample size since only 9703 chains were generated for the NMBM area.

4 CONCLUSION

In this second part of the two-part series we showed how to use the data, prepared in the first part, to run a Multi-Agent Transport Simulation (MATSim) model. We showed how one can combine both private and commercial vehicle populations and simulate them simultaneously over a 24-hour period and obtain rich full day results.

We validated the results with two analyses: comparing the simulated traffic counts with observed traffic counts; and comparing the Vehicle Kilometers Travelled (VKT) of the synthetic commercial vehicle population with observed VKT from the GPS data. Both analyses yielded favourable results.

Having a model that can accurately represent reality for a 24-hour period, allows one to evaluate various policy and infrastructure decisions. The current model can be used as a base case while a new scenario can be modelled and compared to it. One example might be the building of a new connecting highway or new freight corridor. It will entail the addition of the new sections of road to the network and rerunning the simulation model to obtain new results. Various analyses can be done such as diversion from the main roads or change in travel time. Another scenario might be the introduction of toll or toll subsidies and the impact thereof.

REFERENCES

Charypar, C, and Nagel, K, 2005. Generating complete all-day activity plans with genetic algorithms. *Transportation*, 32, pp. 369-397.

Hart, PE, Nilsson, NJ, and Raphael, B, 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics SSC4*, 4(2), pp. 100-107.

MATSim Development Team, 2012. Multi-Agent Transport Simulation (MATSim) website. [online] Available at www.matsim.org [Accessed January 2013].

Van Heerden, Q, and Joubert, JW, 2012. Commercial vehicle behaviour: analysing GPS records. In: 42nd Computers and Industrial Engineering (CIE) Conference, Paper 99.