Scaling traffic flow properties of MATSim to correctly cooperate for population downsampling

Theresa-Maria Mersini and Theresa Ziemke
Transport Systems Planning and Transport Telematics, Technische Universität Berlin, Germany
mersini@vsp.tu-berlin.de, tziemke@vsp.tu-berlin.de

***Abstract*—To increase computational efficiency, large-scale real-world transport scenarios in MATSim are usually run using a sample of the population of a given study region. This requires altering traffic flow properties to maintain congestion patterns similar to a simulation of the entire population. This work systematically investigates the effect of different traffic flow scaling parameters and their relation to the population sampling rate on the results in MATSim. The aim is to derive clear instructions for MATSim users on how to set traffic flow properties for downsampled scenarios appropriately.**

***Index Terms*—agent-based transport simulation, population downsampling, traffic flow scaling, MATSim**

I. Introduction

Agent-based transport models have become essential tools for analyzing and optimizing large-scale transport systems. In the multi-agent transport simulation MATSim [2, 5] individual travelers (agents) iteratively optimize their daily travel plans based on a co-evolutionary approach. Large-scale MATSim scenarios are usually run with a (down)sampled population (e.g., a 10% sample size) to reduce computational complexity and run times. To still model congestion propagation and travelers' decisions realistically, traffic flow properties need to be scaled accordingly. This work explores the scaling of relevant traffic flow parameters in MATSim to enhance the reliability of population downsampling. A full-scale (100%) car traffic scenario of the Lausitz region in Germany serves as a reference [4]. Multiple downsampled scenarios are generated using different sample shares and random seeds. These are simulated with varying traffic flow parameters, specifically flow capacity factor, storage capacity factor, and stuck time. The results are compared to the 100% scenario in terms of runtime, score, travel time, and number of stuck vehicles. Further analyses, including travel-time deviations and distances, and the influence of random numbers on the results, are planned for inclusion.

II. Background and related work

In MATSim, the road network is represented by a directed graph. Each link is equipped with a free-flow travel time, a flow capacity and a storage capacity. The flow capacity determines how many vehicles can leave the link per time unit, the storage capacity determines how many vehicles fit on the link spatially. MATSim's flow model is based on a deterministic queuing model. It handles each link as a spatial first-in-first-out queue. The link travel time is the sum of the free-flow travel time plus the waiting time to leave the queue. A vehicle at the head of a queue is allowed to leave it when its earliest exit time (determined by the free-flow travel time) has passed, the (accumulated) flow capacity of the link is sufficient, and, the storage capacity of the next link is not exceeded. To avoid gridlock situations and ensure a minimal movement even in very congested situations, there is an additional stuck time parameter in MATSim, which determines after how many seconds of non-movement a vehicle that fulfills all above conditions except the storage capacity constraint is allowed to move to the next link.

MATSim incorporates stochasticity at various points, especially in the co-evolutionary process, but also in the traffic flow simulation. A configurable global random seed controls these stochastic elements, ensuring reproducibility.

When downsampling a population, there are two main ways to scale traffic flow properties. One approach is to scale down the infrastructure, reducing road capacities to match the smaller population. Alternatively, vehicle sizes can be scaled up, such that one vehicle represents multiple vehicles, to maintain realistic congestion levels. In this work, we examine the first approach, which is the most common in MATSim scenarios.

Former recommendations suggested scaling storage capacity less than flow capacity by setting

$$\texttt{flowCapacityFactor} = \text{sample share}$$
$$\texttt{storageCapacityFactor} = \text{sample share}^{\alpha},$$

with $\alpha = 0.75$ to prevent breakdowns in small samples [6]. This is in line with [8, 10], who found that for small samples, scaling capacities less than proportionally helps to achieve realistic travel times. However, more recently, flow and storage capacities are usually scaled equally again, i.e., with $\alpha = 1$ [see, e.g., 9]. This aligns with [3], who found no substantial differences in average travel times for a given MATSim scenario when comparing different sample shares using both α values. In contrast, [1] again recommend using $\alpha = 0.75$. However, their study primarily explores how small the sample share can be chosen while still producing robust results, without explicitly analyzing the effect of the α value itself. To derive a clear recommendation how to set the α value to obtain realistic simulation results with downsampled populations, this study will further investigate its effect.

Moreover, initial studies suggest that the stuck time parameter should also be scaled with the sample size – using larger values for smaller samples, as each vehicle represents more agents in that case [7]. So far, stuck time has typically been used as a fixed value across sample sizes in MATSim scenarios. The present work aims to provide a first systematic investigation of this aspect.

III. Method

The Lausitz scenario used as a test case in this study is one of the few real-world MATSim scenarios available as a full-scale (100%) model[1] that is still computationally feasible. It covers both sparsely populated rural areas and dense urban regions, and consists of approximately 470k links and 260k nodes. In total, 1,195,422 agents are included, 13% of which represent freight traffic.

To reduce computational effort and isolate downsampling effects without mode switching, this study deliberately restricts the simulation to non-freight agents with at least one car leg per day. This results in a filtered base population of 305,200 agents, which will be referred to as the 100% benchmark in the following, and from which the sample shares

$$0.01, 0.05, 0.1, 0.25, 0.5, 1.0$$

will be studied.

The lower the sample share is, the greater is the influence of random variation on simulation outcomes, which increases the need for repeated runs [1]. In the given setup, there are two distinct sources of randomness: (1) the selection of agents for a given sample share, and (2) the random number (seed) used during the simulation itself. To better understand the impact of both sources of randomness, ten differently sampled populations are generated for each of the smaller sample shares (0.01 to 0.1). In addition, one sample for both the 0.01 and 0.05 levels is simulated using ten different random seeds.

All simulation runs described above use $\alpha = 1$ to scale storage capacity. To assess the influence of α, one additional run per sample is conducted with $\alpha = 0.75$. Furthermore, another set of runs scales the stuck time inversely with the sample share, instead of keeping it fixed as commonly done in MATSim. This variation is used to examine whether such scaling further improves the results of downsampled scenarios (as described in section II).

All simulations are run with 500 iterations of MATSim's co-evolutionary approach to ensure stable scores and outcomes. Agents can change their plan from their choice set, change routes, or reallocate time, whereas mode choice is disabled.

The results of all runs are compared in terms of runtime, average travel time, travel time distribution, MATSim scores, average travel distances, and the number of times a vehicle is forced onto the next link due to a stuck time violation. While the first measures are inherently comparable across sample shares, the number of stuck time violations is scaled by dividing it by the respective sample share.

[1] In this study, version v2024.2 of the Lausitz MATSim scenario [4] is used with MATSim version 3765 on the high performance cluster of TU Berlin.

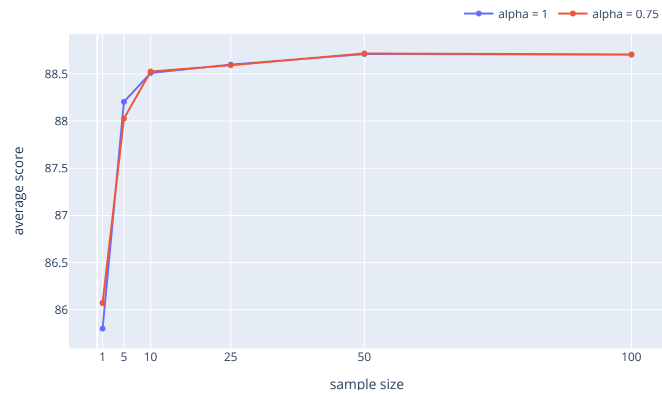

Figure 1. Average score of the experienced plans in the last iteration for different sample sizes. As in all following figures, blue dots show equal scaling of flow and storage capacities ($\alpha = 1$), red dots reduced storage scaling ($\alpha = 0.75$). Values for smaller sample sizes are averaged over multiple random seeds; the 25%, 50%, and 100% values are based on a single run each.

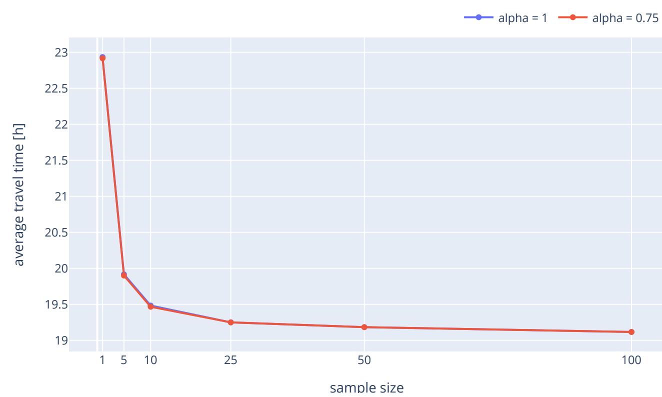

Figure 2. Average travel time in the last iteration for different sample sizes.

IV. Preliminary Results

While not all simulation runs have been completed, the following presents first results for selected measures and run configurations. These allow for a preliminary assessment of downsampling effects and provide a first indication of the expected analytical direction of the full study.

Figure 1 shows the average scores experienced in the last iteration of all MATSim runs described in Section III with constant stuck time parameter. Values for smaller sample sizes are averaged over multiple random seed runs, while for the 25%, 50%, and 100% samples, they are based on a single run each. The scores are quite similar across the different scaling approaches. However, the scores consistently decrease as the sample size decreases and the 5% and, in particular, the 1% sample shows a clear deviation from the larger sample sizes.

A similar trend can be observed in Figure 2: average travel times for the 1% sample differ significantly from those of larger sample sizes. Overall, travel times decrease as the sample size increases, with no noticeable difference between the scaling approaches.

A completely different pattern emerges in Figure 3, where the two scaling approaches differ significantly. As the sample

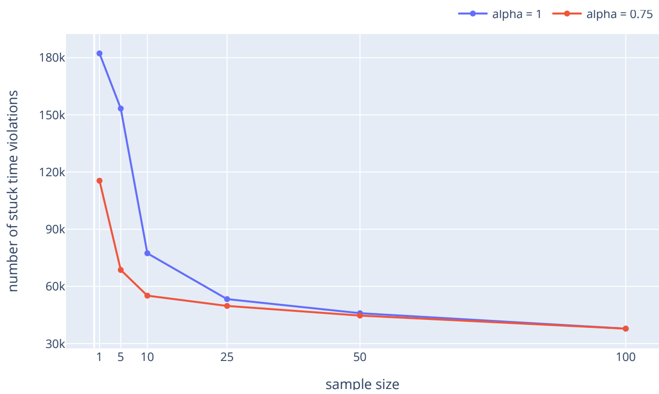

Figure 3. Number of stuck time violations in the last iteration for different sample sizes. Values have been scaled by the inverse sample share to ensure comparability with the full-scale scenario.

size increases, the number of stuck time violations decreases (fewer stuck time violations are generally desirable). In the 100% sample, there are about 40k violations, corresponding to almost 1/8 of the agents getting stuck once per day. In contrast, in the 1% sample, three times as many agents get stuck, and with equal scaling of flow and storage capacity, even more than four and a half times as many. In the latter case, this corresponds to as many stuck events as if half of the agents were stuck once per day.

V. Conclusion

Preliminary comparisons of scores and travel times suggest that even average statistics (such as score and travel time) become unreliable when sampling below 10%. A more detailed analysis would likely show an even stronger reaction to sample size reduction.

So far, the analysis of scores and travel times suggests that adjusting the scaling of storage capacity may not be necessary. However, the stuck time analysis indicates that the similarity in travel times across the scaling approaches is only due to the low (and unscaled) stuck time. Especially for the equal scaling of capacities, the number of stuck time violations increases rapidly with decreasing sample share. This raises the question whether the low stuck time is offsetting congestion effects in smaller sample sizes. We will investigate this further in the upcoming runs with scaled stuck time, as outlined in Section III.

In the next steps, we will also explore the influence of random numbers on the outcomes and analyze additional measures, in order to derive clear guidance on how to appropriately set traffic flow parameters for downsampled scenarios.

References

[1] G. Ben-Dor, E. Ben-Elia, and I. Benenson. “Population downscaling in multi-agent transportation simulations: A review and case study”. In: *Simulation Modelling Practice and Theory* 108 (2021), p. 102233.

[2] A. Horni, K. Nagel, and K. W. Axhausen, eds. *The Multi-Agent Transport Simulation MATSim*. Ubiquity, London, 2016. DOI: 10.5334/baw.

[3] C. Llorca and R. Moeckel. “Effects of scaling down the population for agent-based traffic simulations”. In: *Procedia Computer Science* 151 (2019), pp. 782–787. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.04.106.

[4] *MATSim Lausitz Scenario*. accessed 2025. URL: https://github.com/matsim-scenarios/matsim-lausitz.

[5] MATSim webpage. *MultiAgent Transport Simulation*. accessed 2025. URL: http://www.matsim.org.

[6] T. W. Nicolai. *Using MATSim as a travel model plug-in to UrbanSim*. VSP Working Paper 12-29. 2012.

[7] H. Triebke, M. Kromer, and P. Vortisch. “Bridging the Gap Between Mesoscopic Transport Planning and Microscopic Traffic Simulation: An Analytical and Numerical Analysis of Traffic Dynamics”. In: *Transportation Research Record* 2677.5 (2023), pp. 62–76. DOI: 10.1177/03611981221128284.

[8] D. Ziemke, A. Agarwal, and I. Kaddoura. “Entwicklung eines regionalen, agentenbasierten Verkehrssimulationsmodells zur Analyse von Mobilitätszenarien für die Region Ruhr”. In: *10. Wissenschaftsforum Mobilität : Mobility in Times of Change – Past, Present, Future*. Ed. by Heike Proff. Springer Gabler, Wiesbaden, 2019, pp. 383–410. DOI: 10.1007/978-3-658-26107-8_29.

[9] D. Ziemke, I. Kaddoura, and K. Nagel. “The MATSim Open Berlin Scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data”. In: *Procedia Computer Science* 151 (2019), pp. 870–877. DOI: 10.1016/j.procs.2019.04.120.

[10] D. Ziemke, K. Nagel, and C. Bhat. “Integrating CEMDAP and MATSim to increase the transferability of transport demand models”. In: *Transportation Research Record* 2493 (2015), pp. 117–125. DOI: 10.3141/2493-13.