Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 130 (2018) 900–907

Procedia Computer Science

www.elsevier.com/locate/procedia

The 7th International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications (ABMTrans 2018)

A first look at bridging discrete choice modeling and agent-based microsimulation in MATSim

Sebastian Hörl[a,*], Milos Balac[a], Kay W. Axhausen[a]

[a] *ETH Zurich, Institute for Transport Planning and Systems, Stefano-Franscini-Platz 5, 8094 Zürich, Switzerland*

Abstract

The agent-based transport simulation framework MATSim allows for the simulation of dynamic transport scenarios with agents, that adaptively make travel choices. Regarding mode choice, a heavily randomized process is used to date, which allows for very unrealistic mode decisions in the short run to arrive at consistent mode shares after a large number of iterations. The authors show that implementing a discrete mode choice model may drastically increase the convergence speed of the simulation, but point out that considerable future research is necessary to make travel decisions consistent and to back the process with a strong theoretical foundation.

Keywords: agent-based, simulation, mode choice, discrete choice, MATSim

1. Introduction

Large parts of the transport research community have their main interest in discrete choice modeling, i.e., predicting a traveler's choice for a particular mode, route or destination. Usually, those are based on the expected travel characteristics, such as travel time and cost, in combination with personal preferences and socio-demographic factors. The primary rationale behind the approach is that a statistical model, estimated from stated preference data (e.g., from a travel survey) or revealed preference data (e.g., GPS tracking) can predict the individual travel decisions of people. Ultimately, by modifying individual variables, such as the travel time, one can predict how the travel decisions of people would change. However, these changes in the traffic system are imposed as changes in the exogenous variables of the system and usually do not allow for the analysis of feedback effects.

Microsimulation frameworks such as MATSim[4] simulate individual travelers on a sufficiently small scale, but only with the precision that is needed to observe relevant effects. While MATSim could also be tagged as a mesoscopic framework since the smallest spatial units are road links, frameworks like SUMO[3] or Vissim[7] allow for detailed dynamics on a sub-link level, i.e., all vehicle movements are simulated explicitly. What many of those frameworks have in common is that agents, i.e., the simulated people or vehicles, need to adapt to the conditions of the transport system

[*] Corresponding author.
E-mail address: sebastian.hoerl@ivt.baug.ethz.ch

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2018.04.087

in place. The more people choose a certain route, the larger the travel times will become. Eventually, congestion will emerge. Hence, there is feedback from the agents' decisions, and there is an interaction with the available resources of the transport system. The idea is that agents are continuously allowed to make changes, but by doing so arrive at a global equilibrium, where every agent has made its momentarily best decision, such that no further adjustment is beneficial (see Stochastic User Equilibirum[6]). If there is a subsequent change (either in the behavioral model or in the transport system), agents will start again to make choices until they adapt to the new conditions.

Traditionally, in the case of MATSim, the microsimulation approach has not been easily accessible by the discrete choice modeling community and the MATSim community, to the best knowledge of the authors, did not explore the benefits that a combination of the two approaches may have in practice, although theoretical insights exist (chapters 47 - 49 in the MATSim book[4]). Recently, experiments have been run where MATSim serves as a in-the-loop microsimulation tool with an upper-level plan choice software[1].

The authors of the paper at hand want to explore whether incorporating an explicit discrete choice model into MATSim would improve runtime and convergence of the simulation framework and contribute to the quality of the output results. Furthermore, by enabling discrete choice modellers to implement their models in a straight-forward way into MATSim, the framework could contribute to their research by making analyses of feedback effects possible. In this report, the authors want to provide several options of how a discrete choice model may be incorporated into MATSim explicitly and which implications, restrictions and benefits come along with it. For demonstration, the authors use a rather simple trip-based multinomial logit (MNL) mode choice model and restrict, for now, their analysis to this scope.

2. Mode Choice in MATSim

In order to understand how MATSim works, some nomenclature must be introduced: A *scenario* consists of all data that is available for a certain study, i.e., it is the *network*, consisting of *links* and *nodes*, but also the travel demand of a large set of agents (*persons*), which is called the *population*. Initially, each of these agents has at least one daily *plan*, which consists of all the *activities* and connecting *trips*, that an agent wants to perform during the day.

MATSim uses a co-evolutionary algorithm to reach an equilibrium state of the system[6]. An iteration in MATSim consists of three steps: First, the selected plans of all agents are simulated by executing the information that is saved in them. This stage is backed by a queue-simulation model, which moves vehicles from link to link in the network. Once the capacity or space limitations of a link are reached, traffic slows down and congestion builds up on the upstream links. This way, the choices from the agents' plans directly affect the simulation travel times. Clearly, since this may introduce delays, the outcome of a plan is different than its initial version. In order to compare how well an initial plan worked out, the second phase of a MATSim iteration is the *scoring*. In this step, the observed plan is translated into a utility value (*score*) based on a predefined utility function (e.g., performing an activity is increasing utility, while driving a car or having to wait for a bus is decreasing utility). The final score is assigned to the selected plan of the agent. Over time, agents can collect such plans in their *memory* which has a predefined size of N past plans. The last stage of the iterative process is *replanning*: For each agent, a *replanning strategy* is chosen. This may be a *selection strategy* (i.e., selecting from an agent's memory a plan based on its utility) or an *innovation strategy*, where a certain plan of an agent is duplicated and modified in a specific way (e.g., choosing a different departure time for a trip). Finally, if this leads to a state where an agent has more than N plans in memory, a removal procedure is applied, that chooses a plan to be deleted from the memory. In the next iteration, the selected/modified plans will be executed, scored, replanned, and so on.

2.1. *MATSim as a plan choice algorithm?*

To find parallels between MATSim and discrete choice modeling, one needs to examine how agents choose their plans in the replanning phase. Given the utilities U_i of all the available plans $i \in \{1, ..., N\}$ of an agent, a multinomial selection can be performed:

$$\mathbb{P}(j) = \frac{\exp(U_j)}{\sum_i \exp(U_i)} \tag{1}$$

According to the probability in Formula 1 one plan is selected. Such a selection process is implemented in the `SelectExpBeta` strategy in MATSim. However, as noted in the MATSim book[6], a different strategy has been chosen as the default one for the framework:

$$\begin{aligned} k &\sim \text{Uniform}(1, N) \\ \Delta &= U_{selected} - U_k \\ q(k) &= \xi \exp\left(-\frac{\beta\Delta}{2}\right) \end{aligned} \tag{2}$$

First, a plan (k) is sampled at random from the set of available plans for an agent. Then, the difference in utility (Δ) between the sampled one and the currently selected plan of the agent is computed. Finally, a transition probability ($q(k)$) is constructed that determines whether the selection algorithm should switch to the sampled plan. If $U_k > U_{selected}$, the exponential term becomes larger than one, whereas it becomes smaller than one if the currently selected plan has the better score. The parameter β determines the "softness" of the transition probability and is usually kept at $\beta = 1$ in MATSim. The other parameter, ξ defines the transition probability for $U_{selected} = U_k$ and, in general, should scale the whole expression such that it is smaller than one to behave like a proper probability. Unfortunately, to date, this parameter is hardcoded and cannot be influenced by the framework user although it can influence the selection process in the short run, especially if innovation strategies are kept active. Currently, it is fixed to $\xi = 0.01$.

In the limit of a large number of selections, this default strategy (`ChangeExpBeta`) is supposed to converge against the same stable distribution as the first algorithm[6]. Hence, the default algorithm in MATSim can be described as a selection of plans:

$$\max_i \{U_1 + \varepsilon_1, ..., U_N + \varepsilon_N\} \quad \varepsilon_i \sim \text{Gumbel} \tag{3}$$

Unfortunately, this is only true if the available N plans in the memory are kept constant[6]. As soon as plans are duplicated and modified, some need to be removed in order to maintain a memory size of N, hence a major component of the overall behavior is the plan removal strategy. The default approach in MATSim for plan removal is `WorstPlanSelectionForRemoval`, where one determines the plan with the worst score for each agent that has more than N plans in memory and deletes it.

While the convergence behavior of MATSim is understood quite well for the case where only plan selection is active (i.e. no plans are created or removed), it is not clear how the default removal strategy influences the overall convergence behavior of the simulation. A theoretical framework to understand the process has been set up[2], but not put into practice. For that reason it is difficult to compare the behavior of the classical mode choice in MATSim with our novel discrete choice model implementation. Hence, for now, we will focus on a rather descriptive analysis of both alternatives.

2.2. Subtour Mode Choice

The *Subtour Mode Choice* (SMC) strategy is the default innovation module in MATSim for the modification of transport modes in agents' plans:

- First, a plan from the agent's memory is chosen *uniformly* at random and set as the selected plan of the agent
- An agent's plan is deconstructed into *subtours*, i.e., hierarchical, possibly overlapping sequences of trips with the same start and end location
- SMC chooses one of these subtours and resets the transport modes of all of the included trips to a randomly chosen value. However, the possible modes are limited by continuity constraints: A car can only be used where it has been moved to before.
- In case more information is required (such as a route for travels by car or public transport), the plan is passed down to the `ReRoute` replanning algorithm, which computes a new travel route through the transport network. For the "car" mode, additional variability is introduced by adding slight noise to the edge costs in the network graphs.

A couple of problems are worth noting here:

- The random plan selection seems to be strongly at odds with the advanced selection process that is applied for the pure plan selection. In all iterations where SMC is applied (usually with a probability of 1% to 5%) a plan is selected without any prior knowledge. So far, this issue does not seem to have been stressed in the MATSim community.
- By definition, the deconstruction into subtours does not take into account all decisions an agent could make. The lunch break example, where an agent leaves work to go to a restaurant and comes back half an hour later, is bound to be performed using one single mode of transport, i.e., it is not possible to walk there and return by public transport (since the whole break is one subtour).
- Since modes are sampled at random in SMC, the chance can be high that very unlikely travel plans are created, e.g., an agent walking for 50km instead of taking a train.

3. Discrete mode choice in MATSim

The statistical models that the authors want to examine are trip-based mode choice models. What is known for one choice are usually trip characteristics for each mode, such as travel time and monetary travel costs, as well as additional information by mode, such as the number of transfers or the frequency for public transit alternatives. From these characteristics, a probability for each of the available mode alternatives is constructed. The "availability" of an alternative is either determined by person-based attributes (car ownership, car license, etc.) or by continuity constraints (a car can only be used at a location to which it has been moved before). Especially the latter ones are not considered in most trip-based discrete choice models but need to be taken care of in MATSim.

The proposed replanning algorithm for MATSim is structured as follows: First, a plan is chosen. Here, we follow the usual practice in MATSim to randomly select the plan, while acknowledging that this may not be the ideal approach. Second, the set of all possible chains of modes for the agent's plan is constructed. Third, one of the derived chains is selected and the corresponding modes are applied to the plan. These two stages will be explained in detail in the following sections. Finally, rerouting is performed where necessary.

3.1. Chain Construction

Given the daily plan of an agent, it is known which trips should be performed with origin, destination and departure time. If one iterates over all possible combinations of modes on these trips, one arrives at a set of available mode chains C of size M^N, where M is the number of possible modes and N is the number of trips. The set of feasible chains $C_f \subset C$ can be smaller because some of the available chains can be filtered out due to continuity constraints. An exemplary set of chains for a plan with three trips could then look as follows:

$$C_f = \{(\text{car}, \text{car}, \text{car}), (\text{pt}, \text{pt}, \text{pt}), (\text{pt}, \text{walk}, \text{pt}), ...\} \tag{4}$$

We denote the chains in the set as $c_1, ..., c_K$ and each trip mode in chain k as $t_{k,i}$. The mode choice problem for this agent can then be generalized to choosing one of the alternative chains $c_1, ..., c_K \in C_f$.

3.2. Chain Selection

Since "standard" mode choice models operate on a per-trip basis (although exceptions exist), the authors propose three schemes of incorporating an out-of-the-box discrete mode choice model into MATSim. Given the set of feasible chains as described above, we propose a scheme that is based on a best-response assignment, and two sampling approaches, one based on the total utility of chains and an alternative one based on individual probabilities of trips.

We assume that from the mode choice model, we can get the utility $u_{k,i}(\theta_{k,i})$ for each trip alternative $t_{k,i}$ and the probability of a certain alternative to occur per trip: $\pi_{k,i}(\theta_{k,i})$. Here, $\theta_{k,i}$ are the expected trip characteristics that need to be known, estimated or assumed beforehand. They are the main difference to the SMC approach: While there, choices are made completely at random, here prior knowledge through $\theta_{k,i}$ is introduced, which may include the expected travel time or travel costs. Therefore, there is a significant computational overhead compared to SMC, because each mode alternative for each trip per agent needs to be estimated and routed.

- **A1) Best-response selection.** We introduce the total chain utility $\tilde{u}_k = \sum_i u_{k,i}(\theta_{k,i})$ and select a chain c_k by choosing $k = \arg\max_k\{\tilde{u}_1, ..., \tilde{u}_K\}$. Hence, the chain with the highest total chain utility is chosen. Naturally, this will lead to an oversampling of the best alternative, since no randomness is introduced. The strategy is especially useful since it provides us with an upper bound of what utility can be achieved if the travel characteristics $\theta_{k,i}$ are known deterministically.

- **A2) Total chain utility sampling.** For each trip alternative in each chain, sample $\varepsilon_{k,i} \sim$ Gumbel. This assumes that for each trip there is a certain amount of unobserved randomness, that influences the choice process. Combined with the random samples one obtains $\tilde{z}_k = \sum_i u_{k,i} + \varepsilon_{k,i}$ and selects $k = \arg\max_k\{\tilde{z}_1, ..., \tilde{z}_K\}$. Hence, we introduce a specific amount of randomness to the utility of each mode alternative.

- **A3) Naive chain sampling.** Here, we assume (similar to the discrete choice models), that the mode decision for each trip is made independently of all others. We then define the total chain weight as $\tilde{w}_k = \prod_i \pi_{k,i}$. Hence, $\tilde{w}_k$ describes the likelihood of observing all modes in a chain jointly. Note that the continuity constraint is already fulfilled by construction of the chains, while in a trip-based discrete mode choice model those constraints are usually not taken into account. Finally, given the $\tilde{w}_k$ one can construct chain probabilities as $\tilde{\pi}_k = \tilde{w}_k / \left(\sum_k' \tilde{w}_{k'}\right)$ and sample k from a categorical distribution: $k \sim \text{Cat}(\tilde{\pi}_1, ..., \tilde{\pi}_K)$. An open question is how to define the $\tilde{\pi}_{k,i}$ in this case exactly. On the one hand, one can say that *all* alternatives are taken into account for the choice (except if it is excluded by person-specific constraints), or we can say that the momentarily available modes (governed by the continuity constraints) are considered. Here, for simplicity and similarity to the model definition, we use the former version.

It needs to be noted that these approaches are somewhat "constructed" to make a trip-based discrete choice model fit into the MATSim framework. For instance, the information from the continuity constraints should already be encoded in the model parameters. Further research will need to verify how far the attempts mentioned above make sense from a theoretical viewpoint and what would be a better approach to apply. Especially one can expect that the given chains are highly correlated[8] and nested logit approaches in combinaton with tour-based discrete choice models should be considered in the future.

An interesting idea that could be investigated is "frozen randomness" as has already been applied in the MATSim ecosystem for location choice[5]. There, one would not repetitively sample errors for the choice process, but do this once per trip and alternative and keep those error terms constant.

4. Simulation Example

As an example, we use a mode choice model of the form:

$$U_{i,m}(\theta_i) = \alpha_m + \beta_m \cdot \theta^{tt}_{i,m} + \beta_{cost} \cdot \theta^{cost}_{i,m} \qquad \forall m \in \{\text{car, pt, bike, walk}\} \tag{5}$$

with $\theta^{tt}_{i,m}$ as the expected travel time with mode m and $\theta^{cost}_{i,m}$ as the expected monetary cost per trip. The β_m represent the marginal utility of travel time per mode and β_{cost} is the marginal utility of costs. The constants α_m describe the unexplained utility of the choice model per mode.

These utilities are plugged into a standard multinomial logit model, such that we get the following expression:

$$\pi_{k,i}(\theta_{k,i}) = \frac{\exp\left(U_{i,t_{k,i}}(\theta_{k,i})\right)}{\sum_m \exp\left(U_{i,m}(\theta_{k,i})\right)} \tag{6}$$

The three presented strategies have been implemented in our mode choice framework, which mainly consists of convenient classes in Java to define choice alternatives and the selection dynamics. Furthermore, we provide an easy to use API to generate the continuity-constrained trip chain sets C_f.

The experiments conducted with our framework do not perform any selection based on scores. In 10% of the iterations, the mode choice is performed according to the respective strategy, while in 90% of iterations no further change is applied to the respective agent. While MATSim can still score the experienced plans of the agents these outcome scores do not influence the choice behavior in any way. Please also not that, for these experiments, we do not score the duration of activities.

For comparison, we run two simulations where the scoring *is* influencing the agents' choices through the default selection and removal strategies. We allow each agent to accumulate at maximum five plans for selection in memory and run two versions:

- **R1) Subtour Mode Choice** is enabled (instead of the new strategies) to generate new plans.
- **R2) Chain Sampling**. Here, we use the chain sampling component (Section 3.1) of our framework instead of SMC, but always sample one chain at random from C_f. The choice dynamics are then performed by MATSim through scoring.

The three mode choice strategies, as well as the two reference strategies are run on a 1% MATSim scenario of the city of Zurich. For our tests, the modes of all trips of all agents have been reset to "walking" in the input plans.

4.1. Teleportation-based example

First, we run simulations where all modes are simulated through teleportation. There are no network and congestion dynamics, but the travel times and costs for each mode are obtained by computing the crowfly distance of each trip and by applying calibrated factors to obtain travel times and costs.

Figure 1 shows the population average of the score of the best plan of each agent (dashed). The best-response strategy (A1) establishes an upper bound of the utility because the option with the highest utility/score is chosen. Furthermore, one can see that the reference cases (R1, R2) converge against that value. Therefore, one can say that MATSim is in fact performing optimization of the plan of each agent until each of them finds their best plan. This "best plan" exists and is equal to what (A1) finds because utilities are distinctly predefined through the crowfly distances of the trips. One can see that (R2) converges slower against the upper bound because a larger choice set ("lunch break example") is examined.

Regarding "experienced" scores (solid), i.e., what is executed by the agents, one can see that for R1 and R2 the score is lower than the maximum and also seems to stabilize below it. This is expected: Since R1 and R2 can produce plans that are far off the optimum for 10% of the population in each iteration one must expect a strong offset.

In the two randomized choice strategies (A2, A3) different dynamics can be observed: Plans with low expected utilities are performed very seldom, while plans with high utilities are generated more frequently. The expected utility and how it relates to the introduced randomness then governs the average utility of the plans that are sampled according to the choice model.

One can also see that there is no direct correspondence in mode shares. In Figure 1 (right) one can see that A2 has the lowest share in car trips, while R1/SMC yields the highest one after the best-response selection. So while from the comparison of scores, one may get the impression that the total chain utility approach (A2) behaves most similar to the calibrated SMC, regarding mode share it does not.

When comparing the three choice strategies with the scoring-based reference cases, one can see that the scores stabilize much earlier, because only "smart" choices are being made. These are controlled by an estimated choice behavior which purposefully does *not* always yield the alternative with the best utility.

4.2. Queue simulation-based example

In a second step, all simulations have been repeated, but with "car" being simulated on the network. From iteration to iteration changes in travel times emerge, which make the predictions of the mode choice less accurate.

Figure 2 shows the results of the network simulations. A1 now uses inaccurate (predicted average) travel time estimates to perform the mode choice. Hence, it does not constitute a clean upper bound for the executed and maximum score anymore and produces samples that follow the estimated choice model.

One can see that, again, the choice-based simulations converge substantially faster to a stable state than the scoring-based ones. Even though there is the need to route each alternative per trip and choice repeatedly, the runtime for one iteration is similar because the routing is heavily parallelized (on a per-agent level). One could even think about

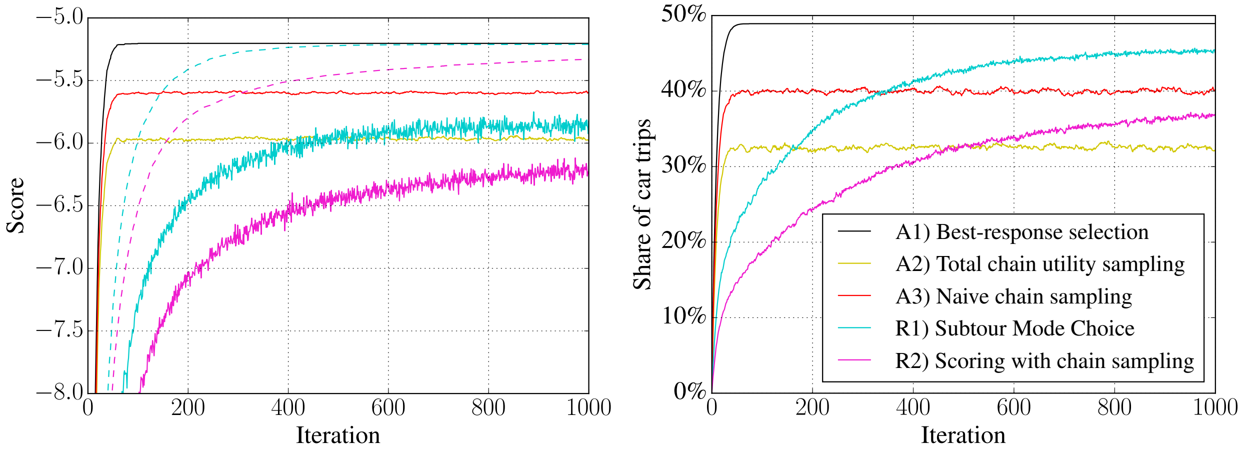

Fig. 1: Teleportation-based simulations. Left: Comparison of the average over all agents' executed plan score (solid) and average over the maximum score of all agents (dashed) for each strategy. Right: Comparison of share of car trips for each strategy.

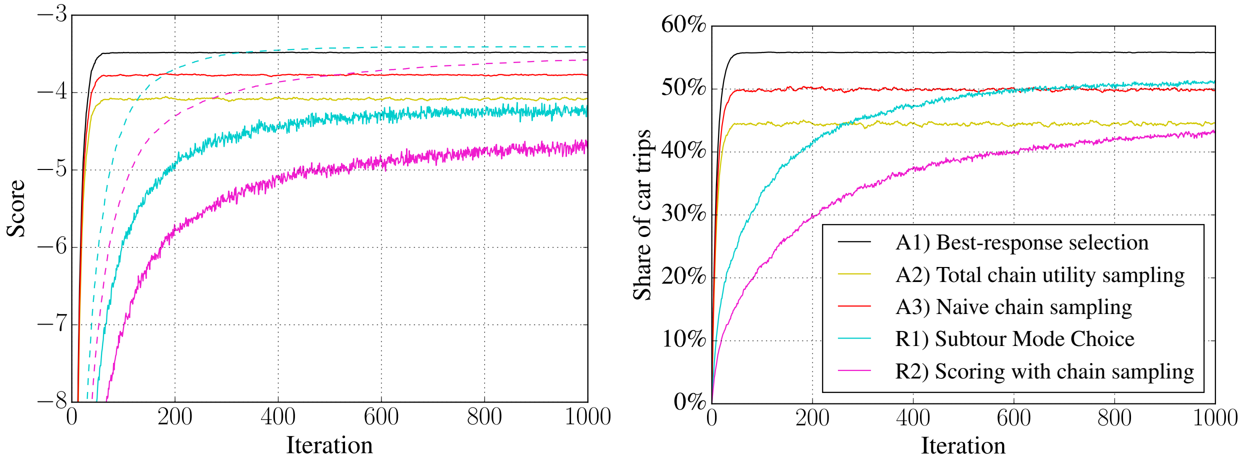

Fig. 2: Network-based simulations. Left: Comparison of the average over all agents' executed plan score (solid) and average over the maximum score of all agents (dashed) for each strategy. Right: Comparison of share of car trips for each strategy.

performing the mode choice parallel to the whole MATSim loop, which could possibly render the overhead in runtime negligible.

5. Discussion & Outlook

With the experiments at hand, the authors were able to dare a first look into how a discrete mode choice model may be integrated into MATSim. While a large part of the work has been setting up the programming infrastructure and providing a first conceptual implementation numerous questions appeared that need to be answered in future research.

From our initial considerations and literature research, we conclude that it is yet not clear if and how the global convergence of the MATSim selection/removal procedure can be formalized, except if innovation strategies are turned off. Hence, it is also not clear whether, in that case, the MATSim loop should be tagged a "plan choice" algorithm, but rather a global "plan optimization" algorithm. Although a utility-based sampling is taking place the main driving force of plan selection seems to be the removal strategy that deletes poorly performing plans and biases the agent's memory toward high scores. This, however, seems to be at odds with the notion that a choice model *should* yield choices with an expected utility rather than the best utility, and finally impedes the comparison of MATSim's standard choice process with ours. Additional research needs to be put into how to increase the comparability of both approaches.

Given that MATSim's selection process can be improved[2], a sensible combination of MATSim's scoring and our mode choice implementation could resemble a kind of importance sampling for mode decisions and should allow for the smooth integration of existing innovation strategies and additional scoring such as for the duration of activities. In terms of convergence, by introducing "smart" mode choices, it is likely that the runtime of MATSim simulations can be reduced considerably, because stable choice distributions are reached in fewer iterations.

Another question that remains is how to "feed" discrete choice models with information. Right now we used average values of travel times, but one could also think of providing distributions of travel characteristics, which would make a mixture-of-logit formulation[2] more appropriate. By comparing the error distributions that are observed while estimating this kind of models and by examining the outcomes of a MATSim iteration, one could further develop metrics for the convergence of a simulation or the ability to replicate reference data.

One main point for future research will be to use our framework in more practical use cases where it is possible to compare the simulated mode shares and travel characteristics carefully with real-world data. Furthermore, we will need to position our approach more thoroughly into the theoretical framework that has been established so far and potentially extend the existing insights.

The paper at hand explicitly looked into trip-based mode choice models, although different choice models may be estimated. Especially tour-based versions may be easier to integrate into MATSim, since continuity constraints translate more naturally from one model to the other.

6. Conclusion

The authors showed that the integration of an explicit choice model into MATSim can dramatically improve the convergence speed of the simulation. The main reason for that is that additional information is incorporated into the choice process rather than relying on purely random mode choices. However, to ensure compatibility with the remaining parts of the MATSim framework on the one hand, and to ensure consistency with the original mode choice model on the other hand, the authors pose numerous open issues and find a need for considerable future research to address these questions.

References

1. Canella, O., Flötteröd, G., and Thelin, J. IHOP3 – Economically consistent simulation of travel behavior with MATSim. *Technical Report* (2017), KTH, Sweco, WSP.
2. Flötteröd, G., and Kickhöfer, B. Choice models in matsim. In *The Multi-Agent Transport Simulation MATSim* (2016), A. Horni, K. Nagel, and K. Axhausen, Eds., Ubiquity Press, London, pp. 337 – 346.
3. German Aerospace Center, Institute of Transportation systems. Simulation of Urban MObility. *http://sumo.dlr.de*. Accessed: 20 January 2018.
4. Horni, A., Nagel, K., and Axhausen, K., Eds. *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, London, 2016.
5. Horni, A., Scott, D., Balmer, M., and Axhausen, K. Location choice modeling for shopping and leisure activities with matsim: combining microsimulation and time geography. *Transportation Research Record: Journal of the Transportation Research Board*, 2135 (2009), 87–95.
6. Nagel, K., and Flötteröd, G. Agent-based traffic assignment. In *The Multi-Agent Transport Simulation MATSim* (2016), A. Horni, K. Nagel, and K. Axhausen, Eds., Ubiquity Press, London, pp. 315 – 326.
7. PTV. PTV Vissim. *http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/*. Accessed: 20 January 2018.
8. Train, K. *Discrete Choice Methods with Simulation*. Cambridge University Press, New York, 2009.