



Integrating discrete choice models with MATSim scoring

Sebastian Hörl

► To cite this version:

Sebastian Hörl. Integrating discrete choice models with MATSim scoring. 9th International Workshop on Agent-based Mobility, Traffic and Transportation Models,(ABMTRANS), Mar 2021, Warsaw, Poland. pp.704-711, 10.1016/j.procs.2021.03.088 . hal-03099994

HAL Id: hal-03099994

<https://hal.science/hal-03099994>

Submitted on 6 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Integrating discrete choice models with MATSim scoring

Sebastian Hörl^{a,*}

^a*Institut de Recherche Technologique SystemX, Palaiseau, France*

Abstract

Agent-based transport simulations rely on realistic representation of agents' movements in the transport system, but also on the simulation of choice making processes that resemble reality well. The MATSim framework uses the scoring-based co-evolutionary algorithm to achieve this, based on the principle of utility maximization. Discrete choice models are another common tool in transport planning which rely on the same principle. Through a toy example, the paper at hand shows that the direct use of discrete choice parameters in MATSim is not a technically sound approach, but that a consistent integration can be achieved by generating deterministic pseudo-errors based on cryptographic hash functions. Further caveats with integrating choice parameters in MATSim scoring are discussed.

Keywords: MATSim, mode choice, discrete choice, hash function, scoring

1. Introduction

The reliability of a behavioural transport simulation depends strongly on two factors: How well and to what detail are the movements of persons or goods simulated, and how close do simulated decisions replicate the choice process of real persons or operators.

5 The MATSim framework (Horni et al., 2016) simulates travellers' choices using an evolutionary algorithm that aims at optimizing daily activity schedules of agents. While the mobility simulation part has been widely extended over the past years, the simulation of agent choices is still mostly based on the principles that have been proposed initially. A common tool to analyse peoples' mobility choices in the transportation research community are discrete choice models. Recently, an extension to MATSim has
10 been proposed which implements such discrete choice models into the modeling loop. As described in (Hörl et al., 2019; Hörl et al., 2018) this comes with a number of advantages, but also disadvantages, as the approach does not fit neatly into the traditional co-evolutionary planning structure of MATSim.

While previous research has looked at how MATSim *fits into* the framework of discrete choice Flötteröd and Kickhöfer (2016), the paper at hand aims at moving towards a closer integration of the co-evolutionary approach of MATSim and discrete choice modeling by investigating how the framework can
15 be *adapted to replicate* the common multinomial logit model for mode choices. In the past, estimated model parameters from such models have been used in the context of MATSim (e.g., Vosooghi et al. (2019)) but always needed adjustment to account for the opportunity cost of time considered in MATSim's approach Nagel et al. (2016) and further calibration. Yet, there are fundamental differences in the
20 simulation dynamics between discrete choice models and the scoring-based approach in MATSim which have not been investigated in detail before. Therefore, the goals of the present workshop contribution are:

- To provide a closer look at the differences between discrete choice models and the scoring-based approach of MATSim
- To propose a method to integrate both concepts in the context of trip-based mode choice models
- To emphasize caveats and difficulties when working with discrete choice models in MATSim

*Corresponding author

E-mail address: sebastian.horl@irt-systemx.fr (Sebastian Hörl).

2. Discrete choice process

A comprehensive and well-written introduction to discrete choice models is given in Train (2009). Here, only the necessary basics to understand the difference between a standard multinomial logit model and MATSim will be presented.

Generally, the multinomial logit model is based on the idea of utility maximization. A decision-maker has $K \in \mathbb{N}$ different alternatives (e.g. taking the car, the bus, or the bike) and each alternative has a number of attributes x (e.g. the trip duration, the cost, ...). In a multinomial logit model, the utility $v_k \in \mathbb{R}$ of a mode is usually expressed as a linear combination of the attributes, including an alternative-specific constant (ASC), e.g.:

$$\begin{aligned} v_{car} &= \beta_{car,ASC} + \beta_{travelTime,car} \cdot x_{travelTime,car} + \beta_{cost} \cdot x_{cost,car} + \dots \\ v_{bike} &= \beta_{bike,ASC} + \beta_{travelTime,bike} \cdot x_{travelTime,bike} + \beta_{slope,bike} \cdot x_{slope,bike} + \dots \end{aligned} \quad (1)$$

All $\beta \in \mathbb{R}$ are model parameters that quantify the influence of the attributes. The assumption is that the utility-maximizing decision maker will choose the alternative with the highest utility. Clearly, the model is a rather simple representation of the decision-making process real people follow. Given a data set, for instance, from a survey, it is never the case that one set of utility functions properly predicts the choice of every respondent. First, because the model may not be complex and detailed enough, second, because people often do not behave rationally and the *homo oeconomicus* in itself is a rather strong assumption.

A multinomial logit model therefore allows variability in the choices, acknowledging that there are taste factors and unobserved preferences leading to errors in the model. The utilities are therefore defined as random variables $U_k = v_k + \sigma \epsilon_{i,k}$ with ϵ being a i.i.d. random error term for respondent i and choice k and $\sigma \in \mathbb{R}^+$ a scaling factor for the error. Given that U_k are random variables, one can now ask for the probability that a certain alternative will be chosen (i.e. that it is the best)

$$\mathbf{P}[U_k \geq U_1 \wedge \dots \wedge U_k \geq U_K] \quad (2)$$

It has been shown McFadden (1974) that if for ϵ an extreme value distribution is chosen (for instance Gumbel, which approximates a normal distribution), the probability can be written in closed form through which the β parameters can be estimated using standard optimization approaches:

$$\mathbf{P}[k] = \frac{\exp(\sigma v_k)}{\sum_{k'} \exp(\sigma v_{k'})} \quad (3)$$

The recent discrete mode choice extension of MATSim makes use of model formulations like Equation 1 to simulate mode choices for the agents. First, all attributes are estimated, second, the utilities are calculated and, third, one alternative is sampled from the distribution over the alternatives defined by Equation 3. This happens after going through a couple of additional steps such as checking availability of alternatives and considering constraints.

3. Scoring-based choice process

MATSim is run in an iterative way. First, a mobility simulation is run with the daily plans of all agents, containing activities at specified location and trips connecting those activities. Due to limited capacity on the roads and other limitations, agents compete against each other for space and time in the transport network, which leads to congestion and similar phenomena. After the mobility simulation, the replanning phase starts, which takes care of the decision-making of the agents. Generally, the idea is to

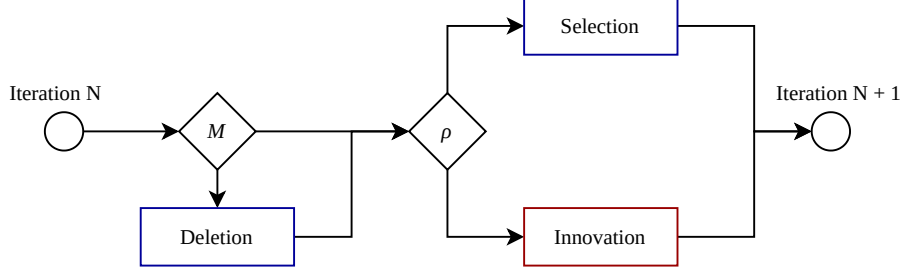


Figure 1: Schema of the co-evolutionary and scoring-based choice process of MATSim with agent memory size M and innovation rate ρ .

modify certain attributes in the agents' plans (departure times from activities, modes chosen for each trip, even locations at which activities happen) such that the agent optimizes the daily plan.

In MATSim, the quality of an agent's plan is described by a score $S \in \mathbb{R}$, consisting of scores for performing activities at the right time for the desired duration and for spending time in transport. A typical score can look like

$$\begin{aligned}
 S = & S_{A1} \\
 & + \beta_{m_1, constant} + \beta_{m_1, travelTime} \cdot x_{1, travelTime} + \dots \\
 & + S_{A2} \\
 & + \beta_{m_2, constant} + \beta_{m_2, travelTime} \cdot x_{2, travelTime} + \dots \\
 & + S_{A3}
 \end{aligned} \tag{4}$$

Here S_A denote scores obtained for activities and m denote the mode used on the connecting trips. While a couple of fixed parameters are commonly used in MATSim, in principle, nothing prevents the user from defining arbitrarily complex scoring functions. The score of a plan is constructed while the mobility simulation is running. It is therefore a descriptive quantity that is only fully known after one plan has been tested in simulation once.

The replanning process of MATSim is shown in Figure ?? . After the mobility simulation of iteration N , an algorithm is run for each agent. Each of the agents has a memory of M plans p_m that have been observed in the past and which have obtained a score $S(p_m)$. In the first step, the replanning process checks whether the agent's memory exceeds the limit $M \in \mathbb{N}$. If so, one of the existing plans is removed. If the plan is removed that was currently selected for execution, a random one among the remaining ones is selected. After, a couple of replanning strategies can be defined which are either "selection" or "innovation" strategies. The former have the task to select a plan for execution in the next mobility simulation among the existing ones. Innovation strategies select one existing plan at random and then apply changes to that plan, effectively creating a new one that is added to memory and selected for execution in the next mobility simulation. Whether to run a selection or innovation procedure in an iteration is defined by a weight defined for each strategy. Assuming for now that only one selection strategy and only one innovation strategy is available, whether to run innovation or not can be described by an "innovation probability" $\rho \in [0, 1]$.

The standard version of the process is designed as follows: The innovation part (red) introduces small random changes to the plans (for instance, changing the transport mode of a trip, the end time of an activity) to generate new plan proposals that will be tested in simulation. The selection process is an interplay between deletion and selection (in blue) which biases the selected plan of an agent towards those with high scores. Effectively, the selection process chooses plans with high scores more frequently than others (**ChangeExpBeta** strategy), and the deletion process removes the plan with the worst score in an agent's memory (**WorstPlanForRemovalSelector**).

Assuming that agents' decisions do not affect other agents (no congestion on the roads, no crowding in public transport) one could run this procedure infinitely long and eventually test any possible combination of plan configurations. The agent memory would then be populated with high probability by multiple

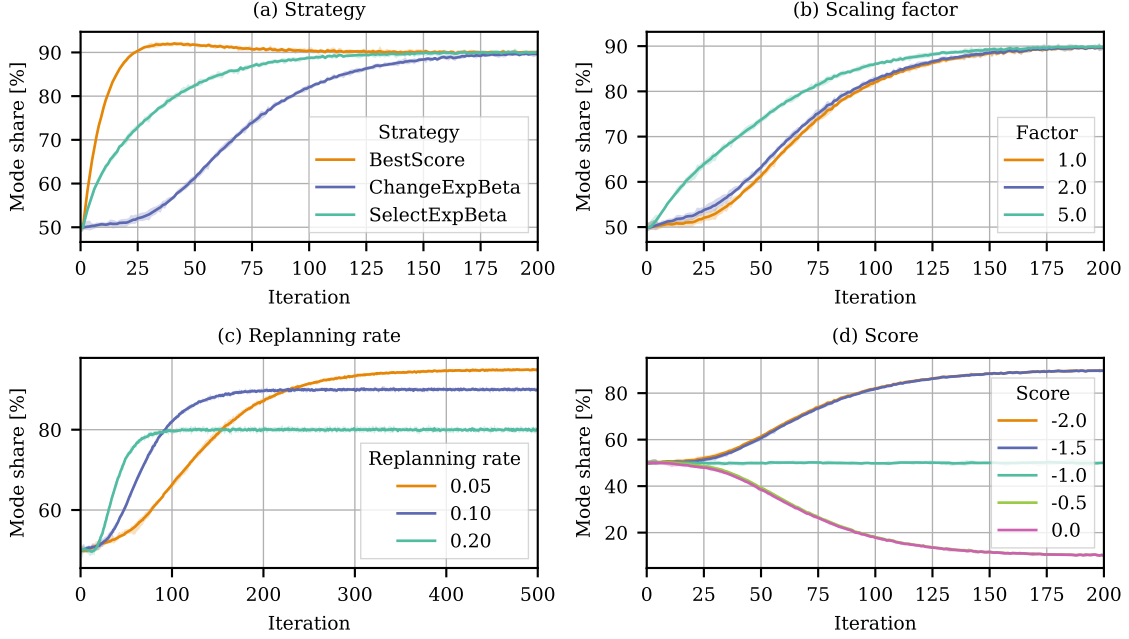


Figure 2: Analysis of the scoring-based choice process of MATSim in the toy example.

instances of the plan configuration with the maximum score. Assuming that the selection procedure always selects the *best* one in memory, there would surely be $M - 1$ instances of that plan in memory. The remaining plan can also be the optimal one, but note that with a probability of ρ in each iteration a modified (sub-optimal) version is generated, *executed* and scored in simulation.

The process becomes more complex if decisions of different agents interact, i.e. if their use of a specific road affects the travel times of other agents. In that case a plan that had a large score in iteration N may have a low score in $N + 10$ because traffic conditions have changed. Still, considering these interactions and fluctuations, MATSim aims at optimizing each agent’s plan based on the currently observed conditions. Which optima exist in that case shall be analyzed elsewhere.

4. Toy example and basic properties

In the following some basic properties of MATSim’s choice process and comparison to discrete choice models are investigated using a toy example¹. We assume that there are two locations (S) and (E) and $N = 10,000$ agents that want to travel from the former to the latter. Two modes of transport A and B are available and there are no interactions between agents. They can, however, individually choose either to use mode A or B. No activity scoring is taking place, but if mode A is used for the trip, the plan receives utility $S_A = -1.0$. If mode B is used, the plan receives utility $S_B = -2.0$. In terms of utility we, therefore, have two states, either S_A or S_B with the former being the favorable one. Initially, both modes are distributed randomly over all agent trips. The standard configuration of the toy example uses the **ChangeExpBeta** selection strategy in MATSim, which compares the currently selected plan with a randomly selected one from memory and accepts the new one based on the difference in score, with a bias towards accepting the one with the larger score. The deletion strategy is, as usual, to delete the worst plan in terms of score. For innovation, we use **ChangeTripMode**, which replaces the current mode of a trip with a randomly chosen one that is *different*. Hence, if A was chosen before, B will be chosen and vice versa. Initially, an innovation rate of $\rho = 0.1$ is chosen and an agent memory size of $M = 3$.

A couple of experiments are run, varying some of the parameters defining the choice process. Each experiment is run for 500 iterations and ten times with different random seeds. Figure ?? shows the mode share of A averaged over ten realizations in each iteration, as well as the minimum and maximum overserved in the ensemble (shaded).

¹All experiments can be repeated with the code available at <https://github.com/sebhoerl/matsim-pseudo-random-errors>

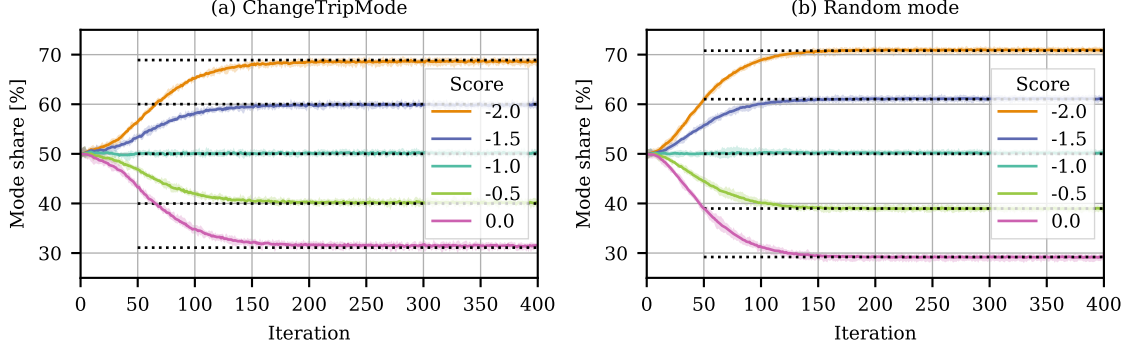


Figure 3: Introduction of pseudo-random errors to scoring in MATSim.

First, we can examine Figure ??a. Starting at the initial state of 50% the mode share stabilizes at a value of 90% after around 200 iterations for each of the strategies. We can see that the choice of strategy affects how the system goes into equilibrium, yet, they all bias towards plans with larger scores. They arrive at the same equilibrium share.

Figure ??b shows how a scaling of both utilities (S_A and S_B) by a factor f (i.e. $S'_X = f \cdot S_X$) affects convergence with the standard `ChangeExpBeta` selector. Clearly, the magnitude of the score has an effect on convergence. Larger scores allow for a clearer distinction of the two modes, while lower ones lead to stronger mixing in the choice process.

More important results, however, show Figures ??c and ??d. The former one shows how the replanning rate affects the final mode share. Apparently, the share of mode A stabilizes at the replanning rate ρ . Moreover, Figure ??d shows different choices for S_B . In the case $S_A = S_B = -1.0$ one can see that both modes are equally probable, which makes sense as their scores are the same. However, for cases where one mode has a larger score than the other, the shares converge to ρ and $1 - \rho$, respectively. The *actual* value of S_B does *not* influence to where the system converges.

Given the explanations on the choice dynamics above, the obtained results are not suprising. In the standard formulation of the scoring function, all agents are the same and in the toy example, and each of them has the same trip. As MATSim aims at maximizing the plan score for each agent, all of them are pushed to the same (optimal) plan configuration. However, with a probability of ρ the plan is innovated. The chosen innovation strategy always chooses the unchosen mode. Hence, after a transient phase, for $100\% \cdot \rho$ of agents the better mode is chosen in most cases. With a probability of $100\% \cdot (1 - \rho)$ the less favorable mode is generated. There is only a slight chance that the less favorable mode was generated, then simulated and reselected again by the selection algorithm.

The rate at which innovation strategies are applied has a major influence on the dynamics of the choice process. It is important to emphasize that a non-zero innovation rate *prevents* the system from visiting an overall optimal state because it continues testing other options (due to the lack of knowledge about when a state is “optimal”).

A major difference between the MATSim process and a discrete choice model is the lack of a notion of “individuality” between the decision-makers. The experiments show that if multiple agents perform a very similar trip, they are very likely to make the same decision. A discrete choice model would still produce a certain variety even if the trips are exactly the same because of the integrated error term, which can either be interpreted as variations in taste of the decision-makers or modeling uncertainty. The reason why large-scale MATSim simulations still show a rich variety in choices comes from the varying trip distances, locations and departure times in the population.

Finally, the experiments show that *technically* it is not a valid approach to convert estimated parameters from a multinomial logit model to MATSim scoring parameters, even if one corrects for opportunity cost of time Nagel et al. (2016) coming from the scoring of activities that is usually not included in discrete mode choice models.

5. Pseudo-random errors

The mode share bias due to innovation seems to be inherent to the process how the co-evolutionary algorithm is structured and may be solved by intelligently reducing the innovation rate. This, however, shall be subject for future research. Here, the issue of variability shall be tackled. Effectively, a method

is proposed that makes it possible to make use of a discrete choice model with the scoring framework of MATSim.

Comparing the structure of scoring in MATSim and discrete choice models, there is one major similarity - both are based on the concept of utility maximization. While MATSim aims at maximizing a deterministic score, the deterministic utility in a multinomial logit model is affected by a random error term that is added on top. From this perspective, MATSim already provides the algorithmic infrastructure to perform the maximization, it just operates on quantities that are different to the choice model formulation.

The easiest approach, therefore, would be to sample a random error everytime a trip is scored in MATSim and to add this error to the score. However, most choice models are estimated with the assumption that if a decision maker takes a certain decision once, he or she will take the same decision again if the situation is the same. For this reason, and to stabilize convergence, we propose to make use of pseudo-random errors that seem random and follow the correct distribution, but are deterministically generated based on high-level information such as the identifier of the agent, the position of a trip in the plan and the chosen mode (c.f. $\epsilon_{i,k}$ where i defines the choice situation, i.e. person i , and k the alternative). So how can we generate error terms that are deterministic, but when viewed over the entirety of all agents and trips in a population follow a specific distribution?

In cryptography, hash functions $O = H(I)$ are used. They take a variable length series of bits as input and return a fixed-size series of output bits. In simple terms, it should be as hard as possible for an attacker to reconstruct the input I from output O . If the hash function is known, the attacker may generate a large number of pairs (I', O') and interpolate them to reconstruct $O \rightarrow I$. To prevent this, most hash functions (like SHA-512) are constructed around the “avalanche effect” Feistel (1973) which says that if only one bit in the input changes, more than 50% of output bits should change. Given a set of inputs I_1, \dots, I_N this means that the hashes in their output domain (for instance, for 4-bit hashes this would be 0000 to 1111) are uniformly distributed.

We can therefore use any hash function $H(\cdot)$ with the correct properties to construct

$$h = H(\text{Agent Id, Trip index, Mode}) \quad (5)$$

and arrive at $\{h_1, \dots, h_\infty\} \sim \text{Uniform}(0,1)$. Now if the choice model was estimated using a Gumbel distribution, all that remains is to transform the uniform error into a Gumbel error using the inverse cumulative distribution function of the error distribution $e = F^{-1}(h)$ and add it to the score of the agent for each trip. As before, the score over multiple iterations is sensitive to changes in attributes (congestion, crowding), but each choice situation includes some error or taste variation. Note that one is not restricted to sample Gumbel-distributed errors. It is perfectly possible to sample i.i.d. normal errors, thus replicating a multinomial probit model in MATSim. Furthermore, additional components can be included in the hash function, which may allow the modeler to define more complex correlation structures as used, for instance, in a nested logit model. These possibilities should be investigated in the future.

The procedure can be implemented as a simple addon to the scoring function in MATSim. Figure ?? shows the toy example with pseudo-random error terms. Note how the mode share is now sensitive to the parameter values. Furthermore, it is possible to describe analytically at which level the mode share will stabilize (given the innovation rate and that we know that all trips are the same). Generally, we know that by performing the score maximization the selection process tends to Equation 3. We can denote the resulting probability (e.g. $\exp(-1.0)/[\exp(-1.0) + \exp(-2.0)]$) as $Q[A]$. Mode A is chosen with probability $Q[A]$ in $100\% \cdot \rho$ of the cases, i.e. when selection is taking place. When innovation is taking place, the probability of emitting $R[A]$ A is dependent on the choice process. Figure ?? shows two examples. In Figure ??b we show a simplified process where the innovation strategy always chooses *at random* a new mode. Hence, the probability in the toy example case is $R[A] = \frac{1}{2}$ as there are two modes:

$$P[A] = (1 - \rho)Q[A] + \rho R[A] = (1 - \rho)Q[A] + \rho R[A] \quad (6)$$

Figure ??b shows the calculated values from this equation as dotted lines. For the **ChangeExpBeta** the case is more complicated as the emitted mode depends on the already chosen mode (i.e., it becomes

the opposite). Hence, the probability of choosing A is $R[A] = P[B]$, or $R[A] = 1 - P[A]$. Inserting the expression in Equation 6 we arrive at

$$P[A] = (1 - \rho)Q[A] + \rho(1 - P[A]) \quad (7)$$

leading to

$$P[A] = \frac{(1 - \rho)Q[A] + \rho}{1 + \rho} \quad (8)$$

Again, the analytical outcome is shown in Figure ??a. It is hence possible to estimate the bias introduced by innovation in this toy example. It remains to explore how these findings can help to generalize to various other cases.

Those more general cases include, for instance:

- How do capacities (congestion and crowding) influence convergence?
- Does the proposed method mitigate other problems (e.g. congestion and oscillations in routing, Tchervenkov et al. (2020)) by avoiding situations in which too many agents have very similar plans and deterministically choose modes?
- How can the proposed method help for other choices (activity end time, location, ...) ?

5.1. Conclusion

The paper at hand reveals a couple of aspects about the choice behaviour in MATSim that are inherent to the approach, but may come as a surprise for users who do not dive deep into the details of the dynamics:

- The lack of agent-specific tastes introduces a tendency for very similar trips to deterministically have the same choice outcomes.
- Because of that, the traditional scoring functionality of MATSim can not easily be used with estimated parameters of a discrete choice model. Only their relative values are taken into account.
- Innovation introduces a bias into the choice process that must be taken into account.

Furthermore, a method based on cryptographic hash functions is proposed which bring a couple of additional insights:

- Using deterministic error terms generated by cryptographic hash functions MATSim scoring can be transformed such that it represents a trip-based multi-nomial logit model.
- For simple cases and informative toy examples, the bias introduced by innovation can be studied and calculated analytically.

For the future, it will be interesting to build on this research to further describe, understand and potentially improve MATSim by the use of toy examples. Promising topics are criteria for convergence and online adjustment of choice process parameters.

References

- Feistel, H., 1973. Cryptography and computer privacy. Scientific american 228, 15–23.
- Flötteröd, G., Kickhöfer, B., 2016. Choice models in matsim, in: Horni, A., Nagel, K., Axhausen, K.W. (Eds.), The Multi-Agent Transport Simulation MATSim. Ubiquity Press, London, pp. 337 – 345.
- Hörl, S., Balac, M., Axhausen, K.W., 2018. A first look at bridging discrete choice modeling and agent-based microsimulation in MATSim. Procedia Computer Science 130, 900 – 907.

- Hörl, S., Balac, M., Axhausen, K.W., 2019. Pairing discrete mode choice models and agent-based transport simulation with MATSim, in: 98th Annual Meeting of the Transportation Research Board, Washington, D.C.
- 250 Horni, A., Nagel, K., Axhausen, K.W. (Eds.), 2016. The Multi-Agent Transport Simulation MATSim. Ubiquity Press, London.
- McFadden, D., 1974. Conditional logit analysis of qualitative choice behaviour. Academic Press, New York. pp. 105–142.
- 255 Nagel, K., Kickhöfer, B., Horni, A., Charypar, D., 2016. A closer look at scoring, in: Horni, A., Nagel, K., Axhausen, K.W. (Eds.), The Multi-Agent Transport Simulation MATSim. Ubiquity Press, London, pp. 23 – 33.
- Tchervenkov, C., Hörl, S., Balac, M., Dubernet, T., Axhausen, K.W., 2020. An improved replanning strategy for congested traffic conditions in matsim. *Procedia Computer Science* 170, 779 – 784.
- Train, K.E., 2009. *Discrete Choice Methods with Simulation*. Cambridge university press, Cambridge.
- 260 Vosooghi, R., Kamel, J., Puchinger, J., Leblond, V., Jankovic, M., 2019. Robo-taxi service fleet sizing: Assessing the impact of user trust and willingness to use. *Transportation* 46, 1997–2015.