

SUBPART SIX

Additional Choice Dimensions

CHAPTER 27

Destination Innovation

Andreas Horni, Kai Nagel and Kay W. Axhausen

27.1 Basic Information

Entry point to documentation:

<http://matsim.org/extensions> → locationchoice

Invoking the module:

<http://matsim.org/javadoc> → locationchoice → RunLocationChoiceBestResponse, RunLocationChoiceFrozenEpsilons classes

Selected publications:

Horni et al. (2012b); Horni (2013)

27.2 Introduction

Generally speaking, destination choice represents an optimization problem, where every agent searches for his or her optimal destination according to an objective function, subject to various constraints such as the agent's travel time budget—as well as interactions with other agents—while competing for space-time slots in the infrastructure. The MATSim destination innovation module provides a problem-tailored heuristic algorithm to solve this problem.

MATSim's iterative base requires a mechanism (the main component of the destination innovation module), ensuring consistent probabilistic choices over the course of iterations.

Unobserved heterogeneity, usually dominant in destination choice, is captured in the adaptable objective function by random error terms (Horni et al., 2012b; Horni, 2013).

As well as considering competition for road infrastructure, the destination choice module can also be configured for activities infrastructure (for example, at shopping malls' parking lots) as shown in Section 27.3.5 and by Horni et al. (2009).

How to cite this book chapter:

Horni, A, Nagel, K and Axhausen, K W. 2016. Destination Innovation. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) *The Multi-Agent Transport Simulation MATSim*, Pp. 165–174. London: Ubiquity Press. DOI: <http://dx.doi.org/10.5334/baw.27>. License: CC-BY 4.0

27.3 Key Issues in Developing the Module

Key issues of integrating destination innovation into MATSim include behavioral and algorithmic problems. On the behavioral side, specification of choice sets for model estimation has not yet been solved. On the algorithmic side, as mentioned above, destination innovation is, in principle, an ordinary optimization problem. However, as agents interact, and choices are embedded in a highly dynamic context, the problem becomes complex, particularly because targeted scenarios are usually large-scale. Thus, as in real-world optimization problems, solutions must be based on problem-tailored heuristics (Michalewicz and Fogel, 2004). Construction of a search space and subsequent evaluation of the search space's elements are important MATSim destination innovation components.

The main component however, is a mechanism to generate consistent random draws over iterations necessary to include the objective function's error terms (see next Section 27.3.1). This mechanism is also applicable to other choice dimensions.

27.3.1 Error Terms

As described in Chapter 49, MATSim—as a utility-maximizing model—is related to the discrete choice framework, meaning that this framework can productively guide the MATSim utility function specification. Utility in discrete choice models is composed of a deterministic part and a random error term representing the unobserved heterogeneity, i.e., it subsumes, both truly, i.e., inherently random, decisions and the modeler's missing knowledge about the choice and its context.

In MATSim, the utility function for route, mode and time innovation does not contain an explicit random error term (yet). This is at least partially compensated through replanning stochasticity, in Chapter 49 denoted by the scale parameter μ and η . An example for this might be: route and time choices are usually subject to significant competition. The co-evolutionary algorithm of MATSim, detailed below, essentially assigns the resources in a random manner to the persons. For example, two identical persons may end up with different routes, according to the order in which they undergo the replanning. Essentially, this means that an (implicit) random term is present in the choice making.

The above, however, does not add enough unobserved heterogeneity to destination choice. Further problems might, or might not, appear when trying to interpret this randomness, since it is added implicitly and somewhat unsystematically. Thus, an explicit random error term ε_{nlq} for every person n , alternative ℓ and activity q , held stable over the iterations, is added to the scoring function during the running of the destination innovation module (Horni, 2013). Research about the necessity of error terms for the remaining choice dimensions is required, as discussed in Section 97.4.6.

27.3.2 Quenched Randomness

Due to random error terms, discrete choices are quantified by probabilities; for example, for the logit model, as $p_{nlq} = \exp(V_{nlq}) / \sum_{j \in L} \exp(V_{njq})$, where V_{nlq} is person n 's systematic utility of alternative ℓ for activity q . When drawing from the distribution specified by p_{nlq} for a population, the aggregate choices are reproduced. This is basically also true when applied in iterative frameworks. However, iterative frameworks are usually associated with some kind of learning or relaxation mechanism, which is heavily distorted by repeatedly and randomly drawing from p_{nlq} in every iteration. In this case, the ε_{nlq} effectively fluctuate from iteration to iteration, which is disastrous for the algorithm's convergence and behaviorally implausible.

Instead, random error terms ε must remain fixed from iteration to iteration. The optimization is then performed as a deterministic search, based on the resulting utilities U_{nlq} , i.e., an alternative ℓ for person n ; activity q is selected as

$$\operatorname{argmax}_{\ell \in \text{choice set}} U_{nlq} = V_{nlq} + \varepsilon_{nlq}.$$

This includes, via the systematic part V_{nlq} , the disutility of traveling to destination ℓ for activity q .

As stated above, random error terms must remain the same over the iterations (also discussed in Chapter 49). In physics, this approach would be called “quenched” (sometimes also “frozen”) randomness; all randomness is computed initially and then attached to particles or destinations, rather than instantaneously generating it, which would be called “annealed” randomness. Two natural approaches for implementing quenched randomness are as follows:

- (a) Freezing the applied *global* sequence of random numbers, meaning that a Monte Carlo method with the same random seed is used before and after introduction of a policy measure and over the course of iterations. Thus, error terms should come out the same way *before* and *after* the introduction of the policy measure. Differences in the outcome can thus be directly attributed to the policy measure.
- (b) Computing and storing a separate ε_{nlq} for every combination of person n , alternative ℓ and activity q .

Both strategies have flaws. Approach (a) is only an option if one is certain about every single aspect of the computational code. Literally, one additional random number, drawn in one run, but not in the other, completely destroys the “quench” for all decisions computed later in the program. Consistency is thus hard to achieve, especially in parallel or even distributed computing environments; substantial machinery is necessary to ensure consistent choices. In a modular environment, as in MATSim, designed for external plugging-in of users’ own modules—possibly drawing their own random numbers—the danger of destroying the quench is prohibitively high and thus approach (a) is impractical.

Approach (b) is certainly more robust. However, for large numbers of decision makers and/or alternatives, storing error terms is difficult. For destination innovation, one quickly has 10^6 decision makers and 10^6 alternatives, resulting in $4 \cdot 10^{12}$ Byte = 4TB of storage space.

One may argue that this should not be a problem, since a normal person will rarely consider more than the order of a hundred alternatives in their choice set, reducing the computational problem. Aside from the necessity of storing every decision maker’s choice set, this converts the computational problem into a conceptual one, since a good method to generate choice sets then needs to be found. With more conceptual progress, this may eventually be an option; at this point, a conceptually simpler approach is preferred.

The solution developed below is generally applicable in econometric microsimulators. The same *stable* error term can be *re-calculated* on the fly by using stable random seeds $s_{nlq} = g(k_n, k_\ell, k_q)$, containing uniformly distributed random numbers associated with k , ℓ , and q . That is, for each person n , a random number k_n is generated and stored; the same is done with each destination ℓ . Value for the activity q can be derived from its index in the plan, possibly combined with the person’s value k_n . This reduces the storage space dramatically, from $N_q \cdot N_n \cdot N_\ell$ to $N_q(N_n + N_\ell)$, where N_n is the number of persons or agents and N_ℓ is the number of destinations and N_q is the average number of discretionary activities in an agent’s plan. This means that storage space is reduced to approximately $2 \cdot 4 \cdot 10^6$ Byte = 8MB, which can be easily stored on any modern machine.

Distribution of these seeds is essentially irrelevant; any error term distribution can be generated from any basic seed distribution. In the current version, $g(k_n, k_\ell, k_q) = (k_n + k_\ell + k_q) \times v_{max}$ is used. v_{max} is the maximum (long) number that can be handled by the specific machine.

To evaluate utility for a person n visiting the destination ℓ for activity q , a sequence of Gumbel-distributed random numbers $seq_{n\ell q}$ is generated on the fly for every person-alternative-activity combination using the seed $s_{n\ell q}$. Some random number generators have problems in the initial phase of drawing, e.g., the first couple of random numbers are correlated or never cover the complete probability space. As in our procedure, the random number generator is constantly re-initialized; for these technical reasons, the error term $\varepsilon_{n\ell q}$ is not derived from the first element, but from the m^{th} element of the sequence $seq_{n\ell q}[m]$. Here, m is set to 10. This procedure is valid, as the set of all m^{th} elements of all different sequences is also a pseudo-random sequence, following the same distribution as the sequences $seq_{n\ell q}$; clearly, *true* random number generators relying on physical phenomena, such as hardware temperature, are not applicable.

27.3.3 Search Space Construction and Evaluation

MATSim destination innovation is based on best-response, rather than random mutation; in every iteration, the best current alternative, *including* the $\varepsilon_{n\ell q}$, is chosen. This works as long as inter-iteration changes are small, which usually happens, given by the relatively small share of agents who re-plan. The best-response approach is adopted due to the usually huge number of alternatives in combination with the search space characteristics. The discrete search landscape is characterized by random noise, because error terms are not spatially correlated (see Figure 27.1(a)). For such problems—as opposed to continuous landscapes (see Figure 27.1(b))—efficient search methods, such as local search methods, generally do not work.

When searching for the best choice, the large number of alternatives—prohibiting exhaustive search—is restrained as follows (for the detailed derivation see Horni, 2013, p.51 ff.). It is assumed that travel costs are always negative and that a person drops activities with negative net utility. Then, the maximum potential travel effort a person is willing to invest is constrained by the maximum error term per person and activity. This approach is promising, as very large values for Gumbel-distributed variables are rare, meaning that a huge space must be searched for only a few persons.

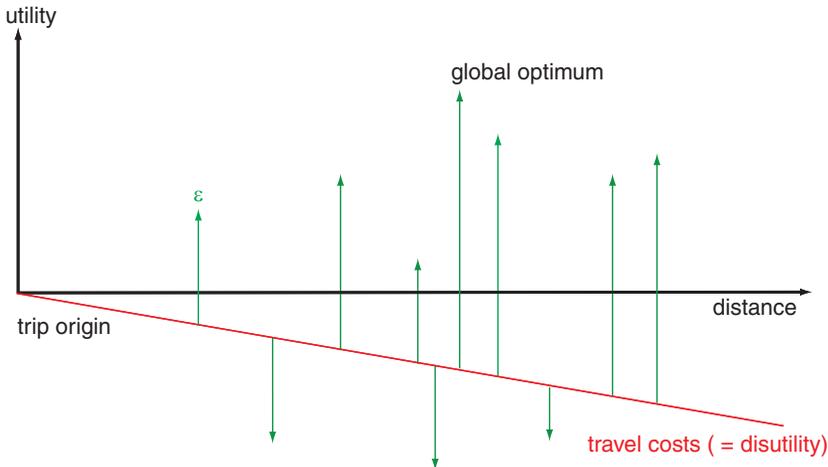
This search space reduction saves a great deal of computation time; however, it is still unfeasible and further speed-ups are necessary. Most computation time is due to travel time calculation, i.e., due to routing, for evaluation of the alternatives in the search space. To reduce these huge routing costs, the Dijkstra (Dijkstra, 1959) routing algorithm is not only applied forward—providing one-to-all travel times—but also *backwards*, using an average estimated arrival time as initial time. This is an approximation; thus, a *probabilistic* best response is applied, justified by the assumption that, during the course of the iterations, the probabilistic choice will reduce the errors incurred by approximating travel times.

With this procedure, the required computational effort is dramatically reduced, allowing application of destination innovation to large-scale scenarios.

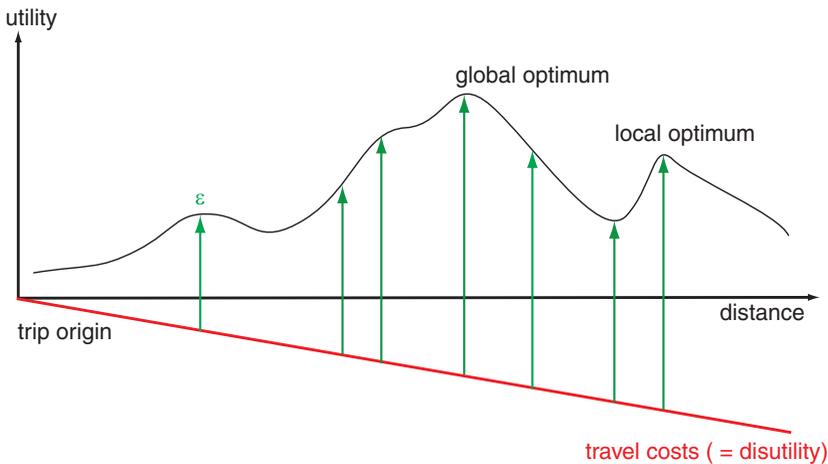
27.3.4 Destination Choice Set Specification

Choice set specification is natural for choices with few alternatives; but in contrast, for problems with a large universal choice set, specifying individual choice sets becomes a challenging computational and behavioral issue. This is particularly true for spatial choices like destination or route choice (e.g., Pagliara and Timmermans, 2009; Thill, 1992; Schüssler, 2010; Frejinger et al., 2009b). Estimates are sensitive to choice sets; at the same time, no established choice set definition procedure exists for spatial problems. This means that choice sets and, hence, estimates are dependent on the modeler.

An important extension of the standard discrete choice modeling approach to treat this problem is formed by stochastic choice set models, founded by Manski (1977); Burnett and Hanson



(a) Uncorrelated error terms.



(b) Spatially correlated error terms.

Figure 27.1: Search space: The search algorithm must be able to handle correlated, as well as uncorrelated, error terms as given by the MNL model. Local search methods, such as hill-climbing algorithms are only able to handle continuous search spaces; thus, for situation (a), a best-response global search algorithm is required.

(1979); Burnett (1980); these integrate the choice set formation step into the estimation procedure by jointly estimating choice set selection and selection of a particular alternative of this choice set (Manski, 1977; Ben-Akiva and Boccara, 1995). Probabilistic choice set formation is conceptually appealing; choice sets are, in principle, not restrained a priori by exogenous criteria, as in standard choice set specification. However, the procedure is generally associated with combinatorial complexity, making it computationally intractable. As a consequence, practical approaches also require mechanisms to reduce complexity of the choice set specification problem (e.g., Ben-Akiva and Boccara, 1995, p.11). Zheng and Guo (2008), for example, make the moderate assumption of continuous store choice sets (i.e., sets without “holes”) around the trip origin, while Ben-Akiva and Boccara (1995)’s random-constraints model exploits additional information on alternatives’ availability for individuals.

In conclusion, the destination innovation set specification problem is still unsolved, meaning that estimated models can only be fully consistently applied for the region where the model was estimated. For MATSim, destination choice model estimation efforts are reported in Horni (2013, Chapter 5).

27.3.5 Facility Load

The influence of interaction in *transport* infrastructure for people's route and departure time choice was recognized almost a century ago (e.g., Pigou, 1920; Knight, 1924; Wardrop, 1952). It can also be reasonably assumed that agent interaction in *activities* infrastructure affects travel choices (Axhausen, 2006). Marketing science provides ample evidence that agent interactions influence utility (positively or negatively) of performing an activity (Baker et al., 1994, p.331), (Eroglu and Harrell, 1986; Eroglu and Machleit, 1990; Eroglu et al., 2005; Harrell et al., 1980; Hui and Bateson, 1991; Pons et al., 2006).

In Horni et al. (2009), based on the Zürich scenario, a model is presented introducing competition for activity infrastructure space-time slots. The actual load is coupled with time-dependent capacity restraints.

Activity location load, computed for 15 minute time bins, is derived from events delivered by the mobsim. The load of one particular iteration, combined with time-dependent activity location capacity restraints, is considered in the agents' choice process of the succeeding iteration. In detail, this means that the utility function term $S_{dur,q}$, described above, is multiplied by $\max(0; 1 - f_{load\ penalty})$, penalizing agents dependent on the load of the location they frequented. $f_{load\ penalty}$ is a power function; this has proved to be a good choice for modeling capacity restraints (remember that the well-known cost-flow function by U.S. Bureau of Public Roads (1964) is a power function). To introduce additional activity location heterogeneity, an attractiveness factor $f_{attractiveness}$ is introduced, defined to be logarithmically dependent on the store size given by the official workplaces census.

Also for demonstration purposes, capacity restraints are exclusively applied to shopping locations; in principle, leisure activity locations could be handled similarly. However, deriving capacity restraints for leisure activity locations is expected to be much more difficult than for shopping locations, because far less data is available for leisure locations and capacity restraints vary much more between different leisure locations than between different shopping activities (hiking versus going to the movies might be a good example).

The model allows assignment of individual time-dependent capacities to the activity locations. For the sake of demonstration, the capacities of all shopping facilities can be set equal, where their values can be derived from the shopping trip information given in the Swiss microcensus (Swiss Federal Statistical Office (BFS), 2006). The total daily capacity is set so that the activity locations located in the Zürich region satisfy the total daily demand with a reserve of 50%. In detail, the capacity restraint function for a location l is as follows:

$$f_{load\ penalty,\ell} = \alpha_l \cdot \left(\frac{load_\ell}{capacity_\ell} \right)^{\beta_\ell}$$

with $\alpha_\ell = 1/1.5^{\beta_\ell}$, $\beta_\ell = 5$. $f_{load\ penalty,\ell}$ is the penalty factor for location ℓ as described above.

Simultaneous computation of all agents' score reduction avoids the last-record problem discussed in Vovsha et al. (2002). There, a sequential choice process is proposed; alternatives are removed from later travelers' choice sets if locations are already occupied by earlier travelers. Thus, travelers' order is specified arbitrarily; the last-record problem (last travelers must go a long distance to find an available location) is significant when modeling heterogeneous travelers.

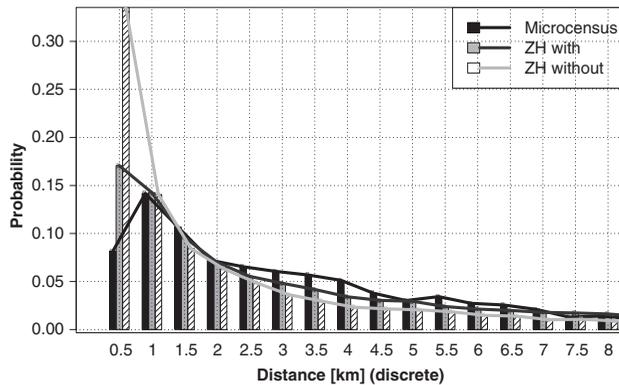
As expected, the constrained model improves result quality by reducing the number of implausibly overcrowded activity locations.

27.4 Application of the Module

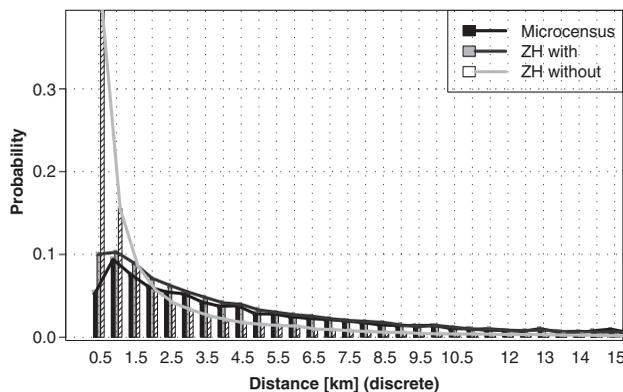
The destination innovation module has been successfully applied for the Zürich scenario (Chapter 56), as reported in Horni (2013, p.99), for the Tel Aviv model (see Chapter 91) and for the MATSim 2030 project. Figure 27.2 and Figure 27.3 show that, through error term scaling, distance distributions can be nicely fitted, decreasing count data error.

27.5 The Module in the MATSim Context

The destination innovation module explicitly incorporates unobserved heterogeneity through random error terms; the standard MATSim utility function, however, does not contain error terms. Randomness measured in empirical data is included implicitly through the simulation process stochasticity, including possible randomness in the choice itself. For destination innovation, this has led to a dramatic underestimation of total travel demand, making inclusion of unobserved heterogeneity inevitable. Clearly, the problem is the impossibility of making all choices at the same level; destination choice is conditional on mode choice which, in turn, is conditional on route choice. Hierarchical choice modeling has clearly showed that randomness, expressed by the logit model scale parameter, needs to be larger in higher level decisions. This chapter addresses replacing the need for more randomness in the choice model by directly including randomness into the utility function; that randomness must be quenched, otherwise the iterative procedure will just average



(a) Shopping trips.



(b) Leisure trips.

Figure 27.2: Error term runs for the Zürich scenario.

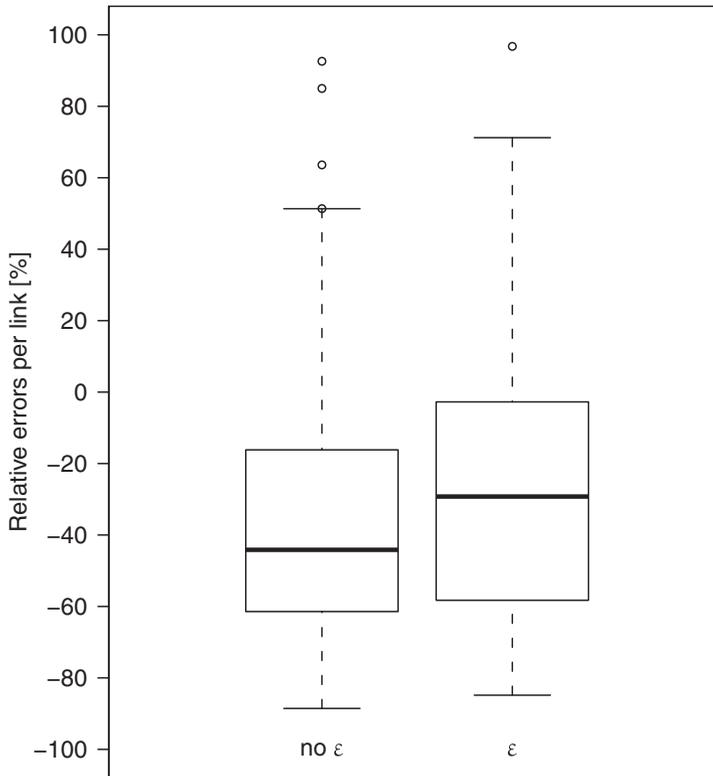


Figure 27.3: Daily traffic volumes for 123 links compared to traffic counts. Per link k the relative error is used, i.e., $(vol_{simulated,k} - vol_{counted,k})/vol_{counted,k}$.

it out. Whether the standard utility function might also profit from the innovations made for this module should be a topic for future research .

MATSim replanning offers different strategies to adapt plans, ranging from random mutation via approximate suggestions to best response answers. Destination innovation is based on best response to handle the sheer size of the alternatives set.

Although the destination innovation utility function is based on discrete choice framework, some conceptual differences about the common discrete choice models application persist. As shown above, there is no drawing from discrete choice models, but instead, maximization of an iteration-stable utility function. The set of alternatives is not necessarily limited a priori; thus, we use the notion of a search space and not of a choice set here.

27.6 Lessons Learned

Two interesting lessons were learned while developing the destination innovation module: first, a lesson on preferences and space interdependence and the necessity to evaluate them in combination. When looking at distance distributions (e.g., Figure 27.2) one might think that the functional form directly represents the preferences, but this is not necessarily the case. In our simulations, it is the result of a *linear* travel disutility, but applied in geographic space, where number of opportunities increases with the *square* of the radius, in other words, with the *square* of travel distance. A similar emergent effect appears when scaling random error terms. Although both negative and

positive error terms are enlarged and the average remains stable, distribution gets more skewed toward the tail; for agents' choices, maximum values—not average values—are relevant.

The second lesson concerns simulation results' variability. Although random elements are not present only in destination choice, it was the largest contributor of endogenous variability when it was developed, necessitating the experiments presented by Horni et al. (2011a) (see also Section 48.4).

27.7 Further Reading

The main information source is Horni et al. (2012b); Horni (2013); technical details and documentation are available at Horni (2016) and in javadoc. Further reading related to destination choice is: Horni et al. (2013b), for parking, or Horni et al. (2012a), about coupling customers' and retailers' choices or, in other words, supply and demand.

