CHAPTER 48

MATSim as a Monte-Carlo Engine

Gunnar Flötteröd

48.1 Introduction

"Agents" that "learn" in a "synthetic reality" is a common term in Artificial Intelligence (Russel and Norvig, 2010) and/or Multi-Agent simulation (Ferber, 1999), but it does not belong to the standard terminology of transport modeling. This chapter explains the functioning of MATSim in terms of modeling and simulation concepts that are more established in the transportation field.

It is important to distinguish between a model and a simulation. A model describes certain aspects of a system; a simulation evaluates a model. For instance, a simple route choice model may state that route A is selected with 25 % probability and route B with 75 % probability. A simulation of this model then draws one or more realizations (route choices) from this distribution. One always needs a model before one can simulate. Possible feedback from simulation to modeling comprises (i) new insights into emergent model properties and (ii) computational constraints that prohibit overly complex model specifications. In MATSim, both kinds of feedback are strong drivers of the modeling.

Consider Figure 48.1, displaying MATSim as a model system comprising a (travel) demand model and a (network) supply model. The travel demand model predicts travelers' behavior, given their information about the network conditions. The network supply model predicts these network conditions using a certain travel behavior chosen by all travelers in the system. This is complemented by the modeling assumption that demand and supply are mutually consistent in the sense that the network conditions resulting from a certain travel behavior are statistically equal to the network conditions that caused this behavior.

Simulation addresses the question of how to identify this state of mutual demand/supply consistency, i.e., it solves the model. The model system shown in Figure 48.1 is complicated—it is nonlinear, stochastic and extremely high-dimensional. The only known operational technique to solve it exploits an additional modeling assumption that justifies the real occurrence of

How to cite this book chapter:

Flötteröd, G. 2016. MATSim as a Monte-Carlo Engine. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) *The Multi-Agent Transport Simulation MATSim*, Pp. 327–336. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.48. License: CC-BY 4.0

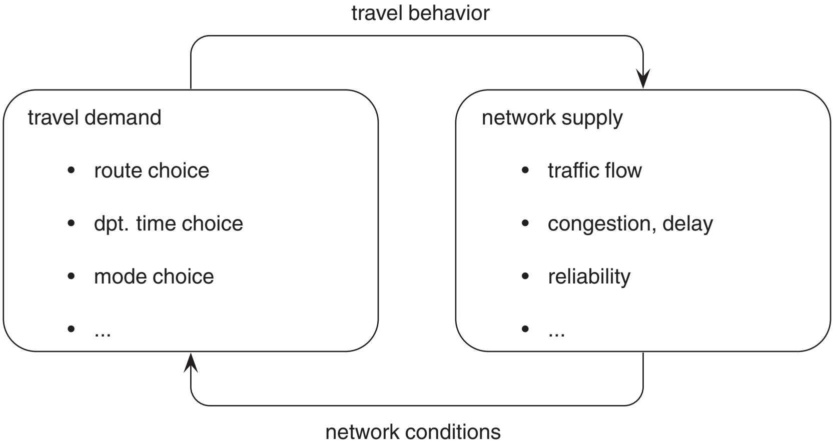

Figure 48.1: Demand/supply perspective on MATSim.

Algorithm 48.1 Iterative scheme to reach demand/supply consistency

```
1. Create a synthetic agent population.
2. Create a synthetic environment.
3. Iterate:
    (a) All agents choose some planned travel behavior.
    (b) All agents execute their travel plans.
    (c) All agents see the resulting network conditions.
```

demand/supply consistency: travelers adjust their behavior for their own benefit and only stop doing that when further improvement is insubstantial. Demand/supply consistency characterizes the outcome of this process jointly for all travelers.

Now, consider Algorithm 48.1, which displays the high-level simulation logic of MATSim. This is indeed a logic that iteratively adjusts travel demand. If this logic adjusts the simulated behavior of the simulated travelers until further simulated improvements are insubstantial, then this logic should approach a state of demand/supply consistency. That is, Algorithm 48.1 may be a valid solution method for the model system shown in Figure 48.1. However, that model system does not specify how demand and supply become consistent; it merely specifies that this eventually happens. The only modeling assumption made is that some process of this type exists. The purpose of Algorithm 48.1 is not to mimic this (unspecified) process; it only identifies the final outcome of that process.

The fact that Algorithm 48.1 mimics real, urban, day-to-day dynamics invites misleading interpretations of the underlying model system. In particular, it is a misconception that there is more than a superficial resemblance between the “learning agents” in MATSim and the (hardly understood) learning processes of real humans. If the notion of “learning” has to be used at all when interpreting Algorithm 48.1, it should be understood as “moving a MATSim model closer to its solution point”. Also see Section 97.3.5.

The remainder of this chapter phrases these statements more technically and explains their implications for the interpretation of MATSim outputs. This presentation is in parts a more technical reformulation of Chapter 47.

48.2 Relaxation as a Stochastic Process

48.2.1 Probabilistic Model Components

Algorithm 48.1 can be written more formally. Denoting the iteration index by k, the following happens in every iteration:

1. All agents choose some planned travel behavior, resulting in the travel demand D^k of the entire agent population.
2. All agents execute their travel plans, resulting in the (time-of-day dependent) network conditions C^k.
3. All agents see the resulting network conditions C^k. As a result, the information Z^k is now available to all agents.

The variables D and Z apply to the population as a whole, comprising all agents. Similarly, the variable C represents network conditions for an entire day and for the entire physical system. Given MATSim's high level of detail, one can think of D, C and Z as placeholders for arbitrarily large and complex data structures. Under MATSim's standard conditions, D corresponds to the set of selected plans, C to the collection of all events and Z to the full plans file including the scores.

Step 1 evaluates the (stochastic) travel behavior model of each agent. Technically, this comprises (i) an optional update of the plan choice set and (ii) the choice of one plan to be executed. Symbolically, this is written as

$$D^k \sim P(D \mid Z^{k-1}), \tag{48.1}$$

meaning that the travel demand of iteration k follows a probability distribution that is conditional on the information Z^{k-1} available to the agents at the end of iteration $k-1$.

Step 2 runs the (stochastic) mobility simulation that moves all agents jointly through the network. In symbols, this becomes

$$C^k \sim P(C \mid D^k), \tag{48.2}$$

meaning that the network conditions of iteration k follow a probability distribution that is conditional on the demand D^k.

Step 3 updates the (possibly stochastic) information available to all agents using the new network conditions C^k. This is written as

$$Z^k \sim P(Z \mid C^k, Z^{k-1}). \tag{48.3}$$

That is, the new information Z^k is not only a transformation of the current network conditions C^k but may also be based on the previously available information Z^{k-1}.

The conditional distributions Equation (48.1)–(48.3) are detailed elsewhere in this book: Chapter 49 describes the plan selection mechanisms leading to $P(D \mid C^{k-1})$, Chapter 50 explains the physical processes underlying $P(C \mid D^k)$, and Chapter 3 specifies at least some of the information update logic behind $P(Z \mid C^k, Z^{k-1})$. A greater level of detail is, however, not necessary in this chapter.

48.2.2 Markov Chain Perspective

Algorithm 48.1 constitutes a discrete time stochastic process. "Discrete-time" because it evolves in stages (from iteration to iteration), stochastic because it evaluates stochastic models. Further, one iteration of this process requires only information about the previous iteration's outcome. This allows the expression of Algorithm 48.1 in terms of a "Markov chain" (Ross, 2006).

In symbols, let X^k be the Markov chain's stochastic state during stage k, and let $P(X^k = x)$ be the probability that the chain is in the concrete state x. Further, let T_y^x be the probability that the chain enters state x in its next stage given that it is currently in state y. The transition from one stage to the next can then be expressed as follows:

$$P(X^{k+1} = x) = \sum_{y} P(X^{k} = y) \cdot T_{y}^{x}. \tag{48.4}$$

Each argument of the sum expresses the probability of the chain being in one particular state y and then entering x. The overall probability of arriving in x results from summing up these probabilities.

Markov chains tend, under certain assumptions sketched in the next section, to stabilize after sufficient iterations, in the sense that a long-term probability $\Pi(x)$ of encountering the process in state x exists. This stationary distribution satisfies

$$\Pi(x) = \sum_{y} \Pi(y) \cdot T_{y}^{x}, \tag{48.5}$$

which essentially results from removing the k-indices from Equation (48.4). Intuitively, removing the stage-indices k means that Equation (48.5) now applies, in the long term, for any stage k.

Given that the long-term behavior of Algorithm 48.1 shapes the predictions made with MATSim, and updated information its characterization in terms of the stationary distribution of a corresponding Markov chain is of interest. To obtain a Markov chain representation of Algorithm 48.1, one needs to specify (i) what variables in MATSim represent the states of that chain and (ii) what transition distribution underlies the MATSim simulation logic.

A state variable must provide sufficient information to simulate a process further into the future. Candidates for MATSim's state space arethe demand D, the network condition C and the information Z. Of these, only the information Z qualifies as a state variable: If one knows Z^k, it is possible to draw the next day's travel demand D^{k+1} based on Equation (48.1), to insert this demand into Equation (48.2) and obtain the network conditions C^{k+1} and to finally use both C^{k+1} and Z^k to obtain an updated Z^{k+1} through Equation (48.3). This last step is what disqualifies D and C as state variables because an evaluation of Equation (48.3) is impossible without having Z in the state space.

Letting $X^k = Z^k$, the transition distribution hence needs to express how the information Z^k available to the population in iteration k carries over to the information Z^{k+1} available in iteration $k+1$. This relationship is given by

$$T_{y}^{x} \quad = \quad \sum_{c} \sum_{d} P(Z^{k+1} = x \mid C^{k} = c, Z^{k} = y)\, P(C^{k} = c \mid D^{k} = d)\, P(D^{k} = d \mid Z^{k} = y). \tag{48.6}$$

Each argument of the double sum represents the probability of one particular sequence of given information y, resulting travel demand d, resulting network conditions c and updated information x. The double sum over all possible travel demand realizations d and network conditions c then accounts for the fact that there are many different such sequences through which one can start out at y and end up at x.

This completes the representation of MATSim in terms of a Markov chain. The next section illustrates practical uses of this representation.

48.3 Existence and Uniqueness of MATSim Solutions

The long-term (stationary) behavior of a Markov chain can be derived from its transition function. This also leads to useful insights for MATSim, despite of the complexity of its transition function Equation (48.6).

Two key properties are aperiodicity and irreducibility. Informally, a Markov chain is aperiodic if all of its states can be visited at irregular times; Figure (48.2) provides an example. It is irreducible if it can reach any other state from any given initial state with one or more transitions; see Figure (48.3) for an example. Aperiodicity and irreducibility are essential when it comes to long-term predictions, where (i) aperiodicity guarantees that the concrete iteration in which one evaluates the simulation does not play a role and (ii) irreducibility ensures that every possible future system state can be reached (predicted) by the simulation. If both properties are given, the Markov chain has the following properties (Ross, 2006):

1. A unique stationary distribution exists. The simulation process attains this distribution after many iterations, independently of its initial state.
2. It is feasible to compute statistics of the stationary distribution from a single simulation run, meaning that it is not necessary to run replications.

With respect to MATSim, the following holds:

- Periodicity is already broken if a nonzero probability of staying in the same state exists. This is likely to be the case in MATSim, for instance because the following sequence of events may occur by chance: (i) No agent uses plan innovation, (ii) all agents select the same plan as in the previous iteration, (iii) the mobility simulation creates identical congestion and travel time patterns as before, meaning that Z^k from Equation (48.3) remains the same as Z^{k-1}. Practically, this means that all plan scores stay unchanged.
  More intuitively: Even if the system returns multiple times exactly to a state where it has been before, it unlikely that it does so in the same number of steps.
- With plan innovation (see Sections 4.5, 4.5.3 and 47.3.2.3) switched on, irreducibility cannot be postulated:
  Every time a new plan is added somewhere, the previous state space subspace where the plan was not available *cannot* be reached any more until that plan is removed; similarly, every time a plan is removed, the previous state space subspace where the plan was available cannot be

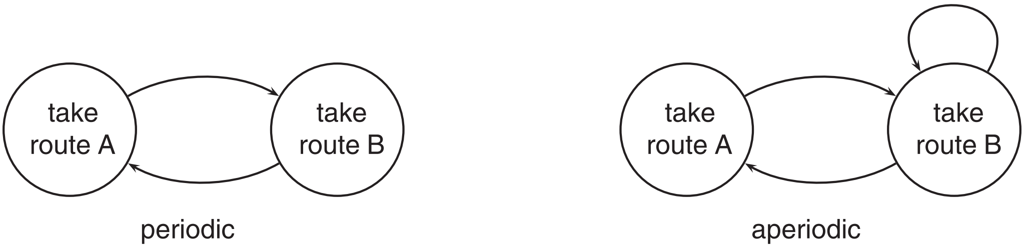

Figure 48.2: Example of (a)periodicity

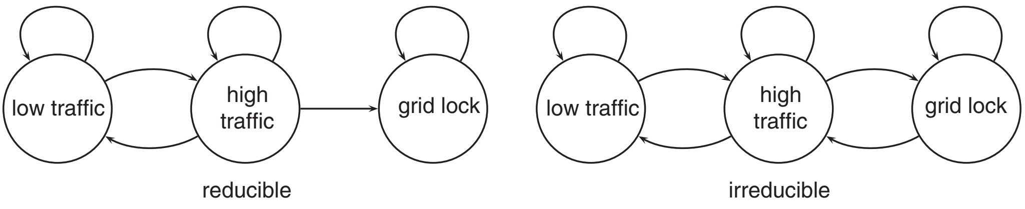

Figure 48.3: Example of (ir)reducibility

reached any more until the plan is re-created again. (Chapter 49 discusses this in greater detail and also suggests a solution for this problem.)

Even if plans creation and removal could be modeled such that irreducibility was guaranteed, the resulting process dynamics would be slow due to the state space size.

- With plan innovation switched off, MATSim in its standard configuration is likely to be irreducible. This is only "likely", because the notion of a "standard configuration" itself is not rigorously specified here. Arguments behind this follow Cascetta (1989), who presents a related result for a much simpler, trip-based traffic simulation that only allows for route choice. Observing that travel plans are, technically, paths in a rather complicated decision network, one can then carry this result over to MATSim. See also Nagel et al. (2000) and Flötteröd et al. (2011).
- When scores are additionally forced to their expected values (Section 3.3.4), the system eventually draws agent behavior from fixed choice distributions, thus varying independently from one iteration to the next.

  If config option `ChangeExpBeta` is used, some correlation is maintained between choices in subsequent iterations, even though the long-term choice distributions remain unchanged.

In summary, a mathematical framework exists allowing a rather rigorous characterization of the outcome of MATSim's relaxation process. It turns out that MATSim, in its current form, is not necessarily a "well-behaved" stochastic process; however, casting it into this framework enables a structured approach to developing the simulation logic further. An example of how to go about this is given in Chapter 49.

48.4 Analyzing Simulation Outputs

Many of the models used in MATSim are stochastic. Examples are the discrete choice models used for plan selection or the randomized selection of the next vehicle to enter a congested downstream link in the mobility simulation. The reason for this randomness is that real mobility and transportation processes are not completely understood. The insertion of randomness represents the uncertainty remaining in the modeling.

This uncertainty may apply to both (i) model inputs, meaning that random variables are computed once before a simulation run and then kept fixed (for instance, the random generation of a synthetic population) and to (ii) processes, meaning that random variables are computed throughout the simulation (for instance, the repeated evaluation of discrete choice models). Technically, if a MATSim scenario is simulated R times with different random seeds one obtains $r = 1 \ldots R$ independent simulation outputs y_r. Note that, while the raw outputs are plans and event files, the actual quantities for which y_r stands here are numerical in the majority of applications.

Given that one has used different random seeds, $y_1, \ldots, y_R$ constitute independent draws from a distribution $\Pi(Y)$. This means that if one performed a huge number of simulation runs and plotted a (possibly multidimensional) histogram of the y values, then this histogram would eventually attain the shape of $\Pi(Y)$. It is important to acknowledge that stochastic simulation outputs are a desirable consequence of stochasticity inserted elsewhere in the simulation; just as a deterministic model output is a truthful representation of its input consequences, a stochastic model output contains a truthful representation of the prediction uncertainty resulting from uncertainties in its input and its process specification.

To help intuition, one may think in the following of y_r as a large vector containing travel times on all links in all one-hour time bins as observed during the last iteration of the rth simulation run. Questions like these may then be asked:

- What travel times can one expect on average?
- What is the travel time variability?
- How probable are travel times beyond some threshold θ?
- ...

This list can be arbitrarily continued. It turns out that most (if not all) of these questions can also be expressed symbolically. For instance:

- What travel times can one expect on average?

$$\mathrm{E}\{Y\} = \sum_y y \cdot \Pi(Y = y) \tag{48.7}$$

  This asks for the expected value of the simulation output distribution.
- What is the travel time variability?

$$\mathrm{VAR}\{Y\} = \sum_y (y - \mathrm{E}\{Y\})^2 \cdot \Pi(Y = y) \tag{48.8}$$

  This asks for the variance (or, for multidimensional outputs, the variance-covariance matrix).
- How probable are travel times beyond some threshold θ?

$$\Pr(Y \geq \theta) = \sum_y \mathbf{1}(y \geq \theta) \cdot \Pi(Y = y) \tag{48.9}$$

  This expression merely sums up the probabilities of all simulation outputs that exceed the threshold.
- ...

This enumeration of symbols reveals a common structure. The mathematical formulation of each question can be written in the form

$$\sum_y m(y) \cdot \Pi(Y = y) \tag{48.10}$$

with different specifications of $m(y)$ (see Table 48.1).

By definition, Equation (48.10) is the expectation $\mathrm{E}\{m(Y)\}$ given that Y is distributed according to its stationary distribution $\Pi(Y)$. Combining this with the observation that the mean over a

quantity of interest	corresponding $m(y)$
$\mathrm{E}\{Y\}$	y
$\mathrm{VAR}\{Y\}$	$(y - \mathrm{E}\{Y\})^2$
$\Pr(Y \geq \theta)$	$\mathbf{1}(y \geq \theta)$
$\dots$	$\dots$

Table 48.1: Examples of m functions.

sample converges to its expectation as the number of samples grows (the Law of Large Numbers), one obtains

$$\begin{aligned} \mathrm{E}\{m(Y)\} &= \sum_y m(y) \cdot \Pi(Y = y) && (48.11) \\ &= \lim_{R\to\infty} \frac{1}{R} \sum_{r=1}^{R} m(y_r) && (48.12) \\ &\approx \frac{1}{R} \sum_{r=1}^{R} m(y_r) \qquad \text{for a finite } R, && (48.13) \end{aligned}$$

where the simulation outputs y_r, $r = 1 \ldots R$, are independent draws from $\Pi(Y)$.

Now recall that initially certain questions about simulation outputs were asked. The Equation (48.11)(first row) represents exactly these questions in a formal way–and Equation (48.13) (last row) provides a simple method for computing answers to these questions. It reads as follows:

1. Define the function $m(y)$ that represents the question of interest.
2. Perform R independent simulation runs and obtain the outputs $y_1, \ldots, y_R$.
3. Compute $m(y_r)$ for all $r = 1 \ldots R$ and average these numbers.

Returning to the example questions, one thus obtains the following:

- What travel times can one expect on average?

$$\mathrm{E}\{Y\} \approx \frac{1}{R} \sum_{r=1}^{R} y_r \qquad (48.14)$$

  Not surprisingly, this turns out to be the mean value over all simulated travel times.

- What is the travel time variability?

$$\mathrm{VAR}\{Y\} \approx \frac{1}{R} \sum_{r=1}^{R} (y_r - \mathrm{E}\{Y\})^2 \qquad (48.15)$$

  This is the empirical variance of the simulated travel times. (Note that in practice $\mathrm{E}\{Y\}$ needs to be replaced by its estimator.)

- How probable are travel times beyond some threshold θ?

$$\Pr(Y \geq \theta) \approx \frac{1}{R} \sum_{r=1}^{R} \mathbf{1}(y_r \geq \theta) \qquad (48.16)$$

  This divides the number of times the threshold was exceeded by the total number of experiments, i.e. it yields the frequency of the event of interest.

- ...

Revisiting Section 48.3, it may be possible to make these computations more efficient. If (i) there is no uncertainty in the model inputs and (ii) the simulation uses fixed choice sets, then it could be feasible to compute the above statistics by averaging over many stationary iterations of a single simulation run instead of having to run a large number of replications to convergence.

Practically, all of this is just a starting point. Important questions, such as how precise these estimates are, how many runs one needs to obtain a certain level of precision, etc. are not answered here; Ross (2006) is a good starting point for further reading.

48.5 Summary

This chapter attempted to clarify certain mechanisms underlying MATSim's iterative solution scheme. The specification of MATSim's model (components) was distinguished from MATSim's iterative solution algorithm. It was stressed that the behavioral day-to-day interpretation of MATSim is not to be taken literally; realism can only be expected from the long-term process behavior.

This long-term behavior was then related to the properties of the iteration logic using the the Markov chain formalism. MATSim was phrased as such a chain, with its state space comprised of the information available for replanning. This representation was exploited to observe that the long-term distribution of MATSim is likely to exist and be unique if the plan choice sets are a priori fixed.

It further was explained that (i) there are good reasons for the stochasticity both in MATSim's inputs and outputs and that (ii) instead of avoiding stochasticity where it constitutes a truthful representation of uncertainty, one should access adequate statistical techniques to make sense of it.