CHAPTER 49

Choice Models in MATSim

Gunnar Flötteröd and Benjamin Kickhöfer

This chapter attempts to reconcile MATSim's mechanisms of plan "mutation", "selection" and "execution", borrowed from evolutionary computation, with a discrete choice modeling perspective.

Discrete choice theory originates in work by Luce and Suppes (1965) and McFadden (1975); Ben-Akiva and Lerman (1985) and Train (2003) are the two standard textbooks in this area. The theory is mainly used to describe individual choices among mutually exclusive alternatives. Discrete choice models typically do not predict individual choices with complete accuracy. Luce and Suppes (1965) distinguishes between two possible interpretations of this phenomenon: (1) People choose randomly among their alternatives, rendering their behavior inherently unpredictable. (2) The choice only *appears* to be random since the model does not perfectly capture the decision process and its relevant decision variables. Both perspectives lead to the same result, the introduction of probabilistic choice models.

Let $\mathcal{U}_n$ be the universal set of all plans that may ever be considered by agent n and let C_n denote that agent's concrete plan choice set. The choice set independent probability that agent n selects plan i for execution can then be written as

$$P_n(i \mid \mathcal{U}_n) = \sum_{C_n \subset \mathcal{U}_n} P_n(i \mid C_n) \cdot P_n(C_n \mid \mathcal{U}_n), \tag{49.1}$$

explained as follows. Selecting a plan requires a plan choice set. The term $P_n(C_n \mid \mathcal{U}_n)$ represents the probability that this concrete choice set is C_n, which must be a subset of $\mathcal{U}_n$. Technically, the MATSim plan *innovation* modules draw from this distribution. The term $P_n(i \mid C_n)$ represents the probability that agent n selects plan i given that its concrete choice set is C_n. Technically, the MATSim plan *selection* modules draw from this distribution. The product of these terms thus

How to cite this book chapter:

Flötteröd, G and Kickhöfer, G. 2016. Choice Models in MATSim. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) *The Multi-Agent Transport Simulation MATSim*, Pp. 337–346. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.49. License: CC-BY 4.0

represents the joint probability that choice set C_n is available and that plan i is chosen from that set. The probability of selecting plan i independently of the concrete choice set then results from summing up the probabilities of selecting it in the presence of all possible choice sets $C_n \subset \mathcal{U}_n$.

It is evident in Equation (49.1) that an agent's behavior depends on both the choice model $P_n(i \mid C_n)$ and the way the choice set is generated through $P_n(C_n \mid \mathcal{U}_n)$. The following two sections will look at each step in more detail.

49.1 Evaluating Choice Models in a Simulated Environment

This section's discussion focuses on the choice distribution $P_n(i \mid C_n)$ for given choice sets. In MATSim, a plan is evaluated and selected based on the score as the sole property of the plan. This is only a technical specification; the scoring and selection protocols are responsible for representing adequate perceptional and behavioral mechanisms. The notions of "choice" and "selection" are subsequently used interchangeably (cf. Section 4.5.2).

The usual selection protocol of MATSim resembles a MNL choice model. Letting S_{ni} be the score of plan i of agent n, one has

$$P_n(i \mid C_n) = \frac{e^{\mu S_{ni}}}{\sum_{j \in C_n} e^{\mu S_{nj}}} \tag{49.2}$$

with μ controlling the preference for higher scores. It is set to one in the remainder of this section.

49.1.1 Case 1: Score is or Converges Towards a Deterministic Value

If the score of a plan was a deterministic number representing an expected value, then Equation (49.2) would constitute a plain MNL choice model with μ taking the role of a scale parameter (see, e.g., Train, 2003, p.45). Such behavior can be approximated in MATSim by the following configuration settings:

- A fixed choice set C_n is eventually obtained by setting the configuration option `fractionOfIterationsToDisableInnovation` below one, meaning that innovation (see Section 49.2) will be switched off for the remaining fraction of iterations beyond the configured value.
- Score convergence to its expectation value can be achieved by setting the configuration option `fractionOfIterationsToStartScoreMSA` below one, meaning that scores will be averaged according to MSA (Method of Successive Averages) for the remaining fraction of iterations.

49.1.2 Case 2: More General

Without the particular configuration mentioned in the previous section, things are somewhat more complicated. Assume that the attribute vector $\mathbf{x}_{ni}$ of the alternatives in Equation (49.2) is defined through (a transformation of) the network conditions observed during the last iteration(s). Assume further that the score is a linear function of these attributes:

$$\begin{aligned} S_{ni} &= \boldsymbol{\beta}^T \mathbf{x}_{ni} && (49.3) \\ &= \boldsymbol{\beta}^T (\mathrm{E}\{\mathbf{x}_{ni}\} + \boldsymbol{\eta}_{ni}) && (49.4) \end{aligned}$$

where $\boldsymbol{\beta}$ is a coefficient vector, superscript T denotes the transpose and $\boldsymbol{\eta}_{ni}$ is a zero mean random vector. In the general case of S_{ni} being a random variable and not just an expected value, one obtains a mixture-of-logit model with the choice distribution

$$P_n(i \mid C_n) \quad = \quad \int \frac{\exp\left(\boldsymbol{\beta}^T \mathrm{E}\{\mathbf{x}_{ni}\} + \boldsymbol{\beta}^T \boldsymbol{\eta}_{ni}\right)}{\sum_{j \in C_n} \exp\left(\boldsymbol{\beta}^T \mathrm{E}\{\mathbf{x}_{nj}\} + \boldsymbol{\beta}^T \boldsymbol{\eta}_{nj}\right)} p(\boldsymbol{\eta}_n) d\boldsymbol{\eta}_n \qquad (49.5)$$

where $p(\boldsymbol{\eta}_n)$ is the probability density function of $\boldsymbol{\eta}_n = (\boldsymbol{\eta}_{ni})_i$, i.e., the joint probability density function of the random disturbances of all alternatives of individual n (Train, 2003, Section 6). This formulation comprises most, if not all, MATSim configurations currently used. It represents the `ExpBetaPlanSelector` and the equivalent `ExpBetaPlanChanger`. It also comprises the `BestPlanSelector`, because that is equivalent to the `ExpBetaPlanSelector` with a very large (infinite) μ. Arbitrary score averaging schemes are also included; this only leads to different instances of $p(\boldsymbol{\eta}_n)$.

Mainstream applications of mixture-of-logit models attempt to combine the tractability of closed-form logit models with the flexibility of simulating arbitrary $p(\boldsymbol{\eta}_n)$ distributions. The distribution of $\boldsymbol{\eta}_n$ is often as simple as a multivariate normal because this already allows for the introduction of rich correlation structures into the underlying random utilities. In MATSim, however, the simulated error term $\boldsymbol{\eta}_n$ is extremely complicated. Revisiting Equation (49.4), it defines the variability of the scores resulting from the fact that the simulated network conditions are stochastic. The distribution from which these network conditions are drawn is defined implicitly through the mobility simulation. It is not available in closed form; one can only draw from it.

Additional complexity results from the simulated network conditions being, in turn, the consequence of simulated travel behavior that is again defined through Equation (49.5). Just as a representation of the mutual demand/supply dependency is essential in transport planning, the circular definition of the $\boldsymbol{\eta}_n$ terms adds realism to MATSim:

1. Assume one could somehow make the simulated network conditions more realistic. The result would be a more realistic distribution $p(\boldsymbol{\eta}_n)$ of the simulated error terms.
2. All else equal, increasing the realism of $p(\boldsymbol{\eta}_n)$ in Equation (49.5) would also increase the realism of the resulting choice distribution.
3. This, in turn, would lead to the selection of more realistic travel plans, meaning that their execution would result in even more realistic network conditions.

However, this positive feedback only applies to the extent to which the error terms in the behavioral model are indeed mobility simulation outputs. Simulated travel time (variability) is such a case. Unobserved preferences of the decision maker, however, are not an output of the mobility simulation and hence need to be differently captured.

It is by no means obvious how the randomness of the simulated network conditions should enter $\boldsymbol{\eta}_n$. The notion of "learning" again enters the picture, cf. Chapter 48. However, if the simulation iterations really represented simulated days then a real human learning model would be needed to combine a sequence of past network conditions into an instantaneous $\boldsymbol{\eta}_n$ realization. Without a sound instance of such a learning model, a learning-based interpretation of Equation (49.5) cannot be given.

Another perspective on this problem is possible, continuing the arguments of Chapter 48. It is stated there that the purpose of MATSim's iterative mechanism is merely to attain a realistic stationary distribution. If so, then the sole purpose of the simulated $\boldsymbol{\eta}_n$s is to yield a realistic stationary choice distribution. To illustrate this perspective, consider the following moving-average score updating rule:

$$\bar{S}_{ni}^{k+1} = \begin{cases} \alpha S_{ni}^{k} + (1-\alpha)\bar{S}_{ni}^{k} & \text{if } n \text{ chose plan } i \\ \bar{S}_{ni}^{k} & \text{otherwise} \end{cases} \tag{49.6}$$

where $\bar{S}^k$ is the filtered score of iteration k and S^k is the concrete score observed in that iteration. The learning rate α controls how strongly the filtered score is smoothed out, thus controlling the variability of $\boldsymbol{\eta}_n$. MATSim enables this mechanism through the `learningRate` parameter.

Assuming – for simplicity – that the unfiltered stationary score S^{∞} fluctuates in stationary conditions independently from iteration to iteration around its expected value, one can derive the following (as demonstrated in this chapter's appendix):

$$\mathrm{E}\{\bar{S}_{ni}^{\infty}\} = \mathrm{E}\{S_{ni}^{\infty}\} \tag{49.7}$$

$$\mathrm{VAR}\{\bar{S}_{ni}^{\infty}\} = \frac{\alpha}{2-\alpha}\mathrm{VAR}\{S_{ni}^{\infty}\}. \tag{49.8}$$

This means that the filtered score is unbiased with respect to the underlying score process and that its variance is in the interval from zero to the variance of the unfiltered score, depending on the chosen α. There is no need to justify this through a learning process. One has merely constructed a parametrization of the distribution $p(\boldsymbol{\eta}_n)$. In the resulting mixture model Equation (49.5), α should be estimated from real data, just like any other model parameter. Even though this apparently has not yet been attempted, techniques necessary for such an endeavor are, in principle, available (Gourieroux et al., 1993).

49.1.3 Expected Maximum Utility

The expected maximum utility of Equation (49.5) is relevant to the microeconomic interpretation of MATSim outputs. A recipe for its computation is described next. Let

$$U_i = V_i + \eta_i + \varepsilon_i \tag{49.9}$$

using the shortcuts $V_i = \boldsymbol{\beta}^T \mathrm{E}\{\mathbf{x}_{ni}\}$, $\eta_i = \boldsymbol{\beta}^T \boldsymbol{\eta}_{ni}$, letting ε_i be the Gumbel error assumed by the multinomial logit model and dropping the n index for brevity. Following this notation, Equation (49.4) is rewritten as

$$S_i = V_i + \eta_i. \tag{49.10}$$

One needs to distinguish between the score of a plan when it is selected and its updated score after it has been executed. To start, it is assumed that the agent receives an expected maximum utility depending on the scores at the time of plan selection, not after plan execution. The expected maximum utility of Equation (49.5) could then be expressed as follows:

$$\mathrm{E}\left\{\max_{i \in C_n} U_i\right\} = \mathrm{E}\left\{\max_{i \in C_n} V_i + \varepsilon_i + \eta_i\right\} \tag{49.11}$$

$$= \mathrm{E}_{\eta}\left\{\mathrm{E}_{\varepsilon}\left\{\max_{i \in C_n} V_i + \varepsilon_i + \eta_i \mid \eta\right\}\right\} \tag{49.12}$$

$$= \mathrm{E}_{\eta}\left\{\ln \sum_{i \in C_n} e^{V_i + \eta_i}\right\}. \tag{49.13}$$

where the law of total expectation is used and E_ε and E_η represent expectations with respect to ε and η, respectively. The remaining argument of the expectation is the expected maximum utility of a multinomial logit choice model given the systematic utilities $V_i + \eta_i$. This expression can be numerically approximated by averaging over many realizations of η_i (i.e. over simulation iterations):

$$\mathrm{E}_\eta \left\{ \ln \sum_{i \in C_n} e^{V_i + \eta_i} \right\} \approx \frac{1}{R} \sum_{r=1}^{R} \ln \sum_{i \in C_n} e^{V_i + \eta_i^r} \tag{49.14}$$

where η_i^r is the realization of η_i in iteration r. This expression holds regardless of the functional form of the mobsim-generated mixture distribution.

Now, one needs to account for the fact that agents can only evaluate *past* information when making a choice leads to a *future* score payoff. Recalling that score variability is represented by the η_i variables in Equation (49.5),

$$\eta_i = \hat{\eta}_i + \gamma_i \tag{49.15}$$

is written with η_i contributing to the score actually received Equation (49.10), $\hat{\eta}_i$ being the agent's prediction of that and γ_i being a random variable capturing the difference between the two.

To express the expected maximum *experienced* utility, one hence needs to add (an estimator of) the expectation of γ_i to Equation (49.14). Using Equation (49.15) and Equation (49.10), one obtains

$$\begin{aligned} \gamma &= \eta_i - \hat{\eta}_i && (49.16)\\ &= (\eta_i + V_i) - (\hat{\eta}_i + V_i) && (49.17)\\ &= S_i - \hat{S}_i && (49.18) \end{aligned}$$

where $\hat{S}_i$ can be interpreted as the agent's prediction of the selected alternative i's score. The expectation of this quantity can again be approximated by averaging, resulting in the following estimator of the expected maximum experienced utility, with $i(r)$ indicating the alternative that was selected in iteration r:

$$\mathrm{E}\left\{ \max_{i \in C_n} U_i^{\text{experienced}} \right\} \approx \frac{1}{R} \sum_{r=1}^{R} \ln \sum_{i \in C_n} e^{\hat{S}_i^r} + \frac{1}{R} \sum_{r=1}^{R} \left(S_{i(r)}^r - \hat{S}_{i(r)}^r \right). \tag{49.19}$$

The second sum of this expression estimates a "cost of uncertainty"; the less predictable the network conditions (and thus the selected plan's future score), the worse off an agent is on average. The usefulness of this expression depends on the simulation's ability to create realistic network condition variability, for instance along the lines of the last paragraphs of Section 49.1.2. Section 51.2.5.5 discusses this a bit further.

49.2 Evolution of Choice Sets in a Simulated Environment

49.2.1 Overview

The choice set of agents can in principle be computed a priori and then held fixed during a MATSim simulation run. However, the pre-computation would have to be done for every relevant system state (e.g., before and after a policy change). Alternatively, MATSim can be used to generate agents' choice sets within the iterative loop (Section 1.2).

As Equation (49.1) shows, the generation of the choice sets affects the simulated choices. The simplest illustration of this mechanism isthat alternatives that never appear in the choice set cannot be chosen. Similarly, including certain alternatives with a low (high) probability in the choice set decreases (increases) their probability of being chosen, given that the choice model is not changed. When a policy study's synthetic choice sets are very different from the alternatives considered in the real world, it is unlikely that the simulation will display correct aggregated quantities or useful sensitivities for policy measures.

These types of biases are well-known in the discrete choice community, even though the focus is there, arguably, more on estimation than simulation. The problem is particularly acute in route choice modeling because the combinatorial size of the universal route choice set prohibits its enumeration. Drawing further from the discrete choice literature (specifically Frejinger and Bierlaire, 2010), different interpretations can be given to "plan mutation" and "plan innovation" in MATSim.

An interpretation of mutation and innovation as *perceptional models* of travel plan choice set formation is hindered by the need to validate them against real and unobservable choice sets. Alternatively, one may assume that travelers consider the universal choice set and that the choice of unfeasible alternatives is impeded by correspondingly low utility values. In this setting, mutation and innovation constitute *sampling techniques* serving the *computational* purpose of reducing the universal choice set to a small, representative subset. However, one still faces the problem from above that the concrete sampling protocol has a concrete effect on the simulated behavior. The cure when *estimating* choice models is to correct for the sampling based on known sampling probabilities (e.g. Ben-Akiva and Lerman, 1985, Chapter 9), even though these probabilities can be rather difficult to obtain (Flötteröd and Bierlaire, 2013; Frejinger et al., 2009a). The problem appears to be less explored when it comes to *simulation*.

MATSim's currently implemented mutation and innovation procedures constitute concrete, yet heuristic, approaches to the choice set generation problem, aiming at valid predictions at the system level. Possible biases induced by these procedures can, however, be difficult to quantify. For example, the current MATSim implementation might, under certain conditions, yield incomplete choice sets and correlated alternatives (also see Chapter 51). To mitigate the effect of strong correlations between alternatives within the choice set, so-called diversity increasing re-planning modules have been tested (see, e.g., Nagel et al., 2014). In the same context, Grether (2014, Chapter 6) and Neumann et al. (2013) have tested path size logit approaches (see, e.g., Daganzo and Sheffi, 1977; Frejinger and Bierlaire, 2007) to maintain diversity in the choice set by penalizing similar alternatives. Still, these approaches are–as of now–ad-hoc solutions, with little theoretical foundation in the simulation context.

It thus seems worthwhile to revisit the plan choice set generation problem from a statistical perspective. The goal of the following presentation is more to establish a corresponding mindset than deliver a complete solution.

49.2.2 Towards Unbiased Choice Set Generation

To make the simulated long-term (stationary) plan choice independent of the plan choice set generation, one may require the following stationary choice distribution:

$$P_n(i \mid \mathcal{U}_n) \quad = \quad \frac{e^{\mu S_{ni}}}{\sum_{j \in \mathcal{U}_n} e^{\mu S_{nj}}}, \tag{49.20}$$

meaning that plans are selected from the universal choice set $\mathcal{U}_n$.

Denoting by $P(C_n \to C'_n)$ the probability that plan mutation/innovation turns the choice set C_n into C'_n, it is possible to enforce the long-term choice distribution Equation (49.20) through an application of the MH (Metropolis-Hastings) algorithm (Hastings, 1970, see also Flötteröd and Bierlaire (2013) for a related approach to a similar problem).

The MH algorithm specifies the transition distribution of a Markov chain so that a desired stationary distribution of that chain is reached. Given that Chapter 48 has established a formulation of MATSim as such a chain, the MH machinery can hence be inserted into the MATSim iterations. A simplification made in the following is that the choice distribution of agent n is considered independent of all other agents.

To make this concrete, let the state space of the algorithm be the tuple $(C_n, i \in C_n)$ consisting of choice set and resulting choice. During each (MATSim) iteration, one first draws a new choice set C'_n, then draws a new choice $i' \in C'_n$ according to the usual model (49.2) and finally accepts the new state (C'_n, i') with probability

$$\phi[(C_n, i), (C'_n, i')] \quad = \quad \min\left\{\frac{P_n(i' \mid \mathcal{U}_n)}{P_n(i \mid \mathcal{U}_n)} \cdot \frac{P(C'_n \to C_n)P_n(i \mid C_n)}{P(C_n \to C'_n)P_n(i' \mid C'_n)}, 1\right\} \tag{49.21}$$

and rejects it otherwise (meaning that the original choice set C_n and choice $i \in C_n$ are maintained).[1] Intuitively, the first fraction introduces a preference for states comprising a more probable choice and the second fraction corrects for the way transitions between states are proposed.

Assume that the plan innovation yields exactly one new plan i_{in} through a against the last iteration. Let the corresponding plan innovation distribution be approximated by $e^{\mu_{\text{inno}} S_{ni}} / \sum_{j \in \mathcal{U}_n} e^{\mu_{\text{inno}} S_{nj}}$ with a very large μ_{inno}. Assume further that i_{in} replaces exactly one uniformly selected plan i_{out}, which implies that the choice set size J is constant and exclude for simplicity the case that the best response innovation reconstructs the removed plan exactly. This leads to

$$P(C_n \to C'_n) \quad = \quad \frac{1}{J} \cdot \frac{e^{\mu_{\text{inno}} S_{ni_{\text{in}}}}}{\sum_{j \in \mathcal{U}_n} e^{\mu_{\text{inno}} S_{nj}}} \tag{49.22}$$

$$P(C'_n \to C_n) \quad = \quad \frac{1}{J} \cdot \frac{e^{\mu_{\text{inno}} S_{ni_{\text{out}}}}}{\sum_{j \in \mathcal{U}_n} e^{\mu_{\text{inno}} S_{nj}}}. \tag{49.23}$$

Inserting this as well as Equation (49.2) and Equation (49.20) into Equation (49.21), one obtains

$$\phi[(C_n, i), (C'_n, i')] \quad = \quad \min\left\{\frac{e^{\mu S_{ni'}}}{e^{\mu S_{ni}}} \cdot \frac{e^{\mu_{\text{inno}} S_{ni_{\text{out}}}} \dfrac{e^{\mu S_{ni}}}{\sum_{j \in C_n} e^{\mu S_{nj}}}}{e^{\mu_{\text{inno}} S_{ni_{\text{in}}}} \dfrac{e^{\mu S_{ni'}}}{\sum_{j \in C'_n} e^{\mu S_{nj}}}}, 1\right\} \tag{49.24}$$

$$= \quad \min\left\{e^{\mu_{\text{inno}}(S_{ni_{\text{out}}} - S_{ni_{\text{in}}})} \cdot \frac{\sum_{j \in C'_n} e^{\mu S_{nj}}}{\sum_{j \in C_n} e^{\mu S_{nj}}}, 1\right\} \tag{49.25}$$

$$\overset{\mu_{\text{inno}} \to \infty}{=} \quad \begin{cases} 1 & \text{if } S_{ni_{\text{out}}} \geq S_{ni_{\text{in}}} \\ 0 & \text{otherwise.} \end{cases} \tag{49.26}$$

[1] The acceptance probability $\phi(X \to X')$ in MH sampling is calculated as

$$\min\left(\frac{w(X') \cdot p_{\text{propose}}(X' \to X)}{w(X) \cdot p_{\text{propose}}(X \to X')}, 1\right),$$

where $p_{\text{propose}}(. \to .)$ is the probability that a certain transition is proposed, and $w(X)$, $w(X')$ are the relative weights of the respective states. It is important to note that w does not have to be normalized; it is sufficient if $w(X)/w(X') = p(X)/p(X')$. $P(C \to C')P(i'|C')$ is the probability that the choice set transitions from C to C' and that i' is selected from the resulting choice set.

Some care is needed when evaluating this expression because it assumes $S_{ni_{\text{in}}}$ and $S_{ni_{\text{out}}}$ to be independent random variables, whereas $S_{ni_{\text{in}}}$ is (due to the best response) always maximal among all alternatives given the most recent iteration. One should thus evaluate this expression by computing either score from the network conditions of a different, randomly selected stationary iteration.

This would allow the selection of plans according to (49.20) from an unconstrained choice set, even though one enumerates only a small subset of the full choice set, which is updated through a computationally efficient best-response mechanism.

In summary, one does the following for each agent in each iteration:

1. Randomly select a given plan for removal and compute a new best-response plan against the last iteration.
2. Is the new plan better than the one selected for removal, based on network conditions from two randomly selected stationary iterations?
   - Yes: Keep the previously selected plan and the previous choice set.
   - No: Remove the randomly selected plan from the choice set, add the newly generated plan and select a new plan from the new choice set.

This (at first glance perhaps counter-intuitive) logic can be explained as follows: Best-response creates new plans that are by chance better than any other plan in a given iteration. Best-response is thus corrected for by accepting the new plan only if it is by chance worse than a randomly selected alternative plan, with both plans being evaluated in randomly selected stationary iterations.

Note that the accuracy of this approach depends on the ability of the best-response plan innovation to create sufficiently variable plans, in the sense that the plan choice set innovation process is irreducible (Ross, 2006, see also Section 48.3 for an intuitive definition of irreducibility).

49.3 Summary

This chapter attempted to phrase MATSim's mechanisms of plan scoring, innovation, mutation and selection in the more mainstream terminology of discrete choice modeling. The implications of evaluating stochastic scores when selecting a plan were explained. The chapter also addressed how simulated choices depend on the way the underlying plan choice sets are generated, and different ways to address this problem were described.

The chapter clearly brought up more issues than it resolved. The take-away message, if any, is probably that even though MATSim agent behavior is roughly based on discrete choice modeling, one needs to be careful when assuming full consistency with a particular discrete choice model.

Appendix: Derivation of Filtered Score Statistics

Writing out the expectation:

$$\begin{aligned} \mathrm{E}\{\bar{S}_{ni}^{k+1}\} &= P_n(i)\mathrm{E}\{\alpha S_{ni}^k + (1-\alpha)\bar{S}_{ni}^k\} + (1-P_n(i))\mathrm{E}\{\bar{S}_{ni}^k\} && (49.27) \\ \Leftrightarrow \mathrm{E}\{\bar{S}_{ni}^{k+1}\} - \mathrm{E}\{\bar{S}_{ni}^k\} &= \alpha P_n(i)(\mathrm{E}\{S_{ni}^k\} - \mathrm{E}\{\bar{S}_{ni}^k\}). && (49.28) \end{aligned}$$

From $\lim_{k\to\infty} \mathrm{E}\{\bar{S}_{ni}^{k+1}\} - \mathrm{E}\{\bar{S}_{ni}^k\} = 0$ then follows

$$\lim_{k\to\infty} \left(\mathrm{E}\{S_{ni}^k\} - \mathrm{E}\{\bar{S}_{ni}^k\}\right) = 0. \qquad (49.29)$$

Proceeding in a similar way for the second moment:

$$\begin{aligned}
\mathrm{E}\{(\bar{S}_{ni}^{k+1})^2\} &= P_n(i)\mathrm{E}\{(\alpha S_{ni}^k + (1-\alpha)\bar{S}_{ni}^k)^2\} \\
&\quad +(1-P_n(i))\mathrm{E}\{(\bar{S}_{ni}^k)^2\} && (49.30) \\
\Leftrightarrow \lim_{k\to\infty} \mathrm{E}\{(\bar{S}_{ni}^{k+1})^2\} - \mathrm{E}\{(\bar{S}_{ni}^k)^2\} &= P_n(i)\alpha\Big[\alpha\mathrm{E}\{(S_{ni}^k)^2\} \\
&\quad +2(1-\alpha)\mathrm{E}\{S_{ni}^k\}^2 - (2-\alpha)\mathrm{E}\{(\bar{S}_{ni}^k)^2\}\Big] && (49.31)
\end{aligned}$$

From $\lim_{k\to\infty} \mathrm{E}\{(\bar{S}_{ni}^{k+1})^2\} - \mathrm{E}\{(\bar{S}_{ni}^k)^2\} = 0$ then follows

$$\lim_{k\to\infty}\mathrm{E}\{(\bar{S}_{ni}^k)^2\} = \frac{\alpha}{2-\alpha}\mathrm{E}\{(S_{ni}^k)^2\} - \frac{2-2\alpha}{2-\alpha}\mathrm{E}\{S_{ni}^k\}^2. \qquad (49.32)$$

The limiting variance then results from inserting of Equation (49.29) and Equation (49.32) into

$$\begin{aligned}
\lim_{k\to\infty}\mathrm{VAR}\{\bar{S}_{ni}^k\} &= \lim_{k\to\infty}\Big[\mathrm{E}\{(\bar{S}_{ni}^k)^2\} - \mathrm{E}\{\bar{S}_{ni}^k\}^2\Big] && (49.33) \\
&= \frac{\alpha}{2-\alpha}\mathrm{VAR}\{(S_{ni}^k)^2\}. && (49.34)
\end{aligned}$$