BHAMSLE: A Breakpoint Heuristic Algorithm for Maximum Simulated Likelihood Estimation of Advanced Discrete Choice Models

Tom Haering

Michel Bierlaire

STRC Conference Paper 2025 **May 21, 2025**

STRC | **25th Swiss Transport Research Conference**
Monte Verità / Ascona, May 14-16, 2025

BHAMSLE: A Breakpoint Heuristic Algorithm for Maximum Simulated Likelihood Estimation of Advanced Discrete Choice Models

Tom Haering
ENAC
EPFL
tom.haering@epfl.ch

Michel Bierlaire
ENAC
EPFL
michel.bierlaire@epfl.ch

May 21, 2025

Abstract

Maximum simulated likelihood estimation (MSLE) is inherently complex due to the presence of multiple local maxima, which hinder standard optimization methods. One solution is to reformulate MSLE as a mixed-integer linear program (MILP), enabling the use of combinatorial techniques to obtain globally optimal solutions. However, this approach introduces two difficulties: (1) the reliance on simulation-based approximation, which is unavoidable when dealing with continuous mixtures and does not pose a fundamental limitation, and (2) the computational intractability of large-scale instances. To address the latter, we adapt the Breakpoint Heuristic Algorithm (BHA), originally developed for choice-based pricing, which has proven effective in solving similar MILPs with high accuracy and reduced computational time. The resulting method, the BHA for MSLE (or BHAMSLE for short), exploits the problem's combinatorial structure by identifying decision-making breakpoints in a coordinate descent framework. Numerical experiments show that BHAMSLE significantly outperforms state-of-the-art global optimization methods that do not exploit this structure. Our approach delivers strong initialization points for estimation, yielding higher log-likelihoods, more stable and interpretable estimates, and improved recovery of latent segments, even in models with mixed parameters and restricted choice sets.

Keywords

Discrete Choice; Heuristic; Latent Class; Maximum Simulated Likelihood Estimation

Preferred citation

Haering, T. and Bierlaire, M. (2025) BHAMSLE: A Breakpoint Heuristic Algorithm for Maximum Simulated Likelihood Estimation of Advanced Discrete Choice Models, paper presented at the *25th Swiss Transport Research Conference (STRC 2025)*, Ascona, May 2025.

Contents

List of Tables

List of Figures

1 Introduction

The estimation of a discrete choice model (DCM) involves determining coefficient values that maximize the log-likelihood of the observed data. This process typically begins with initializing the coefficients, followed by iterative updates through an optimization algorithm until a predefined convergence criterion is met. Consequently, the initialization of coefficients—along with the chosen algorithm—directly influences the trajectory of the estimation process. For widely used models such as the logit, nested logit, and continuous mixtures of logit, this initialization rarely poses a significant issue. In the case of a logit model (and a nested logit with valid nest parameters), the likelihood function can even be shown to be concave, guaranteeing a unique global optimum.

Over the past decade, discrete mixture (or latent class) models have gained significant traction as a powerful framework for capturing unobserved heterogeneity in choice behavior. By explicitly segmenting the population into distinct latent groups, these models allow for the estimation of class-specific preference structures. Unlike continuous mixing approaches, discrete mixtures offer a more interpretable structure by associating individuals with discrete behavioral profiles, each assigned a probability of membership. This makes them a compelling alternative for analyzing preference heterogeneity while maintaining computational efficiency (Greene and Hensher, 2003; Boxall and Adamowicz, 2002). Despite these advantages, a major challenge of discrete mixtures is that they are prone to exhibit a multitude of local optima. Certain model specifications can result in hundreds of potential solutions, making the identification of a globally optimal set of parameters difficult (Peer *et al.*, 2016). As a result, the initialization of the coefficients and the specific estimation algorithm used heavily influence the identified solution.

For more advanced discrete choice models, such as continuous and discrete-continuous mixture models (latent class models with one or more continuous mixtures), the estimation challenge becomes even more pronounced. Unlike standard latent class models, these models lack closed-form expressions for choice probabilities, necessitating the use of simulation-based techniques like maximum simulated likelihood estimation (MSLE) (Train, 2003). When combined with the prevalence of multiple local optima, this reliance on simulation introduces a significant computational burden, further complicating the search for globally optimal parameters. As a result, the estimation of these models is not only sensitive to initialization but also requires robust optimization techniques to navigate their highly irregular likelihood landscapes.

To tackle the issue of numerous local optima, one possible approach is to perform

multiple estimations with diverse initializations. Jung and Wickrama (2008) emphasize the prevalence of local solutions in discrete mixture modeling and advocate for repeated random initialization as a necessary practice. Alternatively, rather than relying on computationally demanding repeated estimations, one can focus on developing more effective strategies for selecting initial values. A well-chosen starting point can mitigate the risk of convergence to suboptimal solutions and improve both the efficiency and reliability of the estimation process, whereas poor initialization can lead to convergence failures, suboptimal likelihood values, or even class misidentification, ultimately affecting the interpretability and validity of the estimated model. Amongst others, Lubke and Muthén (2005) underscore the need for systematic approaches to improve initialization strategies, as better starting points can significantly enhance estimation stability and reduce computational costs.

In complex optimization problems characterized by numerous local optima and reliance on simulation-based evaluations, heuristic global optimization methods have been widely applied. One such approach is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen *et al.*, 2003), a stochastic, population-based algorithm designed for black-box optimization. CMA-ES belongs to the family of evolutionary strategies and adapts the covariance matrix of its search distribution, enabling efficient exploration of high-dimensional, irregular objective functions. The algorithm iteratively samples candidate solutions from a multivariate normal distribution, updating its parameters based on their fitness to progressively guide the search toward the global optimum. Its versatility and robustness make it particularly well-suited for optimization tasks involving nonconvexity, noise, or simulation-based function evaluations. Other heuristic approaches proposed for global optimization include the variable neighborhood search (VNS) framework combined with a trust-region algorithm introduced by Bierlaire *et al.* (2010), which balances diversification and intensification to efficiently locate the global minimum, and Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995), which models solution candidates as a swarm moving through the search space based on social and cognitive influences.

While these broad methods have demonstrated strong performance across a range of optimization problems, they are not specifically tailored to discrete choice models and therefore do not fully exploit the structure inherent to such problems. A general solution approach developed for simulated maximum likelihood estimation specifically was introduced by Fernandez Antolin (2018), framing the problem as a mixed-integer linear program and demonstrating that MSLE can be seen as a choice-based optimization problem. Traditionally, such problems integrate a DCM to account for stochastic behavior

within an optimization context, often targeting endogenous parameters, such as the price of a product, to maximize revenue or other metrics. In the case of MSLE, one instead assumes fixed choice attributes, with the choice model parameters taking on the role of the decision variables, maximizing the simulated likelihood as the objective function. This perspective bridges a gap between choice-based optimization techniques and simulated likelihood estimation, suggesting potential for cross-applications between the two.

Rewriting the problem as a MILP offers the key benefit of ensuring a globally optimal solution. Nonetheless, this reformulation introduces certain difficulties. One is the need for simulation-based approximations—an unavoidable step when working with continuous mixtures, and thus not a fundamental drawback. A more pressing issue is the difficulty of solving large-scale instances exactly due to computational complexity. To address this, we develop a heuristic method that efficiently generates high-quality approximate solutions. These can be used as strong starting points for Newton-type algorithms, which can then further refine the solution and improve the chances of reaching a better local optimum, or even the global one.

To achieve this, we adapt the Breakpoint Heuristic Algorithm (BHA), originally introduced by Haering *et al.* (2024) for choice-based pricing. The BHA has demonstrated the ability to efficiently solve similar MILPs with high accuracy in significantly reduced computational time. The algorithm systematically explores local optima by identifying decision-making breakpoints and can be categorized as a coordinate descent method. Motivated by its effectiveness, we propose an adaptation of BHA tailored to the MSLE problem, termed the Breakpoint Heuristic Algorithm for MSLE (BHAMSLE). By explicitly exploiting the structure of choice models, the algorithm is designed to generate promising initializations for the estimation of discrete mixture models, directly addressing the challenge of numerous local optima. To evaluate its effectiveness, we compare its performance in computing high-quality starting points against CMA-ES, using the latter as a benchmark for global optimization. By contrasting these approaches, we aim to demonstrate that our heuristic reliably identifies superior solutions and is able to recover underlying structures in the data in a reasonable amount of time. This contribution is intended to advance the practical applicability of discrete mixture models, making them more accessible for complex, real-world datasets.

Four test series are performed on a set of discrete and discrete-continuous mixture model estimation problems, where we assess the performance of BHAMSLE vs. CMA-ES as initialization tools for Biogeme, compared to the standard initialization. The remainder of the paper is organized as follows: Section 2 outlines the problem setting and presents

the MSLE problem as a MILP. Section 3 describes BHAMSLE, while Section 4 reports on the case study and computational experiments. Finally, Section 5 offers concluding remarks and essential takeaways.

2 Problem formulation

In this section, we elaborate on the difficulties in latent class model estimation using a concrete example and give the problem formulation for MSLE as a MILP, specifically for the case of discrete-continuous mixtures, as here the added value of our approach is particularly significant. We thus assume that the model specification is a latent class model, where some parameters are distributed across the population. However, it is important to emphasize that the framework is general and can be used for any DCM.

2.1 Latent class choice models

Latent class choice models, or discrete mixtures, are a popular specification within the family of discrete choice models, used to account for unobserved heterogeneity in preferences. Rather than assuming a single set of parameters applies to the entire population, discrete mixtures posit that the population is composed of a finite number of segments—referred to as latent classes—each with its own taste parameters. The class membership of each individual is unobserved and must be inferred from the data.

The input to such models consists of a dataset containing observed choices made by a sample of individuals, along with the values of explanatory variables such as alternative attributes and individual characteristics. A model specification defines how these explanatory variables are mapped to choice probabilities, conditional on a set of parameters that must be estimated. The estimation procedure then yields as output the parameter values that maximize the likelihood of the observed data under the model, as well as the relative size (or probability) of each latent class in the population.

To illustrate, consider a simple mode choice model with two alternatives—car and bus—and one explanatory variable: travel time. Each individual is assumed to belong to one of two latent classes, each characterized by a different sensitivity to travel time. Class

membership is assigned based on directly estimated probabilities, without the influence of any explanatory variables. The deterministic utility for alternative j is specified as the product of a class-specific coefficient $\beta_{\text{time}}^{(s)}$ and the travel time of that alternative. That is, we assume the following deterministic utilities:

$$\begin{aligned} V_{\text{car}}^{(1)} &= \beta_{\text{time}}^{(1)} \cdot \text{traveltime}_{\text{car}}, \\ V_{\text{bus}}^{(1)} &= \beta_{\text{time}}^{(1)} \cdot \text{traveltime}_{\text{bus}}, \end{aligned}$$

$$\begin{aligned} V_{\text{car}}^{(2)} &= \beta_{\text{time}}^{(2)} \cdot \text{traveltime}_{\text{car}}, \\ V_{\text{bus}}^{(2)} &= \beta_{\text{time}}^{(2)} \cdot \text{traveltime}_{\text{bus}}. \end{aligned}$$

We assume that each class follows a standard logit model, and that class membership probabilities are fixed, with π denoting the probability of belonging to class 1 (and $1 - \pi$ for class 2). The probability of observing a given choice is then obtained by averaging over the two classes:

$$P_n = \pi \cdot P_n^{(1)} + (1 - \pi) \cdot P_n^{(2)},$$

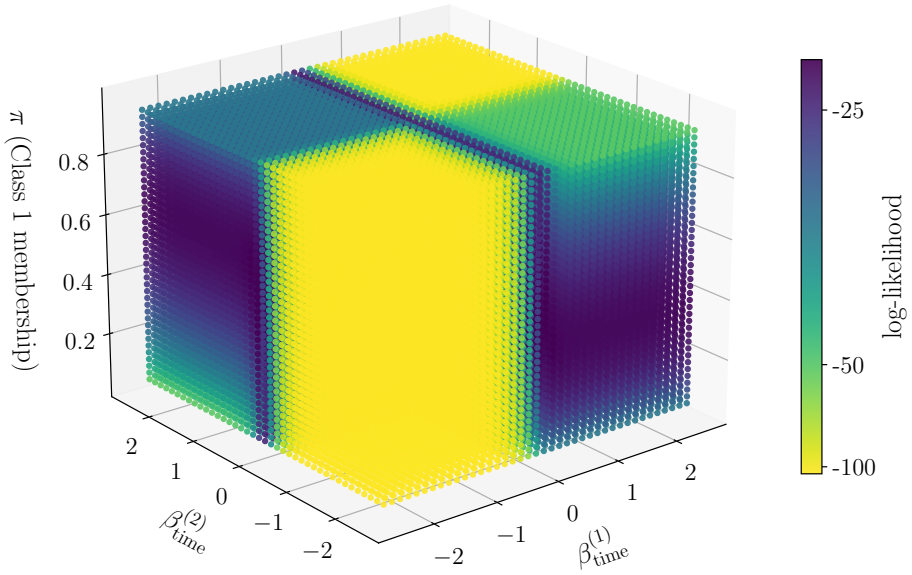
where $P_n^{(s)}$ denotes the choice probability under class s. A well-known challenge with latent class models is the non-convexity of the log-likelihood function. Due to the discrete structure of class membership and the interaction with continuous taste parameters, the likelihood surface may exhibit several local optima. As a motivating example, we construct a synthetic dataset and visualize the likelihood surface to highlight the multimodal nature of the objective and the interaction between taste parameters and class probabilities.

The synthetic dataset was created to reflect three qualitatively distinct behavioral patterns. The first segment, referred to as the rational group, consists of 18 individuals who consistently choose the faster alternative in each choice scenario. This group represents behavior that aligns with the compensatory logic of the logit model, where shorter travel time is always preferred. The second group, called the irrational group, includes 10 individuals who consistently prefer the slower alternative (which may reflect the presence of unobserved factors influencing choice behavior). This introduces a systematic violation of compensatory behavior and creates conflict within the likelihood function that can only be reconciled through latent segmentation. Finally, an ambiguous group of 6 individuals was included to introduce additional complexity. These individuals are exposed to symmetric or nearly symmetric alternatives and exhibit mixed or inconsistent choice behavior, thereby smoothing the likelihood surface and preventing sharp, unrealistic peaks. This configuration creates a meaningful contrast in behavior across the population

while avoiding the use of random sampling. The rational group dominates slightly to favor negative coefficients for one class, while the irrational and ambiguous groups introduce noise and identifiability challenges.

Table 14 in Appendix A provides the complete dataset used to evaluate the log-likelihood, which was computed over a dense grid of parameters $\beta_{\text{time}}^{(1)}$, $\beta_{\text{time}}^{(2)}$, and π. The resulting surface is shown in Figure 1, with the log-likelihood values visualized via color. We observe multiple distinct regions in parameter space with high likelihood values, illustrating the nonconvexity of the estimation problem. Such a landscape can trap local optimization methods, such as gradient-based or Newton-type algorithms, in suboptimal solutions, depending on the initialization.

Figure 1 – Log-likelihood surface for a latent class model with two classes and a single explanatory variable (travel time). $\beta_{\text{time}}^{(1)}$ and $\beta_{\text{time}}^{(2)}$ denote class-specific travel time sensitivities, and π the membership probability for class 1 .



2.2 Mathematical formulation of MSLE as a MILP

Consider a set $\mathcal{N} = \{1, \ldots, N\}$ of individuals, each choosing exactly one alternative amongst a set of choices $\mathcal{J} = \{1, \ldots, J\}$. An individual may have access to only a subset of these alternatives, indicated by their choice set $C_n \subset \mathcal{J}$. The observed choice for individual $n \in \mathcal{N}$ is denoted by $y_n \in \mathcal{J}$. For all $n \in \mathcal{N}$, each alternative is assigned a stochastic utility U_{in}, composed of a deterministic component V_{in} and a random error term ε_{in}. The

deterministic component is represented by the linear combination of alternative attributes and socio-economic characteristics, also referred to as explanatory variables, x_{ink} (where $k \in \mathcal{K} = \{1, \ldots, K\}$ indexes the set of all such factors) with the parameters β_k that are to be estimated. In discrete mixtures, or latent class models, the analyst tests the hypothesis that the population of individuals can be divided into a set of latent classes $\mathcal{S} = \{1, \ldots, S\}$, each characterized by distinct preferences. The available alternatives, the considered attributes and characteristics, and the parameters to be estimated may vary entirely across classes. This yields class-dependent utilities U_{in}^s, given by:

$$U_{in}^s = V_{in}^s + \varepsilon_{in} = \sum_{k \in \mathcal{K}^s} x_{ink}\beta_k + \varepsilon_{in}, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C_n^s,$$

where V_{in}^s represents the deterministic utility component, $\mathcal{K}^s$ the set of attributes and C_n^s the choice set of individual n for class $s \in \mathcal{S}$. We furthermore assume that each individual selects the alternative corresponding to the maximal utility.

To estimate the individual latent class probabilites $\pi_{ns}, n \in \mathcal{N}, s \in \mathcal{S}$, which may depend on explanatory variables, we similarly define scoring functions f_{ns} for every individual $n \in \mathcal{N}$ and latent class $s \in \mathcal{S}$. Each scoring function consists of a deterministic component f_{ns}^d, containing parameters $\alpha_l, l \in \mathcal{L} = \{1, \ldots, L\}$ that require estimation, as well as a stochastic error term δ_{ns}:

$$f_{ns} = f_{ns}^d + \delta_{ns} = \sum_{l \in \mathcal{L}} x_{nsl}\alpha_l + \delta_{ns}, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}.$$

In order to ensure that we can identify a set of probabilities that satisfies $\sum_s \pi_{ns} = 1 \; \forall n \in \mathcal{N}$, it is necessary to normalize one scoring function to be equal to 0. Given a set of utility and scoring functions, we can then derive the probabilities as follows. For an individual n to select alternative i, given their membership in class s, the probability is given by:

$$P_{in}^s = \mathbb{P}(U_{in}^s \geq U_{jn}^s, \; \forall j \in C_n^s), \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C_n^s.$$

Similarly, the latent class probabilities π_{ns} are expressed as:

$$\pi_{ns} = \mathbb{P}(f_{ns} \geq f_{nt}, \; \forall t \in \mathcal{S}), \quad \forall n \in \mathcal{N}, s \in \mathcal{S}.$$

The unconditional probability P_{in} of individual n choosing option i is then described by:

$$P_{in} = \sum_{s \in \mathcal{S}} \pi_{ns} P_{in}^s, \quad \forall \; n \in \mathcal{N}, i \in C_n.$$

We can now write the optimization problem in its stochastic form:

$$\max_{\beta,\pi} \quad \sum_{n\in\mathcal{N}} \ln\left(\sum_{s\in\mathcal{S}} \pi_{ns} P^s_{y_n n}\right) \tag{1}$$

s.t.

$$\sum_{s\in\mathcal{S}} \pi_{ns} = 1, \qquad \forall n \in \mathcal{N}, \tag{2}$$

$$U^s_{in} = \sum_{k\in\mathcal{K}^s} x_{ink}\beta_k + \varepsilon_{in}, \qquad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C^s_n, \tag{3}$$

$$f_{ns} = \sum_{l\in\mathcal{L}} x_{nsl}\alpha_l + \delta_{ns}, \qquad \forall n \in \mathcal{N}, s \in \mathcal{S}, \tag{4}$$

$$P^s_{in} = \mathbb{P}(U^s_{in} \geq U^s_{jn}, \quad \forall j \in C^s_n), \qquad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C^s_n, \tag{5}$$

$$\pi_{ns} = \mathbb{P}(f_{ns} \geq f_{nt},\ \forall t \in \mathcal{S}), \qquad \forall n \in \mathcal{N}, s \in \mathcal{S}, \tag{6}$$

$$\beta_k, \alpha_l \in \mathbb{R}, \qquad \forall k \in \bigcup_{s\in\mathcal{S}} \mathcal{K}^s, l \in \mathcal{L}.$$

The objective function in equation (1) seeks to maximize the log-likelihood of the observed choices. The first constraint in equation (2) ensures that the class membership probabilities sum to one for all individuals, guaranteeing a valid probability distribution over classes. Equations (3) and (4) define the utility and scoring functions U^s_{in} and f_{ns} for each class s, individual n and alternative $i \in C^s_n$, as described above. Finally, equations (5) and (6) specify the choice probabilities P^s_{in} and latent class probabilites π_{ns}.

The latter constraints may present a significant challenge, as the probabilities P^s_{in} and π_{ns} do not have a closed-form expression for certain discrete choice models (for example in continuous mixtures). This issue is typically addressed through simulation techniques (Train, 2003). By generating random draws from known distributions, Monte Carlo simulation can be employed to approximate these probabilities. A general approach for modeling maximum likelihood estimation of arbitrary discrete choice models in a linearized form was proposed by Fernandez Antolin (2018), who formulated the problem as a mixed-integer linear program. We follow a similar approach, extending their framework to discrete and discrete-continuous mixture models.

Let's first address the simulation procedure for a standard error component model, as described in Fernandez Antolin (2018). The key idea is to obtain a deterministic representation of the utility function by simulating the stochastic error terms. Specifically,

for each individual $n \in \mathcal{N}$ and choice alternative $i \in C_n$, the utility function is given by

$$U_{inr} = \sum_{k \in \mathcal{K}} x_{ink}\beta_k + \varepsilon_{inr}, \quad \forall n \in \mathcal{N}, i \in C_n, r \in \mathcal{R},$$

where ε_{inr} represents a random draw from the error term distribution. For instance, in a logit model, the error term follows a Gumbel distribution. The index $r \in \mathcal{R} = \{1, \ldots, R\}$ represents different simulation scenarios. Having a deterministic utility representation allows us to define choice variables per individual, alternative, and scenario:

$$\omega_{inr} = \begin{cases} 1, & \text{if } U_{inr} \geq U_{jnr}, \quad \forall j \in C_n, \\ 0, & \text{otherwise,} \end{cases} \quad \forall n \in \mathcal{N}, i \in C_n, r \in \mathcal{R}.$$

These choice variables enable us to approximate the choice probabilities using an unbiased estimator (Train, 2003):

$$P_{y_n n} \approx \frac{1}{R} \sum_{r \in \mathcal{R}} \omega_{y_n nr},$$

which leads to the following approximation of the objective function:

$$\max_{\beta, \pi_s} \quad \sum_{n \in \mathcal{N}} \ln\left(P_{y_n n}\right) \approx \sum_{n \in \mathcal{N}} \ln\left(\frac{1}{R} \sum_{r \in \mathcal{R}} \omega_{y_n nr}\right) = -NR + \sum_{n \in \mathcal{N}} \ln\left(\sum_{r \in \mathcal{R}} \omega_{y_n nr}\right).$$

In order for the objective to be fully linear, we need to deal with the natural logarithm around the sum of choice variables. This can be achieved through a piece-wise linearization. We refer to Appendix B.1 for more details. Furthermore, Fernandez Antolin (2018) also provide an extension to nested logit models. Our contribution generalizes this approach to any mixture model, extending the methodology to both discrete and continuous-discrete mixture models.

Let's now address the simulation procedure for continuous mixture models. Here, the utility function involves random elements. Those elements include the error term itself, but also distributed parameters. We assume the distributions of those random elements to be known, and that it is possible to generate instances in order to perform simulation. We present the formulation in case of a set of normally distributed parameters $\beta_m, m \in \mathcal{M} \subset \mathcal{K}$. For each simulation scenario $r \in \mathcal{R}$ and individual $n \in \mathcal{N}$, we take a draw σ_{nr} from the standard normal distribution $\mathcal{N}(0, 1)$ and define:

$$\beta_m = \beta_{m,1} + \beta_{m,2}\sigma_{nr} \quad \forall m \in \mathcal{M}$$

Here, $\beta_{m,1}$ represents the mean and $\beta_{m,2}$ the standard deviation of the normal distribution. We can then describe the deterministic utility U_{inr} as:

$$U_{inr} = \sum_{k \in \mathcal{K} \setminus \mathcal{M}} x_{ink}\beta_k + \sum_{m \in \mathcal{M}} x_{inm}\beta_m + \varepsilon_{inr} \quad \forall n \in \mathcal{N},\ i \in C_n,\ r \in \mathcal{R}. \tag{7}$$

For discrete and discrete-continuous mixture models, an additional simulation layer is required to determine latent class membership. To this end, the random error terms δ_{ns} of the scoring functions f_{ns} are also simulated, yielding deterministic functions f_{nsr} for each scenario r:

$$f_{nsr} = \sum_{l \in \mathcal{L}} x_{nsl}\alpha_l + \delta_{nsr} \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, r \in \mathcal{R}. \tag{8}$$

where δ_{nsr} represents a random draw from the error term distribution. Now, in each scenario, the class assignment is fully deterministic and given by the maximal scoring function. Consequently, the utility functions incorporate this latent class allocation as follows:

$$U_{inr} = \sum_{s \in \mathcal{S}} \mathbb{1}_{[f_{nsr} \geq f_{ntr}\ \forall t \in \mathcal{S}]} U^s_{inr}, \tag{9}$$

where U^s_{inr} describes the deterministic utility for a specific class. To incorporate the variables U_{inr} and ω_{inr} into a mixed-integer linear program (MILP), it is necessary to linearize all indicator functions and products. For the sake of readability, we omit the full derivation of this linearization and the complete description of the MILP for MSLE in the case of a discrete-continuous mixture model and refer to Appendices B.2 and B.3 for a detailed exposition.

While solving the MILP formulation guarantees a globally optimal solution, this approach is limited in practice, as only very small instances can be solved within a reasonable timeframe. The MILP furthermore introduces the need for simulation—an issue that is not problematic, as it is unavoidable for continuous mixtures. To address the computational challenge of solving the MILP formulation, we consider leveraging heuristic approaches inspired by existing methodologies. The Breakpoint Heuristic Algorithm (BHA), introduced by Haering *et al.* (2024) for choice-based pricing, has demonstrated the ability to efficiently solve a similar MILP with high accuracy in significantly reduced computational time. This algorithm systematically explores local optima by identifying decision-making breakpoints and can be categorized as a coordinate descent method. Motivated by its effectiveness, we propose adapting the BHA to the MSLE problem in the hope of achieving high-quality approximate solutions efficiently, serving as a strong initialization point for Newton-like

algorithms, which can further refine the solution and increase the likelihood of converging to a superior local maximum, if not the global one. This new algorithm is called the Breakpoint Heuristic Algorithm for MSLE (BHAMSLE), and is presented in the next section.

3 Methodology

In this section, we introduce BHAMSLE. As for the problem formulation, we illustrate the algorithm specifically for the case of discrete-continuous mixtures.

3.1 Breakpoint Heuristic Algorithm for MSLE (BHAMSLE)

BHAMSLE capitalizes on the idea of decision-making breakpoints, more specifically "entry" and "exit" breakpoints for each individual n and scenario r, signifying where the choice variable $\omega_{y_n nr}$ switches to 1 or back to 0. These breakpoints represent a set of local optima that can be enumerated. The method can be categorized as a coordinate descent (ascent), iteratively optimizing one parameter at a time while fixing all others, terminating once no parameter can be improved further. The full algorithm is described in the following procedure:

1. Choose a starting point for the estimation, usually, $\beta_k^* = 0 \ \forall k \in \mathcal{K}, \alpha_l^* = 0 \ \forall l \in \mathcal{L}$, and compute its objective value $o^* = sLL(\beta^*, \alpha^*)$.
2. Set $j = 1$.
3. Fix all parameters with index $\neq j$, i.e. $\beta_k = \beta_k^*$, $k \neq j$ and $\alpha_l = \alpha_l^*$, $l \neq j - K$.

4. Compute the set of breakpoints, initialized as $\mathcal{B} = \{\}$:

   for $n \in \mathcal{N}, r \in \mathcal{R}$ **:**
   - **if** $j \leq K$ **:**
     - **for** $s \in \mathcal{S}$ **:**
       - **if** $f_{nsr} \geq f_{ntr}\ \forall t \in \mathcal{S}$ **:**
         - Compute the segment $[s_1, s_2] \ni \beta_j$ where $U^s_{y_n nr}(\beta) \geq U^s_{inr}(\beta)\ \forall i \in C^s_n$.
         - Add (s_1, n) as an entry breakpoint and (s_2, n) as an exit breakpoint to $\mathcal{B}$.
       - **end**
     - **end**
   - **else**
     - **for** $s \in \mathcal{S}$ **:**
       - **if** $U^s_{y_n nr} \geq U^s_{inr}\ \forall i \in C_n$ **:**
         - Compute the segment $[s_1, s_2] \ni \alpha_j$ where $f_{nsr}(\alpha) \geq f_{ntr}(\alpha)\ \forall t \in \mathcal{S}$.
         - Add (s_1, n) as an entry breakpoint and (s_2, n) as an exit breakpoint to $\mathcal{B}$.
       - **end**
     - **end**
   - **end**

   end

5. Sort $\mathcal{B}$ in ascending order. Define $\Sigma_n = |\{\text{entry point } (x, y) \in \mathcal{B} : x = -\infty, y = n\}|$, $n \in \mathcal{N}$, $o = -N \ln(R) + \sum_n \ln(\Sigma_n)$ and $\mathcal{B} \leftarrow \{(x, y) \in \mathcal{B} : x \neq -\infty\}$. Then evaluate all $b \in \mathcal{B}$:

   for $b \in \mathcal{B}$ **:**
   - **if** b is an entry point **:**
     - $o \mathrel{+}= \ln(\Sigma_n + 1) - \ln(\Sigma_n)$.
   - **else**
     - $o \mathrel{+}= \ln(\Sigma_n - 1) - \ln(\Sigma_n)$.
   - **end**
   - **if** $o > o^*$ **:**
     - $o^* = o$, if $j \leq K$ set $\beta^*_j = b$, else set $\alpha^*_{j-K} = b$.
   - **end**

   end

6. Set $j = j + 1$ (if now $j = K + S$, set $j = 1$) and repeat from step 3.
7. Terminate when no improvement is found over $K + S - 1$ iterations.

The algorithm begins by initializing the parameter estimates and computing the initial simulated log-likelihood (sLL). Each iteration then focuses on optimizing one parameter

at a time, holding the others fixed. For each parameter update, the algorithm constructs a set of entry and exit breakpoints by iterating through all individuals and scenarios and checking where the observed alternative's utility becomes dominant or ceases to be dominant. If the current parameter is not part of the scoring functions, the scoring functions are fixed, and thus, the assigned class is given. We compute the segment of values of the loose parameter that make the utility of the observed alternative the dominant one. The start point of the segment is an entry and the end point an exit breakpoint. If the current parameter belongs to a scoring function, all utilities of alternatives are fixed, implying that we can compute the classes in which the highest utility belongs to the observed alternative. We then derive the segment of values the loose parameter can take that allow the scoring function of these classes to be dominant.

It is worth noting that the algorithm can be slightly simplified if the scoring functions do not depend on explanatory variables, and instead the latent class probabilities are estimated directly as parameters. In this case, let $\gamma_g,\ g \in \mathcal{G} = \{1, \ldots, S-1\}$ represent the parameters that separate the unit interval into S partitions $P_1, \ldots, P_s$. Furthermore, draws from the uniform $[0, 1]$ distribution used to simulate class membership are denoted by $u_{nr}, n \in \mathcal{N}, r \in \mathcal{R}$. The starting values here are $\gamma_g^* = \frac{g}{S},\ g \in \mathcal{G}$, and the algorithm becomes:

1. Choose a starting point for the estimation, usually, $\beta_k^* = 0,\ k \in \mathcal{K}$, $\gamma_g^* = \frac{g}{S},\ g \in \mathcal{G}$, and compute its objective value $o^* = sLL(\pi^*, \beta^*)$.
2. Set $j = 1$.
3. Fix all other parameters $\beta_k = \beta_k^*$, $k \neq j$ and $\gamma_g = \gamma_g^*$, $g \neq j - K$.
4. Compute the set of breakpoints, initialized as $\mathcal{B} = \{\}$:

for $n \in \mathcal{N}, r \in \mathcal{R}$ **:**
  if $j \leq K$ **:**
    for $s \in \mathcal{S}$ **:**
      if $u_{nr} \in P_s$ **:**
        Compute the segment $[s_1, s_2] \ni \beta_j$ where $U^s_{y_n nr}(\beta) \geq U^s_{inr}(\beta)\ \forall i \in C^s_n$.
        Add (s_1, n) as an entry breakpoint and (s_2, n) as an exit breakpoint to $\mathcal{B}$.
      end
    end
  else
    Let $g \leftarrow j - K$.
    if $u_{nr} \in [\gamma^*_{g-1}, \gamma^*_{g+1}]$ **:**
      Let $W \leftarrow \{c \in \{g, g+1\} \mid U^s_{y_n nr} \geq U^s_{inr},\ \forall i \in \mathcal{I}\}$.
      if $W = \{g, g+1\}$ **:**
        Add $(-\infty, n)$ as an entry breakpoint to $\mathcal{B}$.
      elseif $W = \{g\}$ **:**
        Add (u_{nr}, n) as an entry breakpoint to $\mathcal{B}$.
      elseif $W = \{g+1\}$ **:**
        Add (u_{nr}, n) as an exit breakpoint to $\mathcal{B}$.
      end
    end
  end
end

Steps 5, 6, and 7 are the same as before. Now it holds that for latent class parameters γ_g, the entry and exit breakpoints correspond to the draws u_{nr} from the uniform distribution, depending on wether the currently treated parameter allows for the draw to fall into the interval of a class in which $\omega_{y_n nr}$ is switched to 1 or not. This leverages the structure of the problem and ensures a balanced distribution of candidate solutions for the latent class parameter, thereby reducing the likelihood of convergence to extreme values.

Finally, the entry and exit breakpoints are then sorted in ascending order. Subsequently, the log-likelihood contributions are updated as the algorithm traverses the sorted breakpoints, depending on the entry or exit status of the latter. The updated objective value is compared to the previous best value, and the parameter is updated if an improvement is found. This process repeats for all parameters sequentially until no further improvement is achieved after a full pass through all parameters.

Maintaining information on whether a breakpoint represents an entry or exit for a given individual n is crucial, as it enables the efficient processing of breakpoints in ascending order. This allows for the incremental computation of changes in sLL in $\mathcal{O}(1)$ time per breakpoint. In contrast, evaluating the sLL objective function directly at each possible solution (as a general-purpose global optimization algorithm would) necessitates $\mathcal{O}(NR)$ operations. This distinction results in substantial computational savings, particularly for large-scale problems.

The algorithm terminates when no further improvements are found in each coordinate.

4 Results and discussion

To test our approach, we perform experiments on four different setups: discrete and discrete-continuous mixtures of logit with observed vs. synthetically generated choices. All continuous mixtures use a normal distribution for the mixed parameters, and the assignment to latent classes is based solely on estimated probabilities, with no explanatory variables affecting class membership. For each of these four models, we compare using the standard initialization, the general-purpose global optimization algorithm CMA-ES (Covariance Matrix Adaptation Evolution Strategy) and BHAMSLE to find a good starting point for the model estimation using Biogeme. For all initializations, the same set of draws are used in the estimation. The standard initialization value for all parameters is 0, except for the standard deviations in continuous mixtures of logit, which are initialized to 1, and the latent class probabilities, which are initialized to equal probabilities. We employ the CMA-ES implementation provided by the `Evolutionary.jl` package in Julia, with hyperparameters selected to emphasize finding high-quality solutions. Specifically, we choose a considerably large population size ($\lambda = 50$) to enhance exploration, a generous step-size ($\sigma = 66.666$) based on the standard formula $\sigma = \frac{UB-LB}{3}$ with LB, UB $= -100.0, 100.0$, to allow the algorithm to navigate a wide search space efficiently, and a maximum number of iterations (max_iters $= 500$) to achieve a high-quality outcome. To evaluate the objective value, we use Biogeme's simulation module with varying numbers of scenarios R. All tests are performed in a single thread on a computational cluster node with two 2.4 GHz Intel Xeon Platinum 8360Y processors, utilizing 16 GB of RAM. We utilize the latest version of Biogeme, PandasBiogeme 3.2.14, as described by Bierlaire (2023). For each test, we consider sample sizes $N = \{500, 1,000\}$ and numbers of scenarios $R = \{50, 100, 500, 1,000\}$, where for models involving continuous mixtures we increase

the number of scenarios up to $R = 3,000$. For every tuple (N, R) we take 100 samples from the full dataset (and respective distributions) and report the averaged obtained values. For the discrete-continuous mixtures of logit, Biogeme's simulation module with $R = 10,000$ is used to compute the final log-likelihood.

The first dataset is extracted from stated preference data on hypothetical mode choice collected in Switzerland (Bierlaire *et al.*, 2001). Three alternatives are considered: car, rail, and Swissmetro (SM), with "car" being available only to car owners. In the first experiment, we hypothesize that there exists a portion of the population with baseline preferences for alternatives that differ form the rest of individuals. Therefore, separate alternative-specific constants $\text{ASC}'_{\text{car}}, \text{ASC}'_{\text{rail}}$ are estimated for this class. We refer to this class as class 2 and to the base model as class 1. The systematic utility equations for the two classes are:

$$
\begin{aligned}
V^{(1)}_{\text{car}} &= \text{ASC}_{\text{car}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{car}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{car}}, \\
V^{(1)}_{\text{rail}} &= \text{ASC}_{\text{rail}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{rail}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{rail}} + \beta_{\text{headway}} \cdot \text{headway}_{\text{rail}}, \\
V^{(1)}_{\text{SM}} &= \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{SM}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{SM}} + \beta_{\text{headway}} \cdot \text{headway}_{\text{SM}}, \\
V^{(2)}_{\text{car}} &= \text{ASC}'_{\text{car}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{car}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{car}}, \\
V^{(2)}_{\text{rail}} &= \text{ASC}'_{\text{rail}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{rail}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{rail}} + \beta_{\text{headway}} \cdot \text{headway}_{\text{rail}}, \\
V^{(2)}_{\text{SM}} &= \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{SM}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{SM}} + \beta_{\text{headway}} \cdot \text{headway}_{\text{SM}}.
\end{aligned}
$$

The results are presented in Tables 1, 2, and 3. We observe that, CMA-ES does not succeed in providing Biogeme with a starting point better than the standard initialization with the resulting Bio-C log-likelihood values being consistently worse. This trend holds across all tested configurations of N and R. In contrast, Biogeme initialized with BHAMSLE (Bio-B) achieves the highest log-likelihood values in the majority of configurations, with improvements becoming particularly significant starting from $R = 50$. On average, the log-likelihood values achieved by Bio-B improve by approximately 3% compared to Biogeme with default initialization.

For the estimated probabilities of the two latent classes (p_1, p_2), Biogeme with default initialization struggles to distinguish between the two classes, often assigning nearly uniform probabilities. Biogeme initialized with CMA-ES shows more variability in these probabilities but does not achieve the consistency observed with BHAMSLE initialization. When using the BHAMSLE starting points, at low R, the estimated probabilities remain close to uniform, but starting around $R = 50$, Biogeme consistently captures a higher probability for class 1, likely contributing to the improved log-likelihood.

Table 1 – Comparison of achieved log-likelihood values and runtimes using Biogeme with default initialization (Bio), Biogeme with CMA-ES (C) starting point (Bio-C), and Biogeme with BHAMSLE (B) starting point (Bio-B) when estimating a discrete mixture of logit models with observed choices (N = population size, R = number of draws, LL = log-likelihood, T = estimation time in seconds).

N	R	LL-Bio	LL-Bio-C	Gap (%)	LL-Bio-B	Gap (%)	T-Bio	T-C	T-Bio-C	T-B	T-Bio-B
500	1	-390.267	-397.371	-1.82	-390.267	0.00	3	9	3	0	3
500	5	-390.267	-412.224	-5.63	-390.267	0.00	3	9	3	0	3
500	10	-390.267	-397.582	-1.87	-382.090	2.10	3	9	3	0	4
500	20	-390.267	-404.122	-3.55	-377.073	3.38	3	9	3	1	4
500	50	-390.267	-407.086	-4.31	-374.006	4.17	3	9	3	9	3
500	100	-390.267	-409.461	-4.92	-380.120	2.60	3	9	3	8	3
500	500	-390.267	-404.289	-3.59	-374.499	4.04	3	9	3	63	3
500	1,000	-390.267	-391.796	-0.39	-376.737	3.47	3	9	3	238	3
1,000	1	-779.195	-803.627	-3.14	-779.195	0.00	4	18	4	0	4
1,000	5	-779.195	-827.014	-6.14	-779.195	0.00	3	18	3	0	3
1,000	10	-779.195	-813.654	-4.42	-758.929	2.60	3	18	4	1	4
1,000	20	-779.195	-819.721	-5.20	-760.612	2.38	3	18	3	2	5
1,000	50	-779.195	-808.425	-3.75	-761.902	2.22	3	17	4	8	4
1,000	100	-779.195	-820.438	-5.29	-759.129	2.58	3	17	4	15	4
1,000	500	-779.195	-797.136	-2.30	-758.910	2.60	3	16	3	148	4
1,000	1,000	-779.195	-815.383	-4.64	-756.742	2.88	3	18	3	1,035	4

In terms of runtime, all methods terminate in negligibly short amounts of time - except for BHAMSLE which, upwards of $R = 500$, starts to require more time for the estimation. The runtimes for Biogeme and CMA-ES are almost constant across R, due to the fact that no simulation has to be invoked to evaluate the objective function.

Table 3 presents the estimated parameter values for all methods. Several notable differences emerge. The alternative-specific constants (ASCs) exhibit substantial variation across methods, with CMA-ES producing extreme values, particularly for ASC_{car} and ASC'_{car}, which diverge significantly from the estimates obtained using Biogeme-based estimations. Similarly, CMA-ES results in a much larger magnitude for β_{HE}, whereas Biogeme with BHAMSLE (Bio-B) provides more stable estimates that align more closely with expectations. For the cost and travel time sensitivities, β_{cost} and β_{time}, Biogeme with BHAMSLE and default initialization yield relatively consistent estimates, while CMA-ES again shows large deviations, particularly for β_{cost}, which is estimated at a much higher absolute value. These discrepancies may suggest that CMA-ES struggles to navigate the parameter space effectively, potentially contributing to its lower log-likelihood values.

Table 2 – Comparison of latent class probabilities derived using Biogeme with default initialization (Bio), Biogeme with CMA-ES (C) starting point (Bio-C), and Biogeme with BHAMSLE (B) starting point (Bio-B) when estimating a discrete mixture of logit models with observed choices (N = population size, R = number of draws, LL = log-likelihood, T = estimation time in seconds).

N	R	(p_1, p_2)-Bio	(p_1, p_2)-C	(p_1, p_2)-Bio-C	(p_1, p_2)-B	(p_1, p_2)-Bio-B
500	1	(0.50, 0.50)	(0.42, 0.58)	(0.40, 0.60)	(0.50, 0.50)	(0.50, 0.50)
500	5	(0.50, 0.50)	(0.39, 0.61)	(0.43, 0.57)	(0.50, 0.50)	(0.50, 0.50)
500	10	(0.50, 0.50)	(0.39, 0.61)	(0.39, 0.61)	(0.58, 0.42)	(0.72, 0.28)
500	20	(0.50, 0.50)	(0.49, 0.51)	(0.49, 0.51)	(0.53, 0.47)	(0.69, 0.31)
500	50	(0.50, 0.50)	(0.25, 0.75)	(0.00, 1.00)	(0.61, 0.39)	(0.53, 0.47)
500	100	(0.50, 0.50)	(0.56, 0.44)	(0.60, 0.40)	(0.55, 0.45)	(0.72, 0.28)
500	500	(0.50, 0.50)	(0.47, 0.53)	(0.48, 0.52)	(0.67, 0.33)	(0.66, 0.34)
500	1,000	(0.50, 0.50)	(0.36, 0.64)	(0.01, 0.99)	(0.62, 0.38)	(0.61, 0.39)
1,000	1	(0.50, 0.50)	(0.51, 0.49)	(0.51, 0.49)	(0.50, 0.50)	(0.50, 0.50)
1,000	5	(0.50, 0.50)	(0.44, 0.56)	(0.45, 0.55)	(0.50, 0.50)	(0.50, 0.50)
1,000	10	(0.50, 0.50)	(0.45, 0.55)	(0.43, 0.57)	(0.54, 0.46)	(0.77, 0.23)
1,000	20	(0.50, 0.50)	(0.45, 0.55)	(0.45, 0.55)	(0.59, 0.41)	(0.78, 0.22)
1,000	50	(0.50, 0.50)	(0.61, 0.39)	(0.60, 0.40)	(0.63, 0.37)	(0.74, 0.26)
1,000	100	(0.50, 0.50)	(0.42, 0.58)	(0.40, 0.60)	(0.68, 0.32)	(0.62, 0.38)
1,000	500	(0.50, 0.50)	(0.36, 0.64)	(0.38, 0.62)	(0.62, 0.38)	(0.53, 0.47)
1,000	1,000	(0.50, 0.50)	(0.33, 0.67)	(0.33, 0.67)	(0.66, 0.34)	(0.63, 0.37)

In the second experiment, we make use of the same model specification, but now we consider the $\beta_{\text{traveltime}}$ parameter to be normally distributed amongst the population for class 1, resulting in a discrete-continuous mixture of logit. To this end, we denote $\beta^{\text{mixed}}_{\text{traveltime}} = \beta_{\text{traveltime}} + \beta^{\text{std}}_{\text{traveltime}} \cdot U_n$, where $U_n \sim \mathcal{N}(0, 1)$, and replace $\beta_{\text{traveltime}}$ by this new parameter for class 1, keeping everything else the same. We give the new systematic equations below:

$$
\begin{aligned}
V^{(1)}_{\text{car}} &= \text{ASC}_{\text{car}} + \beta^{\text{mixed}}_{\text{traveltime}} \cdot \text{traveltime}_{\text{car}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{car}}, \\
V^{(1)}_{\text{rail}} &= \text{ASC}_{\text{rail}} + \beta^{\text{mixed}}_{\text{traveltime}} \cdot \text{traveltime}_{\text{rail}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{rail}} + \beta_{\text{headway}} \cdot \text{headway}_{\text{rail}}, \\
V^{(1)}_{\text{SM}} &= \beta^{\text{mixed}}_{\text{traveltime}} \cdot \text{traveltime}_{\text{SM}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{SM}} + \beta_{\text{headway}} \cdot \text{headway}_{\text{SM}}, \\
\\
V^{(2)}_{\text{car}} &= \text{ASC}'_{\text{car}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{car}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{car}}, \\
V^{(2)}_{\text{rail}} &= \text{ASC}'_{\text{rail}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{rail}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{rail}} + \beta_{\text{headway}} \cdot \text{headway}_{\text{rail}}, \\
V^{(2)}_{\text{SM}} &= \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{SM}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{SM}} + \beta_{\text{headway}} \cdot \text{headway}_{\text{SM}}.
\end{aligned}
$$

Table 3 – Comparison of average estimated parameter values and log-likelihood over 100 samples with $N = 1,000, R = 1,000$, using Biogeme with default initialization (Bio), CMA-ES, Biogeme with CMA-ES starting point (Biogeme-C), BHAMSLE, and Biogeme with BHAMSLE starting point (Biogeme-B) when estimating a discrete mixture of logit models with observed choices.

Parameter	Biogeme	CMA-ES	Biogeme-C	BHAMSLE	Biogeme-B
ASC_{car}	-0.452	70.802	17.579	-11.987	-12.076
ASC'_{car}	-0.657	38.071	10.754	-2.887	-2.360
$\text{ASC}_{\text{train}}$	-1.066	1.764	1.761	-1.449	-1.571
$\text{ASC}'_{\text{train}}$	-5.299	-11.665	-11.421	-9.955	-5.622
β_{cost}	-1.098	-2.192	-2.195	-0.642	-1.811
β_{HE}	-0.452	-21.005	-5.335	-0.055	-0.399
β_{time}	-0.657	0.407	0.401	-0.284	-0.261
p_1	0.50	0.33	0.33	0.66	0.63
p_2	0.50	0.67	0.67	0.34	0.37
$\text{LL}(\beta)$	-779.195	-821.813	-815.383	-783.631	-756.742

The total number of parameters to estimate thus increases to nine.

The results are shown in Tables 4, 5 and 6. We observe that for smaller values of R, the differences between LL-Bio and LL-Bio-C are small, but the log-likelihood values achieved with Biogeme initialized with CMA-ES (Bio-C) are consistently worse than those achieved with both default initialization (Bio) and BHAMSLE-initialized Biogeme (Bio-B). This suggests that CMA-ES provides suboptimal starting points, which negatively impact the estimation process.

In contrast, Biogeme initialized with BHAMSLE (Bio-B) achieves significantly better log-likelihood values as the number of simulation draws increases. For samples of size $N = 500$, BHAMSLE provides starting points that yield up to 10% better solutions compared to Biogeme with default initialization. For $N = 1,000$, the improvement is smaller but still noticeable, with around 3% better log-likelihood values on average. The number of simulation draws necessary for BHAMSLE to outperform default initialization increases for larger sample sizes, with clear improvements starting from $R = 500$.

It is important to note that in some cases, Biogeme initialized with BHAMSLE can yield worse results than the default initialization, as seen, for example, with $N = 500, R = 100$. This likely stems from the fact that small numbers of simulation draws may not be

Table 4 – Comparison of achieved log-likelihood values and runtimes using Biogeme with default initialization (Bio), Biogeme with CMA-ES (C) starting point (Bio-C), and Biogeme with BHAMSLE (B) starting point (Bio-B) when estimating a discrete-continuous mixture of logit with observed choices (N = population size, R = number of draws, LL = log-likelihood, T = estimation time in seconds).

N	R	LL-Bio	LL-Bio-C	Gap (%)	LL-Bio-B	Gap (%)	T-Bio	T-C	T-Bio-C	T-B	T-Bio-B
500	1	-438.812	-453.410	-2.88	-438.760	0.01	17	25,774	4	0	20
500	5	-431.788	-444.526	-2.41	-428.005	0.88	14	25,025	5	0	13
500	10	-427.414	-442.143	-2.87	-428.099	-0.16	20	25,665	13	0	19
500	20	-426.447	-443.612	-4.30	-426.925	-0.11	23	27,526	19	1	20
500	50	-439.559	-437.951	-3.04	-435.483	0.93	24	26,428	33	3	17
500	100	-431.124	-445.574	-4.82	-433.809	-0.62	23	27,690	37	6	23
500	500	-490.745	-435.685	-2.28	-436.676	11.02	38	37,066	245	46	48
500	1,000	-488.010	-436.262	-2.42	-435.165	10.83	90	48,358	380	107	55
500	3,000	-474.640	-	-	-433.381	8.69	312	>20h	-	347	135
1,000	1	-877.418	-900.399	-2.62	-875.202	0.25	11	26,643	4	0	11
1,000	5	-868.597	-883.455	-2.82	-867.473	0.13	15	26,253	8	0	15
1,000	10	-855.605	-885.834	-3.91	-855.563	0.00	19	28,093	11	1	21
1,000	20	-856.742	-893.263	-3.77	-853.567	0.37	28	27,758	35	2	30
1,000	50	-869.742	-870.438	-0.56	-866.792	0.34	23	28,612	85	7	23
1,000	100	-888.778	-892.162	-3.82	-870.692	2.04	26	30,897	130	14	38
1,000	500	-867.117	-854.552	-0.70	-845.290	2.52	88	48,886	135	96	83
1,000	1,000	-869.915	-	-	-845.012	2.87	166	>20h	-	219	169
1,000	3,000	-868.542	-	-	-843.699	2.86	477	>20h	-	619	493

sufficient to efficiently capture the mixed parameter. For CMA-ES initialization, the log-likelihood values remain consistently below those achieved with either default or BHAMSLE initialization across all configurations.

The differences in the estimated latent class probabilities (p_1, p_2) are again not large, but substantial enough to influence the results. Biogeme with default initialization often fails to approach the true class distribution, which appears to be close to 40% for class 1 and 60% for class 2. Biogeme initialized with CMA-ES produces probabilities that are highly variable and often diverge significantly from the true distribution. In contrast, Biogeme initialized with BHAMSLE consistently converges towards this improved local optimum, demonstrating its effectiveness in guiding the estimation process.

In terms of runtime, CMA-ES initialization is by far the slowest, with runtime increasing as R grows, and even exceeding the 20-hour time limit for larger simulations. BHAMSLE initialization also incurs a runtime overhead compared to Biogeme with default initialization, but the gap is smaller than for the discrete mixture of logit models. On average,

Table 5 – Comparison of latent class probabilities derived using Biogeme with default initialization (Bio), Biogeme with CMA-ES (C) starting point (Bio-C), and Biogeme with BHAMSLE (B) starting point (Bio-B) when estimating a discrete-continuous mixture of logit with observed choices (N = population size, R = number of draws, LL = log-likelihood, T = estimation time in seconds).

N	R	(p_1, p_2)-Bio	(p_1, p_2)-C	(p_1, p_2)-Bio-C	(p_1, p_2)-B	(p_1, p_2)-Bio-B
500	1	(0.45, 0.55)	(0.47, 0.53)	(0.35, 0.65)	(0.50, 0.50)	(0.44, 0.56)
500	5	(0.44, 0.56)	(0.67, 0.33)	(0.67, 0.33)	(0.50, 0.50)	(0.41, 0.59)
500	10	(0.45, 0.55)	(0.99, 0.01)	(0.98, 0.02)	(0.52, 0.48)	(0.43, 0.57)
500	20	(0.46, 0.54)	(1.00, 0.00)	(0.99, 0.01)	(0.49, 0.51)	(0.49, 0.51)
500	50	(0.44, 0.56)	(1.00, 0.00)	(0.98, 0.02)	(0.57, 0.43)	(0.42, 0.58)
500	100	(0.40, 0.60)	(1.00, 0.00)	(1.00, 0.00)	(0.54, 0.46)	(0.41, 0.59)
500	500	(0.33, 0.67)	(0.61, 0.39)	(0.62, 0.38)	(0.43, 0.57)	(0.39, 0.61)
500	1,000	(0.34, 0.66)	(1.00, 0.00)	(1.00, 0.00)	(0.42, 0.58)	(0.38, 0.62)
500	3,000	(0.34, 0.66)	-	-	(0.39, 0.61)	(0.39, 0.61)
1,000	1	(0.39, 0.61)	(0.27, 0.73)	(0.05, 0.95)	(0.50, 0.50)	(0.48, 0.52)
1,000	5	(0.41, 0.59)	(1.00, 0.00)	(0.99, 0.01)	(0.50, 0.50)	(0.45, 0.55)
1,000	10	(0.46, 0.54)	(0.71, 0.29)	(0.72, 0.28)	(0.51, 0.49)	(0.46, 0.54)
1,000	20	(0.45, 0.55)	(1.00, 0.00)	(1.00, 0.00)	(0.41, 0.59)	(0.41, 0.59)
1,000	50	(0.44, 0.56)	(0.51, 0.49)	(0.44, 0.56)	(0.54, 0.46)	(0.41, 0.59)
1,000	100	(0.44, 0.56)	(1.00, 0.00)	(1.00, 0.00)	(0.37, 0.53)	(0.42, 0.58)
1,000	500	(0.44, 0.56)	(1.00, 0.00)	(0.44, 0.56)	(0.41, 0.59)	(0.39, 0.61)
1,000	1,000	(0.43, 0.57)	-	-	(0.40, 0.60)	(0.39, 0.61)
1,000	3,000	(0.43, 0.57)	-	-	(0.41, 0.59)	(0.40, 0.60)

Biogeme with default initialization completes the estimation about 1.5 times faster than BHAMSLE, but the significant improvement in log-likelihood values with BHAMSLE makes this trade-off worthwhile.

Finally, Table 6 presents the estimated parameter values for all methods. As CMA-ES did not converge within the time limit, for $N = 1,000$ and $R = 1,000, 3,000$, we show the results for $R = 500$ for that method instead. Introducing a normally distributed travel time sensitivity $\beta_{\text{traveltime}}$ increases the complexity of the estimation, leading to greater variability in parameter estimates across methods. CMA-ES produces extreme values for several parameters, notably for ASC'_{car} and β_{HE}, which deviate significantly from other estimates. Similarly, the mean of the travel time coefficient, $\beta_{\text{time, mean}}$, is highly unstable under CMA-ES, reaching a large negative value, which may indicate that the optimizer struggles to find a meaningful distribution for this parameter. A key observation concerns the standard deviation of the travel time coefficient, $\beta_{\text{time, std.}}$. Ideally, this

Table 6 – Comparison of average estimated parameter values and log-likelihood over 100 samples with $N = 1,000, R = 3,000,\ 500^{\text{CMA-ES}}$, using Biogeme with default initialization (Bio), CMA-ES, Biogeme with CMA-ES starting point (Biogeme-C), BHAMSLE, and Biogeme with BHAMSLE starting point (Biogeme-B) when estimating a discrete-continuous mixture of logit with observed choices.

Parameter	Biogeme	CMA-ES	Biogeme-C	BHAMSLE	Biogeme-B
ASC_{car}	-4.498	12.427	0.983	-0.257	3.437
ASC'_{car}	-0.108	-31.996	-9.131	-3.736	2.068
$\text{ASC}_{\text{train}}$	-1.548	0.187	-1.042	-1.28	-3.163
$\text{ASC}'_{\text{train}}$	-6.313	-7.328	-7.389	-6.998	-10.272
β_{cost}	-1.286	-1.510	-1.284	-1.301	-2.714
β_{HE}	-0.037	-45.085	0.046	0.731	-6.767
$\beta_{\text{time, mean}}$	0.654	-28.847	-16.483	-16.825	-11.937
$\beta_{\text{time, std.}}$	-5.124	-0.098	-0.166	0.224	-0.432
p_1	0.43	1.00	0.44	0.41	0.40
p_2	0.57	0.00	0.56	0.59	0.60
$\text{LL}(\beta)$	-868.542	-863.920	-854.552	-863.905	-843.699

parameter should capture unobserved heterogeneity in travel time sensitivities. However, CMA-ES yields an estimate close to zero, effectively collapsing the mixed distribution. In contrast, Biogeme with BHAMSLE produces more reasonable values for both the mean and standard deviation, allowing for heterogeneity in preferences to be captured in the model.

The estimated cost sensitivity, β_{cost}, remains relatively consistent across Biogeme-based methods, whereas CMA-ES produces a more negative estimate. The alternative-specific constants also show large variations, with CMA-ES generating results that differ substantially from those obtained with Biogeme's default initialization and BHAMSLE. This pattern suggests that CMA-ES initialization does not provide reliable estimates, further supported by its lower log-likelihood values compared to BHAMSLE.

For the next two tests we consider a different data set. It is extracted from revealed preference data on mode choice collected in London (Hillel *et al.*, 2018). There are four alternatives available to all individuals: walking, cycling, public transport (pt), and driving. This time, instead of using observed choices, we use synthetic choices: In a pre-processing step, using a separately estimated logit model, every individual in the

sample is assigned to either class 1, which represents the base model, or class 2, in which the time-sensitivity parameter $\beta_{\text{traveltime}}$ is divided by a factor of 5 to generate the choice. We therefore estimate a separate travel time sensitivity parameter $\beta'_{\text{traveltime}}$ for that class. The probability to be assigned to class 1 is 70%, and the probability for class 2 is 30%. We investigate which estimation method performs better in discovering these now known latent population segments. The systematic equations for the utilities are:

$$
\begin{aligned}
V^{(1)}_{\text{walking}} &= \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{walking}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{walking}}, \\
V^{(1)}_{\text{cycling}} &= \text{ASC}_{\text{cycling}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{cycling}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{cycling}}, \\
V^{(1)}_{\text{pt}} &= \text{ASC}_{\text{pt}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{pt}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{pt}}, \\
V^{(1)}_{\text{driving}} &= \text{ASC}_{\text{driving}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{driving}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{driving}}, \\
V^{(2)}_{\text{walking}} &= \beta'_{\text{traveltime}} \cdot \text{traveltime}_{\text{walking}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{walking}}, \\
V^{(2)}_{\text{cycling}} &= \text{ASC}_{\text{cycling}} + \beta'_{\text{traveltime}} \cdot \text{traveltime}_{\text{cycling}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{cycling}}, \\
V^{(2)}_{\text{pt}} &= \text{ASC}_{\text{pt}} + \beta'_{\text{traveltime}} \cdot \text{traveltime}_{\text{pt}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{pt}}, \\
V^{(2)}_{\text{driving}} &= \text{ASC}_{\text{driving}} + \beta'_{\text{traveltime}} \cdot \text{traveltime}_{\text{driving}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{driving}}.
\end{aligned}
$$

Thus we have a total of seven parameters to be estimated. The results are presented in Tables 7, 8, 9 and 10. Table 7 shows that Biogeme initialized with CMA-ES (Bio-C) consistently yields worse log-likelihood values compared to both default initialization (Bio) and BHAMSLE-initialized Biogeme (Bio-B). In contrast, starting from $R = 50$, Biogeme initialized with BHAMSLE consistently achieves the best log-likelihood values, improving up to 6% compared to default initialization. This demonstrates the strength of BHAMSLE in providing high-quality starting points, while CMA-ES fails to guide Biogeme effectively.

In terms of estimation times, Biogeme and CMA-ES are again consistently low and independent of R, as no simulation is required. BHAMSLE as a pre-processing step only becomes computationally expensive for $R \geq 500$, and even then, the additional runtime is modest relative to the gains in log-likelihood.

Table 8 highlights the ability of the initialization methods to help Biogeme estimate the correct ratio between the two travel time sensitivity parameters. Biogeme initialized with CMA-ES struggles to produce meaningful ratios, often showing highly erratic and extreme values far from the true ratio of 5. In contrast, with BHAMSLE initialization, Biogeme is able to consistently estimate ratios closer to the true value, particularly starting from $R = 50$. This confirms that BHAMSLE significantly improves Biogeme's capacity to uncover the underlying latent structure. For the estimated class membership probabilities

Table 7 – Comparison of achieved log-likelihood values and runtimes using Biogeme with default initialization (Bio), Biogeme with CMA-ES (C) starting point (Bio-C), and Biogeme with BHAMSLE (B) starting point (Bio-B) when estimating a discrete mixture of logit models with synthetic choices (N = population size, R = number of draws, LL = log-likelihood, T = estimation time in seconds).

N	R	LL-Bio	LL-Bio-C	Gap (%)	LL-Bio-B	Gap (%)	T-Bio	T-C	T-Bio-C	T-B	T-Bio-B
500	1	-524.054	-524.054	0.00	-524.054	0.00	2	8	1	0	3
500	5	-525.484	-525.638	-0.03	-525.484	0.00	2	9	1	0	2
500	10	-525.483	-524.955	0.10	-525.483	0.00	4	9	1	0	1
500	20	-524.371	-535.487	-2.12	-517.502	1.31	2	9	1	2	2
500	50	-523.352	-521.152	0.42	-493.735	5.66	2	8	1	9	2
500	100	-523.352	-527.380	-0.77	-493.267	5.75	3	9	1	24	2
500	500	-525.485	-524.797	0.13	-494.588	5.88	1	9	1	211	1
500	1,000	-525.489	-521.119	0.83	-495.370	5.73	1	10	1	445	1
1,000	1	-1051.925	-1039.297	1.20	-1051.921	0.00	2	16	1	0	2
1,000	5	-1050.030	-1054.650	-0.44	-1050.034	0.00	2	19	2	0	4
1,000	10	-1051.926	-1084.740	-3.12	-1051.929	0.00	2	17	1	0	2
1,000	20	-1051.928	-1042.348	0.91	-1039.099	1.22	2	15	1	4	3
1,000	50	-1051.923	-1051.394	0.05	-988.700	6.01	2	18	2	24	2
1,000	100	-1051.929	-1074.221	-2.12	-987.545	6.12	2	19	2	51	2
1,000	500	-1051.434	-1056.687	-0.50	-989.194	5.92	1	19	1	506	1
1,000	1,000	-1051.926	-1042.242	0.92	-988.494	6.03	2	16	1	1121	1

(p_1, p_2) in Table 9, Biogeme initialized with CMA-ES often converges to incorrect or extreme distributions, including uniform splits or highly biased values unrelated to the true proportions $(0.70, 0.30)$. Biogeme initialized with BHAMSLE, however, consistently guides the estimation process towards the correct class membership probabilities starting from $R = 50$. Biogeme with default initialization struggles to recover the correct proportions, remaining closer to uniform splits.

Table 10 presents the estimated parameter values for all methods. As discussed above, CMA-ES produces an estimate for β'_{time} that is too close to β_{time}, failing to capture the intended difference in sensitivity across the two latent classes. In contrast, Biogeme initialized with BHAMSLE (Bio-B) provides more reliable estimates for both time parameters, consistently maintaining a clear separation between β_{time} and β'_{time}. The alternative-specific constants (ASCs) remain relatively stable across methods, though CMA-ES again introduces some unexpected deviations, particularly for ASC_{pb}, which differs notably from the values obtained with other methods. The cost sensitivity parameter, β_{cost}, is also highly unstable under Biogeme-C, with an exaggerated negative estimate that deviates significantly from the expected range.

Table 8 – Comparison of time-coefficient ratios derived using Biogeme with default initialization (Bio), Biogeme with CMA-ES (C) starting point (Bio-C), and Biogeme with BHAMSLE (B) starting point (Bio-B) when estimating a discrete mixture of logit models with synthetic choices (N = population size, R = number of draws, Ratio = $\beta_{\mathbf{traveltime}} / \beta'_{\mathbf{traveltime}}$).

N	R	Ratio-Bio	Ratio-C	Ratio-Bio-C	Ratio-B	Ratio-Bio-B
500	1	1.00	0.94	0.79	1.00	1.00
500	5	1.00	-3.47	-11.77	1.00	1.00
500	10	1.00	0.59	0.30	1.00	1.00
500	20	1.00	1.06	1.29	0.68	-0.24
500	50	1.00	9.35	-54.83	4.36	6.11
500	100	1.00	0.78	0.57	3.29	4.07
500	500	1.00	-1.58	-34.68	4.15	5.70
500	1,000	1.00	0.57	0.23	4.25	5.70
1,000	1	1.00	7.94	-12.20	1.00	1.00
1,000	5	1.00	1.11	-21.86	1.00	1.00
1,000	10	1.00	0.97	0.69	1.00	1.00
1,000	20	1.00	0.57	0.69	0.65	-0.67
1,000	50	1.00	1.03	1.21	3.95	5.98
1,000	100	1.00	-2.56	-0.17	3.11	3.67
1,000	500	1.00	0.89	-0.21	4.24	5.17
1,000	1,000	1.00	1.30	1.28	4.17	5.20

Overall, the results reinforce the findings from previous experiments: while CMA-ES produces highly variable and often unreliable estimates, BHAMSLE systematically improves Biogeme's ability to recover meaningful parameters. The separation between the two travel time sensitivity parameters is particularly well captured with BHAMSLE, demonstrating its effectiveness in guiding the estimation towards a more accurate latent class segmentation.

For the last experiment, we perform a similar altercation to class 1 as in experiment 1, this time replacing β_{cost} by a normally distributed $\beta_{\text{cost}}^{\text{mixed}} = \beta_{\text{cost}} + \beta_{\text{cost}}^{\text{std}} \cdot U_n$, with $U_n \sim \mathcal{N}(0, 1)$, together with adding a third latent class, which is hypothesized to be "lazy", which in this context means that they do not consider walking or cycling in their choice set. Class 2 remains the same as in the previous experiment. We assign individuals to class 1 with a probability of 50%, class 2 with 30%, and class 3 with 20%. We give the new systematic

Table 9 – Comparison of latent class probabilities derived using Biogeme with default initialization (Bio), Biogeme with CMA-ES (C) starting point (Bio-C), and Biogeme with BHAMSLE (B) starting point (Bio-B) when estimating a discrete mixture of logit models with synthetic choices (N = population size, R = number of draws, LL = log-likelihood, T = estimation time in seconds).

N	R	(p_1, p_2)-Bio	(p_1, p_2)-C	(p_1, p_2)-Bio-C	(p_1, p_2)-B	(p_1, p_2)-Bio-B
500	1	(0.50, 0.50)	(0.62, 0.38)	(0.49, 0.51)	(0.50, 0.50)	(0.50, 0.50)
500	5	(0.50, 0.50)	(0.50, 0.50)	(0.13, 0.87)	(0.50, 0.50)	(0.50, 0.50)
500	10	(0.50, 0.50)	(0.57, 0.43)	(0.43, 0.57)	(0.50, 0.50)	(0.50, 0.50)
500	20	(0.50, 0.50)	(0.57, 0.43)	(0.45, 0.55)	(0.40, 0.60)	(0.06, 0.94)
500	50	(0.50, 0.50)	(0.55, 0.45)	(0.53, 0.47)	(0.50, 0.50)	(0.70, 0.30)
500	100	(0.50, 0.50)	(0.61, 0.39)	(0.39, 0.61)	(0.63, 0.37)	(0.67, 0.33)
500	500	(0.50, 0.50)	(0.59, 0.41)	(0.36, 0.64)	(0.66, 0.34)	(0.71, 0.29)
500	1,000	(0.50, 0.50)	(0.55, 0.45)	(0.31, 0.69)	(0.66, 0.34)	(0.71, 0.29)
1,000	1	(0.50, 0.50)	(0.55, 0.45)	(0.21, 0.79)	(0.50, 0.50)	(0.50, 0.50)
1,000	5	(0.50, 0.50)	(0.63, 0.37)	(0.40, 0.60)	(0.50, 0.50)	(0.50, 0.50)
1,000	10	(0.50, 0.50)	(0.63, 0.37)	(0.46, 0.54)	(0.50, 0.50)	(0.50, 0.50)
1,000	20	(0.50, 0.50)	(0.50, 0.50)	(0.43, 0.57)	(0.46, 0.54)	(0.21, 0.79)
1,000	50	(0.50, 0.50)	(0.61, 0.39)	(0.54, 0.46)	(0.68, 0.32)	(0.79, 0.21)
1,000	100	(0.50, 0.50)	(0.57, 0.43)	(0.50, 0.50)	(0.64, 0.36)	(0.66, 0.34)
1,000	500	(0.50, 0.50)	(0.60, 0.40)	(0.39, 0.61)	(0.65, 0.35)	(0.69, 0.31)
1,000	1,000	(0.50, 0.50)	(0.58, 0.42)	(0.53, 0.47)	(0.69, 0.31)	(0.69, 0.31)

Table 10 – Comparison of average estimated parameter values and log-likelihood over 100 samples with $N = 1,000, R = 1,000$, using Biogeme with default initialization (Bio), CMA-ES, Biogeme with CMA-ES starting point (Biogeme-C), BHAMSLE, and Biogeme with BHAMSLE starting point (Biogeme-B) when estimating a discrete mixture of logit models with synthetic choices.

Parameter	Biogeme	CMA-ES	Biogeme-C	BHAMSLE	Biogeme-B
ASC_{bike}	-3.420	-4.271	-3.892	-3.794	-3.961
ASC_{car}	-0.685	-0.592	-0.912	-0.927	-1.465
ASC_{pb}	-0.356	-0.215	-0.123	-0.374	-0.585
β_{cost}	-0.158	-0.251	-1.983	-0.159	-0.145
β_{time}	-2.496	-2.472	-3.245	-6.296	-6.503
β'_{time}	-2.496	-1.901	-2.535	-1.509	-1.251
p_1	0.50	0.58	0.53	0.69	0.69
p_2	0.50	0.42	0.47	0.31	0.31
$\text{LL}(\beta)$	-1,051.926	-1,046.672	-1,042.242	-1,014.289	-988.494

Table 11 – Comparison of achieved log-likelihood values and runtimes using Biogeme with default initialization (Bio), Biogeme with CMA-ES (C) starting point (Bio-C), and Biogeme with BHAMSLE (B) starting point (Bio-B) when estimating a discrete-continuous mixture of logit with synthetic choices (N = population size, R = number of draws, LL = log-likelihood, T = estimation time in seconds).

N	R	LL-Bio	LL-Bio-C	Gap C (%)	LL-Bio-B	Gap B (%)	T-Bio	T-C	T-Bio-C	T-B	T-Bio-B
500	1	-528.517	-529.759	-0.23	-546.381	-3.38	1	27,195	1	0	1
500	5	-529.650	-527.686	0.37	-576.577	-8.86	6	27,031	3	0	6
500	10	-528.341	-533.415	-0.96	-553.279	-4.72	12	27,007	4	0	9
500	20	-531.867	-530.720	0.22	-534.740	-0.54	23	26,880	6	78	23
500	50	-527.150	-529.982	-0.54	-525.410	0.33	58	27,097	40	131	59
500	100	-530.017	-528.852	0.22	-528.374	0.31	108	27,162	63	271	122
500	500	-529.292	-527.867	0.27	-512.725	3.13	719	32,881	818	1,196	563
500	1,000	-525.036	-528.851	-0.73	-509.862	2.89	1,260	33,086	223	2,084	1,216
500	3,000	-525.564	-526.246	-0.13	-507.590	3.42	1,359	37,063	512	5,543	1,188
1,000	1	-1,051.300	-1,050.300	0.10	-1,053.298	-0.19	3	33,615	3	0	3
1,000	5	-1,051.880	-1,065.680	-1.31	-1,052.301	-0.04	12	33,763	10	0	11
1,000	10	-1,051.470	-1,048.940	0.24	-1,051.155	0.03	26	32,964	12	0	26
1,000	20	-1,049.410	-1,048.650	0.07	-1,051.824	-0.23	45	33,770	24	179	49
1,000	50	-1,050.840	-1,051.740	-0.09	-1,049.790	0.10	137	33,939	47	213	113
1,000	100	-1,054.330	-1,050.760	0.34	-1,030.502	2.26	296	32,651	129	551	240
1,000	500	-1,051.370	-1,053.790	-0.23	-1,016.570	3.31	1,659	38,360	792	2,803	1,171
1,000	1,000	-1,049.230	-1,059.890	-1.02	-1,014.081	3.35	2,805	37,943	459	5,248	2,423
1,000	3,000	-1,059.610	-1,066.090	-0.61	-1,022.137	3.53	2,459	40,720	1030	23,423	2,642

equations for the utilities of classes 1 and 3 below:

$$
\begin{aligned}
V^{(1)}_{\text{walking}} &= \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{walking}} + \beta^{\text{mixed}}_{\text{cost}} \cdot \text{cost}_{\text{walking}},\\
V^{(1)}_{\text{cycling}} &= \text{ASC}_{\text{cycling}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{cycling}} + \beta^{\text{mixed}}_{\text{cost}} \cdot \text{cost}_{\text{cycling}},\\
V^{(1)}_{\text{pt}} &= \text{ASC}_{\text{pt}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{pt}} + \beta^{\text{mixed}}_{\text{cost}} \cdot \text{cost}_{\text{pt}},\\
V^{(1)}_{\text{driving}} &= \text{ASC}_{\text{driving}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{driving}} + \beta^{\text{mixed}}_{\text{cost}} \cdot \text{cost}_{\text{driving}},\\
\\
V^{(3)}_{\text{pt}} &= \text{ASC}_{\text{pt}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{pt}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{pt}},\\
V^{(3)}_{\text{driving}} &= \text{ASC}_{\text{driving}} + \beta_{\text{traveltime}} \cdot \text{traveltime}_{\text{driving}} + \beta_{\text{cost}} \cdot \text{cost}_{\text{driving}}.
\end{aligned}
$$

Thus we have a total of nine parameters to be estimated. Based on the observations from the second experiment, where CMA-ES was not able to complete the estimation of the discrete-continuous mixture of logit for larger instances, we reduce the population size λ from 50 to 20 in this experiment. This choice remains generous given the nine-dimensional parameter space. The results are presented in Tables 11, 12 and 13. Table 11 shows that Biogeme initialized with CMA-ES (Bio-C) produces consistently worse log-likelihood values compared to both default initialization (Bio) and BHAMSLE-initialized Biogeme

Table 12 – Comparison of latent class probabilities derived using Biogeme with default initialization (Bio), Biogeme with CMA-ES (C) starting point (Bio-C), and Biogeme with BHAMSLE (B) starting point (Bio-B) when estimating a discrete-continuous mixture of logit with synthetic choices (N = population size, R = number of draws, LL = log-likelihood, T = estimation time in seconds).

N	R	(p_1, p_2, p_3)-Bio	$(p1, p2, p3)$-C	$(p1, p2, p3)$-Bio-C	(p_1, p_2, p_3)-B	(p_1, p_2, p_3)-Bio-B
500	1	(0.95, 0.05, 0.01)	(1.00, 0.00, 0.00)	(0.97, 0.01, 0.02)	(0.33, 0.33, 0.33)	(0.83, 0.05, 0.12)
500	5	(0.94, 0.05, 0.01)	(1.00, 0.00, 0.00)	(0.91, 0.01, 0.08)	(0.33, 0.33, 0.33)	(0.94, 0.06, 0.00)
500	10	(0.93, 0.05, 0.02)	(0.00, 1.00, 0.00)	(0.98, 0.01, 0.02)	(0.33, 0.33, 0.33)	(0.95, 0.00, 0.05)
500	20	(0.93, 0.07, 0.00)	(1.00, 1.00, 0.00)	(0.93, 0.01, 0.06)	(0.34, 0.34, 0.32)	(0.82, 0.04, 0.14)
500	50	(0.97, 0.02, 0.01)	(0.00, 1.00, 0.00)	(0.91, 0.00, 0.08)	(0.31, 0.39, 0.30)	(0.85, 0.01, 0.14)
500	100	(0.97, 0.03, 0.00)	(0.00, 1.00, 0.00)	(0.99, 0.01, 0.00)	(0.36, 0.36, 0.28)	(0.81, 0.02, 0.17)
500	500	(0.98, 0.02, 0.00)	(1.00, 0.00, 0.00)	(0.91, 0.04, 0.06)	(0.40, 0.42, 0.18)	(0.45, 0.35, 0.20)
500	1,000	(0.96, 0.04, 0.00)	(0.00, 0.00, 1.00)	(1.00, 0.00, 0.00)	(0.36, 0.38, 0.14)	(0.44, 0.31, 0.25)
500	3,000	(0.89, 0.11, 0.00)	(0.00, 1.00, 0.00)	(0.93, 0.07, 0.00)	(0.43, 0.32, 0.25)	(0.48, 0.29, 0.23)
1,000	1	(0.98, 0.02, 0.00)	(0.00, 1.00, 0.00)	(0.93, 0.03, 0.04)	(0.33, 0.33, 0.33)	(0.82, 0.14, 0.04)
1,000	5	(0.87, 0.12, 0.00)	(0.00, 1.00, 0.00)	(0.98, 0.01, 0.01)	(0.33, 0.33, 0.33)	(0.81, 0.18, 0.01)
1,000	10	(0.94, 0.06, 0.00)	(1.00, 0.00, 0.00)	(1.00, 0.00, 0.00)	(0.33, 0.33, 0.33)	(0.97, 0.03, 0.00)
1,000	20	(0.95, 0.05, 0.00)	(1.00, 0.00, 0.00)	(1.00, 0.00, 0.00)	(0.35, 0.35, 0.30)	(0.90, 0.09, 0.00)
1,000	50	(0.93, 0.07, 0.00)	(0.00, 1.00, 0.00)	(0.98, 0.01, 0.01)	(0.32, 0.42, 0.26)	(0.90, 0.10, 0.00)
1,000	100	(0.89, 0.11, 0.00)	(1.00, 0.00, 0.00)	(0.99, 0.01, 0.00)	(0.43, 0.32, 0.25)	(0.81, 0.06, 0.13)
1,000	500	(0.92, 0.08, 0.00)	(1.00, 0.00, 0.00)	(0.96, 0.04, 0.00)	(0.46, 0.28, 0.26)	(0.47, 0.26, 0.27)
1,000	1,000	(0.72, 0.09, 0.19)	(0.00, 0.00, 1.00)	(0.94, 0.04, 0.02)	(0.43, 0.32, 0.25)	(0.49, 0.31, 0.20)
1,000	3,000	(0.91, 0.01, 0.09)	(0.00, 1.00, 0.00)	(0.96, 0.01, 0.03)	(0.46, 0.31, 0.23)	(0.50, 0.31, 0.19)

(Bio-B). On the other hand, Biogeme initialized with BHAMSLE consistently achieves better log-likelihood values starting from $R = 500$, with improvements of up to 3.5% compared to default initialization.

The runtimes for CMA-ES (Bio-C) exhibit low variability across configurations, which can be attributed to both the consistent overhead when calling Biogeme's simulation module to evaluate the objective as well as the fact that for smaller R, the likelihood estimate may be noisier, leading to additional evaluations to converge to a stable solution. In contrast, BHAMSLE as a pre-processing step becomes computationally expensive only for $R \geq 500$, and while it incurs an increased runtime, this is again modest relative to the gains in log-likelihood, especially when compared to the runtime for CMA-ES.

Table 12 highlights the estimated class membership probabilities for each method. The expected probabilities are $(0.50, 0.30, 0.20)$, based on the synthetic choice generation. Biogeme initialized with CMA-ES frequently produces extreme or erratic estimates, often

Table 13 – Comparison of average estimated parameter values and log-likelihood over 100 samples with $N = 1,000, R = 3,000$, using Biogeme with default initialization (Bio), CMA-ES, Biogeme with CMA-ES starting point (Biogeme-C), BHAMSLE, and Biogeme with BHAMSLE starting point (Biogeme-B) when estimating a discrete-continuous mixture of logit with synthetic choices.

Parameter	Biogeme	CMA-ES	Biogeme-C	BHAMSLE	Biogeme-B
ASC_{bike}	-4.041	11.740	-3.386	-3.936	-3.890
ASC_{car}	-18.563	-20.829	-0.524	-1.439	-1.138
ASC_{pb}	-16.195	-6.060	-0.035	-1.281	-0.245
$\beta_{\text{cost, mean}}$	-0.156	-18.890	-0.172	-0.134	-0.175
$\beta_{\text{cost, std.}}$	-2.426	23.218	-3.156	-2.546	-1.277
β_{time}	-1.314	-4.257	-2.285	-5.283	-4.144
β'_{time}	-8.157	-4.738	-4.210	-2.120	-1.843
p_1	0.91	0.00	0.96	0.46	0.50
p_2	0.01	1.00	0.01	0.31	0.31
p_3	0.09	0.00	0.03	0.23	0.19
LL(β)	-1,059.610	-1,063.112	-1,066.090	-1,055.974	-1,022.137

converging to simplistic distributions such as $(1.00, 0.00, 0.00)$. Biogeme with default initialization struggles to align with the true proportions, particularly for smaller R, where the estimated probabilities remain far from the expected values. In contrast, starting from $R = 500$, Biogeme initialized with BHAMSLE closely approaches the true segmentation. For example, at $N = 1,000$ and $R = 3,000$, Biogeme with default initialization estimates the probabilities as $(0.91, 0.01, 0.09)$, while BHAMSLE guides Biogeme to $(0.50, 0.31, 0.19)$, closely matching the expected proportions.

Finally, Table 13 presents the estimated parameter values for all methods. The introduction of both a normally distributed cost sensitivity parameter $\beta_{\text{cost}}^{\text{mixed}}$ and an additional latent class increases the complexity of the estimation, leading to substantial variation across methods. CMA-ES again exhibits unstable behavior, with extreme values for $\beta_{\text{cost, mean}}$ and $\beta_{\text{cost, std.}}$, suggesting that it struggles to effectively capture the heterogeneity in cost sensitivity. The large magnitude of $\beta_{\text{cost, std.}}$ under CMA-ES indicates that it fails to estimate meaningful variation, whereas Biogeme with BHAMSLE initialization produces more reasonable estimates that maintain consistency with previous results. The estimates for the travel time sensitivities β_{time} and β'_{time} also show considerable discrepancies across methods. CMA-ES and Biogeme-C fail to establish a clear separation between the two parameters, whereas BHAMSLE provides a more stable and interpretable distinction.

The gap between β_{time} and β'_{time} is most consistently maintained under BHAMSLE-based initialization, reinforcing its ability to uncover the underlying latent class structure. The alternative-specific constants (ASCs) remain relatively stable across Biogeme-based methods, but CMA-ES produces large negative estimates for ASC_{car} and ASC_{pb}, deviating substantially from reasonably expected values.

Summary of results

The numerical experiments demonstrate the robustness and effectiveness of BHAMSLE in providing high-quality starting points for the estimation of latent class models, even in complex settings involving mixed parameters and multiple latent classes. Across all experiments, the use of BHAMSLE consistently led to improved log-likelihood values compared to standard initializations, with notable improvements ranging from 2% to 10%. Even under challenging conditions such as normally distributed sensitivity parameters and restricted choice sets, the heuristic allowed the estimation process to more accurately recover the latent population segments. In addition to improvements in model fit, BHAMSLE also yielded more stable and interpretable parameter estimates across all experiments. In contrast, Biogeme initialized with CMA-ES consistently underperformed, not only producing poorer log-likelihood values but also generating highly variable and often extreme parameter estimates, particularly for alternative-specific constants and sensitivity parameters. This instability was most pronounced in models with mixed cost or travel time parameters, where CMA-ES struggled to correctly estimate the distributional parameters, often collapsing the variance or producing unrealistic magnitudes. BHAMSLE, on the other hand, facilitated parameter recovery that more accurately reflected the expected segmentation and heterogeneity. Although Biogeme, as a highly optimized software, maintains an advantage in terms of computational time, the presence of numerous local optima in latent class models suggests that random re-initialization strategies or general methods like CMA-ES would likely be more computationally expensive and less effective. This further emphasizes the clear advantage of using BHAMSLE in scenarios where achieving a good fit is crucial, as it not only improves likelihood values but also provides parameter estimates that are more reliable and within a reasonable range.

5 Conclusions

This work aims to improve the estimation of advanced discrete choice models (DCMs) by introducing a new approach to handling multiple local maxima, which often cause unreliable convergence in standard optimization methods. Reformulating the Maximum Simulated Likelihood Estimation (MSLE) problem as a mixed-integer linear program (MILP) provides a structured alternative to continuous optimization, leveraging combinatorial techniques to systematically explore the solution space and identify globally optimal estimates. However, the computational intractability of solving large-scale instances exactly necessitates an alternative solution approach. To this end, we adapt the Breakpoint Heuristic Algorithm (BHA), originally developed for choice-based pricing, as a coordinate descent method that systematically explores local optima through decision-making breakpoints. Extending these principles, we present the Breakpoint Heuristic Algorithm for MSLE (BHAMSLE), designed to generate high-quality solutions that serve as robust initialization points for estimation.

We demonstrate through numerical experiments that this heuristic, by exploiting the structure of the choice problem, performs significantly better than a state-of-the-art global optimization method that does not incorporate this structure. The results show that this tailored approach leads to initialization points for the estimation that lead to up to 10% improved log-likelihood values, more stable and interpretable parameter estimates, and a better recovery of latent population segments, even in complex scenarios with mixed parameters and restricted choice sets. Unlike general-purpose optimization methods, the proposed heuristic avoids extreme or highly variable estimates—particularly for sensitivity parameters and alternative-specific constants.

While this approach induces some additional computational overhead, our findings indicate that this cost is justified by the increased likelihood of identifying high-quality solutions. Even though there is no formal guarantee of reaching the global optimum, spending additional computational time on a structured search significantly enhances estimation reliability. Future research should extend the application of this approach to more complex DCMs, evaluate its performance under different model specifications and real-world datasets, and explore parallelization techniques to further improve computational efficiency.

A Input data for illustrative example of discrete mixtures

Table 14 provides the full input dataset used to construct the likelihood surface shown in Figure 1. Each row corresponds to a synthetic individual characterized by the travel time of two alternatives (car and bus) and their observed choice. The individuals are grouped into three categories based on their decision behavior: the *rational* group contains individuals who consistently choose the faster option, the *irrational* group comprises individuals who consistently select the slower option (possibly reflecting the presence of unobserved factors not captured by the model), and the *ambiguous* group includes individuals facing symmetric alternatives and exhibiting mixed choice behavior. This dataset was used to evaluate the latent class log-likelihood over a grid of parameter values, including class-specific travel time coefficients and the class membership probability. Although simplified, the dataset is carefully constructed to highlight the non-convexity of the likelihood surface and the challenges posed by latent segmentation.

B Linearization and formulation of the MILP

In this section of the appendix, we explain how to linearize various parts and give the full description of the mixed-integer linear program (MILP) formulation of the MSLE problem for discrete-continuous mixture models.

B.1 Linearizing the objective

In order for the MSLE objective to be fully linear, we need to deal with the natural logarithm around the sum of choice variables. This can be achieved through a piece-wise linearization, as demonstrated in Fernandez Antolin (2018). We introduce auxiliary continuous variables z_{in} $\forall n \in \mathcal{N}, i \in C_n$, together with constants $L_r = (1+r)\ln(r) - r\ln(1+r)$ $\forall r \in \mathcal{R}$ and $K_r = \ln(r) - \ln(1+r)$ $\forall r \in \mathcal{R}$, representing the intercepts and slopes. The log-sum can then be written with the following constraints:

$$z_{in} \leq L_r - K_r \sum_{r \in \mathcal{R}} \omega_{inr}, \quad \forall n \in \mathcal{N}, i \in C_n. \tag{10}$$

Table 14 – Input data for illustrative example (travel times in minutes, choices are binary: 1 if bus is chosen, 0 if car is chosen).

Travel time (Car)	Travel time (Bus)	Group	Chosen alternative
10	30	Rational	0
10	28	Rational	0
10	26	Rational	0
10	24	Rational	0
12	18	Rational	0
14	22	Rational	0
15	25	Rational	0
20	22	Rational	0
16	17	Rational	0
30	10	Rational	1
28	10	Rational	1
26	10	Rational	1
24	10	Rational	1
18	12	Rational	1
22	14	Rational	1
25	15	Rational	1
22	20	Rational	1
19	16	Rational	1
30	10	Irrational	0
28	10	Irrational	0
26	10	Irrational	0
24	10	Irrational	0
20	17	Irrational	0
10	30	Irrational	1
10	28	Irrational	1
10	26	Irrational	1
10	24	Irrational	1
20	23	Irrational	1
15	15	Ambiguous	0
15	15	Ambiguous	1
18	18	Ambiguous	0
18	18	Ambiguous	1
22	22	Ambiguous	0
22	22	Ambiguous	1

Together with the direction of optimization, these constraints guarantee that

$$z_{in} = \sum_{r \in \mathcal{R}} \omega_{inr}, \quad \forall n \in \mathcal{N}, i \in C_n.$$

Taking into account that constants in the objective can be ignored, the objective function to maximize in the simulation-based setting, the simulated log-likelihood (sLL), can now be written as:

$$sLL(\beta) = \sum_{n \in \mathcal{N}} z_{y_n n}. \tag{11}$$

B.2 Linearizing the choice variables

To incorporate the variables U_{inr} and ω_{inr} into a MILP, it is necessary to linearize all indicator functions and products. These transformations require the introduction of additional auxiliary binary variables. We first handle the indicator function $\mathbb{1}_{[f_{nsr} \geq f_{ntr} \ \forall t \in \mathcal{S}]}$, determining the maximum scoring function per scenario. To this end, we introduce introduce $\delta^s_{nr} \in \{0, 1\}$ $\forall n \in \mathcal{N}, s \in \mathcal{S}, r \in \mathcal{R}$, defined by the following constraints:

$$f_{nsr} \geq f_{ntr} - M_1^{nsr}(1 - \delta^s_{nsr}), \quad \forall n \in \mathcal{N}, s, t \in \mathcal{S}, r \in \mathcal{R},$$

where M_1^{nsr} is a sufficiently large constant to cover the range between $\max_s f_{nsr}$ and $\min_s f_{nsr}$. As all scoring functions are now fully deterministic, we can define $M_1^{nsr} = f^{\max}_{nsr} - f^{\min}_{nsr}$ where $f^{\max}_{nsr} = \max_s f_{nsr}$ and $f^{\min}_{nsr} = \min_s f_{nsr}$, yielding:

$$f_{nsr} \geq f_{ntr} - (f^{\max}_{nsr} - f^{\min}_{nsr})(1 - \delta^s_{nsr}), \quad \forall n \in \mathcal{N}, s, t \in \mathcal{S}, r \in \mathcal{R}, \tag{12}$$

guaranteeing that

$$\delta^s_{nr} = \begin{cases} 1, & \text{if } f_{nsr} \geq f_{ntr} \ \forall t \in \mathcal{S}, \\ 0, & \text{otherwise.} \end{cases}$$

We apply the same method to linearize the choice variable $\omega_{inr} = \mathbb{1}_{[U_{inr} \geq U_{jnr} \ \forall j \in C_n]}$. We define $\omega_{inr} \in \{0, 1\}$ $\forall n \in \mathcal{N}, i \in C_n, r \in \mathcal{R}$, with the following constraints:

$$U_{inr} \geq U_{jnr} - M_2^{nr}(1 - \omega_{inr}), \quad \forall n \in \mathcal{N}, i, j \in C_n, r \in \mathcal{R},$$

where M_2^{nr} is a sufficiently large constant to cover the range between $\max_i U_{inr}$ and $\min_i U_{inr}$. We again define $M_2^{nr} = U_{\max}^{nr} - U_{\min}^{nr}$ where $U_{\max}^{nr} = \max_i U_{inr}$ and $U_{\min}^{nr} = \min_i U_{inr}$, yielding:

$$U_{inr} \geq U_{jnr} - (U_{\max}^{nr} - U_{\min}^{nr})(1 - \omega_{inr}), \quad \forall n \in \mathcal{N}, i, j \in C_n, r \in \mathcal{R}, \tag{13}$$

and ensuring that

$$\omega_{inr} = \begin{cases} 1, & \text{if } U_{inr} \geq U_{jnr} \ \forall j \in C_n, \\ 0, & \text{otherwise.} \end{cases}$$

Lastly, linearizing the product $\mathbb{1}_{[f_{nsr} \geq f_{ntr} \ \forall t \in \mathcal{S}]} U_{inr}^s = \delta_{nr}^s U_{inr}^s$ is straight-forward due to the binary nature of δ_{nr}^s. We can do so by introducing a new continuous variable $\tilde{U}_{inr}^s$, $\forall n \in \mathcal{N}$, $s \in \mathcal{S}, i \in C_n^s, r \in \mathcal{R}$ with the following constraints:

$$\begin{aligned} \tilde{U}_{inr}^s &\leq U_{inr}^s, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C_n^s, r \in \mathcal{R}, \\ \tilde{U}_{inr}^s &\leq M_3^{nrs} \delta_{nr}^s, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C_n^s, r \in \mathcal{R}, \\ \tilde{U}_{inr}^s &\geq U_{inr}^s - M_3^{nrs}(1 - \delta_{nr}^s), \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C_n^s, r \in \mathcal{R}, \end{aligned}$$

where M_3^{nrs} is a sufficiently large constant to cover the range between $\max_i U_{inr}^s$ and 0. Using the notation from above, we can define $M_3^{nrs} = U_{\max}^{nrs}$, yielding:

$$\tilde{U}_{inr}^s \leq U_{inr}^s, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C_n^s, r \in \mathcal{R}, \tag{14}$$

$$\tilde{U}_{inr}^s \leq U_{\max}^{nrs} \delta_{nr}^s, \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C_n^s, r \in \mathcal{R}, \tag{15}$$

$$\tilde{U}_{inr}^s \geq U_{inr}^s - U_{\max}^{nrs}(1 - \delta_{nr}^s), \quad \forall n \in \mathcal{N}, s \in \mathcal{S}, i \in C_n^s, r \in \mathcal{R}, \tag{16}$$

making it so that

$$\tilde{U}_{inr}^s = \delta_{nr}^s U_{inr}^s.$$

B.3 MSLE as a MILP in the case of a discrete-continuous mixture model

The complete description of MSLE as a MILP in the case of a discrete-continuous mixture model is given below:

$$\max_{\beta,\pi} \quad \sum_{n\in\mathcal{N}} z_{y_n n}$$

s.t.

$$\begin{aligned}
&U^s_{inr} = \sum_{k\in\mathcal{K}^s\setminus\mathcal{M}} x_{ink}\beta_k + \sum_{m\in\mathcal{M}^s} x_{inm}\beta_m + \varepsilon_{inr}, && \forall n\in\mathcal{N}, s\in\mathcal{S}, i\in C^s_n, r\in\mathcal{R},\\
&f_{nsr} = \sum_{l\in\mathcal{L}} x_{nsl}\alpha_l + \delta_{nsr}, && \forall n\in\mathcal{N}, s\in\mathcal{S}, r\in\mathcal{R},\\
&U_{inr} = \sum_{s\in\mathcal{S}} \tilde{U}^s_{inr}, && \forall n\in\mathcal{N}, s\in\mathcal{S}, i\in C_n, r\in\mathcal{R},\\
&f_{nsr} \geq f_{ntr} - (f^{\max}_{nsr} - f^{\min}_{nsr})(1-\delta^s_{nsr}), && \forall n\in\mathcal{N}, s,t\in\mathcal{S}, r\in\mathcal{R},\\
&U_{inr} \geq U_{jnr} - (U^{nr}_{\max} - U^{nr}_{\min})(1-\omega_{inr}), && \forall n\in\mathcal{N}, i,j\in C_n, r\in\mathcal{R},\\
&\tilde{U}^s_{inr} \leq U^s_{inr}, && \forall n\in\mathcal{N}, s\in\mathcal{S}, i\in C^s_n, r\in\mathcal{R},\\
&\tilde{U}^s_{inr} \leq U^{nrs}_{\max}\delta^s_{nr}, && \forall n\in\mathcal{N}, s\in\mathcal{S}, i\in C^s_n, r\in\mathcal{R},\\
&\tilde{U}^s_{inr} \geq U^s_{inr} - U^{nrs}_{\max}(1-\delta^s_{nr}), && \forall n\in\mathcal{N}, s\in\mathcal{S}, i\in C^s_n, r\in\mathcal{R},\\
&z_{in} \leq L_r - K_r \textstyle\sum_{r\in\mathcal{R}} \omega_{inr}, && \forall n\in\mathcal{N}, i\in C_n,\\
&\beta_k, \alpha_l \in \mathbb{R}, && \forall k\in \bigcup_{s\in\mathcal{S}} \mathcal{K}^s, l\in\mathcal{L}\\
&U_{inr}, U^s_{inr}, \tilde{U}^s_{inr}, z_{in} \in \mathbb{R}, && \forall n\in\mathcal{N}, s\in\mathcal{S}, i\in C_n, r\in\mathcal{R},\\
&\omega_{inr}, \delta^s_{nr} \in \{0,1\}, && \forall n\in\mathcal{N}, s\in\mathcal{S}, i\in C_n, r\in\mathcal{R},
\end{aligned}$$

where $\mathcal{K}^s$ represents the set of explanatory variables and $\mathcal{M}^s$ the set of normally distributed parameters considered in class s.

C References

Bierlaire, M. (2023) A short introduction to biogeme, *Technical Report*, **TRANSP-OR 230620**, Transport and Mobility Laboratory, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

Bierlaire, M., K. Axhausen and G. Abay (2001) The acceptance of modal innovation: The

case of swissmetro, paper presented at the *Swiss transport research conference*.

Bierlaire, M., M. Thémans and N. Zufferey (2010) A heuristic for nonlinear global optimization, *INFORMS Journal on Computing*, **22** (1) 59–70.

Boxall, P. C. and W. L. Adamowicz (2002) Understanding heterogeneous preferences in random utility models: a latent class approach, *Environmental and resource economics*, **23**, 421–446.

Eberhart, R. and J. Kennedy (1995) Particle swarm optimization, paper presented at the *Proceedings of the IEEE international conference on neural networks*, vol. 4, 1942–1948.

Fernandez Antolin, A. (2018) Dealing with correlations in discrete choice models, Ph.D. Thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland.

Greene, W. H. and D. A. Hensher (2003) A latent class model for discrete choice analysis: contrasts with mixed logit, *Transportation Research Part B: Methodological*, **37** (8) 681–698.

Haering, T., R. Torres, Legault, Fabian and M. Bierlaire (2024) Heuristics and exact algorithms for choice-based capacitated and uncapacitated continuous pricing, *Technical Report*, **TRANSP-OR 240918**, Transport and Mobility Laboratory, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

Hansen, N., S. D. Müller and P. Koumoutsakos (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es), *Evolutionary computation*, **11** (1) 1–18.

Hillel, T., M. Z. Elshafie and Y. Jin (2018) Recreating passenger mode choice-sets for transport simulation: A case study of london, uk, *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction*, **171** (1) 29–42.

Jung, T. and K. A. Wickrama (2008) An introduction to latent class growth analysis and growth mixture modeling, *Social and personality psychology compass*, **2** (1) 302–317.

Lubke, G. H. and B. O. Muthén (2005) Investigating population heterogeneity with factor mixture models, *Psychological Methods*, **10** (1) 21–39.

Peer, S., J. Knockaert and E. T. Verhoef (2016) Train commuters' scheduling preferences:

Evidence from a large-scale peak avoidance experiment, *Transportation Research Part B: Methodological*, **83**, 314–333.

Train, K. (2003) *Discrete Choice Methods with Simulation*, Cambridge University Press. Http://emlab.berkeley.edu/books/choice.html.