An adaptive Monte Carlo algorithm for computing mixed logit estimators

Bastin, Fabian; Cirillo, Cinzia; Toint, Philippe

Published in:
Computational Management Science

DOI:
10.1007/s10287-005-0044-y

Publication date:
2006

Document Version
Peer reviewed version

An adaptive Monte Carlo algorithm for computing mixed logit estimators

Fabian Bastin*, Cinzia Cirillo†, Philippe L. Toint†

Abstract

Researchers and analysts are increasingly using mixed logit models for estimating responses to forecast demand and to determine the factors that affect individual choices. However the numerical cost associated to their evaluation can be prohibitive, the inherent probability choices being represented by multidimensional integrals. This cost remains high even if Monte Carlo or quasi-Monte Carlo techniques are used to estimate those integrals. This paper describes a new algorithm that uses Monte Carlo approximations in the context of modern trust-region techniques, but also exploits accuracy and bias estimators to considerably increase its computational efficiency. Numerical experiments underline the importance of the choice of an appropriate optimisation technique and indicate that the proposed algorithm allows substantial gains in time while delivering more information to the practitioner.

Keywords: Maximum simulated likelihood estimation; Trust-region algorithms; Monte Carlo samplings; Mixed logit models

Submitted for publication to Computational Management Science

1 Introduction

Discrete choice analysis is an econometric technique for estimating disaggregate demand models. In particular, the multinomial logit and its extensions (see Bhat and Koppelman [9] for a review in the context of travel demand) are widely used, but the more powerful mixed logit models are gaining acceptance among practitioners and researchers. Its main strengths are the possibility to estimate taste variations, to take into account state dependence across observations and to avoid the problem of restricted substitution patterns in standard logit models. However,

*Research Fellow of the National Fund for Scientific Research (FNRS), Department of Mathematics, University of Namur, Belgium.

†Transportation Research Group, Department of Mathematics, University of Namur, Belgium.

these models are more difficult to interpret and the numerical cost associated with their evaluation is significant. In particular the inherent choice probabilities are represented by multidimensional integrals which can only be calculated, in real applications, by simulation techniques. The computed estimators are, under reasonable assumptions, asymptotically consistent and efficient (see Gouriéroux and Monfort [20] and Hajivassiliou and McFadden [22]). Bastin, Cirillo and Toint [5] also show that, for a fixed population size, the Monte Carlo procedure gives solutions converging almost surely towards true maximum likelihood estimators, covering both unconstrained and constrained cases. Unfortunately, the evaluation costs can still be prohibitive due the required sample sizes, as mentioned for instance by Hensher and Greene [24]. These authors however underline the need for speed in practice, in order to allow the exploration of alternative model specifications. As a consequence, current research has turned to the cheaper quasi-Monte Carlo approaches, based on low discrepancy sequences, which have been shown to produce more accurate integration approximations when the number of draws is fixed, for instance in the study of physics problem (Morokoff and Caflish [27]). Bhat [8] and Train [39] for instance advocate using Halton sequences (Halton [23]) for mixed logit models and find they perform much better than pure random draws in simulation estimation. Garrido [18] explores the use of Sobol sequences, while Sándor and Train [33] compare randomised Halton draws and (t, m, s)-nets.

This trend is not without drawbacks. For instance, Bhat [8] points out that the coverage of the integration domain by Halton sequences rapidly deteriorates for high integration dimensions and consequently proposes a heuristic based on the use of scrambled Halton sequences. He also randomises these sequences in order to allow the computation of the simulation variance of the model parameters. Hess, Polak and Daly [25] however show that scrambled Halton methods are very sensitive to the number of draws, and can behave poorly when this number increases. Recently Hess, Train and Polak [26] have proposed the use of modified Latin hypercube sampling and have reported better performance than with any of the Halton based approaches, while other authors have found mitigated results [2, 37]. By contrast, the dimensionality problem is irrelevant in pure Monte Carlo methods, and while computational experiments show that for low dimensional integration quasi-Monte Carlo techniques outperform Monte Carlo integration, the advantage is less clear in high-dimension (Deák [15], Morokoff and Caflish [27]). The same is reported for estimation of mixed logit models, where Monte Carlo methods are again competitive when high-dimensional problems are considered (Hess, Train and Polak [26]). Moreover Monte Carlo techniques also benefit from a credible theory for the convergence of the calibration process, as well as of stronger statistical foundations (see for instance Fishman [17] for a general review, Rubinstein and Shapiro [32], Shapiro [34, 35] for application to stochastic programming, and Bastin, Cirillo and Toint [5] for more specific developments in the context of nonlinear programming and mixed logit problems). In particular, statistical inference on the objective function is possible, while the quality of the results can only be estimated in practice, for quasi-Monte Carlo procedures, by repeating the calibration

process on randomised samples and by varying the number of random draws.

In this paper, we attempt to capitalise on the desirable aspects of pure Monte Carlo techniques while significantly improving their efficiency. Our approach is to propose a new algorithm for stochastic programming using Monte Carlo methods, that is based on the trust-region technique. Trust-region methods are well-known in nonlinear non-convex/non-concave optimisation, and have been proved to be reliable and efficient for both constrained and unconstrained problems. Moreover, the associated theoretical corpus is extensive (see Conn, Gould and Toint [14]). Our efficiency objective led us to adapt the traditional deterministic trust-region algorithm to handle stochasticity and, more importantly, to allow an adaptive variation of the number of necessary random draws. This technique results in an algorithm that is numerically competitive with existing tools for mixed logit models, while giving more information to the practitioner. This underlines the importance of the choice of optimisation algorithm when looking for numerical performances and shows that exploitation of statistical inference is valuable. Therefore quasi-Monte Carlo methods are not the only way to decrease the numerical cost in mixed logit estimation, and future research could benefit from attempts to combine our approach with quasi-Monte Carlo draws, especially when the integrals dimensionality is not too high.

Our exposition is organised as follows. We briefly review the mixed logit problem and some of its properties in Section 2. We then introduce our new algorithm in Section 3 and analyse its convergence properties in Section 4. Section 5 presents our numerical experimentation and discusses its results. Some conclusions and perspectives are finally outlined in Section 6.

2 The Mixed Logit model

2.1 The problem and its approximation

Discrete choice models provide a description of how individuals perform a selection amongst a finite set of alternatives. Let I be the population size and $\mathcal{A}(i)$ the set of available alternatives for individual i, $i = 1, \ldots, I$. For each individual i, each alternative A_j, $j = 1, \ldots, |\mathcal{A}(i)|$ has an associated utility, depending on the individual characteristics and the relative attractiveness of the alternative, which is assumed to have the form

$$U_{ij} = V_{ij} + \boldsymbol{\epsilon}_{ij}, \tag{1}$$

where $V_{ij} = V_{ij}(\beta_j, x_{ij})$ is a function of a vector of model parameters β_j and of x_{ij}, the observed attributes of alternative A_j, while $\boldsymbol{\epsilon}_{ij}$ is a random term reflecting the unobserved part of the utility. Without loss of generality, it can be assumed that the residuals $\boldsymbol{\epsilon}_{ij}$ are random variables with zero mean and a certain probability distribution to be specified. The parameter vectors β_j, $j = 1, \ldots, |\mathcal{A}(i)|$, are assumed to be constant for all individuals but may vary across alternatives. A model parameter is called generic if it is involved in all alternatives, and has the same value for all

of them. Otherwise it is said to be (alternative) specific. Since we can decompose a specific parameter in several parameters taking the same value for a subset of alternatives, and associated to null observations for others, we may assume, without loss of generality, that all parameters are generic. In order to simplify the notation, we will hence omit the subscript j for parameters vectors.

The theory assumes that individual i selects the alternative that maximises his/her utility. Therefore the probability that individual i chooses alternative A_j is given by

$$P_{ij} = P\left[\epsilon_{il} \leq \epsilon_{ij} + \left(V_{ij} - V_{il}\right), \forall A_l \in \mathcal{A}(i)\right].$$

The particular form of the choice probability depends on the random terms $\boldsymbol{\epsilon}_{ij}$ in (1). If we assume that they are independently Gumbel distributed with mean 0 and scale factor 1.0, the probability that the individual i chooses alternative j can be expressed with the logit formula

$$L_{ij}(\beta) = \frac{e^{V_{ij}(\beta)}}{\sum_{l=1}^{|\mathcal{A}(i)|} e^{V_{il}(\beta)}}, \tag{2}$$

where we have simplified our notation by dropping the explicit mention of the dependence of L_{ij} and V_{ij} on the known observations x_{ij}. Formula (2) characterises the classical multinomial logit model.

In the Mixed Multinomial Logit (MMNL) model, the vectors V_i, $i = 1, \ldots, I$, themselves contain random elements, and we will write $V_{ij} = g(\beta, x_{ij}, \boldsymbol{\xi}_{ij})$. This formulation can be exploited in two mathematically identical, yet conceptually different ways. In the error-components formulation (see for instance Walker [40]), the additional vector $\boldsymbol{\xi}_{ij}$ contains a set of Normally-distributed error-components that can be used to induce correlation across alternatives and/or heteroscedasticity in the unobserved parts of utilities across the choice-set. In the more popular random-coefficients formulation (see for example Revelt and Train [31]), the additional error-term is exploited to introduce taste heterogeneity in some of the coefficients across decision-makers, such that β becomes itself a random vector, so we can assume that individual parameters vectors $\beta(i)$, $i = 1, \ldots, I$, are realizations of a random vector $\boldsymbol{\beta}$. Finally, both approaches can be combined, to simultaneously allow for random taste heterogeneity, inter-alternative correlation, and heteroscedasticity. Although the applications presented in this paper concentrate on the random-coefficients formulation, the issues discussed, as well as the solutions presented, can be applied to both formulations. We therefore assume that $\boldsymbol{\beta}$ is itself derived from a random vector $\boldsymbol{\omega}$ and a parameters vector θ, which we express $\boldsymbol{\beta} = \beta(\boldsymbol{\omega}, \theta)$. For example, if $\boldsymbol{\beta}$ is a K-dimensional normally distributed random vector, whose components are mutually independent, we may choose $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_K)$, with $\omega_k \sim N(0,1)$[1], and let θ specify the means and standard deviations of the components of $\boldsymbol{\beta}$. The probability choice is then

[1] $N(\mu, \sigma^2)$ stands for the normal distribution with mean μ and standard deviation σ.

given by

$$P_{ij}(\theta) = E_P\left[L_{ij}\left(\boldsymbol{\omega}, \theta\right)\right] = \int L_{ij}(\omega, \theta) P(d\omega) = \int L_{ij}(\omega, \theta) f(\omega) d\omega, \qquad (3)$$

where P is the probability measure associated with ω and $f(\cdot)$ is its density function.

The vector of parameters θ is then estimated by maximising the log-likelihood function, i.e. by solving the program

$$\max_{\theta} LL(\theta) = \max_{\theta} \frac{1}{I} \sum_{i=1}^{I} \ln P_{ij_i}(\theta), \qquad (4)$$

where j_i is the alternative choice made by the individual i. This involves the computation of $P_{ij_i}(\theta)$ for each individual i, $i = 1, \ldots, I$, which is impractical since it requires the evaluation of one multidimensional integral per individual. The value of $P_{ij_i}(\theta)$ is therefore replaced by a Monte-Carlo estimate obtained by sampling over ω, and given by

$$SP^R_{ij_i}(\theta) = \frac{1}{R} \sum_{r_i=1}^{R} L_{ij_i}(\omega_{r_i}, \theta), \qquad (5)$$

where R is the number of random draws ω_{r_i}, taken from the distribution function of $\boldsymbol{\omega}$. As a result, θ is now computed as the solution of the simulated log-likelihood problem

$$\max_{\theta} SLL^R(\theta) = \max_{\theta} \frac{1}{I} \sum_{i=1}^{I} \ln SP^R_{in_i}(\theta). \qquad (6)$$

We will denote by θ^*_R a solution of this last approximate problem (often called the Sample Average Approximation, or SAA), while θ^* denotes a solution of the true problem (4).

2.2 Convergence of approximations and useful estimators

Bastin, Cirillo and Toint [5] have shown that the mixed logit problem can be viewed as a generalisation of the classical stochastic programming problem, which in turn implies that the estimators derived from the SAA problem converge almost surely towards the true maximum likelihood estimators as the sample size R tends to infinity. For a fixed population size (as is the case in most real applications), they assume that

A.0 the random draws are independently and identically distributed, both for each individual and across them,

A.1 the solutions θ^*_R of the SAA problems (6) remain in some convex compact set S for all R sufficiently large,

A.2 the utilities $V_{ij}(\omega, \cdot)$, $i = 1, \ldots, I$, $j = 1, \ldots, N$, are continuously differentiable for almost every ω,
A.3 for $t = 1, \ldots, m$, $\frac{\partial}{\partial \theta_t} V_{ij}\left(\omega_{r_i}, \theta, x_{ij_i}\right)$, $j = 1, \ldots, N$, is dominated by a P-integrable function.
They deduce in particular that with probability one, as R tends to infinity, there exists some limit point θ^* of $(\theta_R^*)_{R=1}^{\infty}$ that is first-order critical for the true log-likelihood function under some reasonable assumptions, if θ_R^* $(R = 1, \ldots)$ are first order critical for the corresponding SAA problem. Second-order criticality can also be established under additional assumptions.

When the population size tends to infinity (instead of being fixed), it can also be proved that the SAA estimators θ_R^* are asymptotically equivalent to the estimators associated to the true likelihood if R tends to infinity and $\sqrt{I}/R$ tends to 0 (Gouriéroux and Monfort [20]).

It is furthermore possible to estimate the error made by using the SAA problem (6) instead of the true problem (4). If we consider a fixed population size and take an independently and identically distributed sample for each individual, it is possible to show that

$$I^2 \left(LL(\theta) - SLL^R(\theta)\right) \Rightarrow N\left(0, \sqrt{\sum_{i=1}^{I} \frac{\sigma_{ij_i}^2(\theta)}{R\left(P_{ij_i}(\theta)\right)^2}}\right), \tag{7}$$

as R tends to infinity, where $\Rightarrow$ means convergence in distribution and where σ_{ij_i} is the standard deviation of $L_{ij_i}(\theta)$. Therefore, $SLL^R(\theta)$ is an asymptotically unbiased estimator of $LL(\theta)$, and the asymptotic value of the confidence interval radius is given by

$$\epsilon_\delta^R(\theta) = \alpha_\delta \frac{1}{I} \sqrt{\sum_{i=1}^{I} \frac{\sigma_{ij_i}^2(\theta)}{R\left(P_{ij_i}(\theta)\right)^2}}, \tag{8}$$

where α_δ is the quantile of a $N(0,1)$, associated to some level of signification δ. In practice, one typically chooses $\alpha_{0.9} \approx 1.64$ or $\alpha_{0.95} \approx 1.96$ and evaluates $\epsilon_\delta^R(\theta)$ by replacing $\sigma_{ij_i}(\theta)$ and $P_{ij_i}(\theta)$ by their SAA estimators $\sigma_{ij_i}^R(\theta)$ and $P_{ij_i}^R(\theta)$.

Finally, the simulation bias for finite R can be approximated by

$$B^R(\theta) := E[SLL^R(\theta)] - LL(\theta) = -\frac{I\left(\epsilon_\delta^R(\theta)\right)^2}{2\alpha_\delta^2}. \tag{9}$$

Details of these derivations can be again found in Bastin, Cirillo and Toint [5]. For a more general discussion about the asymptotic bias, the reader is referred to Gouriéroux and Monfort [21], Section 3.2.

3 A new algorithm for solving the SAA problem

Solving the SAA problem (6) is very expensive even on modern computers, as pointed in the introduction, since I, the number of multidimensional integrals in

the expression of the objective function, can be large. For the reasons previously discussed, we choose to propose a new, efficient Monte Carlo algorithm that exploits statistical inference to limit the number of draws needed in the early iterations, away from the solution. The main idea is to generate a sample set prior the optimisation process, with $R_{\max}$ i.i.d. random draws per individual. At iteration k, only a (possibly small) subset of this sample set will be used, by selecting R_k of the $R_{\max}$ random draws for each individual (for simplicity, the first R_k). It is important to observe that SLL^R is, in this context, a well defined smooth function for each choice of R, which makes methods for optimising deterministic smooth functions relevant.

The idea to only use a small number of random draws when the iterate is far from the solution is not new in stochastic programming. Shapiro and Homem-de-Mello [36] for instance consider two-stage programs with recourse. They propose an algorithm using independent samples of increasing sizes, and prove the convergence of the method to a point satisfying a first-order criticality statistical test for the true problem. In the trust-region framework, other algorithms ensuring increases of the objective greater than the noise on the objective function's value have also been proposed (see Conn, Gould, Sartenaer and Toint [13] and Conn, Gould and Toint [14], Section 10.6). A direct application of such techniques however usually leads to unmanageable sample sizes, with respect to memory consumption and computational times. To circumvent that problem, we choose here a maximum allowed sample size $R_{\max}$ prior to the optimisation process, generate the corresponding sample set and use a sub-sample of this set at each iteration. The successive sample sets are thus correlated, which allows us to possibly accept increases of the objective smaller than the noise, and to consider a non-monotone sequence of sample sizes.

3.1 A trust-region algorithm with dynamic accuracy

The proposed algorithm is of the trust-region type (see Conn, Gould and Toint [14] for details and an extensive bibliography on optimisation methods of this type). The main idea of a trust-region algorithm is, at a current iterate θ_k, to calculate a trial point $\theta_k + s_k$ by maximising a model m_k of the objective function inside a trust region at each iteration. This region is defined as

$$\mathcal{B}_k = \{\theta \in \mathbb{R}^m \,|\, \|\theta - \theta_k\| \leq \psi_k\}\,,$$

where ψ_k is called the trust-region radius. The predicted and actual increases in objective function values are then compared. If the agreement is sufficiently good, the trial point becomes the new iterate and the trust-region radius is (possibly) enlarged. If this agreement is poor, the trust region is shrunk in order to improve the quality of the model. In addition, if the model approximates the objective function well compared to the accuracy of the objective function itself (which is dependent on the Monte Carlo sample size), we surmise that we could work with a less precise approximation and therefore reduce the sample size. On the other

hand, if the model agreement is poor compared to the precision of the objective function, we increase the sample size in an attempt to correct this deficiency.

A formal description of our algorithm follows.

Algorithm 1: Trust-region maximisation algorithm

Step 0. Initialisation. An initial point θ_0 and an initial trust-region radius ψ_0 are given. The constants η_1, η_2, γ_1, and γ_2 are also given and satisfy

$$0 < \eta_1 \leq \eta_2 < 1 \text{ and } 0 < \gamma_1 \leq \gamma_2 < 1.$$

Set a minimum number of draws $R_{\min} = R^0_{\min}$ and a sample size R_0 satisfying $\|\nabla_\theta SLL^{R_0}(\theta_0)\| \neq 0$ if $\epsilon_\delta^{R_0}(\theta_0) \neq 0$, except if $R_0 = R_{\max}$. Compute $SLL^{R_0}(\theta_0)$ and set $k = 0$, $t = 0$.

Step 1. Stopping test. Stop if $\|\nabla_\theta SLL^{R_k}(\theta_k)\| = 0$ and either $R_k = R_{\max}$, or $\epsilon_\delta^{R_k}(\theta_k) = 0$. Otherwise go to Step 2.

Step 2. Model definition. Define a model $m_k^{R_k}$ of $SLL^{R_k}(\theta)$ in $\mathcal{B}_k$. Compute a new adequate sample size R^+ (see Algorithm 2 below). Set $R^- = R_k$.

Step 3. Step calculation. Compute a step s_k that sufficiently increases the model $m_k^{R_k}$ and such that $\theta_k + s_k \in \mathcal{B}_k$. Set

$$\Delta m_k^{R_k} = m_k^{R_k}(\theta_k + s_k) - m_k^{R_k}(\theta_k).$$

Step 4. Comparison of increases. Compute $SLL^{R^+}(\theta_k + s_k)$ and define

$$\rho_k = \frac{SLL^{R^+}(\theta_k + s_k) - SLL^{R_k}(\theta_k)}{\Delta m_k^{R_k}}. \tag{10}$$

Step 5. Sample size update. If $\rho_k < \eta_1$ and $R_k \neq R^+$, modify R^- or the candidate sample size R^+ to take account of bias and variance differences (see Algorithm 3 below). Recompute ρ_k.

Step 6. Acceptance of the trial point. If $\rho_k < \eta_1$, define $\theta_{k+1} = \theta_k$, $R_{k+1} = R^-$. Otherwise define $\theta_{k+1} = \theta_k + s_k$ and set $R_{k+1} = R^+$; increment t by one.

If $R_{k+1} \neq R^{\max}$, $\|\nabla_\theta SLL^{R_{k+1}}(\theta_{k+1})\| = 0$, and $\epsilon_\delta^{R_{k+1}}(\theta_{k+1}) \neq 0$, increase R_{k+1} to some size less or equal to $R_{\max}$ such that $\|\nabla_\theta SLL^{R_{k+1}}(\theta_{k+1})\| \neq 0$ if $R_{k+1} \neq R_{\max}$, and compute $SLL^{R_{k+1}}(\theta_{k+1})$.

If $R_k = R_{k+1}$ or if sufficient decrease has been observed since the last evaluation of $SLL^{R_{k+1}}$, set $R^{k+1}_{\min} = R^k_{\min}$. Otherwise define $R^{k+1}_{\min} >$

$R^k_{\min}$ (see Algorithm 4 below).

Step 7. Trust-region radius update. Set

$$\psi_{k+1} = \begin{cases} \min\left\{10^{20}, \max(2s_k, \psi_k)\right\} & \text{if } \rho_k \geq \eta_2, \\ \gamma_2 \psi_k & \text{if } \rho_k \in [\eta_1, \eta_2), \\ \gamma_1 \psi_k & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment k by 1 and go to Step 1.

In the current implementation, we have set $\eta_1 = 0.01$, $\eta_2 = 0.75$, $\gamma_1 = 0.5$ and $\gamma_2 = 0.5$. We say that the iteration k is successful if $\rho_k \geq \eta_1$ and very successful if $\rho_k \geq \eta_2$; the variable t is used to record the number of successful iterations. We will refer to Algorithm 1 as the BTRDA algorithm, for basic trust-region with dynamic accuracy, by analogy with the basic trust-region (BTR) algorithm (Conn, Gould and Toint [14], Chapter 6). The two algorithms coincide indeed if we fix R_k to $R_{\max}$.

3.2 Model choice and trial step computation

We use a quadratic model, defined as

$$m^R_k(\theta_k + s) = m^R_k(\theta_k) + \langle g^R_k, s\rangle + \frac{1}{2}\langle s, H_k s\rangle,$$

where

$$m^R_k(\theta_k) = SLL^R(\theta_k) \text{ and } g^R_k = \nabla_\theta SLL^R(\theta_k), \tag{11}$$

and where H_k is a symmetric approximation to $\nabla^2_{\theta\theta} SLL^R(\theta_k)$. In our implementation we use the symmetric BFGS quasi-Newton update to obtain such an approximation, as described in Nocedal and Wright [28], page 198.

The same Hessian approximation scheme as in the BHHH procedure (Berndt et al. [7]) could also be used. However while it delivers significant speed gains for simple cases, its performance and robustness rapidly decreases when dealing with more complex problems (Bastin, Cirillo and Toint [4]). This however suggests that a good preconditioning of the problem could be an important point to investigate in our search for speed efficiency.

The computation of the step s_k is performed using the Steihaug-Toint method (see for instance Conn, Gould, and Toint [14], section 7.5.1, as well as Nocedal and Wright [28], page 75).

3.3 The variable sample size strategy

A crucial ingredient to make our algorithm efficient is to design a technique which adapts the number of draws used to the optimality level of the successive iterates. We now outline our proposed approach.

Prior to the optimisation, the user chooses a maximum sample size $R_{\max}$. A minimum sample size $R^0_{\min}$ is defined to allow estimation of the accuracy; we (arbitrarily) set $R^0_{\min}$ to 36 in our simulations. We also define $R_0 = \max\{R^0_{\min}, 0.1R_{\max}\}$ if $\|\nabla_\theta SLL^{R_0}(\theta_0)\| \neq 0$ and $\epsilon_\delta^{R_0}(\theta_0) \neq 0$, $R_0 = R_{\max}$ otherwise. The choice of R^+ in Step 3 of Algorithm 1 is described below.

Algorithm 2: Candidate sample size selection

Define some constants ν_1 and χ_1 such that $\nu_1, \chi_1 \in (0,1)$. Use (8) to estimate the size needed to obtain a precision equal to the model increase, that is

$$R^s = \max\left\{R^k_{\min}, \left\lceil \frac{\alpha_\delta^2}{(I\Delta m_k^{R_k})^2} \sum_{i=1}^{I} \frac{\left(\sigma_{ij_i}^{R_k}(\theta)\right)^2}{\left(P_{ij_i}^{R_k}(\theta)\right)^2} \right\rceil\right\}.$$

Compute the ratio between the model improvement and the estimated accuracy,

$$\tau_1^k = \frac{\Delta m_k^{R_k}}{\epsilon_\delta^{R_k}(\theta_k)},$$

and the ratio between the current sample size and the suggested sample size for the next iteration:

$$\tau_2^k = \frac{R_k}{\min\{R_{\max}, R^s\}}.$$

Then define

$$R' = \begin{cases} \min\left\{\lceil \chi_1 R_{\max}\rceil, \lceil R^s\rceil\right\} & \text{if } \tau_1^k \geq 1, \\ \min\left\{\lceil \chi_1 R_{\max}\rceil, \lceil \tau_1^k R^s\rceil\right\} & \text{if } \tau_1^k < 1 \text{ and } \tau_1^k \geq \tau_2^k, \\ \lceil \chi_1 R_{\max}\rceil & \text{if } \nu_1 \leq \tau_1^k < 1 \text{ and } \tau_1^k < \tau_2^k, \\ R_{\max} & \text{if } \tau_1^k < \nu_1 \text{ and } \tau_1^k < \tau_2^k. \end{cases}$$

Set $R^+ = \max\{R', R^k_{\min}\}$.

If $\tau_1^k \geq 1$, the model increase is greater or equal to the estimated accuracy, and we then reduce the sample size to the minimum between R^s and $\lceil \chi_1 R_{\max}\rceil$ (in our tests, we set $\chi_1 = 0.5$). The idea to use $\lceil \chi_1 R_{\max}\rceil$ comes from the practical observation that enforcing such a decrease in the proposed sample sizes provides better numerical performance.

If $\tau_1^k < 1$ the improvement is smaller than the precision. However, since the sample has been generated before the optimisation process, a sufficient improvement during several consecutive iterations may lead to a significant improvement compared to the log-likelihood accuracy, while keeping the computational costs lower than if $R_{\max}$ draws were used. We then consider two cases.

- If $\tau_1^k \geq \tau_2^k$, the ratio between the current sample size and the potential next

one is lower than the ratio between the model increase and the estimated error. If the sample size increases, the error decreases for a similar $\Delta m_j^{R_j}$ ($j \geq k$), and therefore τ_1^k increases. We capitalise on τ_1^k by computing a sample size lower than R^s, such that an increase of order $\epsilon_\delta^{R_k}(\theta_k)$ would be reached in approximately $\lceil \tau_1^k \rceil$ iterations if τ_1^j is similar to τ_1^k for j close to k. We therefore propose to use the minimum between $\lceil \chi_1 R_{\max} \rceil$ and $\lceil \tau_1^k R^s \rceil$ as the new sample size.

- If $\tau_1^k < \tau_2^k$, it may nevertheless be cheaper to continue to work with a smaller sample size, defined again as $\lceil \chi_1 R_{\max} \rceil$. This is why we choose to use this smaller sample size as long as τ_1^k is superior to some threshold $\nu_1 > 0$ (set to 0.2 in our tests). Below this threshold, we consider that the increase is too small compared to the log-likelihood accuracy, and we possibly increase the sample size.

If R^+ is not equal to R_k, the computation of

$$SLL^{R^+}(\theta_k + s_k) - SLL^{R_k}(\theta_k)$$

in Algorithm 1 is affected by the change in simulation bias and variance. This can lead to a small or negative ratio ρ_k, even when the model $m_k^{R_k}$ gives a good prediction for the sample size R^k. In particular, $SLL^{R^+}(\theta)$ can be smaller than $SLL^{R_k}(\theta_k)$ for all θ in a neighbourhood of θ_k. It is therefore important to avoid such cases, which motivates the possible redefinition of ρ_k as described in the algorithm below.

Algorithm 3: Sample size revision when $\rho_k < \eta_1$ and $R_k \neq R^+$.

If $R^+ < R_k$ set

$$R^b = \left\lceil \frac{1}{2\Delta m_k^{R_k} I} \sum_{i=1}^{I} \frac{\left(\sigma_{ij_i}^{R_k}(\theta)\right)^2}{\left(P_{ij_i}^{R_k}(\theta)\right)^2} \right\rceil .$$

If $R^+ < R^b < R_k$, set $R^+ = R^b$ and recompute ρ_k from (10).

If the (possibly recomputed) $\rho_k < \eta_1$, compare R^+ and R_k. If $R^+ > R_k$, compute $SLL^{R^+}(\theta_k)$, $\Delta m_k^{R^+}$ and $\epsilon_\delta^{R^+}(\theta_k)$, else if $R^+ < R_k$ compute $SLL^{R_k}(\theta_k + s_k)$. Set R^- to $\max\{R_k, R^+\}$, and redefine

$$\rho_k = \frac{SLL^{R^-}(\theta_k + s_k) - SLL^{R^-}(\theta_k)}{\Delta m_k^{R^-}}.$$

When the number of draws increases ($R^+ > R_k$), the bias decreases in absolute value, but the objective function can still decrease due to the refinement of the sample average approximations. Therefore, we force the algorithm to evaluate

$SLL^{R^+}(\theta_k)$ in order to avoid the accuracy difference effect. The case $R^+ < R_k$ is more subtle since the absolute value of the bias then increases, so that the objective function is usually lower for a fixed θ. If ρ_k is small, we try to circumvent the bias effect by testing another sample size R^b, that corresponds to the sample size giving a bias equal to the predicted increase, using the estimation (9).

While we expect to benefit from smaller sample sizes when we are far from the solution, we ought to be sure that we use a sample size equal to $R_{\max}$ during the final iterations, in order to benefit from the desired accuracy. For this purpose, we increase the minimum sample size when the variable sample size strategy does not provide sufficient numerical gains. This is done as follows. We first define two $R_{\max}$-dimensional vectors v and l, and, at iteration $k = 0$, we set $v(R_0) = SLL_{R_0}(z_0)$, $l(R_0) = 0$, and for $i = 1, \ldots, R_{\max}$, $i \neq R_0$[2], $v(i) = -\infty$, $l(i) = -1$. At the beginning of iteration k, $v(i) = SLL^i\left(\theta_{h(i)}\right)$ where $h(i)$ corresponds to the index of the last iteration for which $R_{h(i)} = i$, and $R_{h(i)-1} \neq R_{h(i)}$ if $h(i) > 0$, or $-\infty$ (a trivial lower bound on the objective function) if the size i has not been used yet. $l(i)$ contains the number of successful iterations until iteration $h(i)$ (included), or -1 if the size i has not been used. Recall also that t contains the total number of successful iterations encountered until the current iteration k (included).

Algorithm 4: Minimum sample size update when $R_k \neq R_{k+1}$.
Let $\gamma_3 \in (0, 1]$ be a constant. If

$$SLL^{R_{k+1}}(\theta_{k+1}) - v(R_{k+1}) \geq \gamma_3 \nu_1 (t - l(R_{k+1})) \epsilon_\delta^{R_{k+1}}(\theta_{k+1}), \qquad (12)$$

set $R_{\min}^{k+1} = R_{\min}^k$. Otherwise increase the minimum sample size: if $R_k < R_{k+1}$, set

$$R_{\min}^{k+1} = \min\left\{ \left\lceil \frac{R_k + R_{k+1}}{2} \right\rceil, R_{\max} \right\},$$

else

$$R_{\min}^{k+1} \in \{R_{k+1} + 1, \ldots, R_{\max}\}.$$

Set $l(R_{k+1}) = t$ and $v(R_{k+1}) = SLL^{R_{k+1}}(\theta_{k+1})$.

Note that we apply a different strategy if the sample size decreases or increases. In the first case, bias difference and loss of precision can explain a decrease or a small increase of the SAA objective, but it is numerically cheaper to continue to use sample sizes as small as possible; in our implementation we then set $R_{\min}^{k+1} = R_{k+1} + 1$. In the second case, we try to avoid poor increases of the SAA objective for large sample sizes since the associated numerical cost is then important, and we then use a more conservative approach.

The constant γ_3 is set to 0.5 in our tests. Note that $R_{\min}^{k+1} > R^k$ if (12) is not

[2]We are in fact only interested in $i \geq R_{\min}^0$, since the sample size used at iteration k ($k \geq 0$) has to be greater or equal to $R_{\min}^0$, but allow i to start from 1 for notational convenience.

satisfied. Moreover, we have that if $R_k \neq R_{k+1}$, $t - l(R_{k+1}) \geq 1$. This is clearly true if $l(R_{k+1}) = -1$, so without loss of generality, assume that $l(R_{k+1}) \geq 0$. At the beginning of iteration k, we have $l(R_i) \leq t$, $i = 1, \ldots, R_{\max}$. If $\rho_k \geq \eta_1$, t is incremented by 1 in Step 6 of Algorithm 1, and $l(R_{k+1}) < t$ in Algorithm 4. If $\rho_k < \eta_1$, from Algorithm 3, $R_k < R_{k+1}$ since reductions of sample sizes can only occur at successful iterations. This also implies that $l(R_{k+1}) < l(R_k) \leq t$.

Finally, we note that, if $R_k \neq R_{\max}$, we cannot exclude the pathological case in which θ_k is a first-order critical point for SLL^{R_k}. If $\epsilon_\delta^{R_k}(\theta_k) \neq 0$, the algorithm does not stop, but since the model is quadratic, no increase is achieved if $-H_k$ is positive definite. The algorithm would then break down. In order to avoid this situation, we therefore force an increase of R_{k+1} in Step 6 when this situation occurs. In practice, we have chosen to set $R_{k+1} = R_{\max}$ if the relative gradient is less than some predefined tolerance (we used 10^{-6}). The relative gradient is defined as

$$g_{rel}(\theta) \stackrel{def}{=} \max_c \left\{ \frac{|[\nabla_\theta SLL^R(\theta)]_c| \ \max\{|\theta_c|, 1.0\}}{\max\{|SLL^R(\theta)|, 1.0\}} \right\},$$

where v_c is the c-th component of the vector v (see Dennis and Schnabel [16], Chapter 7). We should note that this feature was never triggered in our experiments. Indeed, the gradient norm usually changes slowly in the vicinity of such a critical point, and a small gradient typically leads to a small model increase, which itself then causes the sample size to increase and $R_{\max}$ was always reached before our safeguard was activated.

Similarly, we check in Step 0 that we do not produce a first-order critical point associated with a sample size less than $R_{\max}$, excepted if the simulation error is equal to zero (and so the bias).

3.4 Stopping tests

The presence of statistical error requires that the classical stopping tests for unconstrained optimisation, which involve the gradient norm and sometimes the difference between successive iterates or function values, must be considered with caution. In particular they usually lead to final iterations that produce insignificant objective increases compared to the approximation's accuracy. Numerical simulations revealed however that the algorithm can reach an adequate accuracy for a subset of the parameters but then produce small improvements at the maximum sample size during a few iterations, after what good improvements are again obtained, and the desired accuracy achieved on the remaining parameters. This is in particular true for parameters that are hard to estimate, such as small standard deviations since they produce small variations of the simulated likelihood function. It is therefore important not to stop the algorithm prematurely.

The practical stopping criterion used in Step 1 of Algorithm 1 is a modification of the classical test based on the relative gradient: the algorithm is terminated as

soon as

$$g_{rel}(\theta) \leq \max\{tol, \mu_1 \epsilon_\delta^R\},$$

where $0 < \mu_1 < 1$ and $\epsilon_\delta^R(\theta)$ is the estimated log-likelihood accuracy, and either the maximum sample size $R_{\max}$ is used or, in order to consider the multinomial logit case, the estimated log-likelihood accuracy is sufficiently small. The value $\mu_1 = 0.1$ has revealed to be a good compromise, for a signification level δ set to 0.9 in the accuracy estimator. We also stop the algorithm if a (user preset) maximum number of iterations has been reached without convergence, or if the norm of computed step falls under a user-defined signification threshold (we used 10^{-6}).

4 Convergence to solutions of the SAA problem

We now consider the formal convergence properties of our new algorithm (without the stopping tests of Section 3.4) for the solution of the SAA problem, that is that we solve the (deterministic) problem (6), with $R_{\max}$ random draws per individual. We show in particular that the convergence properties can be derived from results known for general trust-region methods.

4.1 Convergence of the sample size

We start by investigating properties of our variable sample size technique and prove the crucial property that R_k converges to $R_{\max}$ as $k \to \infty$, under some regularity assumptions that we now make explicit.

A.4 The utilities $V_{ij}(\omega, \cdot)$ $(i = 1, \ldots, I,\ j = 1, \ldots, J)$ are twice continuously differentiable for almost every ω.

Assumption A.4 implies that for each R, the approximation SLL^R is almost surely twice continuously differentiable.

A.5 The Hessian of each SAA objective is uniformly bounded, that is there exists a positive constant κ_1 such that for all θ and $R = R_{\min}^0, \ldots, R_{\max}$,

$$\left\|\nabla_{\theta\theta} SLL^R(\theta)\right\| \leq \kappa_1.$$

A.6 The Hessian of the model remains bounded within the trust-region, that is there exists a positive constant κ_2 such that for all $\theta \in \mathcal{B}_k$,

$$\left\|\nabla_{\theta\theta} m_k^{R_k}(\theta)\right\| \leq \kappa_2.$$

Note that the approximating objective is bounded above by zero, since we obtain from (5) and the logit formula (2) that:

$$SLL^R(\theta) = \frac{1}{I}\sum_{i=1}^{I} \ln SP_{ij_i}^R(\theta) \leq \frac{1}{I}\sum_{i=1}^{I} \ln 1 = 0, \tag{13}$$

for $R = R_{\min}^0, \ldots, R_{\max}$.

Theorem 1

Suppose that Assumptions A.0, A.4–A.6 hold and that we have

$$\exists\, \kappa > 0 \text{ such that } \epsilon_\delta^{R_k}(\theta_k) \geq \kappa, \tag{14}$$

for all k sufficiently large. Then, either the algorithm converges in a finite number number of iterations with a final number of random draws equal to $R_{\max}$, or the number of iterations is infinite and there exists some j such that for all iterations $i, i \geq j$, R_i is equal to $R_{\max}$.

Proof. Consider the finite case first. From the stopping criteria in Step 1 of Algorithm 1, we cannot stop with a sample size less than $R_{\max}$ as long as (14) is fulfilled, so the result is immediate.

Consider now the infinite case. We first prove that the sample size cannot stay fixed at a value $R_1 < R_{\max}$, after what we show that the maximum sample size must be reached and that the sample size is equal to $R_{\max}$ for k large enough.

Assume, for the purpose of contradiction, that

$$\exists\, k_1 \text{ such that } \forall k \geq k_1, R_k = R_{k_1} < R_{\max}. \tag{15}$$

For a fixed sample size, Algorithm 1 corresponds to the basic trust-region algorithm (Conn, Gould and Toint [14], Chapter 6). Assume first that there are only finitely many successful iterations. Let s be the index of the last successful iteration. Then $\theta_k = \theta_{s+1}$ for all $k > s$. From Assumptions A.0, A.4–A.6 and our model choice, we can apply Theorem 6.4.4 in Conn, Gould and Toint [14] to deduce that $\|\nabla_\theta SLL^{R_{s+1}}(\theta_{s+1})\| = 0$. From Steps 0 and 6 of Algorithm 1, either (14) is violated, or $R_{s+1} = R_{\max}$, and the algorithm stops, violating our assumption that the number of iterations is infinite.

We may therefore assume, without loss of generality, that there is an infinite number of successful iterations. However, from Algorithms 2 and 3, and (14), a necessary condition for $R^+ < R_{\max}$ at iteration k is

$$\Delta m_k^{R_k} \geq \nu_1 \kappa,$$

when $\tau_1^k \geq \nu_1$, or

$$\Delta m_k^{R_k} \geq \kappa \frac{R_{\min}^k}{R_{\max}},$$

when $\tau_1^k \geq \tau_2^k$. Assume that the iteration is successful. Then $R^+ = R_{k+1} = R_{k_1}$ for k large enough and we have from (15) that

$$SLL^{R_{k_1}}(\theta_{k+1}) - SLL^{R_{k_1}}(\theta_k) \geq \eta_1 \Delta m_k^{R_{k_1}} \geq \eta_1 \min\left\{\nu_1 \kappa, \kappa \frac{R_{\min}^0}{R_{\max}}\right\}.$$

Since there is an infinite number of successful iterations, $SLL^{R_{k_1}}(\theta_k)$ converges to infinity, as $k \to \infty$, but this contradicts the fact that $SLL^{R_{k_1}}$ is bounded above, as shown in (13). We have therefore that

$$\text{if } R_{k_1} < R_{\max}, \text{ then there exists } k_2 > k_1 \text{ such that } R_{k_2} \neq R_{k_1}. \tag{16}$$

Assume now by contradiction that

$$\forall k, \text{ there exists } j \geq k \text{ such that } R_j < R_{\max}. \tag{17}$$

From Algorithm 4, $R^k_{\min}$ monotonically increases and is bounded above by $R_{\max}$. Therefore there exists some $R^{\#} = \lim_{k\to\infty} R^k_{\min}$, with $R^{\#} \leq R_{\max}$. Since $R^k_{\min}$ is finite for all k, $R^{\#}$ is reached at some iteration $k_{\#}$ and $R^k_{\min} = R^{k_{\#}} < R_{\max}$ for all $k \geq k_{\#}$. From (16) and (17), there exists an infinite subsequence of iterations such that $R_{k+1} \neq R_k$. Let $m \geq k_{\#}$ be the index of such an iteration. From Algorithm 4 and (14) we have that

$$SLL^{R_{m+1}}(R_{m+1}) - v(R_{m+1}) \geq \gamma_3\nu_1(t - l(R_{m+1}))\epsilon^{R_{m+1}}_{\delta}(\theta_m) \geq \gamma_3\nu_1\kappa, \tag{18}$$

otherwise we would have $R^{m+1}_{\min} > R^m_{\min}$. However each SAA objective is bounded above from (13), and there is a finite number of possible sample sizes. Therefore, (18) can only be satisfied for a finite number of iterations, so we obtain a contradiction if (17) is satisfied. Consequently $R_k = R_{\max}$ for all k large enough. □

4.2 First-order optimality

Having proved that the sample size must be equal to $R_{\max}$ for k large enough, we now prove first-order convergence of the proposed algorithm by applying convergence results known for trust-region methods. For this purpose, we impose a sufficient increase of the model at each iteration:

A.7 For all k

$$m^{R_k}_k(\theta_k + s_k) - m^{R_k}_k(\theta_k) \geq \kappa_3 \|\nabla_\theta SLL^{R_k}(\theta_k)\| \min\left\{\frac{\|\nabla_\theta SLL^{R_k}(\theta_k)\|}{\zeta_k}, \psi_k\right\},$$

for some constant $\kappa_3 \in (0, 1)$ and $\zeta_k = 1 + \max_{x\in\mathcal{B}_k} \left\|\nabla_{xx} m^{R_k}_k(x_k)\right\|$.

Assumption A.6, a classic in trust-region method analysis, is fulfilled by our choice of the Steihaug-Toint step since it ensures a model increase at least as much as that obtained at the approximate Cauchy point (Conn, Gould and Toint [14], page 131). We then obtain our first convergence result.

Theorem 2 (First-order convergence)
Suppose that Assumptions A.0, A.4–A.7 hold and that

$$\exists\, \kappa > 0 \text{ such that } \epsilon^{R_k}_{\delta}(\theta_k) \geq \kappa,$$

for all k sufficiently large. Then, either the algorithm converges in a finite number of iterations to a first-order critical point of $SLL^{R_{\max}}$, or the number of iterations is infinite and

$$\lim_{k\to\infty} \|\nabla_\theta SLL^{R_k}(\theta_k)\| = 0,$$

with $R_k = R_{\max}$ for all k sufficiently large.

Proof. From Theorem 1, we know that $R_k = R_{\max}$ for all k sufficiently large. The first-order convergence then results from the Theorem 6.4.4 in Conn, Gould and Toint [14] in the finite case, and Theorem 6.4.6 in the infinite case. □

From (8), we see that $\epsilon_\delta^R(\theta)$ is equal to 0 if and only if $\sigma_{ij_i}(\theta) = 0$, for $i = 1, \ldots, I$. We then have a multinomial logit model, instead of a mixed logit, and the simulated likelihood function value is then independent of the sampling. Consequently, the fact that (14) is not satisfied is merely an indication that the multinomial logit solution solution is a limit point of the iterates, and that the mixed logit formulation is probably inappropriate. The algorithm could then be terminated with a sample size less than its maximum, as described in Step 1 of Algorithm 1. However, the maximum sample size is always reached in our numerical experimentations, even when testing multinomial logit models. This is explained by the fact that we approximate the $\sigma_{ij_i}(\theta)$ by $\sigma_{ij_i}^R(\theta)$ and that small standard deviations are not easy to recover since their influence in the model is weak. Therefore, the error term remains positive and $R_{\max}$ is always reached in the final iterations.

4.3 Second-order optimality

We conclude our convergence analysis by briefly indicating that, under some additional assumptions, any limit point of the sequence of iterates may be proved to be second-order critical. We first slightly strengthen the conditions governing the trust-region update, imposing that the radius actually increases at very successful iterations:

A.8 If $\rho_k \geq \eta_2$ and $\psi_k \leq 10^{20}$, then $\psi_{k+1} \in [\gamma_4\psi_k, \gamma_5\psi_k]$ for some $\gamma_5 \geq \gamma_4 > 1$.

We also require that the Hessian of the model and that of the simulated log-likelihood asymptotically coincide whenever a first-order limit point is approached.

A.9 We assume that

$$\lim_{k\to\infty} \|\nabla_{\theta\theta} SLL^{R_k}(\theta)_k - \nabla_{\theta\theta} m_k^{R_k}(\theta_k)\| = 0 \text{ whenever } \lim_{k\to\infty} \|g_k^{R_k}\| = 0.$$

Recall that from Theorem 2, R_k is constant for k large enough. The BFGS approximation then satisfies **A.9** under reasonable conditions (Ge and Powell [19]).

Second-order convergence is then ensured if the step uses positive curvature of the model when present. This is expressed formally by the following theorem, where $\lambda_{\max}[A]$ denotes the largest eigenvalue of the matrix A.

Theorem 3 (Second-order convergence)
Suppose that Assumptions A.0, A.4–A.9 hold and that

$$\exists\, \kappa > 0 \text{ such that } \epsilon_\delta^{R_k}(\theta_k) \geq \kappa,$$

for all k sufficiently large. Let k_1 be such that $R_k = R_{\max}$ for all $k \geq k_1$. Assume furthermore that for all $k \geq k_1$, if $\tau_k = \lambda_{\max}\left[\nabla_{\theta\theta} m_k^{R_k}(\theta_k)\right] > 0$, then

$$m_k^{R_k}(x_k + s_k) - m_k^{R_k}(x_k) \geq \pi_1 \tau_k \min\{\tau_k^2, \psi_k^2\},$$

for some constant $\pi_1 \in \left(0, \frac{1}{2}\right)$. Then any limit point of the sequence of iterates is second-order critical for $SLL^{R_{\max}}$.

Proof. Directly follows from Theorem 6.6.8 of Conn, Gould and Toint [14]. □

Note also that the existence of a limit point is ensured if, as is nearly always the case, all iterates lie within a closed, bounded domain $\mathcal{C} \subseteq \mathbb{R}^m$.

5 Numerical assessment

In order to validate our methodology we have developed our own software, called AMLET (for Another Mixed Logit Estimation Tool), available in open-source at `http://www.grt.be/amlet`. AMLET is written in C and is designed to run in a Linux environment, but it can also be used on Windows 2000 and XP under Cygwin. The package allows the user to solve mixed or multinomial logit models from existing data, or to set up simulated data corresponding to a user-defined model structure. AMLET computes parameters estimators, classical tests for goodness of fit (as described in Ben-Akiva and Lerman [6] and Ortúzar and Willumsen [29]), and some specific information such as estimation of the simulation bias and log-likelihood accuracy. Reported results have been obtained on a Pentium 4 2.8Ghz with 1 GB RAM under Linux.

We now discuss the application of our method on a real dataset[3] obtained from the six-week travel diary Mobidrive (Axhausen, Zimmerman, Schönfelder, Rindsfúser and Haupt [1]) collected in the spring and fall 1999 in Karlsruhe and Halle (Germany). In this dataset, we restricted our attention to the observations from

[3]The application described here has been chosen amongst several for illustration purposes. Other numerical experiments both for real and simulated data sets can be found, along with more detailed practical analysis, in Bastin, Cirillo, and Toint [3, 4], Bastin Cirillo and Hess [2], and Pellicanò [30]. All these applications lead to conclusions similar to those discussed here.

Karlsruhe because level of service variables (i.e. time and cost for various modes) were available for this location only. The sample then includes approximately 66 households and 145 individuals. After data cleaning, 5799 records (tours) were retained for calibrating two mixed logit models whose aim is to explain individual modal choice across five alternatives (car driver, car passenger, public transport, walk and bike). The framework applied considers the daily activity chain, in that the individual pattern is divided into tours, which are themselves defined as the sequence of trips starting and ending at home or at work, both being considered as fixed locations. Details and motivation for the model structure can be found in Cirillo and Toint [12]. Note that, as several tours are performed by the same individuals, the data therefore contains significant correlations. For further details on mixed logit on the Mobidrive dataset, see Cirillo and Axhausen [10].

The first model contains 14 parameters, of which four are alternative specific constants (car driver is the base), two describe the household location (urban/suburban location), four the individual characteristics (female and working part time, being married with children, annual car mileage), two the LOS (time and cost) and two represent pattern variables (number of stops and time budget). We specify a mixed logit model with fixed coefficients except for time, cost and time budget, which are expected to vary considerably across observations, and are assumed to be normally distributed[4]. We estimate the model with sample sizes varying form 500 to 4000 random draws per individual, and average the results over 10 simulations. These are summarised in Table 1, where the values in brackets correspond to the t-statistics associated to the estimated parameters. The average value of time is 9.55 DM (about 4.9 euros), which is comparable to that used in other European studies (see TRACE [38]).

The crucially beneficial effect of the variable sample size strategy is illustrated in Figure 1, giving the evolution of the sample size R_k with the iteration index k. The left graph corresponds to a maximum sample size of 1000 while the right graph has been obtained with a maximum of 4000 random draws. Furthermore, Figure 2 shows that the sample size increases towards its maximum value only when the objective function's value is near to its maximum. The graphs correspond again to 1000 (left) and 4000 (right) random draws.

The second model uses the same data set, but is more complex. Its specification has 33 degrees of freedom, resulting from 19 fixed and 7 randomly distributed coefficients. The dimensionality has increased because time parameter is now specific to tour types. (A model with ten normally distributed coefficients was also estimated, but some t-statistics were too small to justify the use of random variables for all coefficients.) The results, obtained (for illustration purposes) by averaging the estimation results over 10 runs, are reported in Table 2. We refer to Cirillo and Axhausen [11] for a complete description of this model.

In order to evaluate the numerical potential of the proposed method, we also estimated both models with the basic trust-region algorithm and with the BFGS

[4]Trials with lognormal distribution for time have given poor results.

Variable		500 MC	1000 MC	2000 MC	3000 MC	4000 MC
Car Passenger (CP)	μ	-1.453 (17.18)	-1.453 (17.15)	-1.453 (17.13)	-1.4544 (17.13)	-1.452 (17.13)
Public Transport (PT)	μ	-0.932 (6.77)	-0.932 (6.77)	-0.932 (6.77)	-0.934 (6.76)	-0.933 (6.77)
Walk (W)	μ	0.109 (0.73)	0.107 (0.73)	0.109 (0.74)	0.109 (0.74)	0.108 (0.73)
Bike (B)	μ	-0.635 (4.63)	-0.636 (4.64)	-0.635 (4.63)	-0.635 (4.62)	-0.636 (4.63)
Urban HH locat. (PT)	μ	0.562 (5.048)	0.561 (5.04)	0.562 (5.04)	0.563 (5.05)	0.561 (5.04)
Suburban HH locat. (W, B)	μ	0.346 (4.05)	0.345 (4.05)	0.346 (4.06)	0.345 (4.05)	0.345 (4.05)
Full-time worker (PT)	μ	0.269 (2.74)	0.270 (2.74)	0.269 (2.73)	0.269 (2.73)	0.269 (2.73)
Female and part-time (CP)	μ	0.915 (8.65)	0.914 (8.64)	0.915 (8.64)	0.916 (8.64)	0.914 (8.63)
Married with children (CD)	μ	0.972 (11.59)	0.971 (11.58)	0.972 (11.58)	0.972 (11.57)	0.972 (11.58)
Annual mileage by car (CD)	μ	0.0520 (15.86)	0.0519 (15.85)	0.0519 (15.87)	0.0519 (15.87)	0.0519 (15.87)
Number of stops (CD)	μ	0.136 (3.01)	0.136 (3.02)	0.136 (3.02)	0.136 (3.02)	0.136 (3.01)
Time	μ	-0.0269 (9.61)	-0.0270 (9.59)	-0.0269 (9.55)	-0.0270 (9.57)	-0.0270 (9.55)
Time	σ	0.0206 (4.99)	0.0206 (5.01)	0.0207 (4.97)	0.0208 (5.00)	0.0208 (4.99)
Cost	μ	-0.169 (12.63)	-0.169 (12.47)	-0.169 (12.41)	-0.169 (12.48)	-0.169 (12.47)
Cost	σ	0.0452 (2.93)	0.0469 (3.05)	0.0468 (3.01)	0.0461 (2.95)	0.0465 (3.02)
Time budget (CD, CP)	μ	-0.125 (8.09)	-0.125 (8.09)	-0.125 (8.08)	-0.125 (8.08)	-0.125 (8.08)
Time budget (CD, CP)	σ	0.115 (5.89)	0.114 (5.77)	0.114 (5.80)	0.115 (5.84)	0.114 (5.74)
Log-likelihood		-1.164617	-1.164667	-1.164724	-1.164690	-1.164738
Bias		-0.000186	-0.000092	-0.000046	-0.000031	-0.000023
Accuracy		0.000417	0.000293	0.000208	0.000170	0.000147

Table 1: Mobidrive: simple model

Variable		500 MC	1000 MC	2000 MC	3000 MC	4000 MC
Car Passenger (CP)	μ	-1.169 (11.77)	-1.167 (11.75)	-1.169 (11.74)	-1.169 (11.74)	-1.168 (11.74)
Public Transport (PT)	μ	-0.761 (3.77)	-0.758 (3.76)	-0.758 (3.75)	-0.757 (3.75)	-0.757 (3.75)
Walk (W)	μ	1.378 (7.04)	1.379 (7.04)	1.382 (7.04)	1.382 (7.04)	1.382 (7.04)
Bike (B)	μ	0.907 (4.74)	0.910 (4.76)	0.912 (4.75)	0.912 (4.75)	0.912 (4.76)
Suburban HH locat. (CD, CP)	μ	0.430 (4.70)	0.430 (4.70)	0.430 (4.69)	0.430 (4.69)	0.429 (4.68)
Urban HH locat. (PT)	μ	0.251 (2.23)	0.249 (2.22)	0.252 (2.24)	0.251 (2.23)	0.251 (2.23)
Age 18-25 (PT)	μ	1.339 (8.50)	1.339 (8.51)	1.340 (8.50)	1.340 (8.50)	1.340 (8.50)
Age 26-35 (CD, CP)	μ	0.337 (2.02)	0.336 (2.02)	0.341 (2.04)	0.340 (2.04)	0.338 (2.03)
Age 51-65 (PT)	μ	0.489 (4.29)	0.489 (4.30)	0.488 (4.28)	0.489 (4.28)	0.489 (4.29)
Full time worker (PT)	μ	-0.182 (1.71)	-0.181 (1.70)	-0.182 (1.70)	-0.182 (1.70)	-0.181 (1.70)
Female and part-time (CP)	μ	0.751 (7.02)	0.748 (6.99)	0.751 (7.00)	0.750 (6.99)	0.749 (6.99)
Married with children (CD)	μ	0.788 (8.85)	0.785 (8.81)	0.788 (8.80)	0.787 (8.79)	0.786 (8.78)
Main car user (CD)	μ	1.101 (11.75)	1.099 (11.73)	1.101 (11.71)	1.100 (11.70)	1.100 (11.71)
Annual mileage by car (CD)	μ	0.0266 (7.24)	0.0265 (7.23)	0.0266 (7.25)	0.0266 (7.24)	0.0266 (7.24)
Number of season tickets (CD)	μ	-0.208 (2.17)	-0.208 (2.18)	-0.207 (2.16)	-0.207 (2.16)	-0.207 (2.16)
Number of stops (CD)	μ	0.180 (3.84)	0.179 (3.83)	0.180 (3.84)	0.180 (3.84)	0.180 (3.84)
Time before principal activity	μ	-0.0431 (9.07)	-0.0431 (9.03)	-0.0433 (9.02)	-0.0433 (9.01)	-0.043 (9.01)
Time before principal activity	σ	0.0314 (4.76)	0.0315 (4.77)	0.0317 (4.77)	0.0318 (4.78)	0.0321 (4.81)
Time principal activity	μ	-0.0358 (6.69)	-0.0357 (6.67)	-0.0358 (6.67)	-0.0359 (6.67)	-0.0359 (6.679)
Time principal activity	σ	0.0513 (5.38)	0.0515 (5.39)	0.0516 (5.38)	0.0517 (5.39)	0.0518 (5.40)
Time post principal activity	μ	-0.00776 (1.49)	-0.00771 (1.48)	-0.00776 (1.49)	-0.00780 (1.50)	-0.00779 (1.50)
Time before work activity	μ	-0.00726 (1.51)	-0.00723 (1.51)	-0.00727 (1.52)	-0.00726 (1.51)	-0.00726 (1.52)
Time work activity	μ	-0.0283 (8.84)	-0.0281 (8.81)	-0.0283 (8.81)	-0.0282 (8.82)	-0.0282 (8.82)
Time work activity	σ	0.00892 (3.27)	0.00887 (3.27)	0.00886 (3.20)	0.00871 (3.14)	0.00875 (3.17)
Time post work activity	μ	-0.0425 (5.70)	-0.0424 (5.68)	-0.0425 (5.67)	-0.0426 (5.68)	-0.0426 (5.68)
Time post work activity	σ	0.0464 (3.22)	0.0461 (3.20)	0.0462 (3.19)	0.0463 (3.20)	0.0463 (3.20)
Cost (CD, PT)	μ	-0.127 (8.73)	-0.126 (8.67)	-0.127 (8.68)	-0.127 (8.65)	-0.127 (8.66)
Cost (CD, PT)	σ	0.0450 (2.23)	0.0449 (2.23)	0.0453 (2.25)	0.0455 (2.21)	0.0452 (2.20)
Time budget (CD, CP)	μ	-0.0398(2.10)	-0.0390 (2.05)	-0.0395 (2.07)	-0.0393 (2.06)	-0.0392 (2.06)
Time budget (CD, CP)	σ	0.0562 (1.96)	0.0540 (1.74)	0.0571 (1.91)	0.0565 (1.85)	0.0555 (1.80)
Sum of travel time (B)	μ	-0.0438 (5.88)	-0.0439 (5.86)	-0.0441 (5.86)	-0.0440 (5.86)	-0.0440 (5.86)
Sum of travel time (B)	σ	0.0448 (4.99)	0.0447 (4.96)	0.0450 (4.97)	0.0449 (4.97)	0.0448 (4.96)
Tour duration (PT)	μ	0.00405 (16.87)	0.00404 (16.85)	0.00405 (16.84)	0.00405 (16.83)	0.00405 (16.83)
Log-likelihood		-1.104536	-1.104654	-1.104555	-1.104560	-1.104513
Bias		-0.000221	-0.000109	-0.000056	-0.000037	-0.000028
Accuracy		0.000455	0.000319	0.000228	0.000186	0.000161

Table 2: Mobidrive: complex model

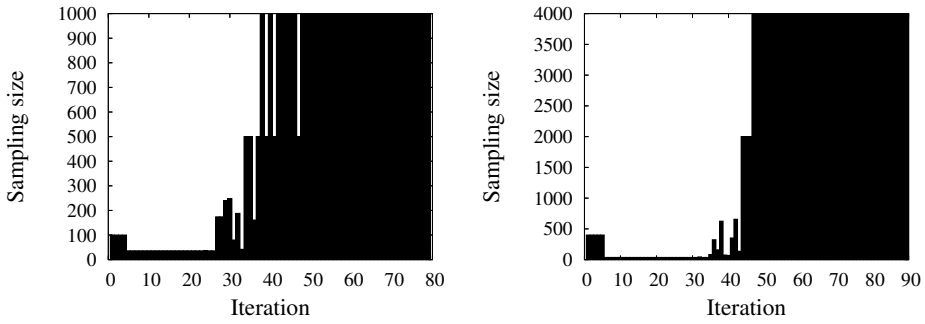

Figure 1: Variation of sample sizes with iterations

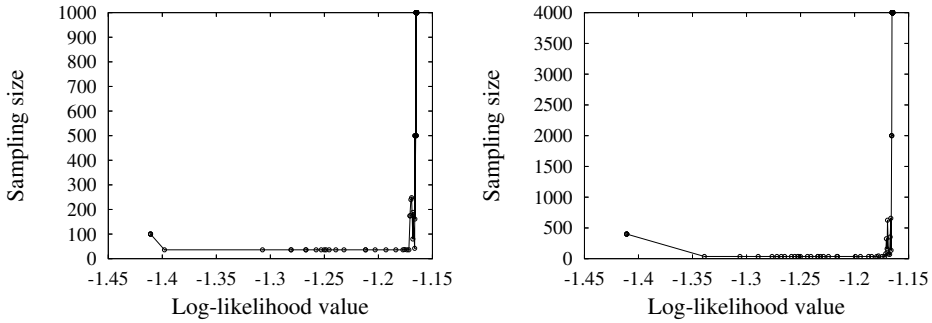

Figure 2: Variation of sample sizes with log-likelihood value

line-search (the More-Thuente step selection was implemented). The draws in both cases are those generated for the adaptive strategy. Resulting computational times are given in Table 3. For the simple model, the trust-region approach and the line-search techniques are quite similar, BFGS line-search being slightly better. The BTRDA approach however delivers a significant speed-up (approximately 35%) compared to both standard techniques. The advantage of the line-search over the trust-region disappears when we consider the more complex model, since the mean optimisation time of the trust-region method is then significantly smaller. Moreover, while the trust-region algorithm always converges, the BFGS line-search frequently fails. We indicate in Table 3 the number of success for the BFGS with the complex model over the 10 runs in brackets, next to the optimisation time. On the other hand, the BTRDA algorithm exhibits more important gains, since the computational times are less than half of those obtained with BTR. This suggests that an adaptive strategy is especially efficient when the number of random variables increases, while it is well known that other techniques, like those based on quasi-Monte Carlo sequences, are often less efficient in such cases. Quasi-Monte Carlo methods can however produce the same accuracy with less random draws, at least in low and medium integration dimensions, but this accuracy is difficult to quantify in practice, while this is easy for Monte Carlo approaches. As our algorithm

$R_{\max}$	Simple model			Complex model		
	BTRDA	BTR	BFGS	BTRDA	BTR	BFGS
500	440s	684s	646s	1093s	2217s	2489s (7)
1000	829s	1403s	1312s	2076s	4310s	4783s (5)
2000	1636s	2753s	2592s	4151s	8500s	10009s (6)
3000	2427s	4089s	3849s	5712s	13146s	15340s (7)
4000	3234s	5581s	5144s	7576s	16741s	21630s (9)

Table 3: Optimisation times

exploits this information, its application to quasi-Monte Carlo techniques is not as direct as with pure random draws. Moreover usual problems in high-dimensional integration with quasi-Monte Carlo methods, such as correlations, do not occur in pure Monte Carlo procedures. Consequently, the latter are often more robust, both theoretically and numerically. Our procedure can therefore be seen as a compromise between speed and the exploitation of theoretical information, while more research is needed to apply the same philosophy to (possibly randomised) quasi-Monte Carlo sequences.

Due to the complexity of the objective function in mixed logit models, the choice of the optimisation procedure is therefore of crucial importance. First of all, the speed of convergence can be dramatically increased if the available information is exploited. In our case, the estimation of the standard deviation allows us to speed up the initial iterations by using smaller samples, and often to successfully terminate earlier. Secondly, important savings can be achieved by taking the problem properties into account. In particular, the use of an optimisation algorithm designed for non-concave problems pays off for mixed logit models. Further computational gains may also be obtained in the detailed organisation of the algorithm, for instance by evaluating the function and its gradient analytically at the same time, instead of successively.

6 Conclusion

In this paper, we have developed a new algorithm for unconstrained stochastic programming using statistical inference to accelerate computations. Convergence of the algorithm is ensured to points satisfying first- and second-order optimality conditions. The method has been applied to the mixed logit estimation problem and we have developed our own package, AMLET, to do so. Numerical experimentation shows that a strategy for using a variable sample size in the estimation of the choice probabilities gives significant gains in optimisation time compared to the classical fixed sample size approach.

This paper opens several further research questions. First of all, comparisons with complex quasi-Monte Carlo methods remain desirable. Secondly, further im-

provements to the variable sample size strategy are likely, possibly yielding additional computational gains. Finally, more research efforts should be devoted to better quantify the statistical accuracy of the estimated parameters compared to the true maximum likelihood estimators.

Acknowledgements

The authors would like to express their gratitude to Kay Axhausen for granting access to the Mobidrive dataset, and to three anonymous referees for their suggestions. Our thanks go also to Marcel Remon for his helpful comments on statistical theory, and to the Belgian National Fund for Scientific Research for the grant that made this research possible for the first author and for its support of the third author during a sabbatical mission.

References

[1] Kay W. Axhausen, Andrea Zimmerman, Stefan Schönfelder, Guido Rindsfüser, and Thomas Haupt. Observing the rythms of daily life: A six week travel diary. *Transportation*, 29(2):95–124, 2002.

[2] Fabian Bastin, Cinzia Cirillo, and Stéphane Hess. Evaluation of optimisation methods for estimating mixed logit models. *Transportation Research Record*, Forthcoming.

[3] Fabian Bastin, Cinzia Cirillo, and Philippe L. Toint. Numerical experiments with AMLET, a new Monte-Carlo algorithm for estimating mixed logit models. Paper presented at the 10th International Conference on Travel Behaviour Research, 2003.

[4] Fabian Bastin, Cinzia Cirillo, and Philippe L. Toint. Application of an adaptive monte carlo algorithm to mixed logit estimation. *Transportation Research Part B*, Submitted.

[5] Fabian Bastin, Cinzia Cirillo, and Philippe L. Toint. Convergence theory for nonconvex stochastic programming with an application to mixed logit. *Mathematical Programming, Series B*, Submitted.

[6] Moshe Ben-Akiva and Steven R. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. The MIT Press, 1985.

[7] Ernst K. Berndt, Bronwyn H. Hall, Robert E. Hall, and Jerry A. Hausman. Estimation and inference in nonlinear structural models. *Annals of Economic and Social Measurement*, 3/4:653–665, 1974.

[8] Chandra R. Bhat. Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model. *Transportation Research*, 35B(7):677–693, 2001.

[9] Chandra R. Bhat and Frank S. Koppelman. Activity-based modeling of travel demand. In Randolph W. Hall, editor, *Handbook of Transportation Science*, pages 35–61, Norwell, USA, 1999. Kluwer Academic Publisher.

[10] Cinzia Cirillo and Kay W. Axhausen. Mode choice of complex tour. In *Proceedings of the European Transport Conference (CD-ROM)*, Cambridge, UK, 2002.

[11] Cinzia Cirillo and Kay W. Axhausen. Values of time? evidence on the distribution of values of travel time savings from a six-week diary. *Transportation Research A*, Submitted.

[12] Cinzia Cirillo and Philippe L. Toint. An activity based approach to the Belgian national travel survey. Technical Report 2001/07, Transportation Research Group, Department of Mathematics, University of Namur, 2001.

[13] Andrew R. Conn, Nicholas I. M. Gould, Dominique Orban, and Philippe L. Toint. Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints. *SIAM Journal on Optimization*, 3(1):164–221, 1993.

[14] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, USA, 2000.

[15] István Deák. Multidimensional integration and stochastic programming. In Y. Ermoliev and R. J.-B. Wets, editors, *Numerical Techniques for Stochastic Optimization*, pages 187–200. Springer Verlag, 1988.

[16] John E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1983.

[17] George S. Fishman. *Monte Carlo: Concepts, Algorithms and Applications*. Springer Verlag, New York, USA, 1996.

[18] Rodrigo A. Garrido. Estimation performance of low discrepancy sequences in stated preferences. Paper presented at the 10th International Conference on Travel Behaviour Research, 2003.

[19] R.-P. Ge and Michael J.D. Powell. The convergence of variable metric matrices in the unconstrained optimization. *Mathematical programming*, 27:123–143, 1983.

[20] Christian Gouriéroux and Alain Monfort. Simulation based econometrics in models with heterogeneity. *Annales d'Economie et de Statistiques*, 20(1):69–107, 1991.

[21] Christian Gouriéroux and Alain Monfort. *Simulation-based Econometric Methods*. Oxford University Press, Oxford, United Kingdom, 1996.

[22] Vassilis A. Hajivassiliou and Daniel L. McFadden. The method of simulated scores for the estimation of LDV models. *Econometrica*, 66(4):863–896, 1998.

[23] John H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.

[24] David A. Hensher and William H. Greene. The mixed logit model: The state of practice. *Transportation*, 30(2):133–176, 2003.

[25] Stéphane Hess, John Polak, and Andrew Daly. On the performance of shuffled Halton sequences in the estimation of discrete choice models. In *Proceedings of European Transport Conference (CD-ROM)*, Strasbourg, France, 2003. PTRC.

[26] Stéphane Hess, Kenneth Train, and John Polak. On the use of a modified latin hypercube sampling (mlhs) approach in the estimation of a mixed logit model for vehicle choice. *Transportation Research B*, Forthcoming.

[27] William J. Morokoff and Russel R. Caflish. Quasi-Monte Carlo integration. *Journal of Computational Physics*, 122(2):218–230, 1995.

[28] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, USA, 1999.

[29] Juan de Dios Ortúzar and Luis G. Willumsen. *Modelling Transport*. John Wiley & Sons, 3rd edition, 2001.

[30] Roberta Pellicanò. *Discrete choice models and Statistical Data Mining for the analysis of the demand for transport. Statistical methods supporting the ingegnering of transport.* PhD thesis, University Federico II of Naples, Naples, Italy, 2005.

[31] David Revelt and Kenneth Train. Mixed logit with repeated choices. *Review of Economics and Statistics*, 80, 1998.

[32] Reuven Y. Rubinstein and Alexander Shapiro. *Discrete Event Systems*. John Wiley & Sons, Chichester, England, 1993.

[33] Zsolt Sándor and Kenneth Train. Quasi-random simulation of discrete choice models. *Transportation Research B*, 38(4):313–327, 2004.

[34] Alexander Shapiro. Stochastic programming by Monte Carlo simulation methods. *SPEPS*, 2000.

[35] Alexander Shapiro. Monte Carlo sampling methods. In A. Shapiro and A. Ruszczyński, editors, *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 353–425. Elsevier, 2003.

[36] Alexander Shapiro and Tito Homem de Mello. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81:301–325, 1998.

[37] Aruna Sivakumar, Chandra R. Bhat, and Giray Ökten. Simulation estimation of mixed discrete choice models using randomized quasi-monte carlo sequences: A comparison of alternative sequences, scrambling method, and uniform-to-normal variate transformation techniques. Paper presented at Transportation Research Board Annual Meeting, 2005.

[38] TRACE, Costs of private road travel and their effects on demand, including short and long term elasticities. Final report to the European Commission, HCG, Den Haag, The Netherlands, 1999.

[39] Kenneth Train. Halton sequences for mixed logit. Working paper No. E00-278, Department of Economics, University of California, Berkeley, 1999.

[40] Joan L. Walker. *Extended Discrete Choice Models: Integrated Framework, Flexible Error Structures, and Latent Variables*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2001.