Evaluation of optimisation methods for estimating mixed logit models

Fabian Bastin
F.N.R.S. Research Fellow
Department of Mathematics
University of Namur
8, Rempart de la Vierge
5000 Namur, Belgium
Tel: +32(0)81 724923
Fax: +32(0)81 724914
fabian.bastin@fundp.ac.be

Cinzia Cirillo
Transportation Research Group
University of Namur
8, Rempart de la Vierge
5000 Namur, Belgium
Tel: +32(0)81 724923
Fax: +32(0)81 724914
cinzia.cirillo@fundp.ac.be

Stephane Hess
Centre for Transport Studies
Imperial College London
Exhibition Road
London SW7 2AZ
Tel: +44(0)20 7594 6105
Fax: +44(0)20 7594 6102
stephane.hess@imperial.ac.uk

Word Count (words 6543 + 4 tables = 7543 words)

ABSTRACT

In this paper, we evaluate the performance of different simulation-based estimation techniques for Mixed Logit modelling. We compare a quasi-Monte Carlo method (modified latin hypercube sampling) to a Monte Carlo algorithm with dynamic accuracy. We also compare the classical BFGS line-search approach to trust-region methods, which have proved to be an extremely powerful approach in nonlinear programming. Numerical tests are performed on two real datasets; stated-preference parking-type data collected in the UK, and revealed preference data for mode-choice collected as part of a German travel diary survey. Several criteria are used in the evaluation of the approximation quality of the log-likelihood function and the accuracy of the results, along with the associated estimation runtime. Results suggest that the trust-region approach outperforms the BFGS approach, and that Monte Carlo methods remain competitive with quasi-Monte Carlo methods in high dimensional problems, especially when an adaptive optimisation algorithm is employed.

1. INTRODUCTION

With the increased use of advanced discrete choice models in the field of transportation, researchers face mounting issues relating to the formulation, estimation and interpretation of the models. One type of discrete choice model that is becoming increasingly popular is the Mixed Multinomial Logit (MMNL) model (18,26), which allows for the representation of random variations in tastes across decision-makers, correlation across alternatives in the unobserved part of utility, and heteroscedasticity in the error-terms. It also allows for an explicit treatment of the repeated choice nature that is inherent to many of the more complex datasets used in transportation.

The fact that the MMNL choice-probabilities take the form of multi-dimensional integrals without a closed-form solution leads to a requirement for numerical techniques, typically simulation, in the estimation and application of the model. Despite major gains in computational power, this dependency on heavy computation is still limiting the applicability of the MMNL model.

Several approaches have been proposed with the aim of reducing the computational overhead. These can be divided into two main categories; changes to the actual estimation process, and changes to the techniques used in the simulation processes. In this paper, we compare the gains in estimation performance that can be obtained when using methods from these two main streams of approaches. The first approach uses recent work by Hess et al. (15) in the field of quasi-Monte Carlo integration, while the second approach is based on an adaptive Monte Carlo algorithm that varies the number of random draws used at a given iteration according to the simulation error and bias of the simulated log-likelihood function, in combination with trust-region approaches (3). We compare the performance of the two methods across two real datasets, using various criteria.

The remainder of this paper is organised as follow. Section 2 reviews the theory behind Mixed Logit models, while estimation techniques are described in Section 3. Details on the datasets used are presented in Section 4. Section 5 presents a discussion of the results, while Section 6 outlines the conclusions and makes some suggestions for future research.

2. THE MIXED MULTINOMIAL LOGIT MODEL

In a random utility model, a decision-maker *n* is faced by a choice among *I* alternatives, characterised by a vector U_n of *I* random utility functions:

$$U_n = \begin{pmatrix} U_{n1} \\ .. \\ U_{nI} \end{pmatrix} = \begin{pmatrix} V_{n1} \\ .. \\ V_{nI} \end{pmatrix} + \begin{pmatrix} \varepsilon_{n1} \\ .. \\ \varepsilon_{nI} \end{pmatrix}, \qquad [1]$$

divided into an observed part, V_n, and an unobserved part, ε_n. The observed part of utility is a function of the tastes of the decision-maker, β, and a vector x_{ni} containing attributes of alternative *i* and socio-demographic characteristics of decision-maker *n* (or interactions of the two), such that $V_{ni}=g(\beta,x_{ni})$. Typically, a linear-in-variables specification is used, such that $V_{ni}= \beta^T x_{ni}$. Under the assumption of utility maximisation, the alternative with the highest utility is chosen. In the Multinomial Logit (MNL) model, the individual error-terms ε_{ni} are assumed to be independently and identically distributed extreme-value, leading to the well-known logit formula for the choice probability of alternative *i*:

$$P_{ni}(V_n) = \frac{e^{V_{ni}}}{\sum_{j=1}^{I} e^{V_{nj}}}. \qquad [2]$$

In the MMNL model, the vector *V* itself contains random elements, and the choice probabilities are rewritten as:

$$P_{ni} = \int_{V_n} P_{ni}(V_n) dV_n, \qquad [3]$$

where the elements in the vector V_n can be rewritten as $V_{ni}=g(\beta,x_{ni})+\xi_{ni}$. This formulation can be exploited in two mathematically identical, yet conceptually different ways. In the error-components formulation (c.f. 28), the additional vector ξ_n 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 regularly used random-coefficients formulation (see for example 23), 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. 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.

In the random-coefficients formulation, β is assumed to be distributed according to $f(\beta|\theta)$, where θ is a vector of parameters of the random distribution, giving for example the mean and standard deviation of the individual elements in β across decision-makers. The choice probabilities in the MMNL model are now given by:

$$P_{ni} = \int_{\beta} P_{ni}(\beta, x_{ni}) f(\beta|\theta) d\beta \;=\; \int_{\beta} \frac{e^{g(\beta, x_{ni})}}{\sum_{j=1}^{I} e^{g(\beta, x_{nj})}} f(\beta|\theta) d\beta . \qquad [4]$$

For practical (numerical) reasons, the log-likelihood is generally used, which, with N decision-makers facing I alternatives, is given by:

$$LL(\theta) = \sum_{n=1}^{N} \sum_{i=1}^{I} d_{ni} \ln(P_{ni}), \qquad [5]$$

where d_{ni} is a dummy variable that is set to *1* if decision-maker n is observed to choose alternative i and *0* otherwise. The standard approach used in estimation is to maximise the log-likelihood given in [5] with regards to θ, such that, at the maximum likelihood estimator θ^*, a necessary (but not sufficient) condition is:

$$\nabla_{\theta} LL(\theta^*) = 0. \qquad [6]$$

The obtained solution θ^* is then said to be first-order critical. If moreover, the Hessian of the log-likelihood is negative semi-definite, the solution is said to fulfil second-order necessary conditions. Finally, a negative definite Hessian is a second-order sufficient condition, ensuring that the solution is a strict local optimum. If the log-likelihood is concave, as is the case for the linear-in-variables MNL model, first-order conditions are sufficient to ensure that the solution is a global solution. Unfortunately, if the utilities are nonlinear or if a mixed-logit formulation is used, the log-likelihood is non-concave, and special care as well as appropriate algorithms are required in the search of a solution.

Except in the case of a trivial distribution function for β, the integrals representing the choice-probabilities [4] do not have a closed-form solution, and require the use of approximation techniques. Typically, the choice probabilities P_{ni} ($i=1,...,I$) need to be replaced by the simulated choice probabilities $\hat{P}_{ni}$ ($i=1,...,I$), given by:

$$\hat{P}_{ni} = \frac{\sum_{r=1}^{R} P_{ni}(\beta_r, x_{ni})}{R}, \qquad [7]$$

where the different values of β_r are independent draws from $f(\beta|\theta)$, for a given value of θ. The simulated log-likelihood (SLL) is then given by:

$$SLL^R(\theta) = \sum_{n=1}^{N} \sum_{i=1}^{I} d_{ni} \ln(\hat{P}_{ni}), \qquad [8]$$

with maximum simulated likelihood estimator (MSLE) denoted by $\hat{\theta}$.

It has been shown (16) that if R rises faster than $\sqrt{N}$, maximum simulate likelihood (MSL) estimation is asymptotically equivalent to maximum likelihood (ML) estimation (in a first-order critical sense). Almost-sure convergence of the estimators is discussed by Bastin et al. (3), for fixed N and increasing N. The use of a fixed number of draws R does inevitably induce simulation bias and variance (26); this is however unavoidable due the absence of a closed-form solution for the MMNL-integrals, but can be minimised by using a sufficiently high number of draws.

As an extension, we look at the treatment of repeated choice observations. Typically, the tastes of a given decision-maker are assumed to stay constant across choice-situations for that respondent, such that tastes vary across individuals, but not across observations for the same individual. The probabilities of the individual choices are then replaced by the probabilities of the observed sequence of choices for each decision-maker. With i_{nt} giving the alternative chosen by decision-maker *n* in choice-situation *t* (*t=1,...,T*$_n$), the probability of the choices made by decision-maker *n*, conditional on β_n, is given by:

$$L_n^{T_n}(\beta) = \prod_{t=1}^{T_n} P_{ni_{nt}}\left(\beta_n, x_{ni_{nt}}\right), \qquad [9]$$

with a corresponding unconditional probability:

$$L_n = \int_\beta L_{nt}^{T_n}(\beta) f(\beta) d\beta \,. \qquad [10]$$

This leads to a new version of the log-likelihood function, given by:

$$LL(\theta) = \sum_{n=1}^{N} \ln\left(L_n\right), \qquad [11]$$

with a corresponding form for the simulated log-likelihood function.

3. ESTIMATION TECHNIQUES

While offering great gains in flexibility, the MMNL model does, as mentioned above, have the disadvantage of yielding choice probabilities that do not in general possess a closed-form solution. Even though recent improvements in computer performance have made the MMNL model a more widely-applicable tool for discrete choice analysis, the estimation and application of the model can still be computationally very expensive. For this reason, considerable efforts have gone into improving the efficiency of the actual estimation processes, with the aim of further reducing this computational overhead. In this paper, we discuss two such approaches: quasi-random numbers and trust-region methods. Our empirical analysis gives an illustration of the performance gains that can be obtained relative to standard Monte Carlo integration, and compares the performance of the two methods in practice.

3.1. Quasi-Monte Carlo integration

In standard Monte Carlo (MC) integration, the draws from $f(\beta|\theta)$ used in equation [7] are based on transformations of pseudo-random numbers, generated uniformly in the interval]0,1[. By their nature, the inherent *random* distribution of these draws across the area of integration leads to uneven coverage (or uniformity), especially when a low number of draws is used. This in turn leads to poor approximation in simulation, which can lead to biased parameter estimates. As the use of a very high number of draws is however often impractical and computationally very expensive, the use of quasi-Monte Carlo (QMC) numbers can be a desirable alternative. By offering a more even spread of points across the area of integration, these deterministically-designed number sequences usually lead to more stable simulation performance, hence enabling the use of a lower number of draws, with corresponding reductions in the computational cost in the actual simulation process.

A large number of different types of quasi-random number sequences have been proposed, especially in the field of numerical and computational statistics. In the field of transportation, only one type of sequence, the Halton sequence, has found widespread application. While Halton sequences perform well in low-dimension (c.f. 5,25), their cyclical nature creates problems with high correlation and poor coverage in high-dimensional applications. Two main transformation methods have been proposed to circumvent these problems: scrambled Halton draws (c.f. 6), which permute the digits on the original elements of a multidimensional sequence, and shuffled Halton sequences (c.f. 12), which randomly permute the order of the original elements. Some effort has also gone into using other quasi-Monte Carlo methods, aiming to minimise some discrepancy measure (usually the star-discrepancy measure); as an example, Garrido (11) proposes the use of Sobol sequences, while Sándor and Train (24) illustrate the performance of (t,m,s)-nets in MMNL estimation. It should remain clear that “the success of QMC methods in practice is due to a clever choice of point sets exploiting the features of the functions that are likely to be encountered, rather than to an unexplainable way of breaking the ‘curse of dimensionality’ ” (17). Therefore, the actual applicability of a particular QMC approach should always be carefully assessed, in particular with respect to the problem to solve, and more research into the use of QMC in ML estimation is still needed.

In this paper we make use of the modified latin hypercube sampling (MLHS) approach proposed by Hess et al. (15). Formally, a one-dimensional sequence of length *N* is obtained by setting

$$\varphi(j) = \frac{j-1}{N} + x, \quad j = 1, \ldots, N\,, \qquad [12]$$

where x is a random number satisfying $0 < x < 1/N$. Multi-dimensional sequences are simply constructed by combination of randomly shuffled one-dimensional sequences (hence disrupting the correlation which would lead to poor coverage), and by using a different shift x in each dimension.

3.2. Trust-region methods with variable numbers of draws

The maximisation of the log-likelihood function in equation [5] can be seen as a generalisation of a classical class of stochastic programming problems (2). A large number of different optimisation algorithms can be used in the maximisation of the SLL [8]. Researchers generally use Newton-Raphson, BHHH, and BFGS line-search methods. The BHHH approach can be much faster than other methods, but can occasionally fail to produce a solution; BFGS on the other hand is usually seen as good compromise between efficiency and robustness.

In this paper, we use basic trust-region (BTR) methods, which have proved to be one of the most powerful approaches in nonlinear programming, and have received a lot of attention and developments during the last decade (see 10 for an exhaustive review of these methods, with more than 300 references since 1990). The main idea of a trust-region algorithm involves the calculation, at iteration k (with current estimate θ_k), of a trial point $\theta_k + s_k$ by approximately maximising a model m_k of the objective function inside a trust region defined as

$$B_k = \{\theta \text{ such that } \|\theta - \theta_k\| \le \Delta_k\}, \qquad [13]$$

where Δ_k is called the trust-region radius. We will use a quadratic model:

$$m_k(s) = SLL^R(\theta_k) + s^T \nabla_\theta SLL^R(\theta_k) + \frac{1}{2} s^T H_k s, \qquad [14]$$

where H_k is a symmetric approximation of the Hessian $\nabla^2_{\theta\theta} SLL^R(\theta_k)$, and where we use the BFGS approximation in the reported numerical experiments. The predicted and actual increases in the value of the objective function are then compared by computing the ratio

$$\rho_k = \frac{SLL^R(\theta_k + s_k) - SLL^R(\theta_k)}{m(\theta_k + s_k) - m(\theta_k)}, \qquad [15]$$

If this ratio is greater than a certain threshold, set to 0.01 in our tests, the trial point becomes the new iterate, and the trust-region radius is (possibly) enlarged. More precisely, if ρ_k is greater than 0.75, we set the trust-region to be the maximum between Δ_k and $2s_k$, otherwise we set $\Delta_k = 0.5\Delta_k$. If the ratio is below the bound, the trial point is rejected and the trust region is shrunk by a factor of 2, in order to improve the correspondence of the model with the true objective function. We have followed Conn et al. (10) in our choice of the parameters. Note however that during the last iterations, the algorithm becomes insensitive to the trust-region radius, as the iterates are close to the solution.

A major advantage of the trust-region approach is that it can easily be adapted to include a variable sample size strategy, as proposed by Bastin et al. (3). The resulting algorithm will be referred to as basic trust-region with dynamic accuracy (BTRDA), since the simulation error is a function of the number of draws. Such an approach is based on the idea of generating a full set of draws prior to optimisation, but to only use part of it during certain stages of the optimisation process. This is motivated by the understanding that the first steps in an optimisation process are rough steps in the general direction of the optimum, requiring a relatively lower level of precision in simulation. The full set of draws is used during the last few iterations; this not only guarantees maximum simulation-precision at this stage of the optimisation, but also means that the problem used at this stage of the optimisation is identical to that used in methods not based on variable sample size strategies. Since the population size N is constant, we can rely on consistency results and expect good estimators, close to the true maximum likelihood estimators, when R is sufficiently large. We will however mainly use the bias and accuracy estimation to define what "sufficiently large" means, since, for instance, just requiring R strictly greater than $\sqrt{N}$ can lead to insufficiently large sample sizes, as we will see later. A practical rule would for instance be to choose R such that the bias and accuracy estimation, averaged over the individuals, are less than some usual tolerance.

We therefore need to know simulation bias and accuracy, which can be computed as follows:

$$\varepsilon_{\delta}^{R}(\theta)=\alpha_{\delta}\sqrt{\sum_{n=1}^{N}\frac{\sigma_{n}^{2}(\theta)}{R\left(L_{n}(\theta)\right)^{2}}}\,, \qquad [16]$$

and

$$B^{R}(\theta)=-\frac{\left(\varepsilon_{\delta}^{R}(\theta)\right)^{2}}{2\alpha_{\delta}^{2}}\,, \qquad [17]$$

respectively, where $\sigma_n^2(\theta)$ is the variance of L_n and α_δ is the quantile of the Gaussian distribution at significance-level δ. In practice, we set δ to 0.9, and replace $L_n(\theta)$ and $\sigma_n^2(\theta)$ by their corresponding statistical estimators (2). Too small a value for δ would imply unrealistic error estimation, while too large a value would produce overly conservative values, implying poor performance of the variable sample size strategy.

We now discuss the main ideas of the implemented strategy, with details and proofs of convergence given by Bastin et al. (4). At a given iteration *k*, using R_k draws per individual, we compute a candidate sample size R^+ in $\left[R_{\min}^{k}, R_{\max}\right]$, where $R_{\max}$ is the final sample size, and $R_{\min}^{k}$ is the minimum number of draws. If the ratio τ_k between the increase in model fit and the estimated accuracy is greater than 1, we set R^+ to the minimum of $\lceil 0.5R_{\max} \rceil$ and the size needed to obtain an accuracy equal to the model increase, denoted by R^s. If the improvement is smaller than the precision, but greater than the ratio between the sample size and R^s, we set R^+ to the minimum of $\lceil 0.5R_{\max} \rceil$ and $\tau_k R^s$, on the grounds that an increase of the order of the estimated accuracy would likely be reached in approximately $\lceil \tau_k \rceil$ iterations. Otherwise, we set R^+ to $\lceil 0.5R_{\max} \rceil$ as long as τ_k is greater than some threshold (set to 0.2 in our applications), and to $R_{\max}$ when this condition is not met. We then compute the simulated log-likelihood function with R^+ draws per individual at the trial iterate. If the ratio ρ_k is less than 0.01, we recompute the simulated log-likelihood at θ_k with R^+ if $R_k < R^+$, in order to take account of variance difference, or, when $R_k > R^+$, we compute a new candidate sample size R^b, corresponding to the size producing a bias equal to the predicted increase, and set $R^+ = R^b$ if $R^+ < R^b < R_k$. The bias is indeed increasing in absolute value when the number of draws is decreased. We then again compute the ratio ρ_k, with updated sample sizes. As a safeguard, we finally increase the minimum sample size if the algorithm exhibits poor performance due to the variations of accuracy and bias when varying the sample size. The algorithm stops when the gradient norm is less than a pre-defined tolerance, or a fraction of the accuracy (0.2 in our tests), where we expect that no more significant increase in the objective function will be achieved.

4. EMPIRICAL APPLICATION

We now describe the framework used to evaluate the performance of the different algorithmic options, in terms of nonlinear programming methods as well as drawings techniques (pseudo-random or quasi-Monte Carlo draws). We first describe the two datasets used in our experimentation.

4.1. Data

To illustrate the differences in performance between the various methods, MMNL models were estimated on two datasets; a stated preference (SP) dataset for parking-type choice collected in the United Kingdom, and a revealed preference (RP) travel-diary dataset collected in Germany. The two datasets are now briefly described.

The first dataset used in the analysis contains the results of an SP survey looking into the choice of parking type in the West Midlands region of the United Kingdom (22). Three separate surveys were conducted in the central business districts of Birmingham and Coventry, as well as in the suburban area of Sutton Coldfield, in 1989. Respondents were selected at street-level with the help of certain screening criteria and quotas (21). They were presented with up to nine hypothetical choice situations involving their revealed parking-type, along with two possible alternative parking-options. The five types of parking used in the survey were free-on-street, charged-on-street, charged-off-street, multi-storey car parking, and illegal parking, where the design ensured that illegal parking was included as an alternative in each choice situation. Four attributes were used in the description of the data; access time (to the parking area), search time (for a parking space), egress time (walking time to final destination) and parking cost (set to zero for the free-on-street alternative). Finally, for the illegal parking option, the cost attribute was replaced by the expected fine, given by the product of the probability of receiving a ticket with the level of fine currently in use. The final sample contains 1335 responses, from 298 respondents, grouped into two purpose groups (work and leisure). The dataset was recently used by Hess and Polak (14) in an MMNL analysis, revealing significant random taste heterogeneity for the majority of variables used in the survey.

The Mobidrive dataset was collected in 1999 in two cities of Germany (Karlsruhe and Halle-Salle), from 160 households and 360 individuals, where each individual was observed during six continuous weeks. For details on data collection techniques and on the descriptive results, see (1). In the present analysis, only the dataset for Karlsruhe is used; appropriate data on the levels of service (LOS) for the used and non-used alternatives were added separately. The days recorded were structured according to the framework proposed by Bhat and Singh (7) and extended by Cirillo and Toint (9). All trips are grouped into tours, and the population is divided into workers (commuters and education) and non-workers. For each worker, the daily chain is divided into *morning, commute* and *evening patterns.* For non-workers, we define the *main activity* as the longest out-of-home activity recorded, where the daily activity chains are represented in relation to this pivotal activity and organised into *morning, principal* and *evening patterns*. With this definition, a total of 5,795 tours were identified, performed by 136 individuals belonging to 66 households, with an average daily number of 1.72 tours per individual.

4.2. Comparison framework

The open-source estimating software AMLET (*Another Mixed Logit Estimation Tool*) was used for the estimation of all models discussed in this paper (c.f. *http://www.grt.be/amlet*). The program allows for classical BFGS line search, but also comprises code for Basic Trust-Region and Basic Trust-Region with dynamic accuracy estimation. In order to as much as possible limit the timing differences between the three algorithms due to the implementation, all algorithms were rewritten directly in the core of AMLET, by taking account of the standard recommendations of the existing literature. For the BTR, we have followed the guidelines proposed by Conn et al. (10), while for the BFGS line search, we have observed the suggestions given by Nocedal and Wright (20), and have taken inspiration from the package L-BFGS-B (29). In particular, we have implemented the efficient More-Thuente line search (19), which is currently considered as the best line-search technique. It is worth noting at this point that the trust-region approach is simpler to implement efficiently than is the case for the line-search method.

Experiments were conducted on a Pentium IV 3.20Ghz, with 2GB of memory, under Linux. The reported times are the CPU runtimes used during the optimisation process, as given by AMLET at the end of estimation. While the three algorithmic options have been tested with Monte Carlo draws, we use the MLHS approach only with BTR, which we observed to deliver significant speed gains by comparison to the BFGS line search, similar to previous experiments on synthetic data (3). As error estimation is not directly available with the MLHS approach, we were however restricted to using the fixed sample size strategy. Unless otherwise stated, the starting points for estimation were obtained by setting all model parameters to 0.1.

5. RESULTS AND DISCUSSION

In this section, we report the estimation results and the evaluation of both simulation and optimisation techniques described in the previous part of the paper. In particular, the performance indicators used are:

- Bias, RMSE and Standard Deviation, as share of the standard error of the associated estimate;
- Estimated accuracy and bias of the log-likelihood (available only for Monte Carlo methods);
- Computational time.

The computation of the bias and the root-mean-squared-error (RMSE) between the estimated value and these “true” values of the parameters aims at testing the ability of the draws to recover the “true” parameters. In order to account for the shape of the log-likelihood function, the RMSE values were expressed as a proportion of the standard error of the true parameters (c.f. 15). For the parking-type choice model, the true parameters were calculated on the basis of 10 runs using 100,000 pseudo-random draws per individual, and the values were averaged over those runs (where the standard errors are given as the square-root of the average of the squared errors). This was found to be sufficient to yield stable parameter values, where the estimated simulation bias and accuracy, averaged over the individuals, were of the order of $–5.10^{-5}$ and 1.10^{-3} respectively. In the Mobidrive dataset, we estimated the true parameters by running the model ten times with 10,000 pseudo-random draws per individual, where a higher number of draws was not possible due to memory limitations and the large number of observations. The tables give the statistics averaged over parameters; parameter-specific results are available from the first author on request.

Ten independent runs were performed in order to produce the performance indicators, and the same random draw sets (for each predefined sample size) were used when comparing BFGS, BTR and BTRDA algorithms. The stopping criterion for BFGS and BTR was set to 10^{-5}, while we stopped the BTRDA algorithm when the gradient was less than a fraction of the estimated accuracy, as explained in Section 3. Reported estimated bias and standard deviations are also expressed in terms of percentages of the optimal log-likelihood values, as given in Tables 1 and 3, by indicating the percentages in brackets, next to the canonical values.

5.1 Parking choice model

In the parking-type choice model (Table 1), we estimated 9 parameters, of which one is fixed and the remaining 8 are randomly distributed. A Normal distribution was used for all four identified alternative specific constants (ASC); the highly significant standard deviations for these coefficients show the extent of taste variation, at least partly reflecting the differences in terms of respondents' attitudes towards the different types of parking. In terms of sensitivity to time, there is significant taste variation only for search and egress times, leading to a fixed coefficient for access time, and *lognormally* distributed coefficients for search and egress times. A lognormal distribution was also used for the cost coefficient, while, due to problems with an overestimated standard deviation when using the lognormal, a normal distribution had to be used for the expected fine coefficient. Although this does imply a probability of ~0.7% of a positive coefficient, this risk is necessary in this case, as very poor results were obtained with all of the alternative distributions. A more exhaustive description of this model and the reasons for the actual specification can be found in Hess and Polak (14).

Bias, RMSE and standard deviations (Table 2) are surprisingly high for the BFGS line search and 1000 pseudo-random draws, when we compare them to those obtained with the trust-region approaches. During our tests, the BFGS method failed to converge with two sets of draws, and we had to choose a starting point close to the solution in order to guarantee convergence. Moreover, two other runs converged to an inferior solution, with a final gradient norm of less than 10^{-5}. This can be explained by the fact that the candidate points produced during the early iterations can become quite large, leading to convergence difficulties in later runs. We argue that this problem comes from the non-concavity and the flatness of the objective function. A good preconditioning of the problem could help to improve the behaviour of the method, but such techniques have not yet been investigated in our tests.

The pseudo-random draws perform surprisingly well when compared to the MLHS draws, since bias, RMSE and standard deviations are comparable when using 1000 draws. It is worth recalling that the dimensionality of the problem is quite high (8 random parameters), and in such cases, it has already been observed that standard Monte Carlo methods are again competitive with quasi-Monte Carlo techniques (15). In terms of computational time, the BTRDA algorithm clearly outperforms any other methods, and the choice of the trust-region methodology also leads to important savings. Even if the BTR algorithm is often reported to be faster than the BFGS line-search techniques in nonlinear programming, the time reduction factor, of approximately two, is still impressive. This suggests that the trust-region approach is more adapted to deal with the shape of the log-likelihood function. This is coherent with theoretical results (27), while the practical differences between trust-region and line-search techniques are usually less or even not appreciable.

We finally note that the proposed model is quite difficult to estimate, as reflected in Table 2, which illustrates large variations between estimates over the 10 runs. Therefore, more runs should probably be used to adequately compare the different methods. Estimation of the accuracy and bias of the log-likelihood optimal values, currently only available with standard Monte Carlo draws, also suggests that we can face estimation difficulties since significant noise is present in the objective, even with 10,000 draws. In particular, their mean values per individual are quite high compared to the usual values used in classical stopping criteria. Note that this information is already available with one simulation, while the other presented criteria can only be obtained by repeating the simulation process over a sufficient number of runs.

5.2 Mobidrive mode choice model

In the Mobidrive mode choice model (Table 3), we estimated 21 parameters, of which four are specified as being normally distributed, namely those associated with time, cost, sum of travel time and time budget. This specification leads to positive value of travel time coefficients for about 10% of the population. It is not clear whether these results are caused by the use of the normal distribution, or are actually revealed by the data. The latter point-of-view is taken by Cirillo and Axhausen (8), based on the observation that other specifications, such as ones using a lognormal distribution for both time and cost and accounting for correlation across those parameters, did not improve the fit of the model and produced very doubtful value of travel time savings. Hess et al. (13) however stress that such negative values of time are not consistent with economic theory, and should be seen as a result or poor distributional assumptions, or data impurities. The discussion of this topic is beyond the scope of the present paper, and the normal distribution was used primarily because it performed well numerically.

Tables 3 and 4 show that the model is very robust, and simulation noise is low, even with a small number of draws. This is due in part to the dataset's inherent properties and to the large number of individuals, but also to the model specification and to the low number of random parameters. This illustrates that the number of draws needed to efficiently estimate a model crucially depends of its characteristics, while the knowledge of simulation bias and accuracy is again a valuable tool in the search of an optimal number of draws. In particular, mean estimated bias and accuracy are small, close to the values usually used as stopping criteria. For the same number of draws per individual, we observe much lower simulation bias and accuracy estimates than in the parking choice models. This is in part consistent with what we expected. The mean error per individual decreases with the population size, but the mean bias only depends on the number of draws. Moreover, we observe that with 1000 draws per individual (a common choice), the ratio between the simulation accuracy and the simulation bias is larger in the Mobidrive dataset. We argue that this mainly comes from the fact that we compare a cross-sectional experiment with a panel data set. The integrand of the unconditional choice probability in the panel case is indeed far more complicated than in the cross-sectional case, which could imply a need for a higher number of draws in order to observe a similar precision and bias. This in turn suggests that care is required when dealing with classical consistency theorems; in particular the required number of draws depends more on the model formulation than on the number of individuals/observations. Preliminary experiments on synthetic data have led to similar observations.

In this application, MLHS draws are competitive compared to standard Monte Carlo approaches, even with the BTRDA technique. The use of 500 and 1000 MLHS draws give smaller bias, RMSE and standard deviation values than those obtained with 2500 pseudo-random draws. This also leads to similar performance as with 5000 pseudo-random draws in terms of bias, and gives better RMSE and standard deviations. We can again partially explain the differences in performances across the two datasets by the low random dimensionality, which usually leads to an advantage for QMC techniques. It should also be noted that the results obtained with 500, 1000 or 2000 draws are quite similar for RMSE and standard deviation (while the bias is smaller for 2000 draws than with 500 of 1000 draws). More research is needed to determine the exact reasons for this observation.

Finally, we observe that trust-region techniques greatly outperform BFGS line search, as in the previous example, and that the choice of optimisation algorithm is a key component of any efforts to reduce computational costs. This is consistent with previous conclusions, obtained on synthetic data (3). The advantage is nevertheless smaller than for the parking study. This can probably be explained by a (mathematically) nicer specification of the model in the Mobidrive study, leading to a better-conditioned log-likelihood, and therefore an easier function to maximise.

6. CONCLUSIONS

In this paper, we have reviewed two techniques designed to lead to computational savings when estimating mixed logit models. We have also discussed the general importance of the choice of optimisation algorithm in the estimation of such model structures.

Our analysis indicates that quasi-Monte Carlo draws, such as MLHS draws, can produce more accurate results than standard Monte Carlo methods with a given number of draws. This may however not always be the case, especially for complex models. This confirms that the form of the integrand has an effect on the quality of the approximation obtained with quasi-Monte Carlo draws. MLHS draws are nevertheless simple to implement, and could easily be implemented in a variety of estimation programs, while adaptive Monte Carlo techniques are far more complicated to develop and implement.

Our results also indicate that the trust-region framework is well adapted to non-concave models; it greatly outperforms classical techniques such as the BFGS line search, and is more robust. The use of an adaptive variable sample size strategy always leads to improvements over standard optimisation approaches, while giving more information to the analyst, in terms of simulation bias as well as standard deviation. This last observation is consistent with previous experiments on synthetic data. Further tests would however be useful to assess the performances when using more complicated models, for instance with highly non-linear utilities. The variable sample size strategy presented in this paper is indeed likely to be affected by the conditioning of the log-likelihood, since it requires sufficiently large steps during the optimisation process. Additionally, it should be noted that, to further increase the reliability of the results and conclusions presented in this paper, a higher number of estimation runs (with different draws) should be used, especially for the more complex models.

In closing, we believe that simple quasi-Monte Carlo techniques represent an interesting first step when dealing with numerical efficiency questions, since important savings can often be achieved. This effort can however be quite useless if a poor optimisation algorithm is used. More research is still needed to evaluate efficient quasi-Monte Carlo techniques for complex, high-dimensional problems. Finally, an important avenue for further research is the combination of quasi-random approaches with variable sample size strategies, in order to benefit from the strengths of both approaches.

References

1. Axhausen, K.W., A. Zimmermann, S. Schönfelder, G. Rindsfüser, and T. Haupt (2002), “Observing the rhythms of daily life: A six-week travel diary”, *Transportation*, 29(2) 95-124.
2. Bastin F., Cirillo C., and Ph. L. Toint (2004a), “Convergence theory for nonconvex stochastic programming with an application to mixed logit”. *Mathematical Programming B. forthcoming.*
3. Bastin F., Cirillo C., and Ph. L. Toint (2004b), “A trust region Monte Carlo algorithm for mixed logit estimation”. Technical Paper, Transportation Research Group, FUNDP, Namur, Belgium.
4. Bastin F., Cirillo C., and Ph. L. Toint (2004), “An adaptive Monte Carlo for computing mixed logit estimators”, *Journal of Computational Management*. Forthcoming.
5. Bhat, C. R. (2001), “Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model”, *Transportation Research*, 35B(7), pp.677-693.
6. Bhat, C. R. (2003), “Simulation Estimation of Mixed Discrete Choice Models Using Randomized and Scrambled Halton Sequences”, *Transportation Research*, 37B(9), pp 837-855.
7. Bhat, C. R. and S.K.Singh (2000), “A comprehensive daily activity-travel generation model system for workers”, *Transportation Research,* 34A(1) 1-22.
8. Cirillo C. and K.W. Axhausen (2004), “Evidence on the distribution of values of travel time savings from a six-week diary”. Arbeitsbericht Verkehrs- und Raumplanung 212, IVT, ETH Zurich, Switzerland.
9. Cirillo C. and Ph. L. Toint (2001), “A multinational comparison of travel behaviour using an activity-based approach”. Technical Paper, Transportation Research Group, FUNDP, Namur, Belgium.
10. Conn, Gould, and Toint (2000), *Trust-Region methods*, SIAM, Philadelphia, USA.
11. Garrido R.A. (2003), “Estimation performance of low discrepancy sequences in stated preferences”. Paper presented at IATBR, Lucerne, Switzerland.
12. Hess, S., Polak, J.W., Daly, A.J. (2003), “On the performance of shuffled Halton sequences in the estimation of discrete choice models” Paper presented at the European Transport Conference, Strasbourg, France.
13. Hess, S., Bierlaire, M. and Polak, J.W. (2005), “Mixed Logit estimation of value of travel-time savings”, Transportation Research 39A(2-3), pp. 221-236.
14. Hess, S. and Polak, J.W. (2004), “Mixed Logit estimation of parking type choice”, paper presented at the 83rd Annual Meeting of the Transportation Research Board, Washington, DC.
15. Hess, S., Train, K., and Polak, J.W. (2004), “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.
16. Lee, L. (1995), “Asymptotic bias in simulated maximum likelihood estimation of discrete choice models”, *Econometric Theory*, 11, pp. 437-483.
17. L'Ecuyer, P. and Lemieux, C. (2002), “Recent advances in randomized quasi-Monte Carlo methods”. Internat. Ser. Oper. Res. Management Sci., *Modeling uncertainty*, 46, Kluwer Academic Publishers, pp 419-474.
18. McFadden, D. and Train, K. (2000), “Mixed MNL Models for discrete response”, *Journal of Applied Econometrics*, 15, pp. 447-470.
19. More, J. J. and Thuente, D. J. (1994), “Line Search Algorithms with Guaranteed Sufficient Decrease**”,** *ACM Transactions on Mathematical Software*, 20(3), pp. 286-307.
20. Nocedal, J. and Wright, S. J. (1999), *Numerical Optimization*, Chapter 8, Springer, New York, NY, USA.
21. Polak, J.W. and Axhausen, K.W. (1989), “The Birmingham CLAMP Stated Preference Survey”, Second Interim Report to Birmingham City Council, Transport Studies Unit, Oxford University.
22. Polak, J.W., Axhausen, K.W., and Errington, T. (1990), “The application of CLAMP to the analysis of parking policy in Birmingham City Centre”, Working Paper 554, Transport Studies Group, Oxford University.
23. Revelt, D. and Train, K. (1998), “Mixed logit with repeated choices”, *Review of Economics and Statistics,* 80, pp. 647–657.
24. Sándor, Z. and Train, K. (2004). “Quasi-random simulation of discrete choice model”. *Transportation Research*, 38B(4), pp. 313-327.
25. Train, K. (1999), “Halton sequences for mixed logit”, Technical Paper, Department of Economics, University of California, Berkeley.
26. Train, K. (2003), *Discrete Choice Methods with Simulation,* Cambridge University Press, Cambridge, MA, USA.
27. Vicente, L. N. (1996) “A Comparison between Line Searches and Trust Regions for Nonlinear Optimization”, *Investigação Operacional*, 16, pp. 173-179.

28. Walker, J. L. (2001), "Extended Discrete Choice Models: Integrated Framework, Flexible Error Structures, and Latent Variables", Ph.D. thesis, MIT, Cambridge, MA.

29. Zhu, C., Byrd, R. H. and Nocedal, J. (1997), "Algorithn 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization", *ACM Transactions on Mathematical Software*, 23(4), pp. 550-560.

Table 1: Parking choice data set, estimation results (PCM)

Parameter	Distr.	MLHS 2000 draws		BTRDA 20000 draws	
		Est.	**t-stat.**	**Est.**	**t-stat.**
ASC Charged on street (N)	μ	-3.0617	3.04	-3.3018	2.92
	σ	3.7468	2.76	3.9796	2.79
ASC Charged off street (N)	μ	0.5339	0.96	0.4853	0.88
	σ	2.2877	3.44	2.1467	3.09
ASC Multi-storey (N)	μ	0.6742	1.04	0.5879	0.91
	σ	3.4751	5.56	3.4657	5.49
ASC Illegal (N)	μ	-6.4490	5.24	-6.4153	4.91
	σ	4.8051	3.83	4.7172	3.79
Time access (F)	μ	-0.1116	3.81	-0.1111	3.79
Time search (LN)	μ	-1.8388	8.15	-1.8321	8.04
	σ	0.8814	3.06	0.8777	2.99
Time Egress (LN)	μ	-1.8863	6.48	-1.9071	6.44
	σ	0.6406	1.39	0.6954	1.73
Time parking fee (LN)	μ	0.9081	4.34	0.9014	4.10
	σ	0.6530	3.20	0.7045	3.30
Time expected fine (N)	μ	-2.3842	3.83	-2.2988	3.74
	σ	1.0626	3.25	1.0449	3.07
Number of observations		1335		1335	
Likelihood at zero		-1466.647		-1466.6	
Likelihood at convergence		-640.286		-640.615	
Number of parameters		17		17	

Table 2: PCM: goodness of model parameters, computational times, LL accuracy and bias

MLHS / BTR*	200 draws	500 draws	1000 draws	2000 draws
Bias as share of standard error	0.4542	0.3068	0.3879	0.1025
RMSE as share of standard error	0.9919	0.7647	0.6949	0.4010
St. Dev. as share of standard error	0.8964	0.7062	0.5773	0.4054
Computational Time (s)	46	115	238	461

Monte-Carlo / BFGS*	1000 draws	5000 draws	10000 draws
Bias as share of standard error	0.5609	0.1296	0.0739
RMSE as share of standard error	1.0258	0.2871	0.1765
Std. Dev. as share of standard error	0.8696	0.2633	0.1597
Computational Time (s)	479	1960	4202
Estimated accuracy of the log-likelihood	2.83351 (0.442%)	1.27682 (0.199%)	0.88883 (0.139%)
Estimated bias of the log-likelihood	-1.48376 (0.232%)	-0.30128 (0.047%)	-0.14600 (0.023%)

Monte-Carlo / BTR*	1000 draws	5000 draws	10000 draws
Bias as share of standard error	0.1988	0.0910	0.0588
RMSE as share of standard error	0.4539	0.2513	0.1698
Std. Dev. as share of standard error	0.4851	0.2437	0.1607
Computational Time (s)	229	1018	2315
Estimated accuracy of the log-likelihood	2.72686 (0.426%)	1.22173 (0.191%)	0.88936 (0.139%)
Estimated bias of the log-likelihood	-1.37417 (0.215%)	-0.27585 (0.043%)	-0.14617 (0.023%)

Monte-Carlo / BTRDA*	1000 draws	5000 draws	10000 draws
Bias as share of standard error	0.2596	0.0854	0.0746
RMSE as share of standard error	0.5603	0.2643	0.1512
Std. Dev. as share of standard error	0.4946	0.2597	0.1280
Computational Time (s)	149	677	1417
Estimated accuracy of the log-likelihood	2.80915 (0.439%)	1.28189 (0.200%)	0.88670 (0.138%)
Estimated bias of the log-likelihood	-1.45836 (0.228%)	-0.30368 (0.047%)	-0.14530 (0.023%)

* draw type / optimisation routine

Table 3: *Mobidrive* data set, estimation results

Parameter	**Alter-native**	**Distr.**	**MLHS**		**BTRDA**	
			1000 draws		**10,000 draws**	
			Est.	**t-stat.**	**Est.**	**t-stat.**
ASC Car passenger	CP	μ	-1.1394	11.78	-1.1390	11.77
ASC Public Transport	PT	μ	-0.7211	3.99	-0.7205	3.99
ASC Walk	W	μ	1.3320	7.69	1.3324	7.70
ASC Bike	B	μ	0.8872	5.24	0.8886	5.25
Sub-Urban HHLD location	CD, CP	μ	0.4439	5.21	0.4428	5.21
Urban HHLD location	PT	μ	0.1950	1.79	0.1950	1.78
Age 18-25	PT	μ	1.3009	8.42	1.3025	8.43
Age 26-35	CD, CP	μ	0.3778	2.31	0.3772	2.31
Age 51-65	PT	μ	0.4562	4.27	0.4568	4.28
Female and part-time	CP	μ	0.7179	6.83	0.7166	6.81
Married with children	CD	μ	0.7938	9.26	0.7922	9.24
Main car user	CD	μ	1.0684	11.66	1.0677	11.64
Annual mileage by car	CD	μ	0.0267	7.55	0.0266	7.54
Number of season tickets	CD	μ	-0.1915	2.06	-0.1915	2.06
Number of stops	CD	μ	0.1596	3.52	0.1591	3.52
Time BP	All	μ	-0.0281	9.99	-0.0281	9.97
Time BP	All	σ	0.0208	5.30	0.0210	5.32
Cost	CD, PT	μ	-0.1233	8.76	-0.1234	8.70
Cost	CD, PT	σ	0.0423	2.18	0.0428	2.16
Time Budget	CD, CP	μ	-0.0330	2.03	-0.0326	2.01
Time Budget	CD, CP	σ	0.0607	2.11	0.0585	1.97
Sum of Travel Time	B	μ	-0.0419	-5.87	-0.0421	5.86
Sum of Travel Time	B	σ	0.0409	4.80	0.0412	4.81
Tour Duration	PT	μ	0.0039	16.50	0.0039	16.49
Number of observations			5795		5795	
Likelihood at zero			-8180.169		-8180.169	
Likelihood at convergence			-6447.956		-6447.400	
Number of parameters			24		24	

Table 4: *Mobidrive:* goodness of model parameters, computational times, LL accuracy and bias

MLHS / BTR*	**200 draws**	**500 draws**	**1000 draws**	**2000 draws**
Bias as share of standard error	0.02690	0.00961	0.01409	0.00311
RMSE as share of standard error	0.04516	0.02486	0.02572	0.02261
St. Dev.	0.03678	0.02387	0.02227	0.02316
Computational Time (s)	381	967	2474	4468

Monte-Carlo / BFGS*	**1000 draws**	**2500 draws**	**5000 draws**
Bias as share of standard error	0.00971	0.01826	0.00546
RMSE as share of standard error	0.06505	0.03428	0.02926
St. Dev.	0.06737	0.02891	0.02996
Computational Time (s)	3024	7610	14746
Estimated accuracy of the log-likelihood	1.61874 (0.0251%)	1.03454 (0.0161%)	0.7237 (0.0112%)
Estimated bias of the log-likelihood	-0.48425 (0.0075%)	-0.19779 (0.0031%)	-0.09738 (0.0015%)

Monte-Carlo / BTR*	**1000 draws**	**2500 draws**	**5000 draws**
Bias as share of standard error	0.01946	0.01076	0.00491
RMSE as share of standard error	0.05128	0.04215	0.03615
St. Dev.	0.04927	0.04244	0.03769
Computational Time (s)	2280	5769	11064
Estimated accuracy of the log-likelihood	1.63525 (0.0254%)	1.02798 (0.0159%)	0.7239 (0.0112%)
Estimated bias of the log-likelihood	-0.49418 (0.0077%)	-0.19529 (0.0030%)	-0.09738 (0.0015%)

Monte-Carlo / BTRDA*	**1000 draws**	**2500 draws**	**5000 draws**
Bias as share of standard error	0.01458	0.00765	0.00504
RMSE as share of standard error	0.06377	0.04622	0.03529
St. Dev.	0.06493	0.04750	0.03665
Computational Time (s)	1381	3215	5857
Estimated accuracy of the log-likelihood	1.62796 (0.0253%)	1.023557 (0.0159%)	0.7255 (0.0112%)
Estimated bias of the log-likelihood	-0.48979 (0.0076%)	-0.19362 (0.0030%)	-0.09726 (0.0015%)

* draw type / optimisation routine