Quasi-Random Simulation of Discrete Choice Models

by
Zsolt Sándor and Kenneth Train
Erasmus University Rotterdam and University of California, Berkeley

December 15, 2004

Abstract

We describe the properties of (t, m, s)-nets and Halton draws. Four types of (t, m, s)-nets, two types of Halton draws, and independent draws are compared in an application of maximum simulated likelihood estimation of a mixed logit model. All of the quasi-random procedures are found to perform far better than independent draws. The best performance is attained by one of the (t, m, s)-nets. The properties of the nets imply that two of them should outperform the other two, and our results confirm this expectation. The two more-accurate nets perform better than both types of Halton draws, while the two less-accurate nets perform worse than the Halton draws.

Econometric Institute Report EI 2004-51

1 Introduction

A choice probability is the expectation of a statistic over a density: $P = \int g(\varepsilon)f(\varepsilon)d\varepsilon$, where f is the density of random factors ε and g is a calculable function. This expectation is simulated by evaluating g at R values of ε and averaging the results. If each evaluation point is drawn independently from f, then the simulated probability is unbiased and consistent for the true probability, with variance inversely proportional to R. In maximum simulated likelihood (MSL) estimation, the simulation induces both bias and variance, with the bias arising from the log transformation of the simulated probability. If R rises faster than the square root of the number of observations, then the effects of simulation disappear asymptotically, and MSL is equivalent to maximum likelihood with exact probabilities (McFadden, 1989; Lee, 1995). However, for fixed R, simulation bias and variance are necessarily present.

Instead of taking independent draws, simulation can potentially be improved by selecting evaluation points more systematically. "Quasi-random" draws are designed to provide better coverage than independent draws over the density for which the integral is defined. This improved coverage usually translates into smaller expected approximation error. In the context of MSL, where bias arises along with variance, quasi-random draws have the potential to reduce the root-mean-squared-error (RMSE) of the estimates against those that would be obtained with the infeasible exact probabilities.

Halton draws have been examined by Bhat (2001; forthcoming) and Train (2000; forthcoming) in the context of MSL on discrete choice models and have been found to provide far more accuracy than a comparable number of independent draws. Halton draws are designed, as described below, to provide good coverage in one dimension. When they are used for multidimensional integration, they provide good coverage over each individual dimension. However, they are not designed to provide good coverage over the multi-dimensional space. For example, in two dimensions, Halton draws are designed to be evenly spaced in the x-dimension and evenly spaced in the y-dimension; however, their spacing in the x-y plane is not an aspect of their design.

(t, m, s)-nets are designed to provide good coverage in the multidimensional space as well as along each individual dimension. Sándor and András (2001) examined these kinds of draws in the calculation of probit probabilities and found them to perform well relative to Halton and independent draws. They did not, however, examine their use in estimation of model parameters.. The purpose of the current paper is to examine (t, m, s)-nets relative to Halton and independent draws in MSL estimation. We conduct a Monte Carlo experiment using data on consumers' choice of vehicle within a mixed logit specification. We compare the RMSE of the estimated parameters for the three kinds of (t, m, s)-nets, two kinds of Halton draws, and independent draws. We find, in summary, that well-designed (t, m, s)-nets perform better than Halton draws, and that both types of draws perform far better than independent draws. The (t, m, s)-net that performs the best is the only net that is also an orthogonal array-based Latin hypercube: its structure arises from the numerical tradition on arrays and Latin hypercubes as well as that of (t, m, s)-nets. These terms are defined in the sections below, followed by the experimental results.

2 Halton draws

We first define a Halton sequence in one dimension and then show how a sequence in multiple dimensions is created. We then describe procedures for introducing randomness, since Halton sequences are deterministic.

2.1 One dimension

A Halton sequence is defined in terms of a base. The sequence in base 10 is most easily explained; sequences in other bases are created the same as in base 10 except with conversion to and from the new base as initial and final steps.

A Halton sequence in base 10 is created in two simple steps:

1. List the integers: 0, 1, 2, 3, 4, 5, 6, 7, 9 , 10, 11, 12 ,13, ...

2. From each integer, create a decimal number by reversing the digits in the integer and putting them after a decimal point. That is, 1 becomes .1, 12 becomes .21, 256 becomes .652. The sequence is now: 0, .1, .2, .3,. .4, .5, .6, .7, .8, .9, .01, .11, .21, .31, ...

Note that the sequence cycles through the unit interval every ten elements. That is, the sequence consists of successive subsequences of length 10 with the elements in each subsequence rising in value and the first element of a subsequence being lower than the last element of the immediately previous subsequence. Note also that successive elements in each subsequence are spaced the same distance apart (1/10 apart to be precise).

A Halton sequence in any other base is created by converting to this base before step 2 and converting back to base 10 after step 2. The steps for a Halton sequence in base 2 are:

1. List the integers (in base 10): 0, 1, 2, 3, 4, 5, 6, 7, 9 , 10, 11, 12 ,13, ...
2. Convert these integers to base 2. The base-2 integers are: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, ...
3. From each base-2 integer, create a base-2 decimal number by reversing the digits in the integer and putting them after the decimal point. The sequence becomes: 0, .1, .01, .11, .001, .101, .011, .111, .0001, .1001, .0101, .1101, .0011, .1011, ...
4. Convert these base-2 decimal numbers back to base 10: 0, 1/2, 1/4, 3/4, 1/8, 5/8, 3/8, 7/8, 1/16, 9/16, 5/16, 13/16, 3/16, 11/16, ... In general, this conversion is calculated as $\sum_{k=1}^{m} d_k/b^k$ where d_k is the k-th digit after the decimal and b is the base.

The sequence cycles every b elements and each element in the cycle is $1/b$ units apart.

2.2 Multiple dimensions

The Halton sequence just described provides points on the unit line. Halton sequences are created in multiple dimensions by using a different base for each dimension. For example, a Halton sequence in two dimensions can be created from the Halton sequences in bases 10 and 2. The sequence is: (0,0), (.1,.5), (.2,.25), (.3,.75),...

The first dimension cycles every 10 elements and the second dimension cycles every 2 elements. Since 10 is a multiple of 2, the cycles overlap and every pair that rises in the second dimension also rises in the first dimension. As a result, the elements are correlated over the two dimensions. To prevent the overlapping of cycles, Halton sequences in multiple dimensions are usually based on prime numbers only, such that none is a multiple of another. For our application, a sequence in five dimensions was created from the Halton sequences for bases 2, 3, 5, 7 and 11.

Even with prime numbers, Halton sequences can evidence correlation over dimensions. For example, a Halton sequence in nine dimensions that uses the primes 17 and 19 (which are the eighth and ninth primes) contains correlation in these dimensions because the cycles are so close as to largely overlap over long segments of the sequence. More importantly, there is no reason to expect that multiple Halton sequences, each of which provides good coverage in one dimension, will, when combined as a sequence of vectors, provide good coverage in multiple dimensions. In contrast, (t, m, s)-nets are created explicitly from concepts regarding coverage in multiple dimensions.

2.3 Randomization

Randomness can be introduced to a Halton sequence in several ways. Wang and Hickernell (2000) suggest a random start procedure. Draw an integer randomly between 0 and some large K and label the draw N_0. For a Halton sequence of length L, create a sequence starting at 0 of length $N_0 + L$ and discard the first N_0 elements. Or, stated differently, create a Halton sequence starting at integer N_0 in step 1 above.

In estimation, a set of evaluation points is used for each observation within the sample. Halton sequences with random starting points can be constructed in two ways for estimation. A "short" sequence can be created for each observation by randomly choosing a starting integer separately for each observation. With this method, which we label HS, each observation has its own randomized Halton sequence. Alternatively, one long sequence for the entire sample can be created from a randomly chosen starting point. With N observations and R evaluation points for each observations, a sequence of length $N * R$ is created from a randomly chosen starting point. Each subsequence of length R is assigned to each observation. We label this procedure HL. The potential advantage of HL arises because Halton sequences are created such that each subsequent point fills in an area that had not been covered by previous points. With one long sequence, the evaluation points for each observation can be self-correcting over observations, as discussed by Train (2000, 2001). The disadvantage of HL is that it contains less randomness than HS.

Tuffin (1996) proposes an alternative randomization procedure for Halton sequences, which is discussed and utilized by Bhat (forthcoming). The Halton sequence is shifted by adding a draw from a standard uniform distribution to each element of the sequence. For any element c of the original sequence in one dimension, the new element is $k = c + \mu$ if $c + \mu < 1$ and $k = c + \mu - 1$ if $c + \mu \geq 1$, where μ is a draw from a standard uniform. Each element is shifted up by μ, and if this shifting pushes the element to the end of the unit line (i.e., to 1), then the shifting "wraps around" back to the beginning of the line (i.e., to 0) and continues. For Halton sequences in multiple dimensions, a separate draw is used for each dimension.

In estimation on a sample of observations, Tuffin's procedure can be applied to a sequence for the entire sample (analogous to HL for a random starting point.) However, if one sequence is created and then randomly shifted separately for each observation (analogous to HS), the relations among the points in each dimension remain the same for all observations. For this reason, we use the random start procedure in our application.

A randomized Halton sequence is a set of draws from the uniform distribution. To obtain draws from density f with cumulative distribution F, each element c is transformed as $F^{-1}(c)$. This inverse-cumulative transformation assumes independence over dimensions; this independence can always be assured by placing the covariance terms within g rather than f.

3 (t, m, s)-nets in base b

The defining terms of a (t, m, s)-net in base b are most readily understood in the context of an example. In our application, there are five dimensions of integration, and we are using 64 evaluation points for each observation. The term s is the dimension of the space, which in our case is 5. The term m relates to the number of points that are contained in the net. In our case, we have 64 points, which is 2^6. The term m is this 6: m is the power to which the base is raised to obtain the length of the sequence. Stated equivalently, a (t, m, s)-net in base b has length b^m. There is a reason for defining the net in terms of m instead of simply the length of the sequence. A (t, m, s)-net can only be constructed for lengths that are some power of the net's base. A (t, m, s)-net in base 2 can only have length 2, 4, 8, 16, 32, 64, 128, etc.; it cannot have a length of, say, 100. In this regard, (t, m, s)-nets are less flexible than Halton sequences, since a Halton sequence can have any length. A net of length 64 can be created in base 4 as well as base 2, with $m = 3$ such that $4^3 = 64$. In our application, we will utilize nets of the form $(t, 6, 5)$ in base 2 and $(t, 3, 5)$ in base 4. We defer the explanation of t until later.

In one dimension, (t, m, s)-nets are created the same as Halton sequences but with an extra step. Recall that in step 3, the sequence is expressed in decimals; for example, in base 2, the decimal sequence is 0, .1, .01, .11, .001, .101, .011, .111, .0001, .1001, .0101, .1101, .0011, .1011, ... For a (t, m, s)-net, each element of this sequence of decimals is transformed in a particular way. Consider the fourth element, .110. Let c be the vector comprised of the digits of this decimal. That is, $c = \langle 1, 1, 0 \rangle$. A new vector, k is created by the transformation $k = Mc$ where M is a "generating matrix" and the matrix multiplication is performed modulo 2 (so that k has elements that take 0 or 1). For example, suppose the transformation matrix is

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Then the digits of the fourth element of the sequence are transformed to become

$$k = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

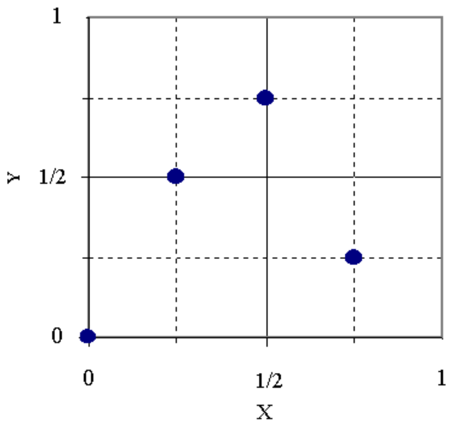
where the top element uses the fact that $1+1$ modulo 2 is 0 (since 2 divided by 2 has no remainder.) The fourth element is changed from .110 to .010.

Nets in multiple dimensions are created by using the same base for all dimensions but applying a different generating matrix in each dimension. In contrast, a Halton sequence uses a different base in each dimension but the same generating matrix for all dimensions (namely, the identity matrix, which doesn't change the sequence). A Halton sequence is not a (t, m, s)-net since its dimensions are defined in different bases whereas all dimensions of a $(t, m.s)$-net are defined in the same base. The desirable properties of a (t, m, s)-net, discussed below, arise from the construction of appropriate generating matrices.

The steps for a (t, m, s)-net in base 2 are:

1. List the integers (in base 10): 0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12 ,13, ...
2. Convert these integers to base 2. The base-2 integers are: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101,...
3. From each base-2 integer, create a base-2 decimal number by reversing the digits in the integer and putting them after the decimal point. The sequence becomes: 0, .1, .01, .11, .001, .101, .011, .111, .0001, .1001, .0101, .1101, .0011, .1011,...
4. Transform each element of the sequence by $k = Mc$, using a different M for each dimension.
5. Convert these base-2 decimal numbers back to base 10.

For interpretation, consider a (0,2,2)-net in base 2. This net has $2^2 = 4$ elements in 2 dimensions. For this illustration, t is set at 0, which provides



the best coverage, as we will see. Each dimension is divided into 2 equal segments (where 2 is the base), creating four squares of size (1/2)x(1/2)=1/4. This division is shown by the dark lines in Figure 1. Then each of the segments in each dimension is divided into two equal subsegments, creating four smaller squares within each of the larger squares, and a total of 16 squares within the unit square. The goal is to place the 4 points in the unit square in a way that gives the best coverage. There are 16 small squares and only 4 points to allocate among them. What provides the best coverage? The points in Figure 1 constitute a (0,2,2)-net, constructed by steps 1-4 above. Each point is at the lower-left corner of one of the small squares; we explain later how these points are moved away from the corners. Note that there is one point in each of the four large squares, and so the four points provide as good coverage over the two dimensions as possible with four points. Note also, that in each dimension, there is one point in each of the four subsegments along that dimension. So the points obtain even coverage over each dimension considered separately.

The coverage properties can be stated more precisely. Each of the four larger squares has size 1/4 and contains one point. Each tall rectangle, of height 1 and width 1/4, has size 1/4 and contains one point. And each long rectangle, of height 1/4 and length 1, has size 1/4 and contains one point. The situation can be described in a way that facilitates generalization. The unit square can be partitioned into subspaces (squares or rectangles) in a variety of ways. Consider any partitioning that satisfies the following conditions: each dimension is segmented into 1, 2 or 4 equal parts (of length 1, 1/2, or 1/4), with possibly a different number of segments in each dimension, but with each subspace created by the segmentation having a size of 1/4. Under any such partitioning, the net contains exactly one point in each subspace.

We can now define t. A (t, m, s)-net in base b contains exactly b^t points in each s-dimensional subspace if the subspaces are created by segmenting each dimension into b^k segments of $1/b^k$ length for some $k = 0, ..., m$ and have volume b^{t-m}. Note that there are b^m/b^t subspaces in any such partitioning, such that, with b^t points in each subspace, the total number of points is b^m. In our (0,2,2)-net in base 2, $t = 0$, such that exactly $2^0 = 1$ point lies in each subspace of size $2^{0-2} = 1/4$ when the subspaces are created by (i) segmenting each dimension into two segments of length 1/2, such that each subspace is a square of size 1/2 by 1/2, or (ii) segmenting the x-dimension into 4 parts of length 1/4 and the y-dimension into 1 part of length 1, such that each subspace is a tall rectangle of size 1/4, or (iii) segmenting the x-dimension into 1 part and the y-dimension into four parts, creating long rectangles of size 1/4. (Stated alternatively: the product of the number of segments over dimensions must equal b^{m-t}. With $b = 2, t = 0$ and $m = 2$, the product must be 4. The possibilities are 2*2, 1*4, and 4*1.)

For a given, m, s and b, a lower t provides better coverage, since the partitioning consists of more subspaces and fewer points in each subspace. (The placement of points within each subspace is not designed to provide good coverage within the subspace, and so better coverage of the unit space is attained with fewer points in more subspaces.) The minimum attainable value of t depends on the values of m, s and b. For any values of t, m, s and b, appropriately specified generating matrices assure that the resulting points have the properties required for a (t, m, s)-net.

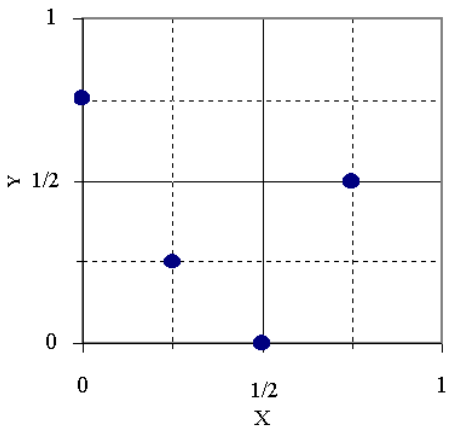
In our application, with 64 points in five dimensions, a t of 0 is attainable for base 4. We therefore create a $(0, 3, 5)$-net in base 4. This net can also be characterized as an orthogonal array-based Latin hypercube (Tang (1993)), as explained by Sándor and András (2001).[1] We refer to this net as O. For base 2, the smallest currently available t for our values of $s = 5$ and $m = 6$ (to obtain $2^6 = 64$ points) is 2. We construct a $(2, 6, 5)$-net in base 2, utilizing generating matrixes from Prisic (forthcoming) based on the construction by Niederreiter and Xing (1996). We call this X. Finally, we create a $(3, 6, 5)$-net in base 2, called N, following Niederreiter's (1988) older

[1]Strictly speaking though, the definition of an orthogonal array latin hypercube does not guarantee that each property of a (0,3,5)-net in base 4 is satisfied because the rectangles of size 1/4 by 1/16 are not guaranteed to contain exactly one point.

proposal. We construct the generating matrices based on the algorithm developed by Bratley et al. (1992). The generating matrices for O, X, and N are given in the appendix. We expect a performance ordering of O followed by X followed by N. We also expect the (t, m, s)-nets to perform better than the Halton draws, since the nets are designed to have good coverage in multiple dimensions while the multidimensional coverage of Halton sequences is not designed but rather occurs by happenstance from the combination of whatever different bases are being used for the dimensions.

3.1 Randomization

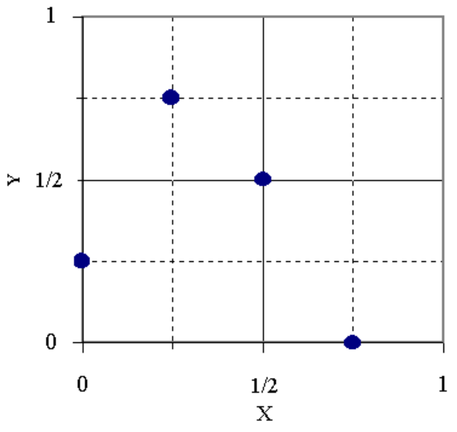
Owen (1995) suggests a procedure for randomizing a $(t, m, s,)$-net that retains the coverage properties of the net. The randomization consists of, first, randomly re-ordering the numbers in the base, and, second, shifting each element by a random amount. Consider first the re-ordering of numbers. In base 2, there are two numbers, 0 and 1, and there are two possibilities for their ordering: either 1 is "larger" than 0, or 0 is "larger" than 1. An equal probability is given for each possible ordering. An ordering is chosen randomly and the numbers are changed to reflect this new ordering. For example, a draw from a standard uniform is taken. If the draw is below .5, the first ordering is used, and the 1's and 0's are left the way they are. If the draw is above .5, the second ordering is used, and, to reflect this ordering, the 1's are changed into 0's and the 0's into 1's. This reordering is performed on each digit of the decimals from step 3, sequentially for each successive digit such that the reordering of each digit depends on the previous digits. The reason why this is so will be apparent in the graphical interpretation below. For example, the first eight elements of the sequence in base 2 from step 3 are: .000, .100, .010, .110, .001, .101, .011, .111. We take a draw from a uniform to determine the ordering for the first digit. If .632 is drawn, then the 1's in the first digit are changed to zeros and vice versa. The sequence becomes: .100, .000, .110, .010, .101, .001, .111, .011. For the second and third digit we apply different random reorderings if the previous digits are different. For reordering the second digit we use two different random reorderings depending on whether the first digit is 0 or 1. If the two draws corresponding to 0 and 1 are 0.115 and 0.821, respectively, and the third digit is not changed, then the sequence becomes: .110, .000, .100, .010, .111, .001, .101, .011. For reordering the third digit we apply four different random reorderings depending on whether the first two digits are 00, 01, 10 or 11. For nets in multiple dimensions, the random reordering is applied to each dimension independently.

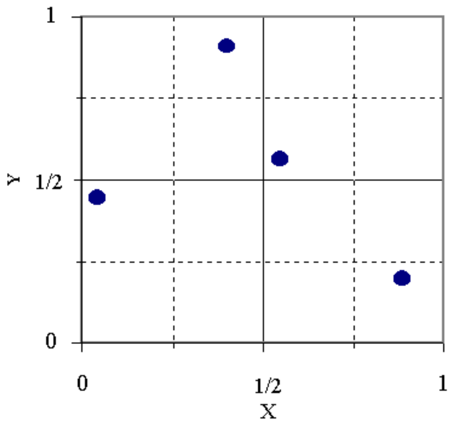


This randomization has a graphical interpretation. In Figure 1, the first dimension is the x-axis. Changing the 1's to 0's and vice versa in the first digit is equivalent to interchanging the two tall 1/2x1 rectangles, which gives Figure 2. Changing the 1's to 0's and vice versa in the second digit is equivalent to interchanging the 1/4x1 rectangles within each of the 1/2x1 rectangles, as shown in Figure 3. Now we can see that using two different random reorderings if the first digit takes 0 and 1 ensures that the random interchange of the 1/4x1 rectangles within the first 1/2x1 rectangle is independent of the random interchange of the 1/4x1 rectangles within the second 1/2x1 rectangle. The same process is then applied to the other dimension (to the long rectangles).

As stated above, a (t, m, s)-net created through appropriate generating matrices gives points on the lower-left corners of the subspaces. Random reordering of the numbers does not change this attribute. To move the points inside their respective subspaces, a draw is taken from a uniform between 0 and $1/b^m$, where $1/b^m$ is the width of each side of the subspace. In our application, a draw is taken between 1 and 1/64. This draw is added to each element of the sequence. A separate draw is taken for each dimension. These random draws, one for each dimension, move all the points the same distant and direction into their respective subspaces. Figure 4 shows the points after random shifting from Figure 3. These draws, which are from a uniform distribution, are transformed to the density of interest the same as with Halton sequences.

Owen-randomization could be applied to Halton sequences as shown by


1
1/2
0
Y
0
1/2
1
X


1
1/2
0
Y
0
1/2
1
X

Wang and Hickernell (2000). The way it is applied is similar to the randomization of (t, m, s)-nets but here the reordering of the different digits is independent, that is, it does not depend on the previous digits. However, the advantage of Owen-randomization is that it preserves the properties of a (t, m, s)-net. Since Halton sequences are not (t, m, s)-nets and do not have their desirable properties, the motivation for this form of randomization is missing. Also, Wang and Hickernell (2000) show that, for five dimensions, the random start procedure tends to be more efficient than Owen-randomization for Halton sequences. It is not obvious how the simpler randomization methods like the random shift or random start procedures can be adapted to (t, m, s)-nets. If these procedures are applied directly then the (t, m, s)-net property of these sequences will not be guaranteed.

4 Application

For our application, we use data described by Train and Hudson (2000) on customers' choice among alternative fueled vehicles. Each of 538 surveyed customers was given a card that listed three vehicles along with descriptions of the vehicles. The customer was asked to state which of the vehicles he/she would choose to buy. Each vehicle was described in terms of its price, operating cost, range, performance, and engine type. Three engine types were included in the experiments: gas internal combustion (ICV), electric (EV), or gas-electric hybrid (HV). Range was defined as the number of miles that the vehicle could be driven before refueling/recharging. Three levels of performance were distinguished, each of which was described in terms of top speed and number of seconds needed to reach 60 miles per hour. These two performance factors were not varied independently. The three performance levels were coded as 0,1 and 1.5, since the analysis in Train (2000) indicates that customers valued the increment from low to medium performance twice as much as the increment from medium to high. Operating cost was denoted in dollars per month. A mixed logit model (Revelt and Train, 1998; Brownstone and Train, 1999) was specified with a fixed coefficient for price and independent normal coefficients for operating cost, range, performance, an EV dummy, and an HV dummy.

The model was first estimated ten times by MSL with 10,000 independent random draws for each observation in each run. The mean of these estimates are designed the "true" estimates, against which the other methods of constructing draws are compared. The variance of these estimates is extremely small such that interpreting the mean as the simulation-free

Table 1: Mixed Logit Model of Vehicle Choice
Average estimates and standard errors over ten sets of draws.
10,000 independent random draws per observation.

Explanatory variable		Estimate	Standard error
Price		-0.1278	0.0209
Operating cost	Mean	-0.0689	0.0152
	Std dev	0.0580	0.0342
Range	Mean	1.0730	0.9472
	Std dev	0.1523	10.3274
EV dummy	Mean	-5.8695	1.8540
	Std dev	3.5259	1.4138
HV dummy	Mean	-0.6757	0.4818
	Std dev	2.0843	0.9619
Performance	Mean	0.8701	0.2727
	Std dev	2.311	0.6713

estimates is warranted. Nevertheless, a more conservative interpretation is that the comparisons reveal how well each method performs relative to taking 10,000 independent draws. Table 1 gives the “true” estimates as well as the mean standard errors. Note that the mean standard errors are not the variance of the estimates over the ten sets of draws. Rather they are the average of the ten standard errors obtained in the ten runs. (More precisely, they are the square root of the average of the squared standard errors.) As such, they reflect sampling variance.

The model was estimated ten times with each of the five types of randomized sequences described above: HL, HS, N, X, and O. Each of these runs used 64 draws per observation. We also estimated the model ten times with 64 independent random draws, which we call R64, and, again with 512 random draws (8 times as many), which we call R512. For each type of draw, the RMSE was calculated for the ten estimates against the “true” estimates. The bias (i.e., the difference between the mean estimates and the “true”) and the standard deviation of the estimates were also calculated.

McFadden (1989) has pointed out that simulation variance is related to sampling variance. In particular, for the method of simulated moments with fixed weights and an accept-reject simulator, simulation variance is proportional to sampling variance. A similar relation can be expected to occur with other simulation-assisted estimators, though not necessarily as direct proportionality. Intuitively, the reason for the relation between simulation and sampling variance is clear: Large sampling variance means that the log-likelihood function is fairly flat near the maximum; and when the likelihood function is fairly flat near its maximum, errors induced by simulation can move the maximum considerably. Conversely, when the likelihood function is highly peaked at its maximum, which means small sampling variance, simulation error is unlikely to move the maximum as much.

To reflect this relation and facilitate comparisons over parameters, RMSE, bias and standard deviation are expressed as a proportion of the standard errors from Table 1. The results are given in Table 2-4. For example, Table 2 indicates that method N provides a RMSE for the first parameter that is 37.7 percent of its standard error.

The methods are reported in these tables in ascending order of performance, based on the average over parameters of the RMSE as a proportion of standard error (the last row of Table 2). One exception to this ordering is R512 which is given last since it uses more draws. The relative performance is similar for each parameter individually as for the average over parameters.

All of the randomized sequences greatly outperform the same number of independent random draws (R64). This result is consistent with, and generalizes, the findings of Bhat (2001) and Train (2000), who found that Halton draws greatly outperform independent random draws.

The (t, m, s)-nets have the expected ordering. O performs better than X, though only moderately so. These two nets have the minimum attainable t for their values of m, s and b. For O, $t = 0$, which means that the net contains only one point in each appropriately defined subspace. For X, $t = 2$ which, since its base is 2, means that 4 points are in each subspace. Dominance cannot be established, since the two sequences have different bases. However, it seems reasonable to expect that 1 point per subspace would provide better coverage than 4 points per (differently defined) subspace. And this expectation is borne out by the results, though only slightly.

Both O and X perform better than N. This result is expected, since N has the same m, s, and b as X but a higher t. As stated above, the N sequence was an early proposal by Niederreiter which he and others later refined, giving X and O. And as also stated above, O is an orthogonal array-based Latin hypercube, which is the culmination of an alternative numerical tradition (namely, the combining of orthogonal arrays with Latin hypercubes.) It is heartening that the best performer is the outcome of two different ways of creating quasi-random draws with good coverage.

Consider now the Halton draws. HS, which contains randomization over observations, performs slightly better than HL. The preferred (t, m, s)-nets (that is, X and O) outperform both of the Halton methods. As discussed above, (t, m, s)-nets are constructed to provide good coverage in multiple

Table 2: RMSE as share of standard error of estimate

R64	N	HS	HL	X	O	R512
0.45624	0.37703	0.17848	0.19757	0.19502	0.17944	0.16082
0.43343	0.25428	0.22577	0.17425	0.18536	0.13843	0.13305
0.31277	0.53546	0.60126	0.21181	0.20657	0.20708	0.13333
0.34205	0.18757	0.12352	0.13154	0.07443	0.14126	0.18009
0.08662	0.07738	0.04337	0.04757	0.05956	0.06015	0.06757
0.49436	0.29077	0.19987	0.23985	0.19512	0.15641	0.21433
0.92837	0.52265	0.21659	0.35563	0.28710	0.23988	0.34307
0.33895	0.17561	0.12252	0.19639	0.16860	0.09083	0.10072
0.52460	0.47166	0.32478	0.33307	0.21315	0.18886	0.18733
0.41280	0.26837	0.18218	0.16642	0.18180	0.11368	0.08774
0.65319	0.33003	0.22955	0.18060	0.27523	0.18548	0.10334
Average RMSE as share						
0.45303	0.31735	0.22254	0.20315	0.18563	0.15468	0.15558

dimensions, while the multidimensional coverage of Halton sequences is the unplanned outcome of its combination of bases. This difference in multiple dimensions is the reason for exploring (t, m, s)-nets, and our results indicate that the exploration is warranted.

As a final note, eight times as many independent draws (R512) are required to reach the same level of performance as the best quasi-random method.

5 Conclusion

In this paper we have shown that quasi-random simulation of the mixed logit model yields more precise estimators than the more traditional independent random simulation. We have investigated the performance of several quasi-random samples and found that for our application the so-called (t, m, s)-nets in base 4 for values $t = 0$, $m = 3$ and $s = 5$ yield the most precise estimates. Our results suggest that eight times as many independent draws are required to reach the same level of performance as this net. In other words, we can save about 87.5% of our computing time when we use this net instead of independent draws while still reaching the same precision of the estimates. We have also provided intuitive explanation why these nets are likely to have a good performance in general. Therefore we recommend the use of these for other mixed logit simulation problems as well.

In situations other than that in our application we may need to use $(0, m, s)$-nets in some other base than 4 and a smaller value of m. It is,

Table 3: Bias as share of standard error of estimate

R64	N	HS	HL	X	O	R512
0.40245	0.08611	0.09516	0.08398	0.11111	0.06619	0.09838
0.40069	0.08515	0.18885	0.08208	0.08873	0.05533	0.10278
0.13339	0.12298	0.34857	0.06168	0.08555	0.15657	0.05996
0.26429	0.12579	0.08235	0.09025	0.01824	0.09293	0.13959
0.06463	0.06130	0.03880	0.04009	0.04839	0.04769	0.05483
0.43984	0.12333	0.14954	0.16833	0.05458	0.03259	0.16535
0.60243	0.23262	0.14708	0.27031	0.08168	0.08610	0.26170
0.28903	0.09243	0.09267	0.13084	0.00559	0.03586	0.05302
0.35815	0.18037	0.01456	0.22657	0.04021	0.07968	0.07457
0.32338	0.00661	0.11248	0.08822	0.07197	0.02226	0.05005
0.43730	0.06143	0.08003	0.00167	0.18745	0.02600	0.08526
Average bias as share						
0.33778	0.10710	0.12274	0.11309	0.07214	0.06374	0.10414

Table 4: Standard deviation as share of standard error of estimate

R64	N	HS	HL	X	O	R512
0.22654	0.38692	0.15917	0.18851	0.16894	0.17581	0.13410
0.17420	0.25256	0.13043	0.16202	0.17154	0.13376	0.08905
0.29821	0.54933	0.51642	0.21359	0.19819	0.14286	0.12553
0.22888	0.14667	0.09704	0.10088	0.07606	0.11214	0.11994
0.06078	0.04977	0.02042	0.02699	0.03659	0.03864	0.04164
0.23787	0.27756	0.13978	0.18009	0.19747	0.16126	0.14374
0.74457	0.49335	0.16759	0.24360	0.29013	0.23601	0.23384
0.18663	0.15740	0.08447	0.15438	0.17762	0.08797	0.09026
0.40405	0.45938	0.34200	0.25734	0.22065	0.18049	0.18115
0.27045	0.28280	0.15107	0.14874	0.17598	0.11751	0.07595
0.51145	0.34181	0.22679	0.19036	0.21243	0.19358	0.06155
Average std dev as share						
0.30397	0.30887	0.18502	0.16968	0.17506	0.14364	0.11789

however, important to keep m as large as possible because this ensures a better performance of the net if $t = 0$. Here we discuss how we can decide optimally on the base in which the net is constructed so that m is as large as possible. This will depend on the dimension of the integral underlying the choice probabilities and on the number of evaluation points that we intend to use. We suppose here that in econometric applications the number of evaluation points does not exceed 500. If the dimension is 5 or lower and we want to use few evaluation points than we recommend our best sample O or with slightly more evaluation points (0,3,6)-nets in base 5. If we can afford even more evaluation points then (0,4,5)-nets in base 4, (0,3,8)-nets in base 7 and (0,3,9)-nets in base 8 should be used. These latter two nets are also suited for dimensions up to 8 and 9. If the dimension is between 6 and 10 and we would like to use few evaluation points then we can only use nets with $m = 2$. Such nets are the (0,2,8)-nets in base 7, (0,2,9)-nets in base 8 or the (0,2,10)-nets in base 9. If the dimension is larger than 10 then for having $m = 3$ we need more evaluation points than 1000. Therefore only nets with $m = 2$ can be our target. Such nets are, with increasing number of evaluation points: (0,2,12)-nets in base 11, (0,2,14)-nets in base 13, (0,2,17)-nets in base 16, (0,2,18)-nets in base 17 and (0,2,20)-nets in base 19. Dimensions higher than 20 can be dealt with by putting several randomized versions of a given net next to each other. This method is called Latin supercube sampling in the literature. Since the nets mentioned above are in bases that are primes or powers of primes, they can be constructed in the way described in Appendix A and Appendix B.

Appendix A: Construction of $(0, m, s)$-nets

Since in our application O, that is, a $(0, 3, 5)$-net in base 4 performs the best, here we provide more information on $(0, m, s)$-nets. More exactly, by means of the simple example discussed in section 3 we explain why the generating matrices yield nets and show how the generating matrices are constructed and can be coded for practical use.

First we present the criteria that the generating matrices should satisfy in order to yield nets. We consider only the case when $t = 0$ due to the reason mentioned above, but we note that the case $t > 0$ can be understood in a similar way. We recall from section 3 that a $(0, m, s)$-net in base b contains exactly one point in each s-dimensional subspace of volume b^{-m}. A subspace contains two points if and only if some digits in base b of the coordinates of the two points are the same. In order to see this more precisely, we consider the example of (0,2,2)-nets in base 2 from Figure 1 in section 3. We have $2^2 = 4$ subspaces in each of the three cases (i), (ii), (iii) on page 9. For each dimension the first digit determines whether the point is in the first or second half of the segment [0,1], and the second digit determines whether the point is in the first or second half of the smaller segment of length 1/2. In case (i) the subspaces are the squares of size 1/2. If two points belong to the same subspace then the first digits of their coordinates have to be the same. In case (ii) the subspaces are the four rectangles of size 1/4x1. If two points belong to the same subspace then the first two digits of their x-coordinate have to be the same. By symmetry, for case (iii) we obtain that if two points belong to the same subspace then the first two digits of their y-coordinate have to be the same. So the generating matrices should be such that they avoid yielding points that have the same coordinates in these three cases.

A sufficient condition for having this is that the generating matrices, denoted by

$$C_x = \begin{pmatrix} c^x_{11} & c^x_{12} \\ c^x_{21} & c^x_{22} \end{pmatrix},\ C_y = \begin{pmatrix} c^y_{11} & c^y_{12} \\ c^y_{21} & c^y_{22} \end{pmatrix},$$

satisfy that the matrix formed by the first rows, i.e.,

$$\begin{pmatrix} c^x_{11} & c^x_{12} \\ c^y_{11} & c^y_{12} \end{pmatrix},$$

as well as C_1 and C_2 are nonsingular modulo 2. These criteria correspond to the cases (i), (ii), (iii), respectively. We recall that the points of the (0,2,2)-net in base 2 are determined by multiplying C_1 and C_2 by the reversed digits of the numbers 0, 1, 2, 3 in base 2. For case (i) we know from the previous paragraph that we need that no two points of the net have their first digit of their x- and y-coordinate equal. If they did that would imply that for two numbers of 0, 1, 2, 3 in reversed digital representation $(d_1, d_2) \neq (e_1, e_2)$ we have

$$c^x_{11}d_1 + c^x_{12}d_2 = c^x_{11}e_1 + c^x_{12}e_2,$$
$$c^y_{11}d_1 + c^y_{12}d_2 = c^y_{11}e_1 + c^y_{12}e_2,$$

which implies

$$c^x_{11}(d_1 - e_1) + c^x_{12}(d_2 - e_2) = 0$$
$$c^y_{11}(d_1 - e_1) + c^y_{12}(d_2 - e_2) = 0.$$

Hence

$$\begin{vmatrix} c_{11}^{x} & c_{12}^{x} \\ c_{11}^{y} & c_{12}^{y} \end{vmatrix} = 0$$

and the matrix formed by the first rows would be singular. This establishes the criterion that if the matrix formed by the first rows of the generating matrices is nonsingular then the net property in case (i) is satisfied. We can proceed in a similar way to show that the other two criteria, namely that C_x and C_y are nonsingular imply that the net properties in cases (ii) and (iii) are satisfied. We note that the generating matrices that yield the net from Figure 1 are

$$C_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; C_y = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Obviously they satisfy the three criteria discussed above.

Next we present the principles on which the construction of the generating matrices is based. We follow Niederreiter's (1988) approach. First we present the general case of a $(0, m, s)$-net in base b and then we particularize it to our example. The construction is fairly abstract and is based on expressing the inverse of polynomials in X as an infinite sum of powers of $1/X$ whose coefficients will eventually be the elements of the generating matrices (X should not be confused with the sequence X and the coordinate x). The polynomials considered for different dimensions have to be different irreducible polynomials, that is, they cannot be expressed as a product of two polynomials. This irreducibility property is analogous to the prime number property, so in this respect the construction of nets is related to that of Halton sequences. Formally, for each dimension i we choose an irreducible polynomial $p_i\left(X\right)$ of degree one and for any $j \geq 1$ we write

$$\frac{1}{p_i\left(X\right)^j} = c_{j1}^{i} X^{-1} + c_{j2}^{i} X^{-2} + c_{j3}^{i} X^{-3} + ..., \qquad (1)$$

where the coefficients c_{jr}^{i} take values from $0, 1, ..., b-1$. Then if the generating matrix corresponding to dimension i is defined as

$$C_i = \begin{pmatrix} c_{11}^{i} & c_{12}^{i} & \cdots & c_{1m}^{i} \\ c_{21}^{i} & c_{22}^{i} & \cdots & c_{2m}^{i} \\ \vdots & \vdots & & \vdots \\ c_{m1}^{i} & c_{m2}^{i} & \cdots & c_{mm}^{i} \end{pmatrix},$$

and $s \leq b$, the sequence constructed using these C_i's is a $(0, m, s)$-net in base b. In the next paragraph we show why this is so in the case of our (0,2,2)-net example. The highest dimension in which a $(0, m, s)$-net in base b can be constructed is b because the number of polynomials of degree one having the coefficient of X equal to 1 is b (in fact, these are: X, $X-1, ..., X-(b-1)$). We note that this method works in the case when b is a prime number or a power of a prime number.

Now we return to our example of (0,2,2)-nets in base 2 and explain for these why the coefficients defined in equation (1) yield nets. For this we need to show that the three matrices, that is, the matrix formed by the first rows of the generating matrices and the two generating matrices, C_1 and C_2, are nonsingular modulo 2. We show this only for the first matrix since for the other two the proof is similar. Suppose by contradiction that this matrix is singular. Then there are numbers f_1 and f_2 taking 0 or 1, but not both equal to 0, such that

$$\begin{aligned} c_{11}^x f_1 + c_{11}^y f_2 &= 0, \\ c_{12}^x f_1 + c_{12}^y f_2 &= 0. \end{aligned}$$

Denote by $p_x(X)$ and $p_y(X)$ the polynomials for the x- and y-coordinates, respectively. Then writing out the equations corresponding to (1) for $j = 1$ we have

$$\begin{aligned} \frac{1}{p_x(X)} &= c_{11}^x X^{-1} + c_{12}^x X^{-2} + c_{13}^x X^{-3} + ..., \\ \frac{1}{p_y(X)} &= c_{11}^y X^{-1} + c_{12}^y X^{-2} + c_{13}^y X^{-3} + ... \end{aligned}$$

Then multiply the first equation by f_1, the second by f_2, add them and let

$$L(X) = \frac{f_1}{p_x(X)} + \frac{f_2}{p_y(X)}. \tag{2}$$

Then we obtain

$$\begin{aligned} L(X) &= (c_{11}^x f_1 + c_{11}^y f_2) X^{-1} + (c_{12}^x f_1 + c_{12}^y f_2) X^{-2} + (c_{13}^x f_1 + c_{13}^y f_2) X^{-3} + ... \\ &= (c_{13}^x f_1 + c_{13}^y f_2) X^{-3} + ... \end{aligned}$$

Since $p_x(X)$ and $p_y(X)$ are of degree one we draw the conclusion that the term with the highest degree in the expression $L(X) p_x(X) p_y(X)$ is X^{-1}. On the other hand, from equation (2) we know that the expression $L(X) p_x(X) p_y(X)$ is a polynomial so its lowest degree term is 1 ($= X^0$). This is only possible if all coefficients of $L(X) p_x(X) p_y(X)$ are 0, which in fact means that $L(X)\, p_x(X)\, p_y(X) = 0$. Hence the fact that $p_x(X)$ and $p_y(X)$ are irreducible polynomials implies that $f_1 = f_2 = 0$, a contradiction. This conclusion and virtually the whole proof applies in the general case of $(0, m, s)$-nets in base b but we can directly see in this example why $f_1 = f_2 = 0$ by noting that $p_x(X) = X$ and $p_y(X) = X - 1$. Then

$$\begin{aligned} L(X)\, p_x(X)\, p_y(X) &= f_1 (X-1) + f_2 X \\ &= (f_1 + f_2) X - f_1, \end{aligned}$$

and $L(X)\, p_x(X)\, p_y(X) = 0$ means that the coefficients of this polynomial are equal to 0, i.e., $f_1 = 0$ and $(f_1 + f_2) = 0$, which eventually implies $f_1 = f_2 = 0$. This establishes that the matrix formed by the first rows of the generating matrices is nonsingular. So we can safely believe that the construction method for the generating matrices yields $(0, m, s)$-nets in base b.

Appendix B: Generating matrices

This appendix presents, first, how the generating matrices for $(0, m, s)$-nets in base b can be constructed if b is a prime number or a power of a prime number, second, as an example, the construction of the generating matrices for the (0,3,5)-nets in base 4 that yield the sample O as well as some ingredients for the non-prime bases 8, 9, 16, and third, the generating matrices applied for constructing the other two samples N and X used in the paper.

The formulas of the generating matrices for $(0, m, s)$-nets in base b are derived in Niederreiter (1992). If the polynomial corresponding to dimension i is $X - b_i$, where b_i is one of the numbers 0, 1, ... , $b-1$, then the elements of C_i are given by:

$$\begin{aligned} c^i_{jr} &= 0 && \text{for } 1 \le r < j, \\ c^i_{jr} &= \binom{r-1}{j-1} b_i^{r-j} && \text{for } r \ge j \ge 1. \end{aligned} \tag{3}$$

The operations in these formulas should be done in a special way. First, in the case of b prime they should be done modulo b, that is, the remainder of the division by b should be computed. If b is not a prime but the power of a prime then the operations in the above formulas are different from those for the prime bases. The reason for this is that in this case some numbers are not "invertible" modulo b. For example, if $b = 4$ none of the numbers

0, 1, 2, 3 multiplied by 2 will give 1 modulo 4, and in this sense 2 is not "invertible.." Second, $\binom{r-1}{j-1} b_i^{r-j}$ should be computed as $b_i^{r-j} + ... + b_i^{r-j}$, where the summation is done $\binom{r-1}{j-1}$ times, where $\binom{n}{k} = \frac{(n-k+1)(n-k+2)...n}{1 \cdot 2 \cdot ... \cdot k}$, and by convention $\binom{n}{0} = 1$ for any nonnegative integer n. For $b_i = 0$ and $r = j$ the convention $0^0 = 1$ should be used.

We turn now to the second topic of this appendix. The addition and multiplication operations for base 4 are presented in Table A1. It is straightforward to code these operations and to compute the elements of the generator matrix using the formulas (3). The generating matrices used for constructing O are:

$$C_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, C_3 = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, C_4 = \begin{pmatrix} 1 & 3 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

For illustration we compute the elements of C_4. The corresponding polynomial is $p_4(X) = X - 3$ so $b_4 = 3$. For $j = r$ we have $\binom{r-1}{j-1} = 1$ and $3^{r-j} = 3^0 = 1$, so the main diagonal elements are equal to 1. For $j = 1, r = 2$ we have $\binom{r-1}{j-1} = \binom{1}{0} = 1$ and $3^{r-j} = 3$, so $c_{12}^4 = 3$. For $j = 1, r = 3$ we have $\binom{r-1}{j-1} = \binom{2}{0} = 1$ and $3^{r-j} = 3^2 = 3 * 3 = 2$ (according to the multiplication rule from Table A1), so $c_{13}^4 = 2 = 2$. For $j = 2, r = 3$ we have $\binom{r-1}{j-1} = \binom{2}{1} = 2$ and $3^{r-j} = 3$, so $c_{23}^4 = 3 + 3 = 0$ (according to the addition rule from Table A1). The elements below the main diagonal are equal to 0.

Table A1: Summation and multiplication rules for base 4

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Using these four generating matrices we can construct a (0,3,4)-net in base 4. However, this is 4-dimensional and we need a 5-dimensional net for our application. For this we use a property that these nets satisfy. Denote the points of the net by the 4-dimensional row vectors $x_0, x_1, ..., x_{63}$. Then according to this property, if we add the one-dimensional sequence 0, 1/64, ..., 63/64 to this net then the new sequence $(0, x_0)$, $(1/64, x_1)$, ..., $(63/64, x_{63})$ will be a (0,3,5)-net in base 4.

We illustrate why this is justified in the case of the (0,2,2)-nets in base 2. For this suppose that a_0, a_1, a_2, a_3 is a (0,2,1)-net in base 2 constructed in the way described above. Then it also holds that a_0, a_1 and a_2, a_3 are (0,1,1)-nets in base 2. This latter property is the formal way of saying that the sequence cycles every 2 elements, as in the case of the Halton sequence. We show that the points $(0, a_0)$, $(1/4, a_1)$, $(1/2, a_2)$, $(3/4, a_3)$ form a (0,2,2)-net in base 2. For this we need to show that in the cases (i), (ii), (iii) (page 9) each subspace contains exactly one point of the sequence. In case (i) take for example the subspace determined by the square [0,1/2]x[1/2,1]. Potentially the first two points of the sequence can belong to this. If both of them or none of them belonged to it then a_0, a_1 could not be a (0,1,1)-net in base 2, and thus only one of them belongs to it. The proof for the other subspaces is similar. In case (ii) take for example the subspace determined by the rectangle [1/2,3/4]x[0,1]. It is obvious that the only point that belongs to this is $(1/2, a_2)$. In case (iii) consider the example when the subspace is determined by the rectangle [0,1]x[0,1/4]. Since a_0, a_1, a_2, a_3 is a (0,2,1)-net in base 2 the interval [0,1/4] contains exactly one of these and therefore the rectangle will also contain exactly one point. This establishes the property for (0,2,2)-nets in base 2 and illustrates why it holds in more general situations.

The addition and multiplication for bases 8, 9, 16 are presented in Tables A2, A3, A4, respectively. Using them in the same way as the tables for base 4, we can construct generating matrices in those bases. We note that the operations are commutative, i.e., $a + b = b + a$ and $a * b = b * a$, so presenting only the upper triangle of the tables is fully informative.

Table A2: Summation and multiplication rules for base 8

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1		0	6	4	3	7	2	5
2			0	7	5	4	1	3
3				0	1	6	5	2
4					0	2	7	6
5						0	3	1
6							0	4
7								0

$*$	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1		1	2	3	4	5	6	7
2			3	4	5	6	7	1
3				5	6	7	1	2
4					7	1	2	3
5						2	3	4
6							4	5
7								6

Table A3: Summation and multiplication rules for base 9

+	0	1	2	3	4	5	6	7	8
0	0	1	2	3	4	5	6	7	8
1		5	8	4	6	0	3	2	7
2			6	1	5	7	0	4	3
3				7	2	6	8	0	5
4					8	3	7	1	0
5						1	4	8	0
6							2	5	1
7								3	6
8									4

*	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1		1	2	3	4	5	6	7	8
2			3	4	5	6	7	8	1
3				5	6	7	8	1	2
4					7	8	1	2	3
5						1	2	3	4
6							3	4	5
7								5	6
8									7

Table A4.a: Summation rules for base 16

+	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		0	5	9	15	2	11	14	10	3	8	6	13	12	7	4
2			0	6	10	1	3	12	15	11	4	9	7	14	13	8
3				0	7	11	2	4	13	1	12	5	10	8	15	14
4					0	8	12	3	5	14	2	13	6	11	9	1
5						0	9	13	4	6	15	3	14	7	12	10
6							0	10	14	5	7	1	4	15	8	13
7								0	11	15	6	8	2	5	1	9
8									0	12	1	7	9	3	6	2
9										0	13	2	8	10	4	7
10											0	14	3	9	11	5
11												0	15	4	10	12
12													0	1	5	11
13														0	2	6
14															0	3
15																0

Table A4.b: Multiplication rules for base 16

$*$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2			3	4	5	6	7	8	9	10	11	12	13	14	15	1
3				5	6	7	8	9	10	11	12	13	14	15	1	2
4					7	8	9	10	11	12	13	14	15	1	2	3
5						9	10	11	12	13	14	15	1	2	3	4
6							11	12	13	14	15	1	2	3	4	5
7								13	14	15	1	2	3	4	5	6
8									15	1	2	3	4	5	6	7
9										2	3	4	5	6	7	8
10											4	5	6	7	8	9
11												6	7	8	9	10
12													8	9	10	11
13														10	11	12
14															12	13
15																14

Finally, we present the generating matrices used for the samples N and X. For N we used again the procedure of constructing a 4-dimensional net and adding the sequence 0, 1/64, ..., 63/64 as its first dimension to obtain a 5-dimensional net. This procedure keeps the t parameter low yielding eventually a (3,6,5)-net in base 2. The four generating matrices are:

$$C_1 = \begin{pmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{pmatrix}, C_2 = \begin{pmatrix} 1&1&1&1&1&1\\ 0&1&0&1&0&1\\ 0&0&1&1&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&1\\ 0&0&0&0&0&1 \end{pmatrix},$$

$$C_3 = \begin{pmatrix} 1&1&0&1&1&0\\ 1&0&1&1&0&1\\ 0&0&1&1&1&1\\ 0&1&1&1&1&0\\ 0&0&0&0&1&1\\ 0&0&0&1&1&1 \end{pmatrix}, C_4 = \begin{pmatrix} 1&1&1&0&0&1\\ 1&1&0&0&1&0\\ 1&0&0&1&0&1\\ 0&0&0&1&1&1\\ 0&0&1&1&1&0\\ 0&1&1&1&0&1 \end{pmatrix}.$$

In the case of the X sequence we use five generating matrices. These are:

$$C_1 = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, C_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix},$$
$$C_3 = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, C_4 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix},$$
$$C_5 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

$$C_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, C_3 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix},$$
$$C_4 = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}, C_5 = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}.$$

References

Bhat, C. (2001), 'Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model', *Transportation Research B* **35**, 677–693.

Bhat, C. (forthcoming), 'Simulation estimation of mixed discrete choice models using randomized and scrambled halton sequences', *Transportation Research* .

Bratley, P., B.L. Fox and H. Niederreiter (1992), 'Implementation and tests of low-discrepancy sequences', *ACM Transactions on Modeling and Computer Simulation* **2**, 195–213.

Brownstone, D. and K. Train (1999), 'Forecasting new product penetration with flexible substitution patterns', *Journal of Econometrics* **89**, 109–129.

Lee, L. (1995), 'Asymptotic bias in simulated maximum likelihood estimation of discrete choice models', *Econometric Theory* **11**, 437–483.

McFadden, D. (1989), 'A method of simulated moments for estimation of discrete response models without numerical integration', *Econometrica* **57**, 995–1026.

Niederreiter, H. (1988), 'Low-discrepancy and low dispersion sequences', *Journal of Number Theory* **30**, 51–70.

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, Philadelphia: SIAM.

Niederreiter, H. and C. Xing (1996), 'Low-discrepancy sequences and global function fields with many rational places', *Finite Fields and their Applications* **2**, 241–273.

Owen, A. (1995), Randomly permuted (t, m, s) -nets and (t,s) - sequences, *in* H.Niederreiter and P.Shiue, eds, 'Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing', Springer-Verlag, New York, pp. 299–317.

Prisic, G. (forthcoming), 'A software implementation of Niederreiter-Xing sequences', Proceedings of MCQMC.

Revelt, D. and K. Train (1998), 'Mixed logit with repeated choices', *Review of Economics and Statistics* **80**, 647–657.

Sándor, Z. and P. András (2001), 'Alternative sampling methods for estimating multivariate normal probabilities', Working Paper, Department of Economics, University of Gronigen, The Netherlands.

Tang, B. (1993), 'Orthogonal array-based Latin hypercubes', *Journal of the American Statistical Association* **88**, 1392–1397.

Train, K. (2000), 'Halton sequences for mixed logit', Working Paper No. E00-278, Department of Economics, University of California, Berkeley.

Train, K. (forthcoming), *Discrete Choice Methods with Simulation*, Cambridge University Press, New York.

Train, K. and K. Hudson (2000), ‘The impact of information in vehicle choice and the demand for electric vehicles in california’, project report, National Economic Research Associates.

Tuffin, B. (1996), ‘On the use of low-discrepancy sequences in monte carlo methods’, *Monte Carlo Methods and Applications* **2**, 295–320.

Wang, X. and F.J. Hickernell (2000), ‘Randomized Halton sequences’, *Mathematical and Computer Modelling* **32**, 887–99.