EFFICIENT LATTICE ASSESSMENT
FOR LCG AND GLP PARAMETER SEARCHES

KARL ENTACHER, THOMAS SCHELL, AND ANDREAS UHL

ABSTRACT. In the present paper we show how to speed up lattice parameter searches for Monte Carlo and quasi-Monte Carlo node sets. The classical measure for such parameter searches is the spectral test which is based on a calculation of the shortest nonzero vector in a lattice. Instead of the shortest vector we apply an approximation given by the LLL algorithm for lattice basis reduction. We empirically demonstrate the speed-up and the quality loss obtained by the LLL reduction, and we present important applications for parameter selections.

1. INTRODUCTION

Quality assessments of integer lattices play an important role in the development of efficient node sets for the approximate calculation of high dimensional integrals using Monte Carlo (MC) and quasi-Monte Carlo (QMC) methods.

Consider the standard domain \( I^s := [0,1]^s \) in dimension \( s \geq 2 \), a point (node) set \( P = \{ \bar{x}_1, \ldots, \bar{x}_N \} \) in \( I^s \), \( N \in \mathbb{N} \), and a function \( f : I^s \rightarrow \mathbb{R} \). The (quasi) Monte Carlo approximation of an integral \( E(f) := \int_I f(x) \, dx \) is computed by the average of the integrand over the point set \( P \),

\[
S_N(f, P) := \frac{1}{N} \sum_{n=1}^{N} f(\bar{x}_n).
\]

Integer lattices with (in a certain sense) optimal resolution or distribution property are classical node sets for QMC integration. Such lattice rules (or lattice methods) yield the approximation error bound \( |E(f) - S_N(f, P)| = O((\log N)^{s-1}/N) \). To obtain this excellent error behavior it is necessary to provide lattices that are optimally chosen with respect to certain measures of uniform distribution \([18, 30, 38]\). An important candidate for such a measure is the spectral test \([12, 19, 24]\), which allows a very efficient and effective quality analysis for lattices up to high dimensions.

Lattice assessments are also used to get reliable linear random number generators to produce node sets for MC integration, the counterpart of QMC. This is due to the fact that different vectors from linear random numbers are contained in lattice...
structures. An analysis of the underlying lattices provides generators with optimal distribution and correlation quality. In comparison to QMC, for the application of random numbers in MC integration, we theoretically may expect an approximation error $O(1/\sqrt{N})$, see [3, 30] for details. The QMC error above is asymptotically better than the MC error, but the latter has the advantage of dimension-independence. However, for explicit applications, the real integration error obviously depends on several additional factors, such as the regularity of the integrand, the constants in the $O$ term, and the dimension $s$ or the maximally computable sample size $N$. For detailed discussions and references on MC and QMC see the monographs [17, 30, 38] and the review articles [3, 34]. For recent results and applications see the series of conference proceedings [32, 31, 33], and also the web-sites http://www.mcqmc.org/ and http://random.mat.sbg.ac.at/links/.

Usually little is known about the regularity of the integrand. Therefore it is important to provide well chosen node sets for MC and QMC. The selection of “good” lattice parameters for the application of lattice rules in QMC, and also for the development of reliable linear random number generators for MC, demands a huge computational effort. It is highly desirable that such lattices should, besides distribution quality, fulfill several additional requirements, such as projection or sub-lattice stability [11, 21, 25, 26]. Even using the fast spectral test it is very hard and sometimes practically impossible to reach all the desired quality requirements [20].

In the present paper we show how to speed up lattice parameter searches for QMC and MC node sets with negligible loss of quality. The spectral test is based on a calculation of the shortest nonzero vector in a lattice [15], which is in general not feasible in polynomial time for increasing dimension. The effort to determine such a shortest vector depends heavily on the given “input” basis of the lattice. In many areas in scientific computing it suffices to apply an approximation of the shortest nonzero vector, which can be obtained by the well known LLL algorithm [6, 27, 35], for example. We apply the latter algorithm to obtain a modified spectral test which is defined by the shortest vector of the LLL reduced lattice basis. The speed-up and the quality loss obtained by the LLL reduction in our application is empirically demonstrated in Section 3. In Section 4 we present important applications for parameter selections. The following section gives introductory notations and concepts.

2. Application and Analysis of Lattices for MC and QMC

2.1. Monte Carlo. Classical node sets for MC methods are obtained by linear congruential random number generators (LCGs). LCGs have been applied extensively for a long time, and they are the most common random number generators. But we have to mention that recent versions use implementations based on a combination of LCGs or multiple recursive generators (MRGs) to get improved quality and huge periods. For many classical and recent examples, see [10, 22]. The definitions and basic properties of linear random number generators are contained in [17, 19, 20, 30]. LCGs are generated by means of the recursion $y_{n+1} \equiv ay_n + b \pmod{m}$, $n \geq 0$, and by an initial seed $y_0$, $a \neq 1$, $b$, $y_0 \in \mathbb{Z}_m$ (the least residue system modulo $m$). Normalized PRNs in $[0,1]$ are obtained by the transformation $x_n := y_n/m$. We consider only multiplicative LCGs ($b = 0$) where the
modulus $m$ is prime and the multiplier $a$ is a primitive root modulo $m$. Therefore the recursion above, with seed $y_0 \neq 0$, produces a sequence of integers in $\mathbb{Z}_m$ with maximal period $m - 1$.

A central property of linear congruential generators in general (this holds also for combined LCGs and for MRGs), is that arbitrary $s$-dimensional vectors

\begin{equation}
\tilde{x}_i := (x_i, x_{i+j_1}, \ldots, x_{i+j_{s-1}}),
\end{equation}

with fixed lags $j_1, \ldots, j_{s-1}$, are contained in certain grid structures [24].

For $j_1 = 1, \ldots, j_{s-1} = s - 1$, the case which has been studied in detail, these vectors are called overlapping $s$-tuples. In our case, for multiplicative LCGs, the latter $s$-tuples produce intersections of a lattice $L_s(a, m)$ with the $s$-dimensional unit cube $I^s$, where

\begin{equation}
L_s(a, m) := \left\{ \sum_{i=1}^{s} k_i \cdot \tilde{b}_i : k_i \in \mathbb{Z} \right\}
\end{equation}

denotes a $s$-dimensional lattice with lattice basis

\begin{equation}
\tilde{b}_1 = (1, a, \ldots, a^{s-1})/m, \tilde{b}_2 = (0, 1, 0, \ldots, 0), \ldots, \tilde{b}_s = (0, 0, \ldots, 0, 1),
\end{equation}

see [15, 19, 20, 29, 30, 36].

In practice, usually nonoverlapping $s$-tuples

\begin{equation}
\tilde{x}_i := (x_{is}, x_{is+1}, \ldots, x_{is+s-1}), \quad i \geq 0,
\end{equation}

are used to produce “independent” random points in $I^s$. If $\gcd(s, m - 1) = 1$, then all possible nonoverlapping tuples $\tilde{x}_i$ originate in the same lattice as above. Note that for the computation of all nonoverlapping $s$-tuples with an LCG one has to generate at least $s$ times the period. If $\gcd(s, m - 1) > 1$, these vectors produce proper subsets of the lattice $L_s(a, N)$ which need not have a pure lattice structure in general [1]. A lattice is obtained if one considers the union of all possible subsets produced by such tuples.

Essentially different lattices $L_s(a', m')$ are obtained for vectors (2.1) with arbitrary lags which occur if, for example, lagged subsequences from the output of an LCG are used [11, 24].

2.2. Quasi-Monte Carlo. A classical method for QMC uses the intersection of the full lattice $L_s(a, m)$, $m = N$, with the $s$-dimensional unit cube $I^s$ as node set for the calculation of the approximation (1.1) of the integral. The application of $L_s(a, m)$ is called the rank-1 lattice rule or the Korobov lattice rule [18, 25, 30, 38]. The quality of $L_s(a, m)$ depends on the generating vector $\tilde{b}_1$ in (2.3). For well chosen vectors $\tilde{b}_1$, the Korobov lattice rule is also called method of “good lattice points” (GLPs). More general lattice rules are for example obtained for different vectors in a basis [24].

\footnote{Note that power-of-two LCGs suffer from strong regularities in the binary representation of the generated numbers, i.e., the least significant bits of these numbers exhibit small periods, see [15]. Hence, such generators should not be used in stochastic simulations.}

\footnote{Note that the difference of MC and QMC in our case lies in the fact that for MC one may use a large modulus $m$ and only a “random” part of the lattice $L_s(a, m)$, while for QMC one may apply the sample size $N$ as modulus and therefore the full lattice $L_s(a, N)$.}
2.3. **Lattice assessment.** The coarseness of the lattice $L_s(a,m)$ may change dramatically if either the dimension $s$ or the multiplier $a$ is varied. To get reliable linear random number generators for a large class of applications, it is necessary to assess the quality of the several lattices produced by various tuple constructions described above. Furthermore, the selection of “good” lattices $L_s(a,m)$ provides excellent QMC node sets as well.

The spectral test (geometric version) is a classical measure for the quality of $s$-dimensional lattices $L_s$. Specifically, this test determines the maximum distance $d_s$ between adjacent hyperplanes, taken over all families of parallel hyperplanes which contain all points of the lattice. The smaller $d_s$, the more regular is the point structure.

Widely used is a normalized spectral test $S_s := d_s^s / d_s^2$, for which $0 \leq S_s \leq 1$ (values near 1 imply a good lattice structure). The constants $d_s$ are absolute lower bounds on $d_s$, see [19, p. 105] and [15, Sect. 7.7]. L’Ecuyer [23] used also certain lower bounds $d_s^*$ for dimensions $s > 8$ in order to compute $S_s$ for arbitrary dimensions.

The algorithm to calculate the spectral test is based on the dual lattice $L^*$ of $L_s$, since the maximal distance between adjacent hyperplanes $d_s$ is equal to the reciprocal of the length of the shortest nonzero vector of the dual lattice [8, 19]. Historically this test is due to Coveyou and MacPherson [7], who used multivariate Fourier analysis to study the quality of LCGs. An efficient implementation of the spectral test for arbitrary multiple recursive generators is given in [24]. A Mathematica package for various spectral test calculations and the C-code of our LLL-spectral test are available from the server [http://www.fh-sbg.ac.at/~entacher/](http://www.fh-sbg.ac.at/~entacher/).

3. **Spectral test approximation with the LLL algorithm**

In this section we want to demonstrate the effects which appear if the shortest nonzero vector in the spectral test calculation is replaced by an approximation obtained by the Lenstra Lenstra Lovász (LLL) basis reduction algorithm [6, 27, 35]. The calculation of the shortest nonzero vectors in a lattice is performed by variants of the Fincke-Pohst algorithm which are in general not polynomial in dimension [14]. Applying the LLL algorithm, an approximation of the shortest vector can be calculated in polynomial time [6, 35, 39]. For recent discussions on the complexity of lattice problems see [2, 4], and also other papers from [9].

For a given basis $B = \{\vec{b}_1, \ldots, \vec{b}_s\}$ of a lattice $L_s$, the LLL algorithm finds a new basis $B' = \{\vec{b}'_1, \ldots, \vec{b}'_s\}$, with

\[
||\vec{b}'_1|| \leq 2^{s-1} \cdot ||\vec{v}||,
\]

where $\vec{b}'_1$ denotes the shortest nonzero vector in $B$, $||.||$ the euclidean norm, and $\vec{v}$ an arbitrary nonzero vector in the lattice $L_s$. Therefore the latter inequality also holds for a shortest vector $\vec{v}_1$ in $L_s$. Cohen [6] quotes: “We see that the vector $\vec{b}'_1$ in a reduced basis is, in a very precise sense, not too far from being the shortest nonzero vector of $L_s$. In fact, it often is the shortest, and when it is not, one can, most of the time, work with $\vec{b}'_1$ instead of the actual shortest vector”. Moreover, Pohst [35] supplements: “Examples show that there exist lattices with LLL reduced bases $\{\vec{b}'_1, \ldots, \vec{b}'_s\}$ with $||\vec{b}'_1||^2 \geq (4/3)^{s-2}||\vec{v}_1||^2$. These observations certainly do

---

For the definition of the dual lattice see subsection [13].

For the definition of the dual lattice see subsection [13].

Further properties of LLL reduced bases are given in [35].
not favour applications of LLL reduced bases. However, the results in practice are in general much better than the worst case estimates.\footnote{We have chosen $M_s$ since the normalization constants $d_s^*$ used for the normalized spectral test are best possible for dimensions $2 \leq s \leq 8$, which is not the case for larger dimensions \cite{22}.}

In the following we demonstrate that the above statements on the good behavior in practice apply also for the spectral test for parameter selection of MC and QMC node sets. Therefore, in most cases, it is sufficient to apply the LLL reduction instead of the calculation of the shortest vector.

Consider our lattices $L_a := L_s(a, p)$, $p$ prime, as defined in Section 2. The LLL spectral test $d_s'$ of $L_s$ will obviously be defined as the reciprocal of the length of the first vector of the LLL reduced dual lattice basis of $L_s$. Similarly to subsection 2.3 we may also use a normalized LLL spectral test $S_s' := d_s^*/d_s'$, $s \geq 2$. Further we will use the notation $M'_j := \min_{2 \leq s \leq j} S_s'$, which in the original form $M_j := \min_{2 \leq s \leq j} S_s$ has often been applied for LCG parameter searches \cite{15, 23}. For fixed prime numbers $p$ we will write $d_s'(a, p)$, $S_s'(a, p)$ and $M'_j(a, p)$ for the corresponding figures of merit for the lattice $L_s(a, p)$.

To exhibit different behavior of the spectral test and its LLL version for our lattices we considered the nearest prime numbers $p_j$ to $2^j$, $9 \leq j \leq 28$, which are for example $\{2^9 - 3, 2^{10} - 3, 2^{11} + 5, \ldots \}$ and, for each of these primes, the set of primitive roots $a$ modulo $p_j$. Note that assessments of lattices $L_s(a, p)$ for MC node set selection are in general restricted to the set $A$ of all primitive roots $a$ modulo $p$ since for such primitive roots the corresponding LCG guarantees maximal period. There are $\phi(p - 1)$ primitive roots where $\phi$ denotes the Euler totient function. Further, there are certain lattices $L_s(a, p)$ for $a \in A$ which are equivalent with respect to the spectral test, and therefore it suffices to assess $L_s(a, p)$ for a number $\phi(p - 1)/2$ of primitive roots $a$ \cite{15, 22}.

Figure 1 shows relative frequencies of the occurrence of different values of $M_s(a, p)$ and $M'_s(a, p)$ (left graphics) and $d_s(a, p)$ and $d'_s(a, p)$, $2 \leq s \leq 8$ (right graphics). For prime numbers $p_j$, $j \leq 18$, we considered all relevant primitive roots modulo $p_j$, and for $p_j$ with $19 \leq j \leq 28$, we have randomly chosen a set of $2^{15}$ primitive roots $a$. The relative number of different values $d_s(a, p)$ and $d'_s(a, p)$, $2 \leq s \leq 8$, is very low (maximum about three percent) but increasing with the dimension. However for the measures $M_s$ and $M'_s$ the frequencies are significantly lower.

In Figure 2 we exhibit the magnitude of the differences of the measures. The left graphic shows the maximal values $\max_a (d_s(a, p)/d'_s(a, p))^2$ for each prime number $p_j$, $12 \leq j \leq 28$, and dimension $2 \leq s \leq 8$. Note that almost all values are between one and two.\footnote{There are some outliers (the largest one equals 7), which for dimension $s = 7$ is still clearly lower than the theoretical bound $2^{(s-1)}$ given in (3.1).} The right graphic shows the mean values of the absolute differences between $S_s(a, p)$ and $S'_s(a, p)$.

From these empirical tests we can confirm the statements of Cohen and Pohst above, i.e., for our applications the vector $\vec{b}_1$ of the LLL reduced dual basis of $L_s(a, b)$ is very often the shortest nonzero vector, and if not the results are much better than the theoretical bounds. Our comparisons have been calculated using a Mathematica implementation of the Fincke-Pohst algorithm by Wilber van der
Figure 1. Relative frequencies of the occurrence of different values of $M_8(a,p)$ and $M_0^8(a,p)$ (left graphics) and $d_s(a,p)$ and $d'_s(a,p)$, $2 \leq s \leq 8$ (right graphics).

Figure 2. The maximal values $\max_a(d_s(a)/d'_s(a))^2$ for each prime number $p_j$, $12 \leq j \leq 28$, and dimension $2 \leq s \leq 8$ (left graphics), and the mean values of the absolute differences of $S_s(a)$ and $S'_s(a)$ (right graphics).

Kallen, University of Utrecht, NL, which is based on a previous LLL reduction. Figure 3 exhibits the time performance of our calculations. The figure displays the mean values of the time used for the calculation of $d_s(a,p)$ divided by the means for $d'_s(a,p)$; the means are taken over 64 different primitive roots $a$ for each prime $p$. The LLL reduction is for small dimensions about a factor 5 faster

[8] Note that for the generation of the timings we also used much larger primes near $2^j$, $16 \leq j \leq 64$, and larger dimensions $s$. 

License or copyright restrictions may apply to redistribution; see http://www.ams.org/journal-terms-of-use
EFFICIENT LATTICE ASSESSMENT FOR PARAMETER SEARCHES

![Image](image.png)

**Figure 3.** Mean values of the time spent for $d_s(a, p)$ divided by the means for $d'_s(a, p)$, the means being taken over 64 different primitive roots $a$ for each prime $p$.

**Table 1.** Multipliers selected via the spectral test from [23] and our results obtained by a random search using LLL reduction.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$a$ [23]</th>
<th>$M_8$ [23]</th>
<th>$a$</th>
<th>$M'_8 = M_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{27} - 39 = 134217689$</td>
<td>45576512</td>
<td>0.75874</td>
<td>45576512</td>
<td>0.75874</td>
</tr>
<tr>
<td>$2^{28} - 57 = 268435399$</td>
<td>150873839</td>
<td>0.74215</td>
<td>150873839</td>
<td>0.74215</td>
</tr>
<tr>
<td>$2^{29} - 3 = 536870909$</td>
<td>520332806</td>
<td>0.75238</td>
<td>435136037</td>
<td>0.75356</td>
</tr>
<tr>
<td>$2^{30} - 35 = 1073741789$</td>
<td>771645345</td>
<td>0.74881</td>
<td>325079677</td>
<td>0.75432</td>
</tr>
<tr>
<td>$2^{31} - 1 = 2147483647$</td>
<td>1583458089</td>
<td>0.72771</td>
<td>981753959</td>
<td>0.73435</td>
</tr>
<tr>
<td>$2^{32} - 5 = 4294967291$</td>
<td>1588635695</td>
<td>0.74530</td>
<td>3265168268</td>
<td>0.74870</td>
</tr>
<tr>
<td>$2^{33} - 9 = 8589934583$</td>
<td>7425194315</td>
<td>0.73666</td>
<td>8137022074</td>
<td>0.75316</td>
</tr>
<tr>
<td>$2^{34} - 41 = 17179869143$</td>
<td>5295517759</td>
<td>0.73607</td>
<td>10771374442</td>
<td>0.73899</td>
</tr>
<tr>
<td>$2^{35} - 31 = 34359738337$</td>
<td>3124199165</td>
<td>0.74740</td>
<td>23314821278</td>
<td>0.75022</td>
</tr>
<tr>
<td>$2^{36} - 5 = 68719476731$</td>
<td>49865143810</td>
<td>0.72011</td>
<td>24365995562</td>
<td>0.75969</td>
</tr>
<tr>
<td>$2^{37} - 25 = 137438953447$</td>
<td>76886758244</td>
<td>0.73284</td>
<td>64192466011</td>
<td>0.73997</td>
</tr>
</tbody>
</table>

than the calculation of the shortest vector, but the speed-up in our Mathematica implementation decreases for increasing dimension and prime size.

We further used Victor Shoup’s C++ implementation of the LLL algorithm [37] and performed the same parameter searches for optimal multipliers with respect to $M_8$ which were carried out in [23, Table 2]. For the exhaustive searches in the latter paper which were computed for prime numbers $p \in \{2^8 - 5, 2^9 - 3, 2^{10} - 3, \ldots, 2^{26} - 5\}$ we got exactly the same multipliers with respect to $M'_8$ and equal values for the measures $M'_8$ and $M_8$. For the random searches for primes
\( p \in \{2^{27} - 39, 2^{28} - 57, \ldots, 2^{37} - 25\} \), for example, we easily obtained improved results with our algorithm. Table \( \text{I} \) shows the results given in [23, Table 2] and our multipliers selected with LLL for the latter primes.

4. Applications

4.1. Projection stable lattice rules. In their recent papers [25, 26], C. Lemieux and P. L’Ecuyer showed that for high quality lattice rules, it is not enough to assess only the underlying lattice \( L_s \). For certain subsets of coordinates \( I = \{i_1, \ldots, i_t\} \subset \{1, \ldots, s\} \), the projection \( L_s(I) \) of \( L_s \) over the \( t \)-dimensional subspace determined by \( I \) should be assessed as well, for many applications. Lattice rules \( L_s(a, p) \) which for example have been selected via \( M_k \) may have poor projection quality, for examples see the papers above. Selection of lattice rules including a quality assessment of a large set of projections may be computationally very expensive. To provide a compromise between computational cost and projection quality Lemieux and L’Ecuyer [26] proposed the following worst case figure of merit for lattice assessments:

\[
M_{j,k} := \min \left[ \min_{2 \leq s \leq j} \frac{d_s}{d_{s(k)}}, \min_{I \in S(k)} \frac{d_I}{d_I} \right],
\]

where \( S(k) := \{I = \{i_1, \ldots, i_t\} : |I| \leq k, i_1 = 1 \text{ and } i_t \leq j\} \). For \( k = 1 \) one has \( M_{j,k} = M_j \). The term \( d_I \) in the definition above denotes the spectral test for the projection \( L_s(I) \), which, in analogy to subsection 2.3, is the reciprocal of the length of a shortest nonzero vector of the dual lattice \( L_s^*(I) \). The dual lattice of \( L_s(I) \) is defined as \( L_s^*(I) := \{\tilde{w} \in \mathbb{R}^s : \tilde{w} \cdot \tilde{v} \in \mathbb{Z} \text{ for all } \tilde{v} \in L_s(I)\} \). The dual of a given lattice basis \( B = \{\tilde{b}_1, \ldots, \tilde{b}_s\} \) is provided by the set of vectors \( B^* = \{\tilde{b}_1, \ldots, \tilde{b}_s^*\} \) such that \( \tilde{b}_i \cdot \tilde{b}_j^* = \delta_{i,j}, \) with \( \delta_{i,j} = 1 \) if \( i = j \) and \( \delta_{i,j} = 0 \) otherwise.

Even for the latter measure it is hard to perform searches for optimal multipliers for reasonable \( p, j \) and \( k \). We applied the measure \( M_{j,k} \) in its faster LLL version \( M'_{j,k} \), which means that the shortest vectors are replaced by their approximations given by the LLL reduction. Table [2] shows examples of exhaustive search results for the best multipliers \( a \) that are primitive roots modulo \( p \), for a given prime \( p \) with \( j = 8 \) and \( k \leq 4 \) and \( j = 16 \) and \( k \leq 8 \). The timings (\(~14 \text{ days of CPU-time}) in the table were achieved using a SGI Power Challenge, equipped with 20 MIPS R10000 processors at 194 MHz running IRIX 6.5., and our LLL implementation [37]. Note that the magnitude of our prime numbers \( p \) is much larger in comparison to the results in [25] [26] (the latter authors used primes lower than \( 2^{17} \)). We also carried out the same search as in [26, Table 1] and verified the results given there with our measure \( M'_{j,k} \).

4.2. Selection of LCG parameters with subsequence stability. Another computationally expensive application of lattice assessments appears for parameter selections of linear random number generators with splitting stability. For several applications of random numbers it is common practice to split the output of an RNG into subsequences. Such subsequences may occur in special simulation setups, in transformation methods for nonuniform random numbers [28] or as

\footnote{Note that for Korobov lattice rules the projections are lattices as well. Moreover, each projection contains the same number of points as the lattice itself (if \( \text{gcd}(a, p) = 1 \), and for special sets of indices \( I \) such projections are identical. The latter properties are called projection-regularity and dimension-stationarity, for details see [25, 26].}
Table 2. Exhaustive search results for optimal multiplier $a$ with respect to the figures of merit $M_{16,k}^t$, $k = 1, 2, 4, 8$, and $M_{8,k}^t$, $k = 1, 2, 3, 4$.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$k$</th>
<th>$a$</th>
<th>$M_{16,k}^t$</th>
<th>Time [s]</th>
<th>$k$</th>
<th>$a$</th>
<th>$M_{8,k}^t$</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>217</td>
<td>1</td>
<td>29223</td>
<td>0.67170</td>
<td>189</td>
<td>1</td>
<td>43165</td>
<td>0.70941</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>22865</td>
<td>0.58550</td>
<td>190</td>
<td>2</td>
<td>52344</td>
<td>0.66695</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>73088</td>
<td>0.24606</td>
<td>248</td>
<td>3</td>
<td>38429</td>
<td>0.47986</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>55810</td>
<td>0.24206</td>
<td>1292</td>
<td>4</td>
<td>9290</td>
<td>0.41693</td>
<td>25</td>
</tr>
<tr>
<td>218</td>
<td>1</td>
<td>195669</td>
<td>0.69174</td>
<td>766</td>
<td>1</td>
<td>166972</td>
<td>0.72539</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>107017</td>
<td>0.59653</td>
<td>767</td>
<td>2</td>
<td>134632</td>
<td>0.70916</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>81970</td>
<td>0.25478</td>
<td>957</td>
<td>3</td>
<td>72153</td>
<td>0.50017</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>203909</td>
<td>0.24084</td>
<td>3656</td>
<td>4</td>
<td>79719</td>
<td>0.43179</td>
<td>99</td>
</tr>
<tr>
<td>219</td>
<td>1</td>
<td>157781</td>
<td>0.67421</td>
<td>842</td>
<td>1</td>
<td>6371</td>
<td>0.72493</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>303495</td>
<td>0.58879</td>
<td>842</td>
<td>2</td>
<td>308445</td>
<td>0.67948</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>117294</td>
<td>0.25579</td>
<td>1050</td>
<td>3</td>
<td>429897</td>
<td>0.53103</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>503284</td>
<td>0.23799</td>
<td>4355</td>
<td>4</td>
<td>195267</td>
<td>0.42274</td>
<td>108</td>
</tr>
<tr>
<td>220</td>
<td>1</td>
<td>246298</td>
<td>0.66738</td>
<td>1728</td>
<td>1</td>
<td>380985</td>
<td>0.71807</td>
<td>192</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>795969</td>
<td>0.62016</td>
<td>1746</td>
<td>2</td>
<td>118096</td>
<td>0.67348</td>
<td>192</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>462565</td>
<td>0.25738</td>
<td>2100</td>
<td>3</td>
<td>325952</td>
<td>0.50054</td>
<td>199</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>50992</td>
<td>0.23811</td>
<td>5877</td>
<td>4</td>
<td>1026371</td>
<td>0.42291</td>
<td>218</td>
</tr>
<tr>
<td>221</td>
<td>1</td>
<td>1043187</td>
<td>0.68608</td>
<td>6767</td>
<td>1</td>
<td>360889</td>
<td>0.72537</td>
<td>733</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>787493</td>
<td>0.62991</td>
<td>6768</td>
<td>2</td>
<td>109078</td>
<td>0.68149</td>
<td>734</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>847810</td>
<td>0.26608</td>
<td>7979</td>
<td>3</td>
<td>1518745</td>
<td>0.54932</td>
<td>748</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>585470</td>
<td>0.24606</td>
<td>20359</td>
<td>4</td>
<td>1236628</td>
<td>0.46299</td>
<td>818</td>
</tr>
<tr>
<td>222</td>
<td>1</td>
<td>2040406</td>
<td>0.67819</td>
<td>6369</td>
<td>1</td>
<td>1406151</td>
<td>0.72226</td>
<td>693</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2088505</td>
<td>0.61795</td>
<td>6375</td>
<td>2</td>
<td>3006043</td>
<td>0.67819</td>
<td>693</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>408602</td>
<td>0.26711</td>
<td>7382</td>
<td>3</td>
<td>4036460</td>
<td>0.53257</td>
<td>709</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3728072</td>
<td>0.23208</td>
<td>21615</td>
<td>4</td>
<td>2059718</td>
<td>0.43735</td>
<td>771</td>
</tr>
<tr>
<td>223</td>
<td>1</td>
<td>3523955</td>
<td>0.69515</td>
<td>28804</td>
<td>1</td>
<td>653276</td>
<td>0.73407</td>
<td>3087</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6033416</td>
<td>0.62230</td>
<td>28823</td>
<td>2</td>
<td>4725434</td>
<td>0.68340</td>
<td>3096</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>7288204</td>
<td>0.25817</td>
<td>33510</td>
<td>3</td>
<td>1235903</td>
<td>0.53696</td>
<td>3145</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>516463</td>
<td>0.23541</td>
<td>53216</td>
<td>4</td>
<td>4071685</td>
<td>0.46114</td>
<td>3382</td>
</tr>
<tr>
<td>224</td>
<td>1</td>
<td>9939730</td>
<td>0.67283</td>
<td>37418</td>
<td>1</td>
<td>10354078</td>
<td>0.74477</td>
<td>3991</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12056378</td>
<td>0.62017</td>
<td>37452</td>
<td>2</td>
<td>4676419</td>
<td>0.70711</td>
<td>3986</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>8036898</td>
<td>0.25713</td>
<td>43349</td>
<td>3</td>
<td>9820243</td>
<td>0.54899</td>
<td>4027</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8226178</td>
<td>0.23096</td>
<td>106800</td>
<td>4</td>
<td>7799995</td>
<td>0.45628</td>
<td>4341</td>
</tr>
<tr>
<td>225</td>
<td>1</td>
<td>31482291</td>
<td>0.68815</td>
<td>116122</td>
<td>1</td>
<td>22119140</td>
<td>0.76177</td>
<td>12332</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>14199480</td>
<td>0.62548</td>
<td>116177</td>
<td>2</td>
<td>26579378</td>
<td>0.71883</td>
<td>12342</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1330563</td>
<td>0.26029</td>
<td>133163</td>
<td>3</td>
<td>26593451</td>
<td>0.58448</td>
<td>12442</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>21798615</td>
<td>0.23281</td>
<td>302623</td>
<td>4</td>
<td>6529775</td>
<td>0.46849</td>
<td>13293</td>
</tr>
</tbody>
</table>

methods to obtain parallel streams of pseudorandom numbers for parallel and distributed simulation [5, 11, 20]. In the case of linear RNGs it frequently occurs that the distribution quality of subsequences is very bad [10, 13, 16, 21]. The latter is due to the fact that overlapping or nonoverlapping $s$-dimensional vectors generated from a single subsequence stream from an LCG are contained in different lattices. Consider for example an LCG with modulus $p$ and multiplier $a$. An analysis of the
Table 3. Random search for LCG parameters $a$, $p$ with subsequence quality.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$a$</th>
<th>$M'$</th>
<th>Time [h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{61} - 1$</td>
<td>576499412011439685</td>
<td>0.300603</td>
<td></td>
</tr>
<tr>
<td></td>
<td>147053512213743586</td>
<td>0.310978</td>
<td></td>
</tr>
<tr>
<td></td>
<td>273635001254884560</td>
<td>0.313225</td>
<td>670</td>
</tr>
<tr>
<td>$2^{64} - 59$</td>
<td>1812254979190146906</td>
<td>0.295821</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12392816392420730400</td>
<td>0.297236</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3401048658419419339</td>
<td>0.316644</td>
<td>1003</td>
</tr>
</tbody>
</table>

underlying lattice $L_s(a, p)$, for certain dimensions $s$, provides a distribution and correlation assessment of the generator’s output stream.

If we also want to assess correlations within and between leaped-subsequences of the generator’s output with step size $k \geq 2$, then we also have to analyze the lattices $L_s(a^k \pmod{p}, p)$; for details see [11, 21]. If one, for example, wants to provide LCG parameters which are also well chosen with respect to subsequence behavior for several step sizes $k$, then a large set of lattices has to be assessed, which obviously requires a considerable computational effort.

As an example, we carried out a random search for multipliers $a$ where the parameters $a$ itself were assessed using the lattice $L_s(a, p)$ and measure $M'_{2^4}(a, p)$ and additionally the subsequence-lattices $L_s(a^k \pmod{p}, p)$ were analyzed using $M'(k) := M'_s(a^k \pmod{p}, p)$ for $2 \leq k \leq 32$, and also for $k \in \{2^j : 6 \leq j \leq 37\}$. Therefore we searched for parameters of LCGs with reliable subsequence behavior for several step-sizes $k$. The random search was carried out using the minimum $M' := \min_k M'(k)$ as quality criterion, for prime numbers $p_1 = 2^{61} - 1$ and $p_2 = 2^{64} - 59$. Table 3 reports the best three multipliers $a$ found for each prime. The CPU-hours given in the table were spent for the entire searches using the SGI Power Challenge mentioned before (18 processors used for $p_1$ and 19 for $p_2$).

5. Conclusion

The spectral test is a classical measure for the assessment of lattices as node sets for Monte Carlo and also for quasi-Monte Carlo methods. The latter test is based on the calculation of the shortest nonzero vector in a lattice, which is carried out by the Fincke-Pohst algorithm. Selection of reliable parameters of such lattices using the spectral test requires considerable computational effort. Therefore, we suggest speeding up such parameter searches by replacing the calculation of the shortest vector by an approximation provided by the well known LLL algorithm for lattice basis reduction, a strategy which is successfully applied in many other areas in scientific computing. We empirically demonstrate that for our applications the LLL algorithm in most of the cases already yields the shortest nonzero vector and, if not, the approximation quality is much better than the theoretical bounds. Perhaps it may be possible to give a theoretical verification of our empirical findings for the special form of our lattices, but this will be a subject for further investigation. We applied the LLL spectral test for important parameter selection strategies and therefore demonstrated the power of our approach.
REFERENCES


School of Telecommunications Engineering, University of Applied Sciences and Technologies, Schillerstr. 30, A-5020 Salzburg, Austria
E-mail address: Karl.Entacher@fh-sbg.ac.at
URL: http://www.fh-sbg.ac.at/~entacher

Department of Scientific Computing, Salzburg University, Hellbrunnerstr. 34, A-5020 Salzburg, Austria

Department of Scientific Computing, Salzburg University, Hellbrunnerstr. 34, A-5020 Salzburg, Austria