MATHEMATICS OF COMPUTATION
Volume 71, Number 239, Pages 1231–1242
S 0025-5718(01)01415-6
Article electronically published on December 21, 2001

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 **M**onte **C**arlo (MC) and **q**uasi–**M**onte **C**arlo (QMC) methods.

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

$$S_N(f,P) := \frac{1}{N} \sum_{n=1}^{N} f(\vec{x}_n). \tag{1.1}$$

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

Received by the editor September 15, 2000.
2000 *Mathematics Subject Classification.* Primary 11Y40, 11-04; Secondary 11K45, 68W40.
Key words and phrases. Monte Carlo and quasi–Monte Carlo methods, lattice rules, good lattice points, random number generation, lattice basis reduction, LLL algorithm, Fincke-Pohst algorithm, spectral test.
The first author was supported by the Austrian Science Fund (FWF), pro. no. S8303-MAT, the second author by the FWF pro. no. P13732-MAT.

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 [26].

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 [19], 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 **l**inear **c**ongruential random number **g**enerators (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 **m**ultiple **r**ecursive **g**enerators (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 [15, 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[1] 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

$$\vec{x}_i := (x_i, x_{i+j_1}, \ldots, x_{i+j_{s-1}}), \tag{2.1}$$

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

$$L_s(a,m) := \left\{ \sum_{i=1}^{s} k_i \cdot \vec{b}_i \ : \ k_i \in \mathbb{Z} \right\} \tag{2.2}$$

denotes a s-dimensional lattice with lattice basis

$$\vec{b}_1 = (1, a, \ldots, a^{s-1})/m,\ \vec{b}_2 = (0,1,0,\ldots,0), \ldots, \vec{b}_s = (0,0,\ldots,0,1), \tag{2.3}$$

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

In practice, usually nonoverlapping s-tuples

$$\vec{x}_i := (x_{is}, x_{is+1}, \ldots, x_{is+s-1}), \quad i \geq 0, \tag{2.4}$$

are used to produce "independent" random points in I^s. If $gcd(s, m-1) = 1$, then all possible nonoverlapping tuples $\vec{x}_i$ 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[2] 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 $\vec{b}_1$ in (2.3). For well chosen vectors $\vec{b}_1$, the Korobov lattice rule is also called method of "**g**ood **l**attice **p**oints" (GLPs). More general lattice rules are for example obtained for different vectors in a basis (2.3).

---

[1]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 [19]. Hence, such generators should not be used in stochastic simulations.

[2]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 $S_s := d_s^*/d_s$, $2 \leq s \leq 8$, 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*[3] 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/`.

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[4]

$$||\vec{b}'_1|| \leq 2^{\frac{s-1}{2}} \cdot ||\vec{v}||, \tag{3.1}$$

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*

[3]For the definition of the dual lattice see subsection 4.1.

[4]Further properties of LLL reduced bases are given in [6].

not favour applications of LLL reduced bases. However, the results in practice are in general much better than the worst case estimates."

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 $L_s := L_s(a,p)$, 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 [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 $\{2^9-3, 2^{10}-3, 2^{11}+5, \dots\}$ 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 primitve roots the corresponding LCG guarantees maximal period. There are $\phi(p-1)$ primitive roots where ϕ 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 [15, 23].

Figure 1 shows relative frequencies of the occurrence of different values of $M_8(a,p)$ and $M'_8(a,p)$ (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[5] M_8 and M'_8 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.[6] 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 Wilberd van der

[5]We have chosen M_8 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 [23].

[6]For dimension $s = 2$ all values are equal to one, which means that in all cases in dimension two the LLL reduction already provides the shortest vector, i.e., $d_s = d'_s$. The latter property can also be seen from the right graphics in Figures 1 and 2.

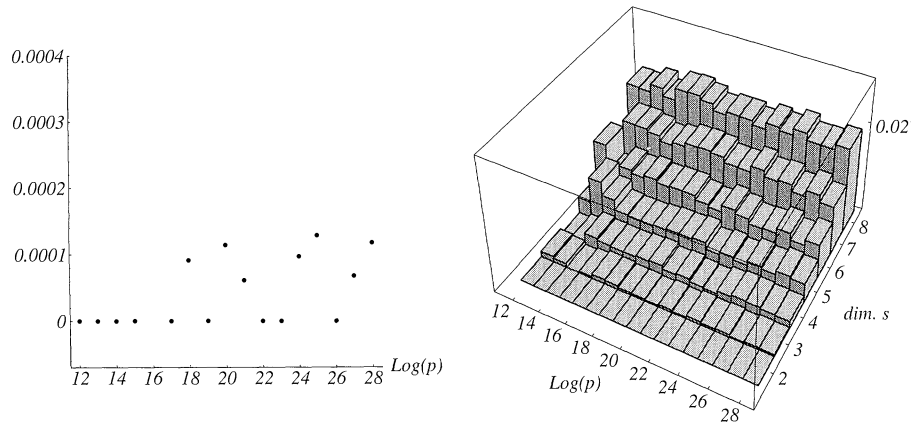

FIGURE 1. Relative frequencies of the occurrence of different values of $M_8(a,p)$ and $M'_8(a,p)$ (left graphics) and $d_s(a,p)$ and $d'_s(a,p)$, $2 \le s \le 8$ (right graphics).

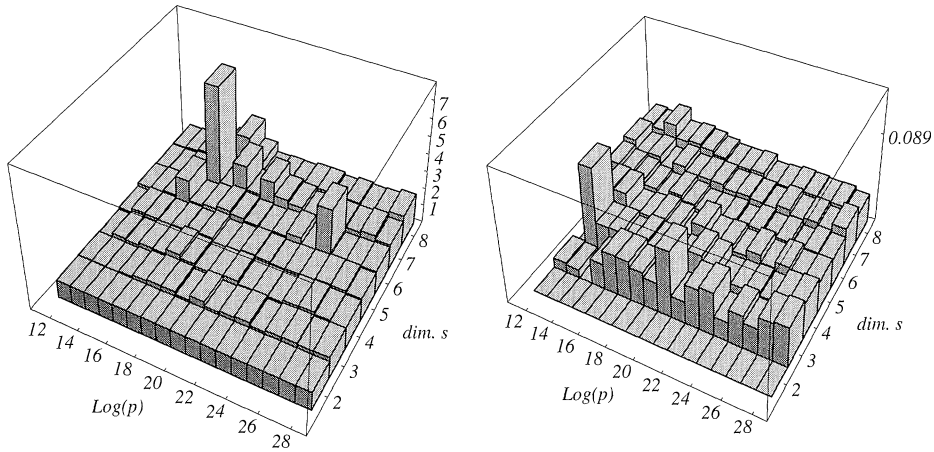

FIGURE 2. The maximal values $\max_a(d_s(a)/d'_s(a))^2$ for each prime number p_j, $12 \le j \le 28$, and dimension $2 \le s \le 8$ (left graphics), and the mean values of the absolute differences of $S_s(a)$ and $S'_s(a)$ (right graphics).

Kallen, University of Utrecht, NL,[7] which is based on a previous LLL reduction. Figure 3 exhibits the time performance of our calculations. The figure displays[8] 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

[7] `http://www.math.ruu.nl/people/vdkallen/kallen.html`.

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

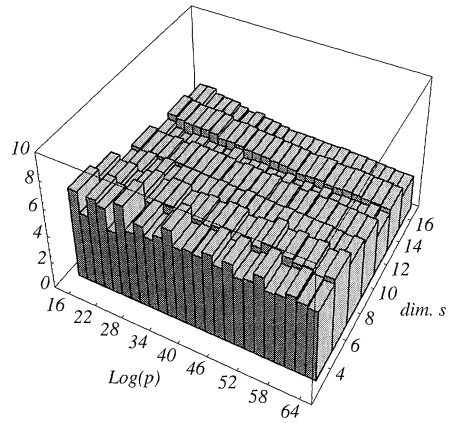

FIGURE 3. Mean values of the time spent for $d_s(a,p)$ divided by the means for $d'_s(a,p)$, the means being taken over 64 different primitive roots a for each prime p.

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

p	a [23]	M_8 [23]	a	$M'_8 = M_8$
$2^{27} - 39 = 134217689$	45576512	0.75874	45576512	0.75874
$2^{28} - 57 = 268435399$	150873839	0.74215	150873839	0.74215
$2^{29} - 3 = 536870909$	520332806	0.75238	435136037	0.75356
$2^{30} - 35 = 1073741789$	771645345	0.74881	325079677	0.75432
$2^{31} - 1 = 2147483647$	1583458089	0.72771	598753959	0.73435
			117879879	0.74309
			629824009	0.74880
			1355089539	0.74972
$2^{32} - 5 = 4294967291$	1588635695	0.74530	3265168268	0.74870
$2^{33} - 9 = 8589934583$	7425194315	0.73666	8137022074	0.75316
$2^{34} - 41 = 17179869143$	5295517759	0.73607	10771374442	0.73899
			1491142424	0.75157
$2^{35} - 31 = 34359738337$	3124199165	0.74740	23314821278	0.75022
$2^{36} - 5 = 68719476731$	49865143810	0.72011	24365995562	0.75969
			46865245638	0.76825
$2^{37} - 25 = 137438953447$	76886758244	0.73284	64192466011	0.73997

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 1 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 projections[9] $L_s(I)$ 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_8 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:

$$M_{j,k} := \min \left[\min_{2 \le s \le j} \frac{d_s^*}{d_s}, \min_{I \in S(k)} \frac{d_{|I|}^*}{d_I} \right], \tag{4.1}$$

where $S(k) := \{I = \{i_1, \ldots i_t\} \ : \ |I| \le k,\ i_1 = 1 \text{ and } i_t \le j\}$. 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 $L_s^*(I) := \{\vec{w} \in \mathbb{R}^s \ : \ \vec{w} \cdot \vec{v} \in \mathbb{Z} \text{ for all } \vec{v} \in L_s(I)\}$. The dual of a given lattice basis $B = \{\vec{b}_1, \ldots, \vec{b}_s\}$ is provided by the set of vectors $B^* = \{\vec{b}_1^*, \ldots, \vec{b}_s^*\}$ such that $\vec{b}_i \cdot \vec{b}_j^* = \delta_{i,j}$, 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 $k \le 4$ and $j = 16$ and $k \le 8$. The timings ($\sim$ 14 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 $M'_{8,k}$.

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

[9] 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 $\gcd(a, p) = 1$), 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 $M'_{16,k}$, $k = 1, 2, 4, 8$, and $M'_{8,k}$, $k = 1, 2, 3, 4$.

p	k	a	$M'_{16,k}$	Time [s]	k	a	$M'_{8,k}$	Time [s]
$2^{17}-1$	1	29223	0.67170	189	1	43165	0.70941	21
	2	22865	0.58550	190	2	52344	0.66695	21
	4	73088	0.24606	248	3	38429	0.47986	23
	8	55810	0.24206	1292	4	9290	0.41693	25
$2^{18}-5$	1	195669	0.69174	766	1	166972	0.72539	85
	2	107017	0.59653	767	2	134632	0.70916	85
	4	81970	0.25478	957	3	72153	0.50017	89
	8	203909	0.24084	3656	4	79719	0.43179	99
$2^{19}-1$	1	157781	0.67421	842	1	6371	0.72493	93
	2	303495	0.58879	841	2	308445	0.67948	92
	4	117294	0.25579	1050	3	429897	0.53103	95
	8	503284	0.23799	4355	4	195267	0.42274	108
$2^{20}-3$	1	246298	0.66738	1728	1	380985	0.71807	192
	2	795969	0.62016	1746	2	118096	0.67348	192
	4	462565	0.25738	2100	3	325952	0.50054	199
	8	50992	0.23811	5877	4	1026371	0.42291	218
$2^{21}-9$	1	1043187	0.68608	6767	1	360889	0.72537	733
	2	787493	0.62991	6768	2	109078	0.68149	734
	4	847810	0.26608	7979	3	1518745	0.54932	748
	8	585470	0.24606	20359	4	1236628	0.46299	818
$2^{22}-3$	1	2040406	0.67819	6369	1	1406151	0.72226	693
	2	2088505	0.61795	6375	2	3060643	0.67819	693
	4	408602	0.26711	7382	3	4036460	0.53257	709
	8	3728072	0.23208	21615	4	2059718	0.43735	771
$2^{23}-15$	1	3523955	0.69515	28804	1	653276	0.73407	3087
	2	6033416	0.62230	28823	2	4725434	0.68340	3096
	4	7288204	0.25817	33510	3	1235903	0.53696	3145
	8	516463	0.23541	53216	4	4071685	0.46114	3382
$2^{24}-3$	1	9939730	0.67283	37418	1	10354078	0.74477	3991
	2	12056378	0.62017	37452	2	4676419	0.70711	3986
	4	8036898	0.25713	43349	3	9820243	0.54899	4027
	8	8226178	0.23096	106800	4	7799995	0.45628	4341
$2^{25}-39$	1	31482291	0.68815	116122	1	22119140	0.76177	12332
	2	14199480	0.62548	116177	2	26579378	0.71883	12342
	4	13305631	0.26029	133163	3	26593451	0.58448	12442
	8	21798615	0.23281	302623	4	6529775	0.46849	13203

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.

p	a	M'	Time [h]
$2^{61}-1$	576499412011439685	0.300603	
	1470535122213743586	0.310978	
	273635001254884560	0.313225	670
$2^{64}-59$	1812254979190146906	0.295821	
	12392816392420730400	0.297236	
	3401048658419419339	0.316644	1003

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 $M'_{24}(a,p)$ and additionally the subsequence-lattices $L_s(a^k \pmod p, p)$ were analyzed using $M'(k) := M'_8(a^k \pmod p, p)$ 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

[1] L. Afflerbach, *The sub-lattice structure of linear congruential random number generators*, Manuscripta Mathematica **55** (1986), 455–465. MR **87k:**65006

[2] J Ajtai, *Generating Hard Instances of Lattice Problems*, Report TR96-007, Electronic Colloquium on Computational Complexity ECCC, 1996, Web-page: [9].

[3] R.E. Caflisch, *Monte Carlo and quasi–Monte Carlo methods*, Acta Numer. **7** (1998), 1–49. MR **2000e:**65019

[4] J Cai, *Some Recent Progress on the Complexity of Lattice Problems*, Report TR99-006, Electronic Colloquium on Computational Complexity ECCC, 1999, Web-page: [9].

[5] P. Coddington, *Random Number Generators for Parallel Computers*, NHSE Review, Second Issue, Northeast Parallel Architectures Center, 1996, Available at: `http://nhse.cs.rice.edu/NHSEreview/RNG/`.

[6] H. Cohen, *A Course in Computational Algebraic Number Theory*, Graduate Texts in Mathematics, vol. 138, Springer-Verlag, 1993. MR **94i:**11105

[7] R.R. Coveyou and R.D. MacPherson, *Fourier analysis of uniform random number generators*, J. Assoc. Comput. Mach. **14** (1967), 100–119. MR **36:**4779

[8] U. Dieter, *How to Calculate Shortest Vectors in a Lattice*, Math. Comp. **29** (1975), no. 131, 827–833. MR **52:**291

[9] ECCC, *Electronic Colloquium on Computational Complexity ECCC*, `http://www.eccc.uni-trier.de/eccc/`.

[10] K. Entacher, *A collection of selected pseudorandom number generators with linear structures – advanced version*, Tech. report, Dept. of Mathematics, University Salzburg, Austria, available at: `http://www.fh-sbg.ac.at/~entacher`, 1998, The previous version is published as technical report 97-1, ACPC–Austrian Center for Parallel Computation, University of Vienna, Austria, 1997.

[11] ———, *Parallel Streams of Linear Random Numbers in the Spectral Test*, ACM Transactions on Modeling and Computer Simulation **9** (1999), no. 1, 31–44.

[12] K. Entacher, P. Hellekalek, and P. L'Ecuyer, *Quasi–Monte Carlo Node Sets from Linear Congruential Generators*, Monte Carlo and Quasi-Monte Carlo Methods 1998 (H. Niederreiter and J. Spanier, eds.), Springer, 2000, pp. 188–198.

[13] K. Entacher, A. Uhl, and S. Wegenkittl, *Linear Congruential Generators for Parallel Monte-Carlo: the Leap-Frog Case.*, Monte Carlo Methods and Appl. **4** (1998), no. 1, 1–16. CMP 98:12

[14] U. Fincke and M. Pohst, *Improved methods for calculating vectors of short length in a lattice, including a complexity analysis*, Math. Comp. **44** (1985), 463–471. MR **86e:**11050

[15] G.S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer Series in Operations Research, vol. 1, Springer, New York, 1996. MR **97g:**65019

[16] P. Hellekalek, *Don't Trust Parallel Monte Carlo*, Twelfth Workshop on Parallel and Distributed Simultation `PADS'98`, May 26th - 29th (Banff, Alberta, Canada), IEEE Computer Society, Los Alamitos, California, 1998, pp. 82–89.

[17] P. Hellekalek and G. Larcher (eds.), *Random and quasi-random point sets*, Lecture Notes in Statistics, vol. **138**, Springer, Berlin, 1998. MR **99g:**11003

[18] F.J. Hickernell, *Lattice Rules: How Well Do They Measure Up*? In [17], pp. 109–166. MR **2000b:**65007

[19] D.E. Knuth, *The art of computer programming*, 2nd ed., vol. 2: Seminumerical Algorithms, Addison-Wesley, Reading, MA, 1981. MR **83i:**68003

[20] P. L'Ecuyer, *Uniform random number generation*, Ann. Oper. Res. **53** (1994), 77–120. MR **95k:**65007

[21] ———, *Bad Lattice Structures for Vectors of Non-Successive Values Produced by Some Linear Recurrences*, INFORMS Journal on Computing **9** (1997), 57–60. MR **98f:**65014

[22] ———, *Random Number Generation*, Handbook of Simulation, Chapter 4 (Jerry Banks, ed.), Wiley, 1998.

[23] ———, *Tables of Linear Congruential Generators of Different Sizes and Good Lattice Structure*, Mathematics of Computation **68** (1999), no. 225, 249–260. MR **99c:**11101

[24] P. L'Ecuyer and R. Couture, *An Implementation of the Lattice and Spectral Tests for Multiple Recursive Linear Random Number Generators*, INFORMS Journal on Computing **9** (1997), no. 2, 209–217. CMP 98:03

[25] P. L'Ecuyer and C. Lemieux, *Variance Reduction via Lattice Rules*, Management Science **46** (2000), no. 9, 1214–1235.
[26] C. Lemieux and P. L'Ecuyer, *On selection criteria for lattice rules and other quasi–Monte Carlo point sets*, Mathematics and Computers in Simulation **55** (2001), 139–148. CMP 2001:11
[27] A.K. Lenstra, H.W. Lenstra, and L. Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen **261** (1982), 515–534. MR **84a:**12002
[28] J. Leydold, H. Leeb, and W. Hörmann, *Higher-Dimensional Properties of Non-Uniform Pseudo-Random Variates*, In [33], pp. 341–355.
[29] G. Marsaglia, *The structure of linear congruential sequences*, Applications of Number Theory to Numerical Analysis (S. K. Zaremba, ed.), Academic Press, New York, 1972, pp. 248–285. MR **53:**14854
[30] H. Niederreiter, *Random number generation and quasi–Monte Carlo methods*, SIAM, Philadelphia, 1992. MR **93h:**65008
[31] H. Niederreiter, P. Hellekalek, G. Larcher, and P. Zinterhof (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 1996*, Lecture Notes in Statistics, vol. 127, Springer, New York, 1998. MR **99d:**65005
[32] H. Niederreiter and P.J.-S. Shiue (eds.), *Monte Carlo and Quasi-Monte Carlo Methods in scientific computing*, Lecture Notes in Statistics, vol. 106, Springer, New York, 1995. MR **97j:**65002
[33] H. Niederreiter and J. Spanier (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 1998*, Springer, Berlin, 2000.
[34] A.B. Owen, *Monte Carlo extension of quasi–Monte Carlo*, Proceedings of the 1998 Winter Simulation Conference (D.J. Madeiros, E.F. Watson, J.S. Carson, and M.S. Manivannan, eds.), 1998, Available from `http://www.informs-cs.org/`, pp. 571–577.
[35] M.E. Pohst, *Computational algebraic number theory*, DMV Seminar Band 21, Birkhäuser Verlag, 1993. MR **94j:**11132
[36] B.D. Ripley, *The lattice structure of pseudo-random number generators*, Proc. Roy. Soc. London Ser. A **389** (1983), 197–204. MR **85i:**65010
[37] V. Shoup, *NTL: A Library for doing Number Theory*, `http://www.shoup.net/`.
[38] I.H. Sloan and S. Joe, *Lattice methods for multiple integration*, Oxford Univ. Press, New York, 1994. MR **98a:**65026
[39] A. Storjohann, *Faster Algorithms for Integer Lattice Basis Reduction*, Technical report, Institute of Scientific Computing, ETH Zürich, 1996, Available at `http://www.inf.ethz.ch/research/wr/`.

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