Algorithmic Construction of Low-Discrepancy Point Sets via Dependent Randomized Rounding

Benjamin Doerr[a], Michael Gnewuch[b], Magnus Wahlström[a]

[a] *Max-Planck-Institut für Informatik, Campus E1 4, 66123 Saarbrücken, Germany*
[b] *Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Christian-Albrechts-Platz 4, 24098 Kiel, Germany, and Department of Computer Science, Columbia University, 1214 Amsterdam Avenue, New York, NY 10027, USA*

Abstract

We provide a deterministic algorithm that constructs small point sets exhibiting a low star discrepancy. The algorithm is based on recent results on randomized roundings respecting hard constraints and their derandomization. It is structurally much simpler than a previous algorithm presented for this problem in [B. Doerr, M. Gnewuch, A. Srivastav, J. Complexity, 21:691–709, 2005]. Besides leading to better theoretical running time bounds, our approach also can be implemented with reasonable effort. We implemented this algorithm and performed numerical comparisons with other known low-discrepancy constructions. The experiments take place in dimensions ranging from 5 to 21 and indicate that our algorithm leads to superior results if the dimension is relatively high and the number of points that have to be constructed is rather small.

Key words: Star discrepancy, low-discrepancy points, randomized rounding, derandomization

1. Introduction

The L^∞-*star discrepancy* or, more shortly, *star discrepancy* of an n-point set T in the d-dimensional unit cube $[0,1]^d$ is given by

$$d^*_\infty(T) := \sup_{x\in[0,1]^d} \left| \frac{1}{n}|T \cap [0,x[| - \mathrm{vol}([0,x[) \right| ,$$

where $[0,x[$ is the d-dimensional anchored half-open box $[0,x_1[\times\ldots\times[0,x_d[$. Here, as in the whole article, the cardinality of a finite set S is denoted by $|S|$

and the ith component of a vector x by x_i. The smallest possible discrepancy of any n-point configuration in $[0,1]^d$ is

$$d^*_\infty(n,d) := \inf_{T\subset[0,1]^d\,;\,|T|=n} d^*_\infty(T)\,.$$

The *inverse of the star discrepancy* is given by

$$n^*_\infty(\varepsilon,d) := \min\{n\in\mathbb{N}\,|\,d^*_\infty(n,d)\le\varepsilon\}\,.$$

The star discrepancy is related to the error of multivariate numerical integration by the Koksma-Hlawka inequality (see, e.g., [11, 20, 22]). The inequality shows that points with small star discrepancy induce quasi-Monte Carlo algorithms with small worst case errors. Since the number of sample points is roughly proportional to the costs of those algorithms, it is of interest to find n-point configurations with small discrepancy and n not too large. In particular, n should not depend exponentially on d.

For fixed dimension d the asymptotically best upper bounds for $d^*_\infty(n,d)$ that have been proved so far are of the form

$$d^*_\infty(n,d)\le C_d\ln(n)^{d-1}n^{-1}\,,\quad n\ge 2\,. \tag{1}$$

These bounds give us no helpful information for moderate values of n, since $\ln(n)^{d-1}n^{-1}$ is an increasing function for $n\le e^{d-1}$. Additionally, point configurations satisfying (1) will in general lead to constants C_d that depend critically on d. (Actually, it is known for some constructions that the constant C'_d in the representation

$$d^*_\infty(n,d)\le\left(C'_d\ln(n)^{d-1}+o(\ln(n)^{d-1})\right)n^{-1}$$

of (1) tends to zero as d approaches infinity, see, e.g., [22, 23, 1]. But to the best of our knowledge, no good bounds have been published for the implicit constant of the o-notation or, respectively, the "whole" constant C_d in (1).)

A bound more suitable for high-dimensional integration was established by Heinrich, Novak, Wasilkowski and Woźniakowski [19], who proved

$$d^*_\infty(n,d)\le cd^{1/2}n^{-1/2}\quad\text{and}\quad n^*_\infty(d,\varepsilon)\le\lceil c^2d\varepsilon^{-2}\rceil\,, \tag{2}$$

where c does not depend on d, n or ε. Here the dependence of the inverse of the star discrepancy on d is optimal. This was also established in [19] by a lower bound for $n^*_\infty(d,\varepsilon)$, which was later improved by Hinrichs [21] to

$n^*_\infty(d,\varepsilon) \geq c_0 d\varepsilon^{-1}$ for $0 < \varepsilon < \varepsilon_0$, where $c_0, \varepsilon_0 > 0$ are constants. The proof of (2) is not constructive but probabilistic, and the proof approach does not provide an estimate for the value of c.

In the same paper the authors proved a slightly weaker bound with an explicitly known small constant k:

$$d^*_\infty(n,d) \leq k d^{1/2} n^{-1/2} \big(\ln(d) + \ln(n)\big)^{1/2}. \tag{3}$$

The proof is again probabilistic and uses Hoeffding's large deviation bound. In [8] the first two authors and Srivastav improved (3) to

$$d^*_\infty(n,d) \leq k' d^{1/2} n^{-1/2} \ln(n)^{1/2}, \tag{4}$$

where k' is smaller than k. (A slightly better bound for the star discrepancy and a corresponding bound for the so-called extreme discrepancy can be found in [15].) Of course the estimate (4) is asymptotically not as good as (2). But the constant k' is small—essentially we have $k' = \sqrt{2}$. A further advantage of this approach is that it can be derandomized. This was done by Srivastav and the first two authors who provided in [8] a deterministic algorithm constructing point sets satisfying (4). The algorithm is based on a quite general derandomization approach of Srivastav and Stangier [29], and is essentially a point-by-point construction using the method of conditional probabilities and so-called pessimistic estimators.

In the proceedings paper [6] the first two authors used a novel approach to randomized rounding presented in [5]. Contrary to the classical one, it allows to generate randomized roundings that respect certain hard constraints. This leads to a construction that needs significantly fewer random variables, which in turn speeds up the randomized construction.

A second speed-up and considerable simplification from the implementational point of view stems from the fact that the general approach in [5] may be derandomized via the more restricted approach of Raghavan [25]. This runs in time $O(mn)$, where n is the number of (random) variables and m the number of constraints.

It thus avoids the general, but more costly solution by Srivastav and Stangier [29], which, from the practical point of view, suffers from a higher running time of $O(mn^2 \log(mn))$ and its extremely high technical demands.

For a given $n \in \mathbb{N}$ the algorithm from [6] computes an n-point set T with discrepancy

$$d^*_\infty(T) \leq (4+\sqrt{3}) \sqrt{n^{-1}\left(\tfrac{1}{2} d \ln(\sigma n) + \ln 2\right)} + 2^{-d\ln(dn)-1} n^{-1} \tag{5}$$

in time $O(d(\sigma n)^d \log(dn))$. Here $\sigma = \sigma(d)$ is less than one and converges to zero as d tends to infinity. In [8] the running time for constructing an n-point set with the same discrepancy order was $O(C^d n^{d+2} \log(d)^d / \log(n)^{d-1})$, C some constant.

A different approach was presented by Kritzer, Pillichshammer and the first two authors in [7], where a component-by-component (CBC) approach is proposed. This algorithm exhibits a reasonably lower running time than the previously mentioned algorithms (which is nevertheless exponential in d), but provides only a weaker discrepancy guarantee of

$$d^*_\infty(T) \leq O(d^{3/2} n^{-1/2} \ln(1 + n/d)^{1/2})$$

for an output set T of size n in dimension d. This algorithm has recently been implemented in [9].

That the running times of these deterministic algorithms are exponential in d may not be too surprising. The problem of constructing n-point sets of such order of discrepancy is related to the problem of approximating the discrepancy of given point sets: Instead of constructing an n-point set deterministically, we may generate n random points, calculate their discrepancy and accept them if a bound of the form (5) is satisfied. Otherwise we proceed by anew generating n random points. The concentration of measure phenomenon guarantees that with high probability we only have to perform a few random trials to end up with a low-discrepancy n-point set. This approach sounds simple, but overlooks the difficulty of calculating (or approximating) high-dimensional star discrepancies. Indeed, it has been shown in [17] that the decision problem whether an arbitrary point set has discrepancy smaller than ε is NP-hard, and recently this was improved to showing that under certain complexity theoretical assumptions, the running time must essentially scale as $n^{\Theta(d)}$ [13]. Also, all deterministic algorithms known so far that approximate the L^∞-star discrepancy of arbitrary given n-point sets has running time exponential in d (see, e.g., [31], the literature mentioned therein, and the discussion in [15]).

Our Results

This paper is an improved and extended version of the conference paper [6], which was purely theoretical without providing an implementation. Contrary to [6], we mainly use the recent derandomization of the first and the third author in [10] of Srinivasan's randomized rounding approach [28].

Compared to Doerr's derandomization approach used in [6], it leads to the slightly better (theoretical) discrepancy bound

$$d^*_\infty(T) \le ((e-1)+\sqrt{3})\sqrt{n^{-1}\left(\tfrac{1}{2}d\ln(\sigma' n)+\ln 2\right)}, \tag{6}$$

where σ' differs only insignificantly from σ in (5). In practice it leads to a better running time and lower rounding errors. Furthermore, it has the advantage that it is technically simpler (in particular, the step of reducing the binary length of the variables to be rounded falls away completely). This leads to a more concise and lucid presentation of the underlying ideas and the actual algorithm.

We implement both versions of the algorithm based on the derandomizations of Doerr's and Srinivasan's approaches to generate randomized roundings [5, 10] and compare them on several test instances. Afterwards we compare the quality of the points generated with the help of the derandomization of Srinivasan's approach to the discrepancy of other constructions, including the recent CBC algorithm proposed in [7] and implemented in [9].

2. Randomized Construction

We start by introducing some useful notation: For arbitrary $n \in \mathbb{N}$ put $[n] := \{1, \ldots, n\}$. If $x, y \in [0,1]^d$, we write $x \le y$ if $x_i \le y_i$ holds for all $i \in [d]$. We write $[x, y] = \prod_{i\in[d]}[x_i, y_i]$ and use corresponding notation for open and half-open intervals. For a point $x \in [0,1]^d$ we denote by $V_x := \prod_{i\in[d]} x_i$ the volume of the box $[0, x]$. Similarly, we denote the volume of a Lebesgue measurable subset S of $[0,1]^d$ by V_S.

2.1. Grids and Covers

Let $0 = q_0 < q_1 < \ldots < q_k = 1$ and

$$G := \{q_i \,|\, 1 \le i \le k\}^d. \tag{7}$$

G is an isotropic (not necessarily equidistant) grid in the d-dimensional unit cube $[0,1]^d$. (By a grid, we shall always mean a finite point set G' in $[0,1]^d$ that can be written as $G' = (G'_0)^d$ for some $G'_0 \subset [0,1]$.) For each $\alpha = (\alpha_1, \ldots, \alpha_d) \in \{0, 1, \ldots, k\}^d$ let $q_\alpha := (q_{\alpha_1}, \ldots, q_{\alpha_d})$, and denote $(\alpha_1 - 1, \ldots, \alpha_d - 1)$ simply by $\alpha - 1$. Let $\delta = \delta(G)$ be the smallest real

number such that for all $y \in [0,1]^d$ there are $x, z \in G \cup \{0\}$ with $y \in [x,z]$ and $V_z - V_x \leq \delta$, i.e.,

$$\delta(G) = \max_{\alpha \in [k]^d} (V_{q_\alpha} - V_{q_{\alpha-1}}). \tag{8}$$

In the language of [8] δ is minimal such that G is a δ–cover. Let us restate the definition from [8]:

A finite set $\Gamma \subset [0,1]^d$ is a *δ-cover* of $[0,1]^d$ if for every $y \in [0,1]^d$ there exist $x, z \in \Gamma \cup \{0\}$ with $V_z - V_x \leq \delta$ and $x \leq y \leq z$.

The helpfulness of δ-covers lies in the fact that one can use them to discretize the discrepancy while controlling the discretization error:

Lemma 1. *Let Γ be a δ-cover of $[0,1]^d$. Then for all n-point sets $T \subset [0,1]^d$*

$$d^*_\infty(T) \leq d^*_\Gamma(T) + \delta\,, \quad \textit{where} \quad d^*_\Gamma(T) := \max_{x \in \Gamma} \left| \frac{1}{n} |T \cap [0,x[| - V_x \right|. \tag{9}$$

The proof is straightforward and can, e.g., be found in [8].

Let now

$$\mathcal{I} := \{[q_{i-1}, q_i[\,|\, 1 \leq i \leq k\}$$

and

$$\mathcal{B} := \{\prod_{i=1}^{d} I_i \,|\, I_1, \ldots, I_d \in \mathcal{I}\}. \tag{10}$$

Note that $\mathcal{B}$ is a partition of $[0,1[^d$ into axis-parallel boxes with upper right corners in G. Let

$$\mathcal{C}_0 := \{[0, g[\,|\, g \in G\}. \tag{11}$$

$\mathcal{C}_0$ is a subset of the set $\mathcal{C}$ of all axis-parallel boxes that are anchored in 0 (these boxes are sometimes called *corners*). If $C \in \mathcal{C}_0$, then denote by $\mathcal{B}(C)$ the set of all $B \in \mathcal{B}$ with $B \subseteq C$.

For $B \in \mathcal{B}$, let $x_B := nV_B$ be the fair number of points to lie in B. We now randomly round (x_B) to integers (y_B) and then choose our point set in a way that it has exactly y_B points in each box B. This rounding is done via a recent extension of the classical randomized rounding method due to Raghavan [25]. We briefly review the basics.

2.2. Randomized Rounding

For a number r we write $\lfloor r \rfloor = \max\{z \in \mathbb{Z} \,|\, z \leq r\}$, $\lceil r \rceil = \min\{z \in \mathbb{Z} \,|\, z \geq r\}$ and $\{r\} = r - \lfloor r \rfloor$. Let $\xi \in \mathbb{R}$. An integer-valued random variable y is called *randomized rounding of* ξ if it obeys the probability distribution

$$\begin{aligned} \Pr(y = \lfloor \xi \rfloor + 1) &= \{\xi\}, \\ \Pr(y = \lfloor \xi \rfloor) &= 1 - \{\xi\}. \end{aligned}$$

Since only the fractional part of ξ is relevant, we often may ignore the integer part and then have $\xi \in [0, 1]$. In this case, a randomized rounding y of ξ satisfies

$$\begin{aligned} \Pr(y = 1) &= \xi, \\ \Pr(y = 0) &= 1 - \xi. \end{aligned}$$

For $\xi \in \mathbb{R}^n$, we call $y = (y_1, \ldots, y_n)$ *randomized rounding of* ξ if y_j is a randomized rounding of ξ_j for all $j \in [n]$. We call y *independent randomized rounding* of ξ, if the y_i are mutually independent random variables.

Independent randomized rounding was introduced by Raghavan [25] and since has found numerous applications. It takes its strength from the fact that sums of independent random variables are strongly concentrated around their mean. This allows to bound the deviation of a weighted sum of the ξ_i from the corresponding sum of the y_i (this is done via so-called Chernoff bounds).

Independent randomized rounding can be derandomized. That is, one can transform the above sketched approach into a deterministic rounding algorithm (at the price of a slightly higher running time) that guarantees large deviation bounds comparable to those that randomized rounding satisfies with high probability.

For our purposes, independent randomized rounding is not fully satisfactory since we would like to construct exactly n points. In other words, we prefer to have

$$\sum_{B \in \mathcal{B}} y_B = \sum_{B \in \mathcal{B}} x_B = n \tag{12}$$

without any deviation. Fortunately, this can be achieved relatively easy with two recent extensions of randomized rounding.

The historically first way of generating such non-independent randomized roundings is due to Srinivasan [28] (see [12] for the journal version). Contrary

to classical randomized rounding, the authors of [12, 28] did not provide a derandomization of these randomized roundings.

This was overcome in [5], where the first author gave a different way of generating randomized roundings satisfying hard constraints like (12). This approach is structurally simpler, because it reduces the problem to the independent case. In consequence, one can use Raghavan's derandomization. The price for this are slightly higher rounding errors in the derandomized version (compared to Raghavan's derandomization for the setting without hard constraints) and a slightly larger running time (both compared to Raghavan's derandomization and Raghavan's and Srinivasan's randomized version).

In [10], the first and third author finally gave a derandomization also of Srinivasan's dependent randomized roundings. It satisfies the same error guarantees as Raghavan's derandomization. In experiments [10], we observed rounding errors that typically were slightly better, with a high dependence on how the problem instances actually look like. Since for the rounding problems occurring in this paper the derandomization given in [10] mostly produces better errors and leads to better runnung times, we focus in the following on this one.

Since it greatly eases the presentation, we start by describing Srinivasan's randomized roundings [28] and then build on this to explain the derandomization of [10].

Theorem 2 ([28])**.** *Let $\xi \in [0,1]^N$ such that $\sum_{i=1}^N \xi_i \in \mathbb{N}$. Then in time $O(N)$ a randomized rounding y of ξ can be generated such that $\Pr(\sum_{i=1}^N y_i = \sum_{i=1}^N \xi_i) = 1$ and for all $a \in [0,1]^N$, $Y := \sum_{i=1}^N a_i y_i$, $\mu := E(Y) = \sum_{i=1}^N a_i \xi_i$ and all $\delta \in [0,1]$,*

$$\begin{aligned} \Pr(Y > (1+\delta)\mu) &< \exp(-\tfrac{1}{3}\mu\delta^2), \\ \Pr(Y < (1-\delta)\mu) &< \exp(-\tfrac{1}{2}\mu\delta^2). \end{aligned}$$

Such roundings can in fact be generated relatively easy. We start by setting $y_i := \xi_i$. We repeat the following pair-rounding step until all y_i are integral (that is, in $\{0,1\}$).

Pair-rounding: Find $i, j \in [N]$ such that y_i and y_j are not integral. Choose $\delta \in [0,1]$ minimal such that at least one of $y_i + \delta$ and $y_j - \delta$ becomes integral. Likewise, choose $\varepsilon \in [0,1]$ minimal such that at least one of $y_i - \varepsilon$ and $y_j + \varepsilon$ becomes integral. With probability $\varepsilon/(\delta+\varepsilon)$, set $(y_i, y_j) := (y_i + \delta, y_j - \delta)$, otherwise set $(y_i, y_j) := (y_i - \varepsilon, y_j + \varepsilon)$.

By definition, each pair-rounding iteration does not change the sum of the y_i. Also, the expected value of some variable after one pair-rounding is equal to its original value. Hence these two properties also hold for the whole rounding process, showing that we satisfy the hard constraint $\sum_i y_i = \sum_i \xi_i$ and actually do randomized rounding, that is, have $E(y_i) = \xi_i$ for all i. Note that the latter is an equivalent definition for randomized rounding if y_i only takes the values $\lfloor \xi_i \rfloor$ and $\lceil \xi_i \rceil$.

It is slightly more involved (see [28]) to show that the y_i fulfill certain negative correlation properties. By a result of Panconesi and Srinivasan [24], this implies the usual Chernoff bounds that are known for independent random variables.

Since our roundings do not change the sum of all variables, Theorem 2 easily implies the following bound, which in our setting is more convenient.

Lemma 3. *Assume the setting of Theorem 2, but allow the ξ_i to be arbitrary non-negative numbers. Let $n := \sum_{i=1}^{N} \xi_i$. Then for all $\lambda \geq 0$, we have*

$$\Pr(|Y - \mu| > \lambda) < 2\exp(-\tfrac{1}{3}\lambda^2/n).$$

Proof. We may assume $\lambda \leq n$, as Y never exceeds n due to the fact that $\sum_{i=1}^{N} y_i = n$ by construction and $a_i \in [0,1]$ for all $i \in [N]$. Let us first assume that $\xi \in [0,1]^N$.

Let $N' = N + \lceil n - \mu \rceil$. For $i = N+1, \ldots, N'$ let $\xi_i = 1$ and $a_i = (n-\mu)/(N'-N)$. Let $y_1, \ldots, y_{N'}$ be a randomized rounding of $\xi_1, \ldots, \xi_{N'}$ as in Theorem 2. Note that since $\xi_{N+1} = \ldots = \xi_{N'} = 1$, we have $y_{N+1} = \ldots = y_{N'} = 1$ with probability one. In particular, there is a natural one-one correspondence between the randomized roundings of $\xi_1, \ldots, \xi_{N'}$ and those of $\xi_1, \ldots, \xi_N$. This allows us to not really distinguish between them.

Let $Y' = \sum_{i=1}^{N'} a_i \xi_i$ and $\mu' = E(Y')$. Note that $\mu' = n$ by construction. Hence with $\delta = \lambda/n$, the first bound of Theorem 2 yields

$$\begin{aligned} \Pr(Y - \mu > \lambda) &= \Pr(Y' - \mu' > \lambda) \\ &= \Pr(Y' > (1+\delta)\mu') \\ &< \exp(-\tfrac{1}{3}n\delta^2) = \exp(-\tfrac{1}{3}\lambda^2/n). \end{aligned}$$

The second bound of Theorem 2 analogously yields $\Pr(Y - \mu < -\lambda) < \exp(-\frac{1}{2}\lambda^2/n)$. Both estimates give this lemma (still assuming $\xi \in [0,1]^N$).

For arbitrary $\xi \in \mathbb{R}^N_{\geq 0}$, define $\tilde{\xi} \in [0,1]^N$ by $\tilde{\xi}_i = \{\xi_i\} = \xi_i - \lfloor \xi_i \rfloor$ for all $i \in [N]$. Let $\tilde{y}$ be a randomized rounding of $\tilde{\xi}$ as in Theorem 2. Define y

through $y_i = \tilde{y}_i + \lfloor \xi_i \rfloor$. Then y is a randomized rounding of ξ and satisfies $\sum_{i=1}^N y_i = \sum_{i=1}^N \xi_i$ with probability one. By construction, $Y - \mu = \sum_{i=1}^N a_i \tilde{y}_i - \sum_{i=1}^N a_i \tilde{\xi}_i$, so we may apply the claim for the $[0,1]^N$ case. This yields an error bound of $2\exp(-\frac{1}{3}\lambda^2/\tilde{n})$ with $\tilde{n} = \sum_{i=1}^N \tilde{\xi}_i$, which immediately implies the bound of the lemma since $\tilde{n} \le n$. □

2.3. Construction of the Point Set

Let G, $\delta = \delta(G)$, $\mathcal{B}$, and $\mathcal{C}_0$ be as defined in (7), (8), (10), and (11), respectively. We use Theorem 2 to generate random variables (y_B) as randomized roundings of (x_B). Since by definition the x_B, $B \in \mathcal{B}$, are non-negative, the y_B, $B \in \mathcal{B}$, are non-negative integers. Let T be an n-point set in the unit cube such that for all $B \in \mathcal{B}$ the intersection $T \cap B$ contains exactly y_B points. (At the moment we do not care about the actual placement of the y_B points inside B, since this does not affect our analysis. Later we will assume that the points of our set T are placed independently and uniformly random in B.)

Lemma 4. *Let $C \in \mathcal{C}_0$. Then for all non-negative λ we have*

$$\Pr\left(\left||C \cap T| - nV_C\right| > \lambda\right) < 2\exp\left(-\frac{\lambda^2}{3n}\right).$$

Proof. By construction, we have $|C \cap T| = \sum_{B \in \mathcal{B}(C)} y_B$ and $nV_C = \sum_{B \in \mathcal{B}(C)} x_B$. Hence,

$$\left||C \cap T| - nV_C\right| = \left|\sum_{B \in \mathcal{B}(C)} (y_B - x_B)\right|.$$

Since the x_B are non-negative and $n = \sum_{B \in \mathcal{B}} x_B$, we get from Lemma 3

$$\Pr\left(\left|\sum_{B \in \mathcal{B}(C)} (y_B - x_B)\right| > \lambda\right) < 2\exp\left(-\frac{\lambda^2}{3n}\right).$$

□

Theorem 5. *For all $p \in (0,1]$, the point set T satisfies*

$$\Pr\left(d_\infty^*(T) > \sqrt{3n^{-1}\ln(2p^{-1}|\mathcal{B}|)} + \delta\right) < p. \tag{13}$$

Proof. By Lemma 1, we have $d^*_\infty(T) \le d^*_G(T) + \delta$. If $\lambda = \sqrt{3n \ln(2p^{-1}|\mathcal{B}|)}$, then we obtain from Lemma 4

$$\Pr(\big||C \cap T| - nV_C\big| > \lambda) < p\,|\mathcal{B}|^{-1} \quad \text{for all } C \in \mathcal{C}_0.$$

Hence, since $|\mathcal{C}_0| = |\mathcal{B}|$,

$$\Pr(d^*_G(T) > \lambda/n) \le \sum_{C \in \mathcal{C}_0} \Pr\left(\big||C \cap T| - nV_C\big| > \lambda\right) < p.$$

□

Choice of Parameters

Note that inequality (13) depends on the parameters p and δ (in particular, $|\mathcal{B}|$ depends on δ). For our derandomization procedure we may choose $p = 1$. In the following we make a reasonable choice for δ to get a version of inequality (13) that only depends on d and n.

Let $d \ge 2$. In [8, Thm.2.3] a δ-cover in form of a non-equidistant grid $G = \{q_1, \ldots, q_k\}^d$ was constructed satisfying

$$k = k(\delta, d) = \left\lceil \frac{d}{d-1} \frac{\ln(1 - (1-\delta)^{1/d}) - \ln \delta}{\ln(1-\delta)} \right\rceil + 1 \le \left\lceil \frac{d}{d-1} \frac{\ln d}{\delta} \right\rceil + 1\,. \quad (14)$$

The explicit construction goes as follows: Put $r_0 := 1$ and $r_1 := (1-\delta)^{1/d}$. If $r_i > \delta$, then define $r_{i+1} := (r_i - \delta) r_1^{1-d}$. If $r_{i+1} \le \delta$, then put $\kappa(\delta, d) := i + 1$, otherwise proceed by calculating r_{i+2}. Then $k = \kappa(\delta, d) + 1$ and $q_{k-i} = r_i$.

For this grid G and $\delta \le 1/2$ we get

$$\ln |\mathcal{B}| = \ln |G| = d \ln k \le d(\ln \delta^{-1} + \ln \ln d + \ln 4)\,. \quad (15)$$

Choosing

$$\delta = \left(3n^{-1}(d\,(\ln \ln d + \ln 8) + \ln 2)\right)^{1/2} \quad (16)$$

leads to

$$\ln |\mathcal{B}| \le \frac{d}{2} \ln(\sigma n)\,, \quad \text{where} \quad \sigma = \sigma(d) := \frac{16(\ln d)^2}{3(d(\ln \ln d + \ln 8) + \ln 2)}\,. \quad (17)$$

An elementary analysis shows that σ takes its maximum in $d = 6$ and therefore $\max_{d \ge 2} \sigma(d) < 1.0272$.

Let us assume that for a given dimension $d \geq 2$ the number of points n is large enough to imply $\delta \leq 1/2$. Then, for $p = 1$,

$$\sqrt{3n^{-1}\ln(2p^{-1}|\mathcal{B}|)} + \delta \leq 2\sqrt{3n^{-1}\left(\frac{d}{2}\ln(\sigma n) + \ln 2\right)}.$$

Our choices of G, δ, and p result in the following corollary.

Corollary 6. *Let δ be as in (16), and let G be the corresponding δ-cover from [8, Thm. 2.3]. Furthermore, let $\sigma = \sigma(d)$ be as in (17). Then*

$$\Pr\left(d^*_\infty(T) > 2\sqrt{3n^{-1}\left(\tfrac{1}{2}d\ln(\sigma\, n) + \ln 2\right)}\right) < 1. \tag{18}$$

Remark 1. Note that we are using the non-equidistant grid G from [8, Thm. 2.3] as δ-cover. In [8, 15, 16], also δ-covers were constructed that had no grid structure. These δ-covers are superior in the sense that they need fewer points. For the approach we use in this paper, however, they cannot be applied. The reason is that in Lemma 4, we heavily use the fact that corners (elements from $\mathcal{C}_0$) are the union of all boxes (elements from $\mathcal{B}$) they have a non-trivial intersection with.

Remark 2. Since for comparison we will also use the randomized roundings of [5] and their derandomization, let us briefly outline the differences. It is based on the observation that randomized rounding with a cardinality constraint is simple, if all numbers to be rounded are zero or 1/2. In this case, we just choose a random set of half of the 1/2-numbers, and round exactly these up to one. This can be extended to (almost) arbitrary numbers via their binary expansion. That is, if all numbers $x_1, \ldots, x_n$ to be rounded have a finite binary expansion $x_j = \sum_{k=1}^{\ell} a_{kj}2^{-i}$, $a_{kj} \in \{0,1\}$, of length at most ℓ, we may use the $\{0, 1/2\}$ solution to round the vector $\tilde{x} = (a_{\ell 1}/2, \ldots, a_{\ell n}/2)$ in a randomized rounding fashion to some $\tilde{y} \in \{0,1\}^n$, and update $x := x - 2^{\ell-1}(\tilde{x} - \tilde{y})$. This has binary length $\ell - 1$. Iterating this defines a rounding scheme. Apart from the running time increase by a factor of ℓ, it satisfies the same properties as the method by Srinivasan.

3. Derandomized Construction

The randomized roundings of Theorem 2 and hence the whole construction of the n-point set T described in Section 2.3 can be derandomized using

the method of pessimistic estimators of Raghavan [25], as shown in [10]. We will here give an outline of the resulting deterministic rounding algorithm. For our purpose it is sufficient to consider the case of matrices with entries from $\{0,1\}$. We also restrict ourselves to one concrete definition of the pessimistic estimators (see below), rather than the general form shown in [25].

Let $A \in \{0,1\}^{m\times n}$ and $\xi \in \mathbb{R}^n_{\geq 0}$ such that $\sum_{i=1}^n \xi_i \in \mathbb{Z}$. We will compute a rounding y of ξ such that $\sum_{i=1}^n y_i = \sum_{i=1}^n \xi_i$, where for each $i \in [m]$, we have

$$|(A\xi)_i - (Ay)_i| < \delta_i (A\xi)_i \tag{19}$$

where δ_i are *error tolerances* fixed in the algorithm. To achieve this, we define *pessimistic estimators* P_i^+, P_i^- following Raghavan [25]. Let $p_i = \{\xi_i\}$ and $\mu_i = \sum_{j=1}^n A_{ij} p_j$. The pessimistic estimators are defined according to

$$\begin{aligned} P_i^+ &= (1+\delta_i)^{-(1+\delta_i)\mu_i} \prod_{j:A_{ij}=1} (1+\delta_i p_j) \\ P_i^- &= (1+\delta_i)^{(1-\delta_i)\mu_i} \prod_{j:A_{ij}=1} \left(1 + \left(\frac{1}{1+\delta_i} - 1\right) p_j\right). \end{aligned}$$

We shall need only a simple observation. Let $q \in \{0,1\}^n$, and let Q_i^+, Q_i^- be the values of P_i^+ and P_i^-, respectively, calculated on values q_j instead of p_j, with μ_i unchanged. Then $Q_i^+ \geq 1$ if and only if $\sum_j A_{ij} q_j \geq (1+\delta_i)\mu_i$, and $Q_i^- \geq 1$ if and only if $\sum_j A_{ij} q_j \leq (1-\delta_i)\mu_i$.

By *updating* the pessimistic estimators for some adjustment $p_j \leftarrow x$, we shall mean the operation of replacing the factor $(1+\delta_i p_j)$ in P_i^+ by $(1+\delta_i x)$, and analogously for P_i^-, for each $i \in [m]$ such that $A_{ij} = 1$. Again, μ_i is unchanged.

The rounding algorithm proceeds as follows.

(i) Initialize $p_j = \{\xi_j\}$.

(ii) Set the error tolerances δ_i such that for each $i \in [m]$, $P_i^+, P_i^- < 1/(2m)$. Let $U = \sum_{i=1}^m (P_i^+ + P_i^-)$.

(iii) Let $J = \{j \in [n] : p_j \notin \{0,1\}\}$. While $|J| \geq 2$:

  (a) Pick $j, j' \in J$.

  (b) Let $(p_j^{(i)}, p_{j'}^{(i)})$, $i = 1, 2$, be the two possible outcomes of the pair-rounding step described in Section 2.2 with respect to the pair of variables $(p_j, p_{j'})$. Let U_i, $i = 1, 2$, be the sum of the pessimistic estimators updated according to the corresponding outcome.

(c) Pick $i \in \{1,2\}$ to minimize U_i. Let $p_j \leftarrow p_j^{(i)}$, $p_{j'} \leftarrow p_{j'}^{(i)}$ and update J, the pessimistic estimators, and U.

(iv) Let $y_j = \lfloor \xi_j \rfloor + p_j$.

Note that in step (ii) the error tolerance δ_i can be chosen independently for each i and that we clearly have $U < 1$. Furthermore, it was shown in [10, Sect. 3.1] that the minimum U_i of $\{U_1, U_2\}$ appearing in step (iii.c) satisfies $U_i \leq U$. After the loop, since $\sum_j \xi_j \in \mathbb{Z}$, $J = \emptyset$ holds and $p_j \in \{0,1\}$ for each $j \in [n]$. By our previous observation, $(Ap)_i \geq (1+\delta_i)\mu_i$ if and only if $P_i^+ \geq 1$, and analogously for the lower bound. Since $U < 1$ is maintained throughout the algorithm and since the pessimistic estimators are non-negative, this cannot occur. The process thus produces a rounding satisfying equation (19). Also note that as in the randomized rounding, the value of $\sum_i p_i$ is kept constant throughout the process.

Finally, though the order in which variables are picked in step (iii.a) is not important for the theoretical bound, it was found in [10] that in practice, this can have a big effect on the size of the rounding error for certain instances (in particular, for the kind of instances created in this paper; see [10] for details). The best order was found to be one such that the tree formed by the pairings is a balanced binary tree (so that each probability p_j is adjusted only $O(\log n)$ times).

Using the bounds on δ_i derived by Raghavan [25], we get the following theorem.

Theorem 7. *Let $A \in \{0,1\}^{m \times n}$. Let $\xi \in \mathbb{R}^n_{\geq 0}$ such that $\sum_{i=1}^n \xi_i \in \mathbb{Z}$. Then a rounding y of ξ such that $\sum_{i=1}^n y_i = \sum_{i=1}^n \xi_i$ and*

$$\forall i \in [m] : |(A\xi)_i - (Ay)_i| \leq (e-1)\sqrt{\max\{(A\xi)_i, \ln(2m)\}\ln(2m)}$$

can be computed in time $O(mn)$.

The rounding errors we are interested in are all of the kind $\sum_{B \in \mathcal{B}(C)}(x_B - y_B)$ for some $C \in \mathcal{C}_0$. Hence the matrix encoding all these errors is an $|\mathcal{C}_0| \times |\mathcal{B}|$ matrix having entries 0 and 1 only. More precisely, we consider the matrix $A = (a_{C,B})_{C \in \mathcal{C}_0, B \in \mathcal{B}}$, where $a_{C,B} = 1$ if $B \subseteq C$ and $a_{C,B} = 0$ else. For each $C \in \mathcal{C}_0$ we have

$$(Ax)_C = \sum_{B \in \mathcal{B}(C)} x_B \leq \sum_{B \in \mathcal{B}} x_B = n\,.$$

Thus, if $n \geq \ln(2|\mathcal{C}_0|)$, we get from Theorem 7 the bound

$$|(Ax)_C - (Ay)_C| \leq (e-1)\sqrt{n \ln(2|\mathcal{C}_0|)}\,.$$

If $n \leq \ln(2|\mathcal{C}_0|)$, this bound holds trivially, since always $|(Ax)_C - (Ay)_C| \leq n$. Hence we obtain

$$d_G^*(T) = \max_{C \in \mathcal{C}_0} \frac{1}{n} |(Ax)_C - (Ay)_C| \leq (e-1)\sqrt{n^{-1} \ln(2|\mathcal{C}_0|)}. \tag{20}$$

Altogether we get the following theorem.

Theorem 8. *Let $n \in \mathbb{N}$ be given. Let G be an isotropic grid of the form (7), and let $\delta = \delta(G)$, $\mathcal{B}$, and $\mathcal{C}_0$ be as in (8), (10), and (11), respectively. Then there is a deterministic algorithm that*

(i) computes a point set $T \subseteq [0,1]^d$ that has exactly n points;
(ii) $d_\infty^*(T) \leq (e-1)\sqrt{n^{-1} \ln(2|\mathcal{C}_0|)} + \delta$;
(iii) has running time $O(|\mathcal{B}||\mathcal{C}_0|)$.

We get the following corollary.

Corollary 9. *Let $n \in \mathbb{N}$ be given. Let δ be as in (16), and let G be the corresponding δ-cover from [8, Thm. 2.3]. Furthermore, let σ be as defined in (17). There is a deterministic algorithm that*

(i) computes a point set $T \subseteq [0,1]^d$ that has exactly n points;
(ii) $d_\infty^*(T) \leq ((e-1)+\sqrt{3})\sqrt{n^{-1}\left(\frac{d}{2}\ln(\sigma\, n) + \ln 2\right)}$;
(iii) has running time $O(d\, \ln(d\, n)(\sigma\, n)^d)$.

Remark 3. As mentioned above, the randomized roundings of [5] can also be derandomized. The corresponding analogues of Theorem 7 and 8 and Corollary 9 can be found in [6]. With the default values that we will use in our experiments, the derandomized algorithm guarantees the discrepancy bound (5).

4. Numerical Experiments

In this section, we present the results of our numerical experiments, comparing the quality of our point sets to those produced by previous methods.

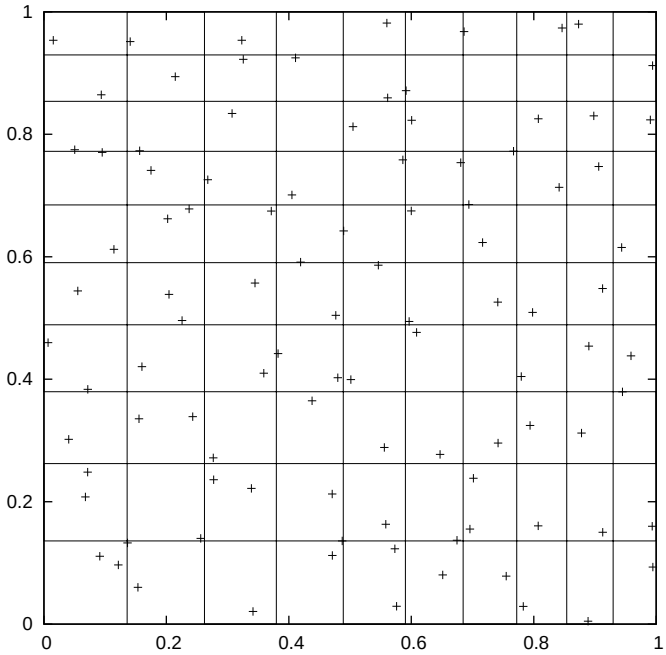

Figure 1: Example point set and grid, 100 points

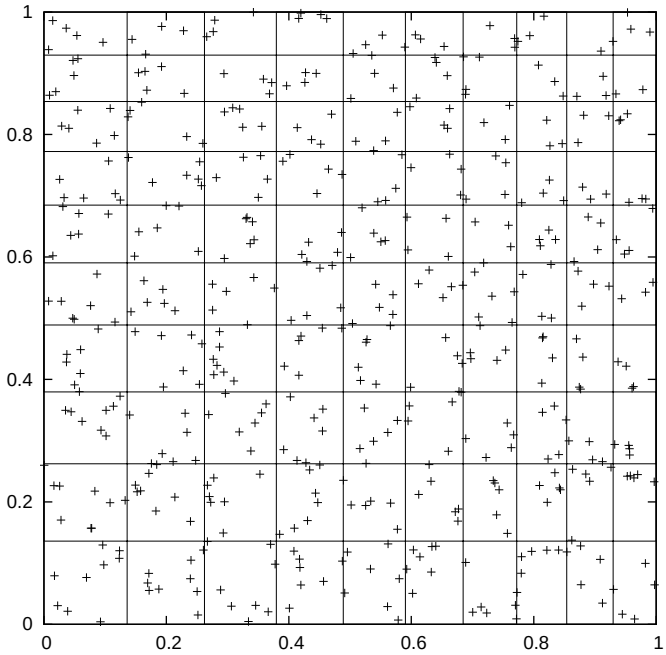

Figure 2: Example point set and grid, 500 points

d	k	n range	Randomized, time	Derandomized, time
9	2	0–120	< 0.01s	0.1s
9	3	130–200	0.05s	36s
15	2	0–180	0.1s	37s
20	2	0–240	1s	10.5 hours

Table 1: Times required for point set creation (default settings)

Figures 1 and 2 illustrate our approach in dimension $d = 2$. The grid shown has the lowest value of δ possible with our approach for $k = 10$, with $\delta = 0.1359$; as previously described, the number of points inside each grid box is decided by the rounding procedure, and the actual placement of the points within the box is done uniformly at random.

As the time required for the derandomized procedure grows as $k^{2d} = |\mathcal{B}||\mathcal{C}_0|$ for grid size k and d dimensions, we are limited in our choice of k for the interesting values of d – see Table 1 for the running times required for some settings. The time is determined by the values of d and k; the table also includes the corresponding range of n under the default grid setting (i.e. with $d = 9$ and $n \leq 120$, the default grid size is $k = 2$).

4.1. Parameter tuning

Before we compare our new point sets against those created by other methods, we first examine the effects of using different parameter settings in our algorithm. The parameters we consider are the method to use for the rounding step – the randomized method of Srinivasan [28], the derandomization of this method by the first and third authors [10], or the randomized or derandomized version of the method of the first author [5] – and the size to use for the grid. In the latter case, while the running time is frequently a limiting factor for the derandomized methods, using a size which is one step higher than the default can still be feasible.

Table 2 shows the range of discrepancies encountered for these settings. For each row of the table, one point set was created for each value of n in the range indicated. Table 3 shows the discrepancy introduced by the rounding step alone, i.e., the largest discrepancy found among boxes whose upper right corners are lying on the grid. We see that the derandomized methods have a clear advantage over the randomized methods in general, and that the derandomized methods, but not the randomized, provide better point sets with

d	n	k	Method	Lowest value	Median	Highest value
7	145–155	3	Srinivasan, derandomized	0.119	0.139	0.163
7	145–155	4	Srinivasan, derandomized	0.118	0.134	0.155
7	145–155	3	Srinivasan, randomized	0.141	0.155	0.173
7	145–155	4	Srinivasan, randomized	0.137	0.157	0.216
7	145–155	3	Doerr, derandomized	0.124	0.135	0.157
7	145–155	4	Doerr, derandomized	0.124	0.133	0.164
7	145–155	3	Doerr, randomized	0.136	0.157	0.191
7	145–155	4	Doerr, randomized	0.136	0.156	0.208
9	85–95	2	Srinivasan, derandomized	0.214	0.221	0.291
9	85–95	3	Srinivasan, derandomized	0.193	0.212	0.258
9	85–95	2	Srinivasan, randomized	0.204	0.233	0.313
9	85–95	3	Srinivasan, randomized	0.216	0.233	0.282
9	85–95	2	Doerr, derandomized	0.188	0.220	0.253
9	85–95	3	Doerr, derandomized	0.187	0.207	0.254
9	85–95	2	Doerr, randomized	0.211	0.226	0.244
9	85–95	3	Doerr, randomized	0.199	0.232	0.292

Table 2: Comparison of parameter settings. Exact discrepancy values, showing lowest, median, and highest results of eleven runs.

d	n	k	Method	Lowest value	Median	Highest value
7	145–155	3	Srinivasan, derandomized	0.023	0.026	0.030
7	145–155	4	Srinivasan, derandomized	0.033	0.036	0.040
7	145–155	3	Srinivasan, randomized	0.049	0.058	0.096
7	145–155	4	Srinivasan, randomized	0.068	0.091	0.163
7	145–155	3	Doerr, derandomized	0.025	0.030	0.040
7	145–155	4	Doerr, derandomized	0.038	0.041	0.056
7	145–155	3	Doerr, randomized	0.055	0.069	0.090
7	145–155	4	Doerr, randomized	0.065	0.084	0.119
9	85–95	2	Srinivasan, derandomized	0.016	0.025	0.036
9	85–95	3	Srinivasan, derandomized	0.048	0.056	0.064
9	85–95	2	Srinivasan, randomized	0.043	0.051	0.096
9	85–95	3	Srinivasan, randomized	0.100	0.119	0.187
9	85–95	2	Doerr, derandomized	0.026	0.032	0.050
9	85–95	3	Doerr, derandomized	0.057	0.071	0.093
9	85–95	2	Doerr, randomized	0.044	0.060	0.070
9	85–95	3	Doerr, randomized	0.073	0.130	0.189

Table 3: Comparison of parameter settings. Rounding error components only, showing lowest, median, and highest results of eleven runs.

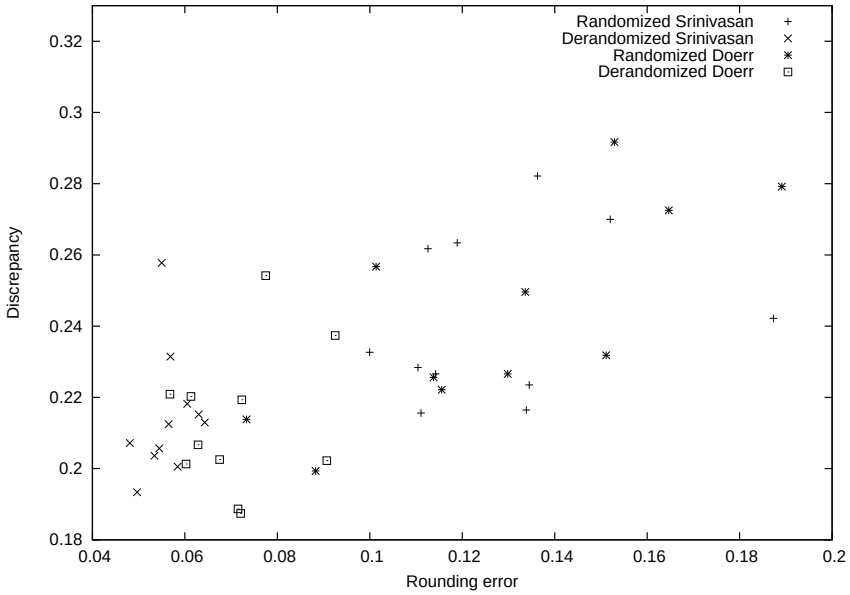

Figure 3: Observed combinations of rounding error and discrepancy for $d = 9$, $k = 3$, $85 \leq n \leq 95$.

a higher value of k. For the rounding error part, we also find that the derandomization of Srinivasan's method behaves better than the derandomized version of Doerr's method, going in line with the results of [10, Fig. 3], where the same type of experiment was performed. However, this difference is not so visible after the noise of the random point placement has been added. In fact, it even seems that the difference in the final discrepancy between derandomized and randomized methods is smaller than the difference in the rounding error component.

To examine the connection closer, we plotted our data for $d = 9$, $k = 3$ (as this is the case with the biggest differences in rounding error) in Figure 3. Every point in the figure represents one instance. Some correlation between rounding error and discrepancy is visible, but the connection is not absolute.

In the sequel, the derandomized method we use will be the derandomization of Srinivasan's method, as this is faster than Doerr's method, produces smaller rounding errors, and produces no visible negative effects in terms of discrepancy. For the derandomization of Srinivasan's approach the choice of k has some impact; we investigate this further in the next paragraph. For the randomized method, there does not seem to be a big difference, but we will

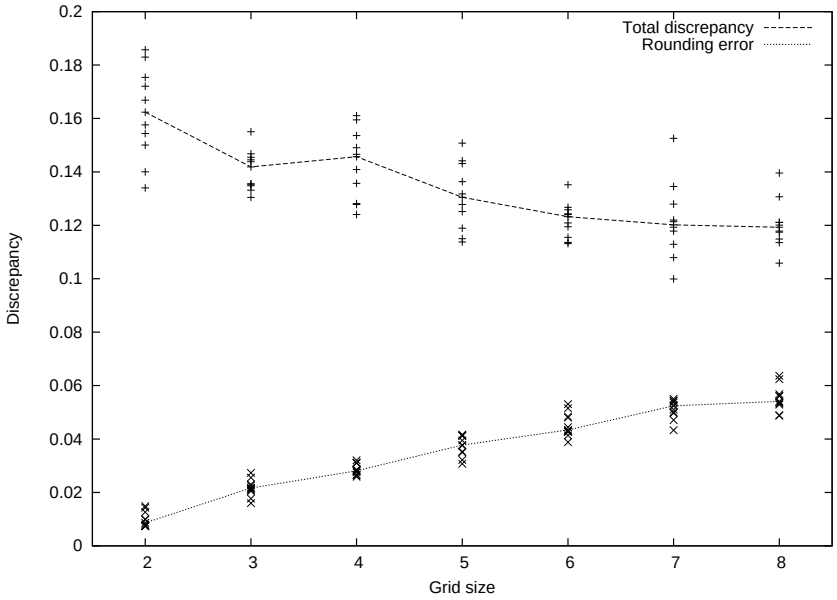

Figure 4: Discrepancy vs grid size; $d = 5$, n around 100

use Srinivasan's method, with the same value of k as in the derandomized method.

Values of k

The choice for the size k^d of the grid $G = \{q_i \,|\, 1 \leq k\}^d$ is based on a balancing of the theoretical bounds for the rounding error and the placement error. As these bounds are purely theoretical, they can be expected to be overly pessimistic compared to the errors occurring in practice, at least with respect to the constants involved. If it should be the case that one of these estimates is more pessimistic than the other, then the prescribed choice of k would not be the best possible value.

We already saw some suggestion of this, as picking a value of k one point higher than the prescribed choice lead to improved point sets for the derandomized methods in the experiments above. To test the issue in more detail, we ran experiments varying the value of k. To be able to handle the number of resulting variables k^d for reasonable values of k, and to calculate the exact discrepancy, we need d and n not to be too high, but at the same time we want a d high enough that the results are relevant for higher values of d. We

chose $d = 5$ and $95 \leq n \leq 105$ (with a range for n to introduce some variation in the deterministic but irregular result of the rounding procedure), and varied k from 2 to 8, calculating the rounding error and the exact discrepancy for each combination of parameters. The results are plotted in Figure 4; the actual values are plotted as crosses, and their median value as a line. Letting k be induced by the value of δ from (16) for this range of parameters leads to $k = 3$. As the plot shows, even using $k = 8$ for these settings shows no signs of being higher than the optimal choice.

Thus, when feasible, we will use a higher setting for k than the default setting. We will indicate the value of k in the tables.

4.2. Discrepancy tests

We will now show the outcomes of the experiments, but first we explain the experimental setup.

Our own point sets are generated as described in the previous section, using randomized and derandomized versions of the rounding procedure. The point sets we compare our points against are generated by five methods. The first is a randomized variant of the method proposed in [7] to generate points in a component-by-component (CBC) fashion, rather than the all-at-once generation used in this paper, resulting in somewhat worse bounds on the discrepancy, but a better running time. The randomized variant we use here is described in detail in [9, Sect. 3.4]; we refer to its output points as CBC points. The other point sets are pure Monte Carlo (i.e. points placed independently and uniformly at random), Halton-Hammersley points (HH-points) [18], Sobol-points [26], and Faure-nets shuffled through a Gray code (Gray-Faure points), where we take the first n points after shuffling. The Halton-Hammersley points are also called Hammersley point sets, see [22]. For the Sobol points, we use an implementation and parameters from [27]; the Gray-Faure points are produced by an implementation by Thiémard [30]. The Gray-Faure points are created with an integer parameter *skip*, controlling the shuffling. As we observed that a skip parameter of 0 or 1 tends to generate point sets with higher discrepancy, we use a random setting ranging from 2 to the maximum value. To even out irregularities in the outcomes of the deterministic parts of the algorithms, we throughout create eleven point sets for each line in the tables, with the number of points ranging from $n - 5$ to $n + 5$, for a target number n of points, and report the median observed.

Because of the difficulty of determining or even usefully approximating the star discrepancy of a high-dimensional point set (see, e.g., [15, 17, 13])

we are quite limited in our choice of dimension and number of points for our experiments. Since theory suggests that our method gains advantage over the classical methods with increasing dimension, we still wish to evaluate our point sets for higher dimensions, and so we are forced to use gradually less precise methods as d and n increase. We use three different methods for this:

- For exact calculation of the discrepancy, we use a speedup of the basic n^d-time enumeration algorithm [2, 3]; though the speedup over the naïve method is significant (by a factor large enough to be difficult to measure), we are still only able to use it for $d < 10$.

- For nearly exact calculation of the discrepancy, we implemented a variant of an algorithm by Dobkin, Eppstein, and Mitchell [4]. To gain speed our version does not calculate the discrepancy exactly, but allows for an imprecision of order d/n. We refer to values calculated with this method as *approximated discrepancy* (though note that the use of this term is not precise in the sense of approximation algorithms).

- The final method we use is a combination of Thiémard's bracketing cover method [31] for upper bounds, and a random experiment for better lower bounds; the random experiment is a modification by C. Winzen and the second author [14, 33] of the method of Winker and Fang [32]. We refer to these bounds as *estimated discrepancy*. Thiémard's method uses an error parameter ϵ such that the upper bound given is guaranteed to be at most ϵ higher than the lower bound, which the method also produces (although the lower bound from the random experiment is superior). However, the dependency of the running time on ϵ is quite bad in higher dimension.

We will present the outcomes of the experiments when calculated using these three methods in turn. We will let d vary from 7 to 21, presenting data with around 150 points as well as some cases with fewer points. In the latter cases, in the exact and approximated discrepancy values, the number of points is largely determined by what the respective method for bounding the discrepancy can handle.

Table 4 contains the range of discrepancy values we encountered for the various methods and settings where the exact method could be used. In the cases $d = 7$, $145 \leq n \leq 155$ and $d = 9$, $85 \leq n \leq 95$, the data for our method is repeated from Table 2.

d	n	Method	Lowest value	Median	Highest value
7	65–75	Gray-Faure	0.150	0.172	0.224
7	65–75	Sobol	0.164	0.180	0.189
7	65–75	Derandomized ($k = 4$)	0.169	0.197	0.223
7	65–75	HH	0.212	0.223	0.240
7	65–75	Randomized ($k = 4$)	0.182	0.232	0.313
7	65–75	Monte Carlo	0.215	0.232	0.316
7	65–75	CBC	0.201	0.247	0.299
7	145–155	Sobol	0.098	0.102	0.114
7	145–155	Gray-Faure	0.099	0.112	0.205
7	145–155	HH	0.115	0.119	0.123
7	145–155	Derandomized ($k = 4$)	0.118	0.134	0.155
7	145–155	Randomized ($k = 4$)	0.141	0.155	0.173
7	145–155	Monte Carlo	0.160	0.182	0.212
7	145–155	CBC	0.165	0.195	0.236
9	85–95	Sobol	0.170	0.173	0.178
9	85–95	Gray-Faure	0.173	0.193	0.217
9	85–95	Derandomized ($k = 3$)	0.193	0.212	0.258
9	85–95	Randomized ($k = 3$)	0.216	0.233	0.282
9	85–95	CBC	0.213	0.238	0.283
9	85–95	Monte Carlo	0.218	0.244	0.274
9	85–95	HH	0.233	0.245	0.259

Table 4: Comparison of methods, exact discrepancy values. Lowest, median, and highest results of eleven runs.

d	n	Method	Lower bound	Random test	Upper bound
9	145–155	Sobol	0.119	0.123	0.151
9	145–155	Gray-Faure	0.130	0.139	0.167
9	145–155	Derandomized ($k = 3$)	0.163	0.163	0.184
9	145–155	HH	0.171	0.172	0.197
9	145–155	Randomized ($k = 3$)	0.170	0.173	0.199
9	145–155	CBC	0.170	0.186	0.203
9	145–155	Monte Carlo	0.194	0.199	0.228
12	65–75	Derandomized ($k = 2$)	0.276	0.278	0.333
12	65–75	Gray-Faure	0.290	0.287	0.338
12	65–75	Randomized ($k = 2$)	0.280	0.291	0.343
12	65–75	Sobol	0.292	0.294	0.328
12	65–75	CBC	0.294	0.307	0.350
12	65–75	Monte Carlo	0.332	0.344	0.390
12	65–75	HH	0.394	0.387	0.431

Table 5: Comparison of methods, approximated discrepancy. The values are the medians of eleven experiments, using the Dobkin et al. method [4] for lower and upper bounds columns, and the random experiment [33] for the "random" column.

Pushing to higher values of d, we show in Table 5 the outcomes of experiments using the nearly exact variant of the method of Dobkin et al.. The columns of the table show the lower and upper bounds provided by the algorithm, as well as the highest discrepancy found by the random experiment (which is also a lower bound, and sometimes of better quality, but quite irregular); every entry is the median of the eleven runs, taken independently (i.e. the value in the "lower bound" column does not necessarily come from the same point set as the value in the "upper bound" column for any line of the table).

Our last set of experiments is shown in Tables 6 and 7, giving estimated discrepancy values for those sizes where only the third method applies. The entries are again medians taken independently. The upper bounds come from Thiémard's algorithm, with error parameter $\epsilon = 0.25$ for $d = 12$, $\epsilon = 0.35$ for $d = 15$, $\epsilon = 0.4$ for $d = 18$, and $\epsilon = 0.45$ for $d \geq 20$. This was chosen as a trade-off between the tightness of the bound and the running time. With these settings, the time for the algorithm varies from about half an hour to about four hours per point set; because of the fast growth of the running time, significantly better bounds were not reasonably attainable.

Looking at the tables overall, the tendencies are perhaps easiest seen when using Monte Carlo as a baseline for comparison. Thus we see a general trend that settings of high dimension and relatively few points are advantageous to our current method, while the opposite is true for the Halton-Hammersley and Gray-Faure points. This is as suggested by the theoretical discrepancy bounds. The Halton-Hammersley points in particular cannot be recommended for $d \geq 9$, but the Gray-Faure points also lose their advantage against the Monte Carlo points for $d \geq 15$. This tendency can not be observed for Sobol points, which performed always better than the Monte Carlo points. Nevertheless, if the ratio n/d is relatively small, as it is the case for $d = 12$ and 15, $65 \leq n \leq 75$, and $d = 21$, $95 \leq n \leq 105$, than the bounds on the discrepancies indicate that our current method performs better even than the Sobol points.

Regarding the CBC points, it seems that the gain we get in speed really is paid for by a higher discrepancy, but we must also mention that the results of [9] indicated that the theoretical analysis which led to the choice of grid in [7] was overly pessimistic, and so the exact grid used is perhaps not optimal in practice.

We have no explanation for the seeming inversion in quality between the randomized and derandomized methods for $d = 15$, $145 \leq n \leq 155$, where the bounds for the points created by the randomized method are suddenly better than for those created by the derandomized method. The errors introduced in the rounding step are roughly twice as big for the randomized as for the derandomized methods (of

d	n	Method	Random test	Upper bound
12	145–155	Gray-Faure	0.156	0.356
12	145–155	Sobol	0.169	0.366
12	145–155	Derandomized ($k = 2$)	0.199	0.391
12	145–155	Randomized ($k = 2$)	0.208	0.404
12	145–155	CBC	0.225	0.402
12	145–155	Monte Carlo	0.221	0.420
12	145–155	HH	0.279	0.447
15	65–75	Derandomized ($k = 2$)	0.322	0.568
15	65–75	CBC	0.332	0.570
15	65–75	Sobol	0.336	0.568
15	65–75	Randomized ($k = 2$)	0.333	0.572
15	65–75	Monte Carlo	0.370	0.596
15	65–75	Gray-Faure	0.369	0.618
15	65–75	HH	0.572	0.778
15	95–105	Sobol	0.258	0.514
15	95–105	Derandomized ($k = 2$)	0.265	0.523
15	95–105	Randomized ($k = 2$)	0.283	0.536
15	95–105	CBC	0.305	0.552
15	95–105	Gray-Faure	0.308	0.563
15	95–105	Monte Carlo	0.315	0.561
15	95–105	HH	0.436	0.682
15	145–155	Sobol	0.198	0.473
15	145–155	Randomized ($k = 2$)	0.215	0.487
15	145–155	Derandomized ($k = 2$)	0.227	0.501
15	145–155	Gray-Faure	0.238	0.503
15	145–155	CBC	0.240	0.502
15	145–155	Monte Carlo	0.247	0.520
15	145–155	HH	0.360	0.620

Table 6: Comparison of methods, estimated discrepancy. The values are the medians of eleven experiments, using [31] for upper bounds, and [33] for lower bounds. The parameter ϵ for the upper bound was set to $\epsilon = 0.25$ for $d = 12$ and $\epsilon = 0.35$ for $d = 15$.

d	n	Method	Random test	Upper bound
18	95–105	Sobol	0.293	0.569
18	95–105	Derandomized ($k = 2$)	0.300	0.578
18	95–105	Randomized ($k = 2$)	0.312	0.600
18	95–105	CBC	0.324	0.605
18	95–105	Monte Carlo	0.322	0.626
18	95–105	Gray-Faure	0.336	0.630
18	95–105	HH	0.535	0.790
18	145–155	Sobol	0.230	0.524
18	145–155	Derandomized ($k = 2$)	0.233	0.541
18	145–155	Randomized ($k = 2$)	0.252	0.562
18	145–155	CBC	0.268	0.568
18	145–155	Monte Carlo	0.269	0.580
18	145–155	Gray-Faure	0.276	0.577
18	145–155	HH	0.454	0.699
20	145–155	Sobol	0.239	0.578
20	145–155	Derandomized ($k = 2$)	0.254	0.596
20	145–155	Randomized ($k = 2$)	0.264	0.600
20	145–155	CBC	0.278	0.624
20	145–155	Monte Carlo	0.279	0.631
20	145–155	Gray-Faure	0.279	0.648
20	145–155	HH	0.459	0.736
21	95–105	Derandomized ($k = 2$)	0.299	0.630
21	95–105	Sobol	0.315	0.627
21	95–105	CBC	0.335	0.648
21	95–105	Randomized ($k = 2$)	0.335	0.661
21	95–105	Gray-Faure	0.342	0.660
21	95–105	Monte Carlo	0.342	0.664
21	95–105	HH	0.628	0.823

Table 7: Comparison of methods, estimated discrepancy. The values are the medians of eleven experiments, using [31] for upper bounds, and [33] for lower bounds. The parameter ϵ for the upper bound was set to $\epsilon = 0.4$ for $d = 18$ and $\epsilon = 0.45$ for $d = 20$ and $d = 21$.

size roughly 0.1 for the former, 0.05 for the latter). Thus if for the latter the resulting discrepancies are really higher, then this has to be due to some worse local random placement of the points. Unfortunately, the coarseness of our discrepancy bounds prevents us from making definite conclusions.

Acknowledgment

The first and third authors gratefully acknowledge support from the German Science Foundation (DFG) via its priority programme "SPP 1307: Algorithm Engineering", grant DO 479/4-1.

The second author gratefully acknowledges support from the DFG under grant GN 91/3-1 and GN 91/4-1.

References

[1] E. Atanassov, On the discrepancy of the Halton sequences, Math. Balkanica (N.S.) 18 (2004) 15–32.

[2] P. Bundschuh, Y. C. Zhu, A method for exact calculation of the discrepancy of low-dimensional point sets I, Abh. Math. Sem. Univ. Hamburg 63 (1993) 115–133.

[3] L. de Clerk, A method for exact calculation of the star-discrepancy of plane sets applied to the sequence of Hammersley, Monatsh. Math. 101 (1986) 261–278.

[4] D. P. Dobkin, D. Eppstein, D. P. Mitchell, Computing the discrepancy with applications to supersampling patterns, ACM Trans. Graph. 15 (1996) 354–376.

[5] B. Doerr, Generating randomized roundings with cardinality constraints and derandomizations, in: B. Durand, W. Thomas (Eds.), Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS'06), LNiCS 3884, 2006, pp. 571–583.

[6] B. Doerr, M. Gnewuch, Construction of low-discrepancy point sets of small size by bracketing covers and dependent randomized rounding, in: A. Keller, S. Heinrich, H. Niederreiter (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer, Berlin Heidelberg, 2008, pp. 299–312.

[7] B. Doerr, M. Gnewuch, P. Kritzer, F. Pillichshammer, Component-by-component construction of low-discrepancy point sets of small size, Monte Carlo Methods Appl. 14 (2008) 129–149.

[8] B. Doerr, M. Gnewuch, A. Srivastav, Bounds and constructions for the star discrepancy via δ-covers, J. Complexity 21 (2005) 691–709.

[9] B. Doerr, M. Gnewuch, M. Wahlström, Implementation of a component-by-component algorithm to generate small low-discrepancy samples, in: P. L'Ecuyer, A. B. Owen (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2008, Springer, Berlin Heidelberg, 2009, pp. 323–338.

[10] B. Doerr, M. Wahlström, Randomized rounding in the presence of a cardinality constraint, in: Proceedings of ALENEX'09, 2009, pp. 162-174.

[11] M. Drmota, R. F. Tichy, Sequences, Discrepancies and Applications, LNiM 1651, Springer, Berlin Heidelberg New York, 1997.

[12] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, Dependent rounding and its applications to approximation algorithms, J. ACM 53 (2006) 324–360.

[13] P. Giannopoulos, C. Knauer, M. Wahlström, and D. Werner, Hardness and inapproximability results for discrepancy and related problems parameterized by the dimension. Work under submission.

[14] M. Gnewuch, A randomized algorithm to compute the star-discrepancy of a given n-point set, Manuscript, Kiel, 2007.

[15] M. Gnewuch, Bracketing numbers for axis-parallel boxes and applications to geometric discrepancy, J. Complexity 24 (2008) 154–172.

[16] M. Gnewuch, Construction of minimal bracketing covers for rectangles, Electron. J. Combin. 15(1) (2008) Research Paper 95.

[17] M. Gnewuch, A. Srivastav, C. Winzen, Finding optimal volume subintervals with k points and calculating the star discrepancy are NP-hard problems, J. Complexity 25 (2009) 115–127.

[18] J. M. Hammersley, Monte Carlo methods for solving multivariable problems, Ann. New York Acad. Sci. 86 (1960) 844–874.

[19] S. Heinrich, E. Novak, G. W. Wasilkowski, H. Woźniakowski, The inverse of the star-discrepancy depends linearly on the dimension, Acta Arith. 96 (2001) 279–302.

[20] F. J. Hickernell, I. H. Sloan, G. W. Wasilkowski, On tractability of weighted integration over bounded and unbounded regions in $\mathbb{R}^s$, Math. Comp. 73 (2004) 1885–1901.

[21] A. Hinrichs, Covering numbers, Vapnik-Červonenkis classes and bounds for the star-discrepancy, J. Complexity 20 (2004) 477–483.

[22] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, SIAM, Philadelphia, 1992.

[23] H. Niederreiter, C. Xing, Low-discrepancy sequences and global function fields with many rational places, Finite Fields Appl. 2 (1996) 241–273.

[24] A. Panconesi, A. Srinivasan, Randomized distributed edge coloring via an extension of the Chernoff–Hoeffding bounds, SIAM J. Comput. 26 (1997) 350–368.

[25] P. Raghavan, Probabilistic construction of deterministic algorithms: Approximating packing integer programs, J. Comput. Syst. Sci. 37 (1988) 130–143.

[26] I. M. Sobol, The distribution of points in a cube and the approximate evaluation of integrals, Zh. Vychisl. Mat. i Mat. Fiz. 7 (1967) 784–802. (Russian)

[27] F. Kuo, Sobol sequence generator. Retrieved Dec 1, 2009 from `http://web.maths.unsw.edu.au/~fkuo/sobol/index.html`.

[28] A. Srinivasan, Distributions on level-sets with applications to approximations algorithms, in: Proceedings of FOCS'01, 2001, pp. 588–597.

[29] A. Srivastav, P. Stangier, Algorithmic Chernoff-Hoeffding inequalities in integer programming, Random Structures Algorithms 8 (1996) 27–58.

[30] E. Thiémard, Economic generation of low-discrepancy sequences with a b-ary Gray code, in: EPFL-DMA-ROSO, RO981201, 1998, available at `http://rosowww.epfl.ch/papers/grayfaure/`.

[31] E. Thiémard, An algorithm to compute bounds for the star discrepancy. J. Complexity 17 (2001) 850-880.

[32] P. Winker, K. T. Fang, Application of threshold-accepting to the evaluation of the discrepancy of a set of points, SIAM J. Numer. Anal. 34 (1997) 2028-2042.

[33] C. Winzen, Approximative Berechnung der Sterndiskrepanz, Diploma thesis [German], Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Kiel, 2007.