

Calculation of Discrepancy Measures and Applications

Carola Doerr, Michael Gnewuch and Magnus Wahlström

Abstract In this book chapter we survey known approaches and algorithms to compute discrepancy measures of point sets. After providing an introduction which puts the calculation of discrepancy measures in a more general context, we focus on the geometric discrepancy measures for which computation algorithms have been designed. In particular, we explain methods to determine L_2 -discrepancies and approaches to tackle the inherently difficult problem to calculate the star discrepancy of given sample sets. We also discuss in more detail three applications of algorithms to approximate discrepancies.

1 Introduction and Motivation

In many applications it is of interest to measure the quality of certain point sets, e.g., to test whether successive pseudo-random numbers are statistically independent, see, e.g., [82, 87, 99], or whether certain sample sets are suitable for multivariate numerical integration of certain classes of integrands, see, e.g., [25]. Other areas where the need of such measurements may occur include the generation of low-discrepancy samples, the design of computer experiments, computer graphics, and stochastic programming. (We shall describe some of these applications in more detail in Section 5.) A particularly useful class of quality measures, on which we want to focus in this book chapter, is the class of discrepancy measures. Several different

Carola Doerr
Université Paris Diderot - Paris 7, LIAFA, Paris, France and
Max Planck Institute for Informatics, Saarbrücken, Germany, e-mail: winzen@mpi-inf.mpg.de

Michael Gnewuch
School of Mathematics and Statistics, University of New South Wales, Sydney, NSW, 2052, Australia, e-mail: m.gnewuch@unsw.edu.au

Magnus Wahlström
Max Planck Institute for Informatics, Saarbrücken, Germany, e-mail: wahl@mpi-inf.mpg.de

discrepancy measures are known. Some of them allow for an efficient evaluation, others are hard to evaluate in practice. We shall give several examples below, but before doing so, let us provide a rather general definition of a geometric discrepancy measure.

Let (M, Σ) be a measurable space. Now let us consider two measures μ and ν defined on the σ -algebra Σ of M . A typical situation would be that μ is a rather complicated measure, e.g., a continuous measure or a discrete measure supported on a large number of atoms, and ν is a simpler measure, e.g., a discrete probability measure with equal probability weights or a discrete (signed) measure supported on a small number of atoms. We are interested in approximating μ by the simpler object ν in some sense and want to quantify the approximation quality. This can be done with the help of an appropriate discrepancy measure.

Such situations occur, e.g., in numerical integration, where one has to deal with a continuous measure μ to evaluate integrals of the form $\int_M f d\mu$ and wants to approximate these integrals with the help of a quadrature formula

$$Qf = \sum_{i=1}^n v_i f(x^{(i)}) = \int_M f d\nu; \quad (1)$$

here $\nu = \nu_Q$ denotes the discrete signed measure $\nu(A) = \sum_{i=1}^n v_i 1_A(x^{(i)})$, with 1_A being the characteristic function of $A \in \Sigma$. Another instructive example is scenario reduction in stochastic programming, which will be discussed in more detail in Section 5.3.

To quantify the discrepancy of μ and ν one may select a subset \mathcal{B} of the σ -algebra Σ , the class of *test sets*, to define the *local discrepancy* of μ and ν in a test set $B \in \mathcal{B}$ as

$$\Delta(B; \mu, \nu) := \mu(B) - \nu(B),$$

and the *geometric L_∞ -discrepancy* of μ and ν with respect to \mathcal{B} as

$$\text{disc}_\infty(\mathcal{B}; \mu, \nu) := \sup_{B \in \mathcal{B}} |\Delta(B; \mu, \nu)|. \quad (2)$$

Instead of considering the geometric L_∞ -discrepancy, i.e., the supremum norm of the local discrepancy, one may prefer to consider different norms of the local discrepancy. If, e.g., the class of test sets \mathcal{B} is endowed with a σ -algebra $\Sigma(\mathcal{B})$ and a probability measure ω on $\Sigma(\mathcal{B})$, and the restrictions of μ and ν to \mathcal{B} are measurable functions, then one can consider for $p \in (0, \infty)$ the *geometric L_p -discrepancy* with respect to \mathcal{B} , defined by

$$\text{disc}_p(\mathcal{B}; \mu, \nu) := \left(\int_{\mathcal{B}} |\Delta(B; \mu, \nu)|^p d\omega(B) \right)^{1/p}.$$

In some cases other norms of the local discrepancy may be of interest, too.

In the remainder of this chapter we restrict ourselves to considering discrete measures of the form

$$v(B) = \sum_{i=1}^n v_i 1_B(x^{(i)}), \quad \text{where } v_1, \dots, v_n \in \mathbb{R} \text{ and } x^{(1)}, \dots, x^{(n)} \in M. \quad (3)$$

In the case where $v_i = 1/n$ for all i , the quality of the probability measure $v = v_X$ is completely determined by the quality of the “sample points” $x^{(1)}, \dots, x^{(n)}$. The case where not all v_i are equal to $1/n$ is of considerable interest for numerical integration or stochastic programming (see Section 5.3). As already mentioned above, in the case of numerical integration it is natural to relate a quadrature rule Qf as in (1) to the signed measure $v = v_Q$ in (3). The quality of the quadrature Q is then determined by the sample points $x^{(1)}, \dots, x^{(n)}$ and the (integration) weights v_1, \dots, v_n .

Let us provide a list of examples of specifically interesting discrepancy measures.

- *Star discrepancy.* Consider the situation where $M = [0, 1]^d$ for some $d \in \mathbb{N}$, Σ is the usual σ -algebra of Borel sets of M , and μ is the d -dimensional Lebesgue measure λ^d on $[0, 1]^d$. Furthermore, let \mathcal{C}_d be the class of all axis-parallel half-open boxes anchored in zero $[0, y) = [0, y_1) \times \dots \times [0, y_d)$, $y \in [0, 1]^d$. Then the L_∞ -star discrepancy of the finite sequence $X = (x^{(i)})_{i=1}^n$ in $[0, 1]^d$ is given by

$$d_\infty^*(X) := \text{disc}_\infty(\mathcal{C}_d; \lambda^d, v_X) = \sup_{C \in \mathcal{C}_d} |\Delta(C; \lambda^d, v_X)|,$$

where

$$v_X(C) := \frac{1}{n} \sum_{i=1}^n 1_C(x^{(i)}) \quad \text{for all } C \in \Sigma. \quad (4)$$

Thus v_X is the counting measure that simply counts for given Borel sets C the number of points of X contained in C . The star discrepancy is probably the most extensively studied discrepancy measure. Important results about the star discrepancy and its relation to numerical integration can, e.g., be found in [5, 24, 37, 94, 99] or the book chapters [7, 16, 25].

Since we can identify \mathcal{C}_d with $[0, 1]^d$ via the mapping $[0, y) \mapsto y$, we may choose $\Sigma(\mathcal{C}_d)$ as the σ -algebra of Borel sets of $[0, 1]^d$ and the probability measure ω on $\Sigma(\mathcal{C}_d)$ as λ^d . Then for $1 \leq p < \infty$ the L_p -star discrepancy is given by

$$d_p^*(X) := \text{disc}_p(\mathcal{C}_d; \lambda^d, v_X) = \left(\int_{[0, 1]^d} \left| y_1 \cdots y_d - \frac{1}{n} \sum_{i=1}^n 1_{[0, y)}(x^{(i)}) \right|^p dy \right)^{1/p}.$$

In the last few years also norms of the local discrepancy function different from L_p -norms have been studied in the literature, such as suitable Besov, Triebel-Lizorkin, Orlicz, and BMO¹ norms, see, e.g., [8, 68, 69, 91, 92, 132] and the book chapter [7].

The star discrepancy is easily generalized to general measures v . For an application that considers measures v different from v_X see Section 5.3.

Notice that the point 0 plays a distinguished role in the definition of the star discrepancy. That is why 0 is often called the anchor of the star discrepancy.

¹ BMO stands for “bounded mean oscillation”.

There are discrepancy measures on $[0, 1]^d$ similar to the star discrepancy that rely also on axis-parallel boxes and on an anchor different from 0, such as the *centered discrepancy* [65] or *quadrant discrepancies* [66, 102]. Such kind of discrepancies are, e.g., discussed in more detail in [101, 103].

- *Extreme discrepancy.* The extreme discrepancy is also known under the names *unanchored discrepancy* and *discrepancy for axis-parallel boxes*. Its definition is analogue to the definition of the star discrepancy, except that we consider the class of test sets \mathcal{R}_d of all half-open axis-parallel boxes $[y, z) = [y_1, z_1) \times \cdots \times [y_d, z_d)$, $y, z \in [0, 1]^d$. We may identify this class with the subset $\{(y, z) \mid y, z \in [0, 1]^d, y \leq z\}$ of $[0, 1]^{2d}$, and endow it with the probability measure $d\omega(y, z) := 2^d dy dz$. Thus the L_∞ -extreme discrepancy of the finite sequence $X = (x^{(i)})_{i=1}^n$ in $[0, 1]^d$ is given by

$$d_\infty^e(X) := \text{disc}_\infty(\mathcal{R}_d; \lambda^d, \nu_X) = \sup_{R \in \mathcal{R}_d} |\Delta(R; \lambda^d, \nu_X)|,$$

and for $1 \leq p < \infty$ the L_p -extreme discrepancy is given by

$$d_p^e(X) := \text{disc}_p(\mathcal{R}_d; \lambda^d, \nu_X) = \left(\int_{[0, 1]^d} \int_{[0, z)} \left| \prod_{i=1}^d (z_i - y_i) - \frac{1}{n} \sum_{i=1}^n 1_{[y, z)}(x^{(i)}) \right|^p 2^d dy dz \right)^{1/p}.$$

The L_2 -extreme discrepancy was proposed as a quality measure for quasi-random point sets in [96].

To avoid confusion, it should be mentioned that the term “extreme discrepancy” is used by some authors in a different way. Especially in the literature before 1980 the attribute “extreme” often refers to a supremum norm of a local discrepancy function, see, e.g., [83, 138]. Since the beginning of the 1990s several authors used the attribute “extreme” to refer to the set system of unanchored axis-parallel boxes, see, e.g., [96, 99, 103].

- *G-discrepancy.* The G - or G -star discrepancy is defined as the star discrepancy, except that the measure μ is in general not the d -dimensional Lebesgue measure λ^d on $[0, 1]^d$, but some probability measure given by a distribution function G via $\mu([0, x)) = G(x)$ for all $x \in [0, 1]^d$. This is

$$\text{disc}_\infty(\mathcal{C}_d; \mu, \nu_X) = \sup_{C \in \mathcal{C}_d} \left| G(x) - \frac{1}{n} \sum_{i=1}^n 1_C(x^{(i)}) \right|.$$

The G -discrepancy has applications in quasi-Monte Carlo sampling, see, e.g., [105]. Further results on the G -star discrepancy can, e.g., be found in [54].

- *Isotrope discrepancy.* Here we have again $M = [0, 1]^d$ and $\mu = \lambda^d$. As set of test sets we consider \mathcal{I}_d , the set of all closed convex subsets of $[0, 1]^d$. Then the *isotrope discrepancy* of a set X is defined as

$$\text{disc}_\infty(\mathcal{I}_d; \lambda^d, \nu_X) := \sup_{R \in \mathcal{I}_d} |\Delta(R; \lambda^d, \nu_X)|. \quad (5)$$

This discrepancy was proposed by Hlawka [71]. It has applications in probability theory and statistics and was studied further, e.g., in [4, 97, 113, 124, 143].

- *Hickernell's modified L_p -discrepancy.* For a finite point set $X \subset [0, 1]^d$ and a set of variables $u \subset \{1, \dots, d\}$ let X_u denote the orthogonal projection of X into the cube $[0, 1]^u$. Then for $1 \leq p < \infty$ the *modified L_p -discrepancy* [65] of the point set X is given by

$$D_p^*(X) := \left(\sum_{\emptyset \neq u \subseteq \{1, \dots, d\}} d_p^*(X_u)^p \right)^{1/p}, \quad (6)$$

and for $p = \infty$ by

$$D_\infty^*(X) := \max_{\emptyset \neq u \subseteq \{1, \dots, d\}} d_\infty^*(X_u). \quad (7)$$

In the case where $p = 2$ this discrepancy was already considered by Zaremba in [142]. We will discuss the calculation of the modified L_2 -discrepancy in Section 2 and present an application of it in Section 5.2.

The modified L_p -discrepancy is an example of a weighted discrepancy, which is the next type of discrepancy we want to present.

- *Weighted discrepancy measures.* In the last years *weighted discrepancy measures* have become very popular, especially in the study of tractability of multivariate and infinite-dimensional integration, see the first paper on this topic, [118], and, e.g., [25, 26, 53, 67, 89, 103] and the literature mentioned therein.

To explain the idea behind the weighted discrepancy let us confine ourselves to the case where $M = [0, 1]^d$ and $\nu = \nu_X$ is a discrete measure as in (4). (A more general definition of weighted geometric L_2 -discrepancy, which comprises in particular infinite-dimensional discrepancies, can be found in [53].) We assume that there exists a one-dimensional measure μ^1 and a system \mathcal{B}_1 of test sets on $[0, 1]$. For $u \subseteq \{1, \dots, d\}$ we define the product measure $\mu^u := \otimes_{j \in u} \mu^1$ and the system of test sets

$$\mathcal{B}_u := \left\{ B \subseteq [0, 1]^u \mid B = \prod_{j \in u} B_j, B_j \in \mathcal{B}_1 \right\}$$

on $[0, 1]^u$. Again we denote the projection of a set $X \subseteq [0, 1]^d$ to $[0, 1]^u$ by X_u . Put $\mathcal{B} := \mathcal{B}_{\{1, \dots, d\}}$ and $\mu := \mu^{\{1, \dots, d\}}$. Let $(\gamma_u)_{u \subseteq \{1, \dots, d\}}$ be a family of *weights*, i.e., of non-negative numbers. Then the *weighted L_∞ -discrepancy* $d_{\infty, \gamma}^*(X)$ is given by

$$d_{\infty, \gamma}^*(X) := \text{disc}_{\infty, \gamma}(\mathcal{B}; \mu, \nu_X) = \max_{\emptyset \neq u \subseteq \{1, \dots, d\}} \gamma_u \text{disc}_\infty(\mathcal{B}_u; \mu^u, \nu_{X_u}). \quad (8)$$

If furthermore there exists a probability measure $\omega = \omega^1$ on \mathcal{B}_1 , put $\omega^u := \otimes_{j \in u} \omega^1$ for $u \subseteq \{1, \dots, d\}$. The *weighted L_p -discrepancy* $d_{p, \gamma}^*(X)$ is then defined by

$$d_{p,\gamma}^*(X) := \text{disc}_{p,\gamma}(\mathcal{B}; \mu, \nu_X) = \left(\sum_{\emptyset \neq u \subseteq \{1, \dots, d\}} \gamma_u \text{disc}_p(\mathcal{B}_u; \mu^u, \nu_{X_u})^p \right)^{1/p},$$

where

$$\text{disc}_p(\mathcal{B}_u; \mu^u, \nu_{X_u})^p = \int_{\mathcal{B}_u} |\mu^u(B_u) - \nu_{X_u}(B_u)|^p d\omega^u(B_u).$$

Hence weighted discrepancies do not only measure the uniformity of a point set $X \subset [0, 1]^d$ in $[0, 1]^d$, but also take into consideration the uniformity of projections X_u of X in $[0, 1]^u$. Note that Hickernell's modified L_p -discrepancy, see (6), is a weighted L_p -star discrepancy for the particular family of weights $(\gamma_u)_{u \subseteq \{1, \dots, d\}}$ where $\gamma_u = 1$ for all u .

Other interesting discrepancy measures in Euclidean spaces as, e.g., discrepancies with respect to half-spaces, balls, convex polygons or rotated boxes, can be found in [5, 13, 17, 94] and the literature mentioned therein. A discrepancy measure that is defined on a flexible region, i.e., on a certain kind of parametrized variety $M = M(m)$, $m \in (0, \infty)$, of measurable subsets of $[0, 1]^d$, is the *central composite discrepancy* proposed in [19]. For discrepancy measures on manifolds as, e.g., the *spherical cap discrepancy*, we refer to [10, 25, 37] and the literature listed there.

There are further figures of merits known to measure the quality of points sets that are no geometric discrepancies in the sense of our definition. Examples include the classical and the dyadic *diaphony* [62, 145] or the figure of merit $R(z, n)$ [78, 99, 115], which are closely related to numerical integration (see also the comment at the beginning of Section 3.4). We do not discuss such alternative figures of merit here, but focus solely on geometric discrepancy measures. In fact, we confine ourselves to the discrepancies that can be found in the list above. The reason for this is simple: Although deep theoretical results have been published for other geometric discrepancies, there have been, to the best of our knowledge, no serious attempts to evaluate these geometric discrepancies efficiently. Efficient calculation or approximation algorithms were developed almost exclusively for discrepancies that are based on axis-parallel rectangles, such as the star, the extreme or the centered discrepancy, and weighted versions thereof. We briefly explain in the case of the isotrope discrepancy at the beginning of Section 3 the typical problem that appears if one wants to approximate other geometric discrepancies than those based on axis-parallel rectangles.

This book chapter is organized as follows: In Section 2 we consider L_2 -discrepancies. In Section 2.1 we explain why many of these discrepancies can be calculated exactly in a straightforward manner with $O(n^2 d)$ operations, where n denotes (as always) the number of points in X and d the dimension. In Section 2.2 we discuss some asymptotically faster algorithms which allow for an evaluation of the L_2 -star and related L_2 -discrepancies in time $O(n \log n)$ (where this time the constant in the big- O -notation depends on d). The problem of calculating L_2 -discrepancies is the one for which the fastest algorithms are available. As we will see in the following sections, for $p \neq 2$ there are currently no similarly efficient methods known.

In Section 3 we discuss the calculation of the L_∞ -star discrepancy, which is the most prominent discrepancy measure. To this discrepancy the largest amount of research has been devoted so far, both for theoretical and practical reasons. We remark on known and possible generalizations to other L_∞ -discrepancy measures. In Section 3.1 we present elementary algorithms to calculate the star discrepancy exactly. These algorithms are beneficial in low dimensions, but clearly suffer from the curse of dimensionality. Nevertheless, the ideas used for these algorithms are fundamental for the following Subsections of Section 3. In Section 3.2 we discuss the more sophisticated algorithm of Dobkin, Eppstein and Mitchell, which clearly improves on the elementary algorithms. In Section 3.3 we review recent results about the complexity of exactly calculating the star discrepancy. These findings lead us to study approximation algorithms in Section 3.4. Here we present several different approaches.

In Section 4 we discuss the calculation of L_p -discrepancy measures for values of p other than 2 and ∞ . This Section is the shortest one in this book chapter, due to the relatively small amount of research that has been done on this topic.

In Section 5 we discuss three applications of discrepancy calculation and approximation algorithms in more detail. These applications are the quality testing of points (Section 5.1), the generation of low-discrepancy point sets via an optimization approach (Section 5.2), and scenario reduction in stochastic programming (Section 5.3). The purpose of this section is to show the reader more recent applications and to give her a feeling of typical instance sizes that can be handled and problems that may occur.

2 Calculation of L_2 -Discrepancies

L_2 -discrepancies are often used as quality measures for sets of sample points. One reason for this is the fact that geometric L_2 -discrepancies are equal to the worst-case integration error on corresponding reproducing kernel Hilbert spaces and the average-case integration error on corresponding larger function spaces, see the research articles [43, 53, 65, 67, 102, 118, 141, 142] or the surveys in [25, 103].

An additional advantage of the L_2 -star discrepancy and related L_2 -discrepancies is that they can be explicitly computed at cost $O(dn^2)$, see Section 2.1 below. Faster algorithms that are particularly beneficial for lower dimension d and larger number of points n will be presented in Section 2.2.

2.1 Warnock's Formula and Generalizations

It is easily verified by direct calculation that the L_2 -star discrepancy of a given n -point set $X = (x^{(i)})_{i=1}^n$ in dimension d can be calculated via Warnock's formula [135]

$$d_2^*(X) = \frac{1}{3^d} - \frac{2^{1-d}}{n} \sum_{i=1}^n \prod_{k=1}^d (1 - (x_k^{(i)})^2) + \frac{1}{n^2} \sum_{i,j=1}^n \prod_{k=1}^d \min\{1 - x_k^{(i)}, 1 - x_k^{(j)}\} \quad (9)$$

with $O(dn^2)$ arithmetic operations. As pointed out in [43, 93], the computation requires a sufficiently high precision, since the three terms in the formula are usually of a considerably larger magnitude than the resulting L_2 -star discrepancy. A remedy suggested by T. T. Warnock [136] is to subtract off the expected value of each summand in formula (9) (assuming that all coordinate values $x_k^{(i)}$ are uniformly and independently distributed) and to add it back at the end of the computation. This means we write down (9) in the equivalent form

$$\begin{aligned} d_2^*(X) = & \frac{1}{n} \left[\frac{1}{2^d} - \frac{1}{3^d} \right] - \frac{2^{1-d}}{n} \sum_{i=1}^n \left[\prod_{k=1}^d (1 - (x_k^{(i)})^2) - \left(\frac{2}{3} \right)^d \right] \\ & + \frac{1}{n^2} \left(\sum_{\substack{i,j=1 \\ i \neq j}}^n \left[\prod_{k=1}^d \min\{1 - x_k^{(i)}, 1 - x_k^{(j)}\} - \frac{1}{3^d} \right] + \sum_{i=1}^n \left[\prod_{k=1}^d (1 - x_k^{(i)}) - \frac{1}{2^d} \right] \right), \end{aligned} \quad (10)$$

and calculate first the terms inside the brackets [...] and sum them up afterwards. These terms are, in general, more well-behaved than the terms appearing in the original formula (9), and the additional use of Kahan summation [79] helps to further reduce rounding errors [136].

For other L_2 -discrepancies similar formulas can easily be deduced by direct calculation. So we have, e.g., that the extreme L_2 -discrepancy of X can be written as

$$\begin{aligned} d_2^e(X) = & \frac{1}{12^d} - \frac{2}{6^d n} \sum_{i=1}^n \prod_{k=1}^d (1 - (x_k^{(i)})^3) - (1 - x_k^{(i)})^3 \\ & + \frac{1}{n^2} \sum_{i,j=1}^n \prod_{k=1}^d \min\{x_k^{(i)}, x_k^{(j)}\} \min\{1 - x_k^{(i)}, 1 - x_k^{(j)}\}, \end{aligned} \quad (11)$$

cf. [60, Sect. 4], and the weighted L_2 -star discrepancy of X for *product weights* $\gamma_u = \prod_{j \in u} \gamma_j$, $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_d \geq 0$, as

$$\begin{aligned} d_{2,\gamma}^*(X) = & \prod_{k=1}^d \left(1 + \frac{\gamma_k^2}{3} \right) - \frac{2}{n} \sum_{i=1}^n \prod_{k=1}^d \left(1 + \gamma_k^2 \frac{1 - (x_k^{(i)})^2}{2} \right) \\ & + \frac{1}{n^2} \sum_{i,j=1}^n \prod_{k=1}^d \left(1 - \gamma_k^2 \min\{1 - x_k^{(i)}, 1 - x_k^{(j)}\} \right), \end{aligned} \quad (12)$$

cf. also [25, (6)]. In particular, the formula holds for the modified L_2 -discrepancy (6) that corresponds to the case where all weights γ_j , $j = 1, 2, \dots$, are equal to 1. Notice that formulas (11) and (12) can again be evaluated with $O(dn^2)$ arithmetic operations. In the case of the weighted L_2 -star discrepancy this is due to the sim-

ple structure of the product weights, whereas for an arbitrary family of weights $(\gamma_u)_{u \subseteq \{1, \dots, d\}}$ the cost of computing $d_{2, \gamma}^*(X)$ exactly will usually be of order $\Omega(2^d)$.

2.2 Asymptotically Faster Methods

For the L_2 -star discrepancy S. Heinrich [60] developed an algorithm which is asymptotically faster than the direct calculation of (9). For fixed d it uses at most $O(n \log^d n)$ elementary operations; here the implicit constant in the big- O -notation depends on d . This running time can be further reduced to $O(n \log^{d-1} n)$ by using a modification noted by K. Frank and S. Heinrich in [43].

Let us start with the algorithm from [60]. For a quadrature rule

$$Qf = \sum_{i=1}^n v_i f(x^{(i)}), \quad \text{with } v_i \in \mathbb{R} \text{ and } x^{(i)} \in [0, 1]^d \text{ for all } i,$$

we define the signed measure ν_Q by

$$\nu_Q(C) := Q(1_C) = \sum_{i=1}^n v_i 1_C(x^{(i)}) \quad \text{for arbitrary } C \in \mathcal{C}_d.$$

Then it is straightforward to calculate

$$\begin{aligned} d_2^*(Q) &:= \text{disc}_2(\mathcal{C}_d; \lambda^d, \nu_Q) \\ &= \frac{1}{3^d} - 2^{1-d} \sum_{i=1}^n v_i \prod_{k=1}^d (1 - (x_k^{(i)})^2) + \sum_{i,j=1}^n v_i v_j \prod_{k=1}^d \min\{1 - x_k^{(i)}, 1 - x_k^{(j)}\}. \end{aligned} \tag{13}$$

If we are interested in evaluating this generalized version of (9) in time $O(n \log^d n)$ or $O(n \log^{d-1} n)$, it obviously only remains to take care of the efficient calculation of

$$\sum_{i,j=1}^n v_i v_j \prod_{k=1}^d \min\{y_k^{(i)}, y_k^{(j)}\}, \quad \text{where } y_k^{(i)} := 1 - x_k^{(i)} \text{ for } i = 1, \dots, n.$$

In the course of the algorithm we have actually to take care of a little bit more general quantities: Let $A = ((v_i, y^{(i)}))_{i=1}^n$ and $B = ((w_j, z^{(j)}))_{j=1}^m$, where $n, m \in \mathbb{N}$, $v_i, w_j \in \mathbb{R}$ and $y^{(i)}, z^{(j)} \in [0, 1]^d$ for all i, j . Put

$$D(A, B, d) := \sum_{i=1}^n \sum_{j=1}^m v_i w_j \prod_{k=1}^d \min\{y_k^{(i)}, z_k^{(j)}\}.$$

We allow also $d = 0$, in which case we use the convention that the “empty product” is equal to 1.

The algorithm is based on the following observation: If $d \geq 1$ and $y_d^{(i)} \leq z_d^{(j)}$ for all $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$, then

$$D(A, B, d) = \sum_{i=1}^n \sum_{j=1}^m (v_i y_d^{(i)}) w_j \prod_{k=1}^{d-1} \min\{y_k^{(i)}, z_k^{(j)}\} = D(\tilde{A}, \tilde{B}, d-1),$$

where $\tilde{A} = ((\tilde{v}_i, \tilde{y}^{(i)}))_{i=1}^n$, $\tilde{B} = ((w_i, \tilde{z}^{(i)}))_{i=1}^m$ with $\tilde{v}_i = (v_i y_d^{(i)})$ and $\tilde{y}^{(i)} = (y_k^{(i)})_{k=1}^{d-1}$ and $\tilde{z}^{(i)} = (z_k^{(i)})_{k=1}^{d-1}$. Hence we have reduced the dimension parameter d by 1. But in the case where $d = 0$, we can simply calculate

$$D(A, B, 0) = \left(\sum_{i=1}^n v_i \right) \left(\sum_{j=1}^m w_j \right) \quad (14)$$

with cost of order $O(n+m)$. This observation will be exploited by the algorithm proposed by Heinrich to calculate $D(A, B, d)$ for given $d \geq 1$ and arrays A and B as above.

We describe here the version of the algorithm proposed in [60, Sect. 2]; see also [93, Sect. 5]. Let μ denote the median of the d th components of the points $y^{(i)}$, $i = 1, \dots, n$, from A . Then we split A up into two smaller arrays A_L and A_R with A_L containing $\lfloor n/2 \rfloor$ points $y^{(i)}$ (and corresponding weights v_i) satisfying $y_d^{(i)} \leq \mu$ and A_R containing the remaining $\lceil n/2 \rceil$ points (and corresponding weights) satisfying $y_d^{(i)} \geq \mu$. Similarly, we split up B into the two smaller arrays B_L and B_R that contains the points (and corresponding weights) from B whose d th components are less or equal than μ and greater than μ , respectively.

Since we may determine μ with the help of a *linear-time median-finding algorithm* in time $O(n)$ (see, e.g., [1, Ch. 3]), the whole partitioning procedure can be done in time $O(n+m)$. With the help of this partitioning we can exploit the basic idea of the algorithm to obtain

$$D(A, B, d) = D(A_L, B_L, d) + D(A_R, B_R, d) + D(\tilde{A}_L, \tilde{B}_R, d-1) + D(\tilde{A}_R, \tilde{B}_L, d-1), \quad (15)$$

where, as above, \tilde{A}_L is obtained from A_L by deleting the d th component of the points $y^{(i)}$ and substituting the weights v_i by $v_i y_d^{(i)}$, and \tilde{A}_R is obtained from A_R by also deleting the d th component of the points $y^{(i)}$, but keeping the weights v_i . In an analogous way we obtain \tilde{B}_L and \tilde{B}_R , respectively.

The algorithm uses the step (15) recursively in a *divide-and-conquer* manner. The “conquer” part consists of three base cases, which are solved directly.

The first base case is $m = 0$; i.e., $B = \emptyset$. Then $D(A, B, d) = 0$.

The second one is the case $d = 0$ already discussed above, where we simply use formula (14) for the direct calculation of $D(A, B, 0)$ at cost at most $O(n+m)$.

The third base case is the case where $|A| = 1$. Then we can compute directly

$$D(A, B, d) = v_1 \sum_{j=1}^m w_j \prod_{k=1}^d \min\{y_k^{(1)}, z_k^{(j)}\}.$$

This computation costs at most $O(m)$.

An inductive cost analysis reveals that the cost of this algorithm to calculate $D(A, B, d)$ is of order $O((n+m)\log^d(n+1))$, see [60, Prop. 1]. As already said, the implicit constant in the big- O -notation depends on d . J. Matoušek provided in [93] a running time analysis of Heinrich's algorithm that also takes care of its dependence on the dimension d and compared it to the cost of the straightforward calculation of (13). From this analysis one can conclude that Heinrich's algorithm reasonably outperforms the straightforward method if n is larger than (roughly) 2^{2d} ; for details see [93, Sect. 5]. Moreover, [60, 93] contain modifications of the algorithm and remarks on a practical implementation. Furthermore, Heinrich provides some numerical examples with the number of points ranging from 1,024 to 65,536 in dimensions ranging from 1 to 8 comparing the actual computational effort of his method and the direct calculation of (13), see [60, Sect. 5]. In these examples his method was always more efficient than performing the direct calculation; essentially, the advantage grows if the number of points increases, but shrinks if the dimension increases.

As pointed out by Heinrich in [60, Sect. 4], his algorithm can be modified easily to calculate L_2 -extreme discrepancies instead of L_2 -star discrepancies with essentially the same effort. Furthermore, he describes how to generalize his algorithm to calculate " r -smooth" L_2 -discrepancies, which were considered in [108, 125]. (Here the smoothness parameter r is a non-negative integer. If $r = 0$, we regain the L_2 -star discrepancy. If $r > 0$, then the r -smooth discrepancy is actually not any more a geometric L_2 -discrepancy in the sense of our definition given in Section 1.)

Heinrich's algorithm can be accelerated with the help of the following observation from [43]: Instead of employing (14) for the base case $D(A, B, 0)$, it is possible to evaluate already the terms $D(A, B, 1)$ that occur in the course of the algorithm. If we want to calculate $D(A, B, 1)$, we assume that the elements $y^{(1)}, \dots, y^{(n)} \in [0, 1]$ from A and $z^{(1)}, \dots, z^{(m)} \in [0, 1]$ from B are already in increasing order. This can be ensured by using a standard sorting algorithm to preprocess the input at cost $O((n+m)\log(n+m))$. Now we determine for each i an index $v(i)$ such that $y^{(i)} \geq z^{(j)}$ for $j = 1, \dots, v(i)$ and $y^{(i)} < z^{(j)}$ for $j = v(i) + 1, \dots, m$. If this is done successively, starting with $v(1)$, then this can be done at cost $O(n+m)$. Then

$$D(A, B, 1) = \sum_{i=1}^n v_i \left(\sum_{j=1}^{v(i)} w_j z^{(j)} + y^{(i)} \sum_{j=v(i)+1}^m w_j \right), \quad (16)$$

and the right hand side can be computed with $O(n+m)$ operations if the inner sums are added up successively. Thus the explicit evaluation of $D(A, B, 1)$ can be done at total cost $O(n+m)$. Using this new base case (16) instead of (14) reduces the running time of the algorithm to $O((n+m)\log^{d-1}(n+1))$ (as can easily be checked by adapting the proof of [60, Prop. 1]).

The original intention of the paper [43] is in fact to efficiently calculate the L_2 -star discrepancies (13) of *Smolyak quadrature rules*. These quadrature rules are also known under different names as, e.g., *sparse grid methods* or *hyperbolic cross points*, see, e.g., [47, 100, 119, 125, 137, 144] and the literature mentioned therein. Frank and Heinrich exploit that a d -dimensional Smolyak quadrature rule is uniquely determined by a sequence of one-dimensional quadratures, and in the special case of composite quadrature rules even by a single one-dimensional quadrature. Their algorithm computes the L_2 -star discrepancies of Smolyak quadratures at cost $O(N \log^{2-d} N + d \log^4 N)$ for a general sequence of one-dimensional quadratures and at cost $O(d \log^4 N)$ in the special case of composite quadrature rules; here N denotes the number of quadrature points used by the d -dimensional Smolyak quadrature. This time the implicit constants in the big- O -notation do not depend on the dimension d . With the help of their algorithm Frank and Heinrich are able to calculate the L_2 -star discrepancy for extremely large numbers of integration points as, e.g., roughly 10^{35} points in dimension $d = 15$. For the detailed description of the algorithm and numerical experiments we refer to [43].

Notice that both algorithms from [43, 60] use as a starting point formula (13). Since the three summands appearing in (13) are of similar size, the algorithms should be executed with a sufficiently high precision to avoid cancellation effects.

Notes

Related to the problem of calculating L_2 -discrepancies of given point sets is the problem of computing the smallest possible L_2 -discrepancy of all point sets of a given size n . For the L_2 -star discrepancy and arbitrary dimension d the smallest possible discrepancy value of all point sets of size n was derived in [111] for $n = 1$ and in [85] for $n = 2$.

Regarding the L_2 -star discrepancy, one should mention that this discrepancy can be a misleading measure of uniformity for sample sizes n smaller than 2^d . For instance, Matoušek pointed out that for small n the pathological point set that consists of n copies of the point $(1, \dots, 1)$ in $[0, 1]^d$ has almost the best possible L_2 -star discrepancy, see [93, Sect. 2]. A possible remedy is to consider a weighted version of the L_2 -star discrepancy instead, as, e.g., the modified L_2 -discrepancy.

Matoušek's observation may also be interpreted in the context of numerical integration. The L_2 -star discrepancy is equal to the worst-case error of quasi-Monte Carlo (QMC) integration on the unanchored Sobolev space. More precisely, we have for a finite sequence $X = (x^{(i)})_{i=1}^n$ in $[0, 1]^d$ that

$$d_2^*(X) = \sup_{f \in B} \left| \int_{[0,1]^d} f(x) \, dx - Q(f) \right|,$$

where B is the norm unit ball of the unanchored Sobolev space and Q is the QMC algorithm

$$Q(f) = \frac{1}{n} \sum_{i=1}^n f(x^{(i)});$$

see, e.g., [103, Chapter 9]. Now Matoušek's observation indicates that if for given n smaller than 2^d one is interested in minimizing the worst-case integration error with the help of a general quadrature rule of the form (1), then one should not use QMC algorithms with equal integration weights $1/n$. In fact, already normalized QMC algorithms with suitably chosen equal integration weights $a = a(n, d) < 1/n$ as stated in [103, (10.12)] improve over conventional QMC algorithms with weights $1/n$; for a detailed discussion see [103, Sect. 10.7.6]. This suggests that for n smaller than 2^d the L_2 -star discrepancy modified by substituting the factor $1/n$ by $a < 1/n$ from [103, (10.12)] may be a better measure of uniformity than the L_2 -star discrepancy itself.

3 Calculation of L_∞ -Discrepancies

In this section we survey algorithms which can be used to calculate or approximate the L_∞ -star discrepancy. Most of these algorithms have a straightforward extension to other " L_∞ -rectangle discrepancies", as, e.g., to the extreme discrepancy discussed above, the centered discrepancy [65], or other quadrant discrepancies [66, 103]. Algorithms for the L_∞ -star discrepancy are also necessary to compute or estimate weighted L_∞ -star discrepancies. Let us, e.g., assume that we are interested in finding tight upper or lower bounds for the weighted L_∞ -discrepancy $d_{\infty, \gamma}^*(X)$, as defined in (8). Then we may divide the family of weights $(\gamma_u)_{u \subseteq \{1, \dots, d\}}$ into a set S of suitably small weights and a set L of larger weights and use the fact that the star discrepancy has the following monotonicity behavior with respect to the dimension: If $u \subseteq v$, then $d_\infty^*(X_u) \leq d_\infty^*(X_v)$. We can use the algorithms discussed below to calculate or bound the discrepancies $d_\infty^*(X_u)$, $u \in L$. The remaining discrepancies $d_\infty^*(X_v)$, $v \in S$, corresponding to the less important weights can be upper-bounded simply by 1 and lower-bounded by $\max_{u \in L; u \subset v} d_\infty^*(X_u)$ (or even by 0 if the weights are negligible small).

In general, it is not easy to calculate L_∞ -discrepancies as defined in (2); the cardinality of the system \mathcal{B} of test sets is typically infinite. Since we obviously cannot compute the local discrepancies for an infinite number of test boxes, we usually have to find a finite subset $\mathcal{B}_\delta \subset \mathcal{B}$ such that $\text{disc}_\infty(\mathcal{B}; \mu, \nu) = \text{disc}_\infty(\mathcal{B}_\delta; \mu, \nu)$ or at least $\text{disc}_\infty(\mathcal{B}; \mu, \nu) \leq \text{disc}_\infty(\mathcal{B}_\delta; \mu, \nu) + \delta$ for sufficiently small δ . (This "discretization method" is also important for finding upper bounds for the best possible discrepancy behavior with the help of *probabilistic proofs*, see, e.g., [3] or the book chapter [52].) For most systems \mathcal{B} of test sets this is not a trivial task. If one is, for instance, interested in the isotrope discrepancy, see (5), it is not completely obvious to see how the system \mathcal{I}_d can be substituted by a finite set system that leads to a (arbitrarily) close approximation of $\text{disc}_\infty(\mathcal{I}_d; \lambda^d, \nu_X)$. In [97] H. Niederreiter pointed out that it is sufficient to consider the smaller system of test sets \mathcal{E}_d of all

open and closed polytopes P contained in $[0, 1]^d$ with the property that each face of P is lying entirely on the boundary of $[0, 1]^d$ or contains a point of X . Note that \mathcal{E}_d still consists of infinitely many test sets and that further work has to be done before this observation can be used for a concrete algorithm to approximate the isotrope discrepancy.

For the star discrepancy it is easier to find useful discretizations, as we will show below.

3.1 Calculating the Star Discrepancy in Low Dimension

Let us have a closer look at the problem of calculating the L_∞ -star discrepancy: Let $X = (x^{(i)})_{i=1}^n$ be some fixed finite sequence in $[0, 1]^d$. For convenience we introduce for an arbitrary point $y \in [0, 1]^d$ the short-hands

$$V_y := \prod_{i=1}^d y_i,$$

and

$$A(y, X) := \sum_{i=1}^n 1_{[0, y)}(x^{(i)}), \quad \text{as well as} \quad \bar{A}(y, X) := \sum_{i=1}^n 1_{[0, y]}(x^{(i)}),$$

i.e., V_y is the volume of the test box $[0, y)$, $A(y, X)$ the number of points of X lying inside the half-open box $[0, y)$, and $\bar{A}(y, X)$ the number of points of X lying in the closed box $[0, y]$. Let us furthermore set

$$\delta(y, X) := V_y - \frac{1}{n}A(y, X) \quad \text{and} \quad \bar{\delta}(y, X) := \frac{1}{n}\bar{A}(y, X) - V_y.$$

Putting $\delta^*(y, X) := \max\{\delta(y, X), \bar{\delta}(y, X)\}$, we have

$$d_\infty^*(X) = \sup_{y \in [0, 1]^d} \delta^*(y, X).$$

We define for $j \in \{1, \dots, d\}$

$$\Gamma_j(X) := \{x_j^{(i)} \mid i \in \{1, \dots, n\}\} \quad \text{and} \quad \bar{\Gamma}_j(X) := \Gamma_j(X) \cup \{1\},$$

and put

$$\Gamma(X) := \Gamma_1(X) \times \dots \times \Gamma_d(X) \quad \text{and} \quad \bar{\Gamma}(X) := \bar{\Gamma}_1(X) \times \dots \times \bar{\Gamma}_d(X).$$

We refer to $\Gamma(X)$ and to $\bar{\Gamma}(X)$ as *grids induced by X* .

Lemma 1. *Let $X = (x^{(i)})_{i=1}^n$ be a sequence in $[0, 1]^d$. Then*

$$d_{\infty}^*(X) = \max \left\{ \max_{y \in \overline{\Gamma}(X)} \delta(y, X), \max_{y \in \Gamma(X)} \overline{\delta}(y, X) \right\}. \quad (17)$$

Formulas similar to (17) can be found in several places in the literature—the first reference we are aware of is [97, Thm. 2].

Proof. Consider an arbitrary test box $[0, y)$, $y \in [0, 1]^d$, see Figure 1.

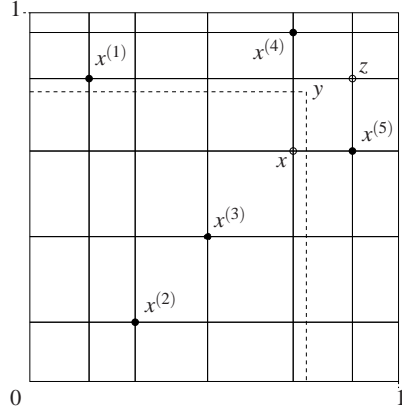


Fig. 1 Some set $X = (x^{(i)})_{i=1}^5$ in $[0, 1]^2$, a test box $[0, y)$, and $x \in \Gamma(X)$, $z \in \overline{\Gamma}(X)$ with $x \leq y \leq z$.

Then for every $j \in \{1, \dots, d\}$ we find a maximal $x_j \in \Gamma_j(X) \cup \{0\}$ and a minimal $z_j \in \overline{\Gamma}(X)$ satisfying $x_j \leq y_j \leq z_j$. Put $x = (x_1, \dots, x_d)$ and $z = (z_1, \dots, z_d)$. We get the inequalities

$$V_y - \frac{1}{n}A(y, X) = V_y - \frac{1}{n}A(z, X) \leq V_z - \frac{1}{n}A(z, X),$$

and

$$\frac{1}{n}A(y, X) - V_y = \frac{1}{n}\overline{A}(x, X) - V_y \leq \frac{1}{n}\overline{A}(x, X) - V_x.$$

Observing that $\overline{A}(x, X) = 0$ if $x_j = 0 \notin \Gamma_j(X)$ for some $j \in \{1, \dots, d\}$, we see that the right hand side of (17) is at least as large as $d_{\infty}^*(X)$.

Let us now show that it cannot be larger than $d_{\infty}^*(X)$. So let $y \in \Gamma(X)$ be given. Then we may consider for a small $\varepsilon > 0$ the vector $y(\varepsilon)$, defined by $y(\varepsilon)_j = \min\{y_j + \varepsilon, 1\}$ for $j = 1, \dots, d$. Clearly, $y(\varepsilon) \in [0, 1]^d$ and

$$\lim_{\varepsilon \rightarrow 0} \left(\frac{1}{n}A(y(\varepsilon), X) - V_{y(\varepsilon)} \right) = \frac{1}{n}\overline{A}(y, X) - V_y.$$

These arguments show that (17) is valid. \square

Lemma 1 shows that an enumeration algorithm would provide us with the exact value of $d_{\infty}^*(X)$. But since the cardinality of $\Gamma(X)$ for almost all X is n^d , such an

algorithm would be infeasible for large values of n and d . Indeed, for a random n -point set X we have almost surely $|\Gamma(X)| = n^d$, resulting in $\Omega(n^d)$ test boxes that we have to take into account to calculate (17). This underlines that (17) is in general impractical if n and d are large. There are some more sophisticated methods known to calculate the star discrepancy, which are especially helpful in low dimensions. In the one-dimensional case H. Niederreiter derived the following formula, see [97] or [98].

Theorem 1 ([97, Thm. 1]). *Let $X = (x^{(i)})_{i=1}^n$ be a sequence in $[0, 1]$. If $x^{(1)} \leq x^{(2)} \leq \dots \leq x^{(n)}$, then*

$$d_{\infty}^*(X) = \max_{i=1}^n \max \left\{ \frac{i}{n} - x^{(i)}, x^{(i)} - \frac{i-1}{n} \right\} = \frac{1}{2n} + \max_{i=1}^n \left| x^{(i)} - \frac{2i-1}{2n} \right|. \quad (18)$$

Proof. The first identity follows directly from (17), since for $y = 1 \in \overline{\Gamma}(X)$ we have $V_y - \frac{1}{n}A(y, X) = 0$. The second identity follows from

$$\max \left\{ \frac{i}{n} - x^{(i)}, x^{(i)} - \frac{i-1}{n} \right\} = \left| x^{(i)} - \frac{2i-1}{2n} \right| + \frac{1}{2n}, \quad i = 1, \dots, n.$$

□

Notice that (18) implies immediately that for $d = 1$ the set $\{\frac{1}{2n}, \frac{3}{2n}, \dots, \frac{2n-1}{2n}\}$ is the uniquely determined n -point set that achieves the minimal star discrepancy $1/2n$.

In higher dimension the calculation of the star discrepancy unfortunately becomes more complicated.

In dimension $d = 2$ a reduction of the number of steps to calculate (17) was achieved by L. De Clerk [22]. In [11] her formula was slightly extended and simplified by P. Bunschuh and Y. Zhu.

Theorem 2 ([22, Sect. II],[11, Thm. 1]). *Let $X = (x^{(i)})_{i=1}^n$ be a sequence in $[0, 1]^2$. Assume that $x_1^{(1)} \leq x_1^{(2)} \leq \dots \leq x_1^{(n)}$ and rearrange for each $i \in \{1, \dots, n\}$ the numbers $0, x_2^{(1)}, \dots, x_2^{(i)}, 1$ in increasing order and rewrite them as $0 = \xi_{i,0} \leq \xi_{i,1} \leq \dots \leq \xi_{i,i} \leq \xi_{i,i+1} = 1$. Then*

$$d_{\infty}^*(X) = \max_{i=0}^n \max_{k=0}^i \max \left\{ \frac{k}{n} - x_1^{(i)} \xi_{i,k}, x_1^{(i+1)} \xi_{i,k+1} - \frac{k}{n} \right\}. \quad (19)$$

The derivation of this formula is mainly based on the observation that in dimension $d \geq 2$ the discrepancy functions $\delta(\cdot, X)$ and $\overline{\delta}(\cdot, X)$ can attain their maxima only in some of the points $y \in \overline{\Gamma}(X)$ and $y \in \Gamma(X)$, respectively, which we shall call *critical points*. The formal definition we state here is equivalent to the one given in [56, Sect. 4.1].

Definition 1. Let $y \in [0, 1]^d$. The point y and the test box $[0, y)$ are $\delta(X)$ -critical, if we have for all $0 \neq \varepsilon \in [0, 1 - y_1] \times \dots \times [0, 1 - y_d]$ that $A(y + \varepsilon, X) > A(y, X)$. The

point y and the test box $[0, y]$ are $\overline{\delta}(X)$ -critical, if we have for all $\varepsilon \in [0, y) \setminus \{0\}$ that $\overline{A}(y - \varepsilon, X) < \overline{A}(y, X)$. We denote the set of $\delta(X)$ -critical points by $\mathcal{C}(X)$ and the set of $\overline{\delta}(X)$ -critical points by $\overline{\mathcal{C}}(X)$, and we put $\mathcal{C}^*(X) := \mathcal{C}(X) \cup \overline{\mathcal{C}}(X)$.

In Figure 1 we have, e.g., that $y = (x_1^{(5)}, x_2^{(4)})$ and $y' = (x_1^{(5)}, 1)$ are $\delta(X)$ -critical points, while $y'' = (x_1^{(5)}, x_2^{(3)})$ is not. Furthermore, $\overline{y} = (x_1^{(3)}, x_2^{(3)})$ and $\overline{y}' = (x_1^{(5)}, x_2^{(1)})$ are $\overline{\delta}(X)$ -critical points, while $\overline{y}'' = (x_1^{(4)}, x_2^{(2)})$ is not. This shows that, in contrast to the one-dimensional situation, for $d \geq 2$ not all points in $\Gamma(X)$ and $\overline{\Gamma}(X)$ are critical points.

With a similar argument as in the proof of Lemma 1, the following lemma can be established.

Lemma 2. *Let $X = (x^{(i)})_{i=1}^n$ be a sequence in $[0, 1]^d$. Then $\mathcal{C}(X) \subseteq \overline{\Gamma}(X)$ and $\overline{\mathcal{C}}(X) \subseteq \Gamma(X)$, as well as*

$$d_{\infty}^*(X) = \max \left\{ \max_{y \in \mathcal{C}(X)} \delta(y, X), \max_{y \in \overline{\mathcal{C}}(X)} \overline{\delta}(y, X) \right\}. \quad (20)$$

For a rigorous proof see [22, Sect. II] or [56, Lemma 4.1].

The set of critical test boxes may be subdivided further. For $j = 0, 1, \dots, n$ put

$$\mathcal{C}^k(X) := \{y \in \mathcal{C}(X) \mid A(y, X) = k\},$$

and

$$\overline{\mathcal{C}}^k(X) := \{y \in \overline{\mathcal{C}}(X) \mid \overline{A}(y, X) = k\}.$$

Then

$$d_{\infty}^*(X) = \max_{k=1}^n \max \left\{ \max_{y \in \mathcal{C}^{k-1}(X)} \delta(y, X), \max_{y \in \overline{\mathcal{C}}^k(X)} \overline{\delta}(y, X) \right\}, \quad (21)$$

see [22, Sect. II]. For $d = 2$ De Clerk characterized the points in $\mathcal{C}^k(X)$ and $\overline{\mathcal{C}}^k(X)$ and derived (19) under the assumption that $|\Gamma_j(X)| = n$ for $j = 1, \dots, d$ [22, Sect. II]. Bundschuh and Zhu got rid of this assumption, which resulted in a technically more involved proof [11, Sect. 2].

De Clerk used her formula (19) to provide explicit formulas for the discrepancies of *Hammersley point sets* [59] for arbitrary basis b and $n = b^m$, $m \in \mathbb{N}$, see [22, Sect. III]. Her results generalize the formulas for the star discrepancy of two-dimensional Hammersley sets in basis 2 provided by Halton and Zaremba [58]. To establish the explicit formula, she used a recursion property of two-dimensional Hammersley point sets X of size b^m [39] and the facts that these sets are symmetric with respect to the main diagonal of the unit square [109] and that their discrepancy function $\delta(\cdot, X)$ is never positive [45], which implies that for the calculation of the discrepancy of these sets one only has to consider $\overline{\delta}(X)$ -critical test boxes.

In dimension $d = 3$ Bundschuh and Zhu provided a formula similar to (19).

Theorem 3 ([11, Thm. 2]). *Let $X = (x^{(i)})_{i=1}^n$ be a sequence in $[0, 1]^3$. Put $x^{(0)} := (0, 0, 0)$ and $x^{(n+1)} := (1, 1, 1)$, and assume that $x_1^{(1)} \leq x_1^{(2)} \leq \dots \leq x_1^{(n)}$. For $i \in$*

$\{1, \dots, n\}$ rearrange the second components $x_2^{(0)}, x_2^{(1)}, \dots, x_2^{(i)}, x_2^{(n+1)}$ in increasing order and rewrite them as $0 = \xi_{i,0} \leq \xi_{i,1} \leq \dots \leq \xi_{i,i} \leq \xi_{i,i+1} = 1$ and denote the corresponding third components $x_3^{(i)}$, $i = 0, 1, \dots, i, n+1$ by $\tilde{\xi}_{i,0}, \tilde{\xi}_{i,1}, \dots, \tilde{\xi}_{i,i+1}$. Now for fixed i and $k = 0, 1, \dots, i$ rearrange $\tilde{\xi}_{i,0}, \tilde{\xi}_{i,1}, \dots, \tilde{\xi}_{i,k}, \tilde{\xi}_{i,i+1}$ and rewrite them as $0 = \eta_{i,k,0} \leq \eta_{i,k,1} \leq \dots \leq \eta_{i,k,k} \leq \eta_{i,k,k+1} = 1$. Then

$$d_\infty^*(X) = \max_{i=0}^n \max_{k=0}^i \max_{\ell=0}^k \max \left\{ \frac{k}{n} - x_1^{(i)} \xi_{i,k} \eta_{i,k,\ell}, x_1^{(i+1)} \xi_{i,k+1} \eta_{i,k,\ell+1} - \frac{k}{n} \right\}. \quad (22)$$

The method can be generalized to arbitrary dimension d and requires for generic point sets roughly $O(n^d/d!)$ elementary operations. This method was, e.g., used in [139] to calculate the exact discrepancy of particular point sets, so-called rank-1 lattice rules (cf. [24, 99, 115]), up to size $n = 236$ in dimension $d = 5$ and to $n = 92$ in dimension $d = 6$. But as pointed out by P. Winker and K.-T. Fang, for this method instances like, e.g., sets of size $n \geq 2000$ in $d = 6$ are infeasible. This method can thus only be used in a very limited number of dimensions.

A method that calculates the exact star discrepancy of a point set in a running time with a more favorable dependence on the dimension d , namely time $O(n^{1+d/2})$, was proposed by D. P. Dobkin, D. Eppstein, and D. P. Mitchell in [27]. We discuss this more elaborate algorithm in the next subsection.

3.2 The Algorithm of Dobkin, Eppstein, and Mitchell

In order to describe the algorithm of Dobkin, Eppstein, and Mitchell [27], we begin with a problem from computational geometry. In *Klee's measure problem*, the input is a set of n axis-parallel rectangles in \mathbb{R}^d , and the problem is to compute the volume of the union of the rectangles. The question of the best possible running time started with V. Klee [81], who gave an $O(n \log n)$ -time algorithm for the 1-dimensional case, and asked whether this was optimal; it was later found that this is the case [44]. The general case was considered by J. L. Bentley [6], who gave an $O(n^{d-1} \log n)$ -time algorithm for $d \geq 2$, thus (surprisingly) giving an algorithm for the $d = 2$ case with an asymptotic running time matching that of the $d = 1$ case. By the lower bound for $d = 1$, Bentley's algorithm is tight for $d = 2$, but as we shall see, not for $d \geq 3$.

The essential breakthrough, which also lies behind the result of [27], was given by Overmars and Yap [107]. They showed that \mathbb{R}^d can be partitioned (in an input-dependent way) into $O(n^{d/2})$ regions, where each region is an axis-parallel box, such that the intersection of the region with the rectangles of the input behaves in a particular regular way. Let us fix some terminology. Let $C = [a, b]$ be a region in the decomposition. A *slab* in C is an axis-parallel box contained in C which has full length in all but at most one dimension, i.e., a box $\prod_{j=1}^d I_j$ where $I_j = [a_j, b_j]$ for all but at most one j . A finite union of slabs is called a *trellis*. Overmars and Yap show the following result.

Theorem 4 ([107]). *Let d be a fixed dimension. Given a set of n rectangles in \mathbb{R}^d , there is a partitioning of the space into $O(n^{d/2})$ regions where for every region, the intersection of the region with the union of all rectangles of the input forms a trellis.*

An algorithm to enumerate this partition, and a polynomial-time algorithm for computing the volume of a trellis, would now combine into an $O(n^{d/2+c})$ -time algorithm for Klee's measure problem, for some constant $c > 0$; Overmars and Yap further improve it to $O(n^{d/2} \log n)$ time (and $O(n)$ space) by using dynamic data structures and careful partial evaluations of the decomposition. Recently, T. M. Chan [12] gave a slightly improved running time of $n^{d/2} 2^{O(\log^* n)}$, where $\log^* n$ denotes the iterated logarithm of n , i.e., the number of times the logarithm function must be iteratively applied before the result is less than or equal to 1.

While no direct reductions between the measure problem and discrepancy computation are known, the principle behind the above decomposition is still useful. Dobkin et al. [27] apply it via a dualization, turning each point x into an orthant (x, ∞) , and each box $B = [0, y)$ into a point y , so that x is contained in B if and only if the point y is contained in the orthant. The problem of star discrepancy computation can then be solved by finding, for each $i \in \{1, \dots, n\}$, the "largest" and "smallest" point $y \in \overline{F}(X)$ contained in at most respectively at least i orthants; here largest and smallest refer to the value V_y defined previously, i.e., the volume of the box $[0, y)$. Note that this is the dual problem to (21). As Dobkin et al. [27] show, the partitioning of \mathbb{R}^d of Overmars and Yap can be used for this purpose. In particular, the base case of a region B such that the intersections of the input rectangles with B form a trellis, corresponds for us to a case where for every point x there is at most one coordinate $j \in \{1, \dots, d\}$ such that for any $y \in B$ only the value of y_j determines whether $x \in [0, y)$. Given such a base case, it is not difficult to compute the maximum discrepancy relative to points $y \in B$ in polynomial time.

We now sketch the algorithm in some more detail, circumventing the dualization step for a more direct presentation. For simplicity of presentation, we are assuming that for each fixed $j \in \{1, \dots, d\}$, the coordinates $x_j^{(1)}, \dots, x_j^{(n)}$ are pairwise different. We will also ignore slight issues with points that lie exactly on a region boundary.

We will subdivide the space $[0, 1]^d$ recursively into regions of the form $[a_1, b_1] \times \dots \times [a_i, b_i] \times [0, 1]^{d-i}$; call this a *region at level i* . We identify a region with $[a, b]$; if the region is at level $i < d$, then for $j > i$ we have $a_j = 0$ and $b_j = 1$. For $j \in \{1, \dots, i\}$, we say that a point x is *internal in dimension j* , relative to a region $[a, b]$ at level i , if x is contained in $[0, b)$ and $a_j < x_j < b_j$. We will maintain two invariants as follows.

1. For every point x contained in the box $[0, b)$, there is at most one coordinate $j \in \{1, \dots, i\}$ such that x is internal in dimension j .
2. For any region at level $i > 0$ and any $j \in \{1, \dots, i\}$ there are only $O(\sqrt{n})$ points internal in dimension j .

Once we reach a region at level d , called a *cell*, the first condition ensures that we reach the above-described base case, i.e., that every point contained in the box $[0, b)$ is internal in at most one dimension. The second condition, as we will see, ensures that the decomposition can be performed while creating only $O(n^{d/2})$ cells.

The process goes as follows. We will describe a recursive procedure, subdividing each region at level $i < d$ into $O(\sqrt{n})$ regions at level $i + 1$, ensuring $O(\sqrt{n^d}) = O(n^{d/2})$ cells in the final decomposition. In the process, if $[a, b]$ is the region currently being subdivided, we let X_b be the set of points contained in the box $[0, b)$, and X_I the subset of those points which are internal in some dimension $j \leq i$. We initialize the process with the region $[0, 1]^d$ at level 0, with X_b being the full set of points of the input, and $X_I = \emptyset$.

Given a region $[a, b]$ at level $i < d$, with X_b and X_I as described, we will partition along dimension $i + 1$ into segments $[\xi_j, \xi_{j+1}]$, for some $\xi_j, j \in \{1, \dots, \ell\}$, where ℓ is the number of subdivisions. Concretely, for each $j \in \{1, \dots, \ell - 1\}$ we generate a region $[a^{(j)}, b^{(j)}] = [a_1, b_1] \times \dots \times [a_i, b_i] \times [\xi_j, \xi_{j+1}] \times [0, 1]^{d-i-1}$ at level $i + 1$, set $X_b^{(j)} = X_b \cap [0, b^{(j)})$ and $X_I^{(j)} = (X_I \cap [0, b^{(j)})) \cup \{x \in X_b^{(j)} : \xi_j < x_{i+1} < \xi_{j+1}\}$, and recursively process this region, with point set $X_b^{(j)}$ and internal points $X_I^{(j)}$. Observe that the points added to $X_I^{(j)}$ indeed are internal points, in dimension $i + 1$. The coordinates ξ_j are chosen to fulfill the following conditions.

$$\text{For all } x \in X_I, \text{ we have } x_{i+1} \in \{\xi_1, \dots, \xi_\ell\}. \quad (23)$$

$$\text{For all } j \in \{1, \dots, \ell - 1\}, \text{ we have } |\{x \in X_b : \xi_j < x_{i+1} < \xi_{j+1}\}| = O(\sqrt{n}). \quad (24)$$

We briefly argue that this is necessary and sufficient to maintain our two invariants while ensuring that $\ell = O(\sqrt{n})$ at every level. Indeed, if condition (23) is not fulfilled, then the point x is internal in two dimensions in some region $[a^{(j)}, b^{(j)}]$, and if condition (24) is not fulfilled for some j , then $|X_I^{(j)}|$ is larger than $O(\sqrt{n})$ in the same region. To create a set of coordinates ξ_j fulfilling these conditions is not difficult; simply begin with the coordinate set $\{x_{i+1} : x \in X_I\}$ to fulfill (23), and insert additional coordinates as needed to satisfy (24). This requires at most $|X_I| + (n/\sqrt{n})$ coordinates; thus $\ell = O(\sqrt{n})$ as required, and one finds inductively that both invariants are maintained at every region created at level $i + 1$.

To finally sketch the procedure for computing the discrepancy inside a given cell, let $B = [a, b]$ be the cell, X_b the points contained in $[0, b)$, and X' the set of points in $[0, b)$ which are not also contained in $[0, a]$. The points X' must thus be internal in at least one dimension, so by invariant 1, the points X' are internal in exactly one dimension $j \in \{1, \dots, d\}$. Note that if a point $x \in X'$ is internal in dimension j , and hence has $x_i \leq a_i$ for every $i \in \{1, \dots, d\}, i \neq j$, then for any $y \in (a, b)$ we have that $x \in [0, y)$ if and only if $y_j > x_j$; that is, if x is internal in dimension j , then the membership of x in $[0, y)$ is determined only by y_j . Now, for each $j \in \{1, \dots, d\}$, let $X'_j = \{x \in X' : a_j < x_j < b_j\}$ be the points internal in dimension j ; note that this partitions X' . We can then determine $X \cap [0, y)$ for any $y \in (a, b)$ by looking at the coordinates y_j independently, that is,

$$[0, y) \cap X = ([0, a] \cap X_b) \cup \bigcup_{j=1}^d \{x \in X'_j : x_j < y_j\}.$$

This independence makes the problem suitable for the algorithmic technique of *dynamic programming* (see, e.g., [20]). Briefly, let $f_j(y) = |X_b \cap [0, a]| + \sum_{k=1}^j |\{x \in X'_k : x_k < y_k\}|$. For $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, d\}$, let $p(i, j)$ be the minimum value of $\prod_{k=1}^j y_k$ such that $f_j(y) \geq i$, and let $q(i, j)$ be the maximum value of $\prod_{k=1}^j y_k$ such that $f_j(y) \leq i$ and $y \in (a, b)$. By sorting the coordinates x_j of every set X'_j , it is easy both to compute the values of $p(\cdot, 1)$ and $q(\cdot, 1)$, and to use the values of $p(\cdot, j)$ and $q(\cdot, j)$ to compute the values of $p(\cdot, j+1)$ and $q(\cdot, j+1)$. The maximum discrepancy attained for a box $[0, y]$ for $y \in (a, b)$ can then be computed from $p(\cdot, d)$ and $q(\cdot, d)$; note in particular that for $y \in (a, b)$, we have $f_d(y) = |X \cap [0, y]|$. For details, we refer to [27]. Discrepancy for boxes $[0, y]$ with $y_j = 1$ for one or several $j \in \{1, \dots, d\}$ can be handled in a similar way.

Some slight extra care in the analysis of the dynamic programming, and application of a more intricate form of the decomposition of Overmars and Yap, will lead to a running time of $O(n^{d/2+1})$ and $O(n)$ space, as shown in [27].

3.3 Complexity Results

We saw in the previous sections that for fixed dimension d the most efficient algorithm for calculating the star discrepancy of arbitrary n -point sets in $[0, 1]^d$, the algorithm of Dobkin, Eppstein, and Mitchell, has a running time of order $O(n^{1+d/2})$.

So the obvious question is if it is possible to construct a faster algorithm for the discrepancy calculation whose running time does *not* depend exponentially on the dimension d .

There are two recent complexity theoretical results that suggest that such an algorithm does not exist—at least not if the famous hypotheses from complexity theory, namely that $P \neq NP$ and the stronger *exponential time hypothesis* [74], are true.

3.3.1 Calculating the Star Discrepancy is NP-Hard

Looking at identity (17), we see that the calculation of the star discrepancy of a given point set X is in fact a *discrete optimization problem*, namely the problem to find an $y \in \overline{I}$ that maximizes the function value

$$\delta^*(y, X) = \max \left\{ \delta(y, X), \overline{\delta}(y, X) \right\}.$$

In [55] it was proved that the calculation of the star discrepancy is in fact an NP-hard optimization problem. Actually, a stronger statement was proved in [55], namely that the calculation of the star discrepancy is NP-hard even if we restrict ourselves to the easier sub-problem where all the coordinates of the input have finite binary expansion, i.e., are of the form $k2^{-\kappa}$ for some $\kappa \in \mathbb{N}$ and some integer $0 \leq k \leq 2^\kappa$. To explain this result in more detail let us start by defining the *coding length* of a real number from the interval $[0, 1)$ to be the number of digits in its binary expansion.

Informally speaking, the class NP is the class of all *decision problems*, i.e., problems with a true-or-false answer, for which the instances with an affirmative answer can be decided in polynomial time by a *non-deterministic Turing machine*²; here “polynomial” means polynomial in the coding length of the input. Such a non-deterministic Turing machine can be described as consisting of a non-deterministic part, which generates for a given instance a polynomial-length candidate (“certificate”) for the solution, and a deterministic part, which verifies in polynomial time whether this candidate leads to a valid solution.

In general, the NP-hardness of an optimization problem U is proved by verifying that deciding the so-called *threshold language* of U is an NP-hard decision problem (see, e.g., [73, Sect. 2.3.3] or, for a more informal explanation, [46, Sect. 2.1]). Thus it was actually shown in [55] that the following decision problem is NP-hard:

Decision Problem STAR DISCREPANCY

Instance: Natural numbers $n, d \in \mathbb{N}$, sequence $X = (x^{(i)})_{i=1}^n$ in $[0, 1]^d$, $\varepsilon \in (0, 1]$

Question: Is $d_\infty^*(X) \geq \varepsilon$?

Notice that the input of STAR DISCREPANCY has only finite coding length if the binary expansion of all coordinates and of ε is finite. The standard approach to prove NP-hardness of some decision problem is to show that another decision problem that is known to be NP-hard can be reduced to it in polynomial time. This approach was also used in [55], where the graph theoretical problem DOMINATING SET was reduced to STAR DISCREPANCY. The decision problem DOMINATING SET is defined as follows.

Definition 2. Let $G = (V, E)$ be a graph, where V is the finite set of vertices and $E \subseteq \{\{v, w\} \mid v, w \in V, v \neq w\}$ the set of edges of G . Let M be a subset of V . Then M is called a *dominating set* of G if for all $v \in V \setminus M$ there exists a $w \in M$ such that $\{v, w\}$ is contained in E .

Decision Problem DOMINATING SET

Instance: Graph $G = (V, E)$, $m \in \{1, \dots, |V|\}$

Question: Is there a dominating set $M \subseteq V$ of cardinality at most m ?

The decision problem DOMINATING SET is well studied in the literature and known to be NP-complete, see, e.g., [46].

Before we explain how to reduce DOMINATING SET to STAR DISCREPANCY, let us introduce another closely related decision problem, which will also be important in Section 3.4.2.

Decision Problem EMPTY HALF-OPEN BOX

Instance: Natural numbers $n, d \in \mathbb{N}$, sequence $X = (x^{(i)})_{i=1}^n$ in $[0, 1]^d$, $\varepsilon \in (0, 1]$

Question: Is there a $y \in \overline{T}(X)$ with $A(y, X) = 0$ and $V_y \geq \varepsilon$?

As in the problem STAR DISCREPANCY the coding length of the input of EMPTY HALF-OPEN BOX is only finite if the binary expansion of all coordinates and of ε is finite. For the reduction of DOMINATING SET it is sufficient to consider these instances with finite coding length. We now explain how to reduce DOMINATING SET

² NP stands for “non-deterministic polynomial time”.

to EMPTY HALF-OPEN BOX in polynomial time; this proof step can afterwards be re-used to establish that DOMINATING SET can indeed be reduced to STAR DISCREPANCY.

Theorem 5 ([55, Thm. 2.7]). *The decision problem EMPTY HALF-OPEN BOX is NP-hard.*

Proof. Let $G = (V, E), m \in \{1, \dots, |V|\}$ be an instance of DOMINATING SET. We may assume that $V = \{1, \dots, n\}$ for $n := |V|$. Let $\alpha, \beta \in [0, 1)$ have finite binary expansions and satisfy $\beta < \alpha^n$; for instance, we may choose $\alpha = 1/2$ and $\beta = 0$. For $i, j \in \{1, \dots, n\}$ put

$$x_j^{(i)} := \begin{cases} \alpha, & \text{if } \{i, j\} \in E \text{ or } i = j, \\ \beta, & \text{otherwise.} \end{cases}$$

and put $x^{(i)} := (x_j^{(i)})_{j=1}^n \in [0, 1)^n$ and $X := (x^{(i)})_{i=1}^n$. We shall show that there is a dominating set $M \subseteq V$ of cardinality at most m if and only if there is a $y \in \overline{\Gamma}(X)$ such that $A(y, X) = 0$ and $V_y \geq \alpha^m$.

Firstly, assume that there is a dominating set $M \subseteq V$ of cardinality at most m . Put

$$y_j := \begin{cases} \alpha, & \text{if } j \in M, \\ 1, & \text{otherwise.} \end{cases}$$

and $y := (y_j)_{j=1}^n$. Then $y \in \overline{\Gamma}(X)$ and $V_y = \alpha^{|M|} \geq \alpha^m$. Hence it suffices to prove that $[0, y) \cap X = \emptyset$. Now for each $i \in M$ we have $x_i^{(i)} = \alpha$, i.e., $x^{(i)} \notin [0, y)$. For every $i \in V \setminus M$ there is, by definition of a dominating set, a $v \in M$ such that $\{i, v\} \in E$, implying $x_v^{(i)} = \alpha$ which in turn yields $x^{(i)} \notin [0, y)$. Therefore $A(y, X) = 0$.

Secondly, assume the existence of a $y \in \overline{\Gamma}(X)$ such that $A(y, X) = 0$ and $V_y \geq \alpha^m$. Recall that $y \in \overline{\Gamma}(X)$ implies that $y_j \in \{\beta, \alpha, 1\}$ for all j . Since $\beta < \alpha^n \leq V_y$, we have $|\{j \in \{1, \dots, n\} \mid y_j \geq \alpha\}| = n$. Putting $M := \{i \in \{1, \dots, n\} \mid y_i = \alpha\}$, we have $|M| \leq m$. Since $A(y, X) = 0$, we obtain $|M| \geq 1$, and for each $i \in \{1, \dots, n\}$ there exists a $v \in M$ such that $\{i, v\} \in E$ or $i \in M$. Hence M is a dominating set of G with size at most m . \square

In [55] further decision problems of the type ‘‘maximal half-open box for k points’’ and ‘‘minimal closed box for k points’’ were studied; these problems are relevant for an algorithm of E. Thiérmard that is based on integer linear programming, see Section 3.4.2.

Theorem 6 ([55, Thm. 3.1]). *STAR DISCREPANCY is NP-hard.*

Sketch of the proof. Due to identity (17) the decision problem STAR DISCREPANCY can be formulated in an equivalent way: Is there a $y \in \overline{\Gamma}(X)$ such that $\delta(y, X) \geq \varepsilon$ or $\overline{\delta}(y, X) \geq \varepsilon$? The NP-hardness of this equivalent formulation can again be shown by polynomial time reduction from DOMINATING SET. So

let $V = \{1, \dots, n\}$, and let $G = (V, E), m \in \{1, \dots, n\}$ be an instance of DOMINATING SET. We may assume without loss of generality $n \geq 2$ and $m < n$. Put $\alpha := 1 - 2^{-(n+1)}$, $\beta := 0$, and

$$x_j^{(i)} := \begin{cases} \alpha, & \text{if } \{i, j\} \in E \text{ or } i = j, \\ \beta, & \text{otherwise.} \end{cases}$$

The main idea is now to prove for $X := (x^{(i)})_{i=1}^n$ that $d_\infty^*(X) \geq \alpha^m =: \varepsilon$ if and only if there is a dominating set $M \subseteq V$ for G with $|M| \leq m$.

From the proof of Theorem 5 we know that the existence of such a dominating set M is equivalent to the existence of a $y \in \overline{\Gamma}(X)$ with $A(y, X) = 0$ and $V_y \geq \alpha^m$. Since the existence of such a y implies $d_\infty^*(X) \geq \alpha^m$, it remains only to show that $d_\infty^*(X) \geq \alpha^m$ implies in turn the existence of such a y . This can be checked with the help of Bernoulli's inequality, for the technical details we refer to [55].

3.3.2 Calculating the Star Discrepancy is W[1]-Hard

Although NP-hardness (assuming $P \neq NP$) excludes a running time of $(n+d)^{O(1)}$ for computing the star discrepancy of an input of n points in d dimensions, this still does not completely address our running time concerns. In a nutshell, we know that the problem can be solved in polynomial time for every fixed d (e.g., by the algorithm of Dobkin, Eppstein, and Mitchell), and that it is NP-hard for arbitrary inputs, but we have no way of separating a running time of $n^{\Theta(d)}$ from, say, $O(2^d n^2)$, which would of course be a huge breakthrough for computing low-dimensional star discrepancy.

The usual framework for addressing such questions is *parameterized complexity*. Without going into too much technical detail, a parameterized problem is a decision problem whose inputs are given with a *parameter* k . Such a problem is *fixed-parameter tractable* (FPT) if instances of total length n and with parameter k can be solved in time $O(f(k)n^c)$, for some constant c and an arbitrary function $f(k)$. Observe that this is equivalent to being solvable in $O(n^c)$ time for every fixed k , as contrasted to the previous notion of polynomial time for every fixed k , which also includes running times such as $O(n^k)$. We shall see in this section that, unfortunately, under standard complexity theoretical assumptions, no such algorithm is possible (in fact, under a stronger assumption, not even a running time of $O(f(d)n^{o(d)})$ is possible).

Complementing the notion of FPT is a notion of parameterized hardness. A *parameterized reduction* from a parameterized problem \mathcal{Q} to a parameterized problem \mathcal{Q}' is a reduction which maps an instance I with parameter k of \mathcal{Q} to an instance I' with parameter k' of \mathcal{Q}' , such that

1. (I, k) is a true instance of \mathcal{Q} if and only if (I', k') is a true instance of \mathcal{Q}' ,
2. $k' \leq f(k)$ for some function $f(k)$, and

3. the total running time of the reduction is FPT (i.e., bounded by $O(g(k)||I||^c)$, for some function $g(k)$ and constant c , where $||I||$ denotes the total coding length of the input).

It can be verified that such a reduction and an FPT-algorithm for \mathcal{Q}' imply an FPT-algorithm for \mathcal{Q} .

The basic hardness class of parameterized complexity, analogous to the class NP of classical computational complexity, is known as W[1], and can be defined as follows. Given a graph $G = (V, E)$, a *clique* is a set $X \subseteq V$ such that for any $u, v \in X$, $u \neq v$, we have $\{u, v\} \in E$. Let k -CLIQUE be the following parameterized problem.

Parameterized Problem k -CLIQUE

Instance: A graph $G = (V, E)$; an integer k .

Parameter: k

Question: Is there a clique of cardinality k in G ?

The class W[1] is then the class of problems reducible to k -CLIQUE under parameterized reductions, and a parameterized problem is W[1]-hard if there is a parameterized reduction to it from k -CLIQUE. (Note that, similarly, the complexity class NP can be defined as the closure of, e.g., CLIQUE or 3-SAT under standard polynomial-time reductions.) The basic complexity assumption of parameterized complexity is that $\text{FPT} \neq \text{W}[1]$, or equivalently, that the k -CLIQUE problem is not fixed-parameter tractable; this is analogous to the assumption in classical complexity that $\text{P} \neq \text{NP}$.

For more details, see the books of R. G. Downey and M. R. Fellows [36] or J. Flum and M. Grohe [42]. In particular, we remark that there is a different definition of W[1] in terms of problems solvable by a class of restricted circuits (in fact, W[1] is just the lowest level of a hierarchy of such classes, the so-called *W-hierarchy*), which arguably makes the class definition more natural, and the conjecture $\text{FPT} \neq \text{W}[1]$ more believable.

P. Giannopoulos et al. [48] showed the following.

Theorem 7 ([48]). *There is a polynomial-time parameterized reduction from k -CLIQUE with parameter k to STAR DISCREPANCY with parameter $d = 2k$.*

Thus, the results of [48] imply that STAR DISCREPANCY has no algorithm with a running time of $O(f(d)n^c)$ for any function $f(d)$ and constant c , unless $\text{FPT} = \text{W}[1]$.

A stronger consequence can be found by considering the so-called exponential-time hypothesis (ETH). This hypothesis, formalized by Impagliazzo and Paturi [74], states that 3-SAT on n variables cannot be solved in time $O(2^{o(n)})$; in a related paper [75], this was shown to be equivalent to similar statements about several other problems, including that k -SAT on n variables cannot be solved in time $O(2^{o(n)})$ for $k \geq 3$ and that 3-SAT cannot be solved in time $O(2^{o(m)})$, where m equals the total number of clauses in an instance (the latter form is particularly useful in reductions).

It has been shown by J. Chen et al. [14, 15] that k -CLIQUE cannot be solved in $f(k)n^{o(k)}$ time, for any function $f(k)$, unless ETH is false. We thus get the following corollary.

Corollary 1 ([48]). STAR DISCREPANCY for n points in d dimensions cannot be solved exactly in $f(d)n^{o(d)}$ time for any function $f(d)$, unless ETH is false.

In fact, it seems that even the constant factor in the exponent of the running time of the algorithm of Dobkin, Eppstein and Mitchell [27] would be very difficult to improve; e.g., a running time of $O(n^{d/3+c})$ for some constant c would imply a new faster algorithm for k -CLIQUE. (See [48] for more details, and for a description of the reduction.)

3.4 Approximation of the Star Discrepancy

As seen in Section 3.3.1 and 3.3.2, complexity theory tells us that the exact calculation of the star discrepancy of large point sets in high dimensions is infeasible.

The theoretical bounds for the star discrepancy of low-discrepancy point sets that are available in the literature describe the asymptotic behavior of the star discrepancy well only if the number of points n tends to infinity. These bounds are typically useful for point sets of size $n \gg e^d$, but give no helpful information for moderate values of n . To give concrete examples, we restate here some numerical results provided in [76] for bounds based on inequalities of *Erdős-Turán-Koksma-type* (see, e.g., [24, 99]).

If the point set $P_n(z)$ is an n -point rank-1 lattice in $[0, 1]^d$ with generating vector $z \in \mathbb{Z}^d$ (see, e.g., [99, 115]), then its discrepancy can be bounded by

$$\begin{aligned} d_\infty^*(P_n(z)) &\leq 1 - \left(1 - \frac{1}{n}\right)^d + T(z, n) \leq 1 - \left(1 - \frac{1}{n}\right)^d + W(z, n) \\ &\leq 1 - \left(1 - \frac{1}{n}\right)^d + R(z, n)/2; \end{aligned} \quad (25)$$

here the quantities $W(z, n)$ and $R(z, n)$ can be calculated to a fixed precision in $O(nd)$ operations [76, 78], while the calculation of $T(z, n)$ requires $O(n^2d)$ operations (at least there is so far no faster algorithm known).

S. Joe presented in [76] numerical examples where he calculated the values of $T(z, n)$, $W(z, n)$, and $R(z, n)$ for generators z provided by a *component-by-component algorithm* (see, e.g., [84, 104, 116, 117]) from [76, Sect. 4]. For dimension $d = 2$ and 3 he was able to compare the quantities with the exact values

$$d_\infty^*(P_n(z)) - \left[1 - \left(1 - \frac{1}{n}\right)^d\right] =: E(z, n). \quad (26)$$

In $d = 2$ for point sets ranging from $n = 157$ to 10,007, the smallest of the three quantities, $T(z, n)$, was 8 to 10 times larger than $E(z, n)$. In dimension $d = 3$ for point sets ranging from $n = 157$ to 619 it was even more than 18 times larger. (Joe

used for the computation of $E(z, n)$ the algorithm proposed in [11] and was therefore limited to this range of examples.)

Computations for $d = 10$ and 20 and $n = 320,009$ led to $T(z, n) = 1.29 \cdot 10^4$ and $5.29 \cdot 10^{13}$, respectively. Recall that the star discrepancy and $E(z, n)$ are always bounded by 1. The more efficiently computable quantities $W(z, n)$ and $R(z, n)$ led obviously to worse results, but $W(z, n)$ was at least very close to $T(z, n)$.

This example demonstrates that for good estimates of the star discrepancy for point sets of practicable size we can unfortunately not rely on theoretical bounds.

Since the exact calculation of the star discrepancy is infeasible in high dimensions, the only remaining alternative is to consider approximation algorithms. In the following we present the known approaches.

3.4.1 An Approach Based on Bracketing Covers

An approach that approximates the star discrepancy of a given set X up to a user-specified error δ was presented by E. Thiémond [128, 130]. It is in principle based on the generation of suitable δ -bracketing covers (which were not named in this way in [128, 130]). Let us describe Thiémond's approach in detail.

The first step is to "discretize" the star discrepancy at the cost of an approximation error of at most δ . The corresponding discretization is different from the one described in Section 3.1; in particular, it is completely independent of the input set X . The discretization is done by choosing a suitable finite set of test boxes anchored in zero whose upper right corners form a so-called δ -cover. We repeat here the definition from [31].

Definition 3. A finite subset Γ of $[0, 1]^d$ is called a δ -cover of the class \mathcal{C}_d of all axis-parallel half-open boxes anchored in zero (or of $[0, 1]^d$) if for all $y \in [0, 1]^d$ there exist $x, z \in \Gamma \setminus \{0\}$ such that

$$x \leq y \leq z \quad \text{and} \quad V_z - V_x \leq \delta.$$

Put

$$N(\mathcal{C}_d, \delta) := \min\{|\Gamma| \mid \Gamma \text{ } \delta\text{-cover of } \mathcal{C}_d.\}$$

Any δ -cover Γ of \mathcal{C}_d satisfies the following approximation property:

Lemma 3. Let Γ be a δ -cover of \mathcal{C}_d . For all finite sequences X in $[0, 1]^d$ we have

$$d_\infty^*(X) \leq d_\Gamma^*(X) + \delta, \quad (27)$$

where

$$d_\Gamma^*(X) := \max_{y \in \Gamma} |V_y - A(y, X)|$$

can be seen as a discretized version of the star discrepancy.

A proof of Lemma 3 is straightforward. (Nevertheless, it is, e.g., contained in [31].)

In [50] the notion of δ -covers was related to the concept of bracketing entropy, which is well known in the theory of empirical processes. We state here the definition for the set system \mathcal{C}_d of anchored axis-parallel boxes (a general definition can, e.g., be found in [50, Sect. 1]):

Definition 4. A closed axis-parallel box $[x, z] \subset [0, 1]^d$ is a δ -bracket of \mathcal{C}_d if $x \leq z$ and $V_z - V_x \leq \delta$. A δ -bracketing cover of \mathcal{C}_d is a set of δ -brackets whose union is $[0, 1]^d$. By $N_{\square}(\mathcal{C}_d, \delta)$ we denote the *bracketing number* of \mathcal{C}_d (or of $[0, 1]^d$), i.e., the smallest number of δ -brackets whose union is $[0, 1]^d$. The quantity $\ln N_{\square}(\mathcal{C}_d, \delta)$ is called the *bracketing entropy*.

The bracketing number and the quantity $N(d, \delta)$ are related to the *covering* and the *L_1 -packing number*, see, e.g., [31, Rem. 2.10].

It is not hard to verify that

$$N(\mathcal{C}_d, \delta) \leq 2N_{\square}(\mathcal{C}_d, \delta) \leq N(\mathcal{C}_d, \delta)(N(\mathcal{C}_d, \delta) + 1) \quad (28)$$

holds. Indeed, if \mathcal{B} is a δ -bracketing cover, then it is easy to see that

$$\Gamma_{\mathcal{B}} := \{x \in [0, 1]^d \setminus \{0\} \mid \exists y \in [0, 1]^d : [x, y] \in \mathcal{B} \text{ or } [y, x] \in \mathcal{B}\} \quad (29)$$

is a δ -cover. If Γ is a δ -cover, then

$$\mathcal{B}_{\Gamma} := \{[x, y] \mid x, y \in \Gamma \cup \{0\}, [x, y] \text{ is a } \delta\text{-bracket}, x \neq y\}$$

is a δ -bracketing cover. These two observations imply (28).

In [50] it is shown that

$$\delta^{-d}(1 - O_d(\delta)) \leq N_{\square}(\mathcal{C}_d, \delta) \leq 2^{d-1}(2\pi d)^{-1/2}e^d(\delta^{-1} + 1)^d, \quad (30)$$

see [50, Thm. 1.5 and 1.15]. The construction that leads to the upper bound in (30) implies also

$$N_{\square}(\mathcal{C}_d, \delta) \leq (2\pi d)^{-1/2}e^d\delta^{-d} + O_d(\delta^{-d+1}) \quad (31)$$

(see [50, Remark 1.16]) and

$$N(\mathcal{C}_d, \delta) \leq 2^d(2\pi d)^{-1/2}e^d(\delta^{-1} + 1)^d. \quad (32)$$

For more information about δ -covers and δ -bracketing covers we refer to the original articles [31, 50, 51] and the survey article [52]; [51] and [52] contain also several figures showing explicit two-dimensional constructions.

The essential idea of Thiémond's algorithm from [128, 130] is to generate for a given point set X and a user-specified error δ a small δ -bracketing cover $\mathcal{B} = \mathcal{B}_{\delta}$ of $[0, 1]^d$ and to approximate $d_{\infty}^*(X)$ by $d_{\Gamma}^*(X)$, where $\Gamma = \Gamma_{\mathcal{B}}$ as in (29), up to an error of at most δ , see Lemma 3.

The costs of generating \mathcal{B}_{δ} are of order $\Theta(d|\mathcal{B}_{\delta}|)$. If we count the number of points in $[0, y]$ for each $y \in \Gamma_{\mathcal{B}}$ in a naive way, this results in an overall running time of $\Theta(dn|\mathcal{B}_{\delta}|)$ for the whole algorithm. As Thiémond pointed out in [130], this

orthogonal range counting can be done in moderate dimension d more effectively by employing data structures based on *range trees*, see, e.g., [21, 95]. This approach reduces in moderate dimension d the time $O(dn)$ per test box that is needed for the naive counting to $O(\log^d n)$. Since a range tree for n points can be generated in $O(C^d n \log^d n)$ time, $C > 1$ some constant, this results in an overall running time of

$$O((d + \log^d n)|\mathcal{B}_\delta| + C^d n \log^d n).$$

As this approach of orthogonal range counting is obviously not very beneficial in higher dimension (say, $d > 5$), we do not further explain it here, but refer for the details to [130].

The smallest bracketing covers \mathcal{T}_δ used by Thiémond can be found in [130]; they differ from the constructions provided in [50, 51], see [51, 53]. He proved for his best constructions the upper bound

$$|\mathcal{T}_\delta| \leq e^d \left(\frac{\ln \delta^{-1}}{\delta} + 1 \right)^d,$$

a weaker bound than (31) and (32), which both hold for the δ -bracketing covers constructed in [50]. Concrete comparisons of Thiémond's bracketing covers with other constructions in dimension $d = 2$ can be found in [51], where also optimal two-dimensional bracketing covers are provided.

The lower bound in (30) proved in [50] immediately implies a lower bound for the running time of Thiémond's algorithm, regardless how cleverly the δ -bracketing covers are chosen. That is because the dominating factor in the running time is the construction of the δ -bracketing cover $|\mathcal{B}_\delta|$, which is of order $\Theta(d|\mathcal{B}_\delta|)$. Thus (30) shows that the running time of the algorithm is exponential in d . (Nevertheless smaller δ -bracketing covers, which may, e.g., be generated by extending the ideas from [51] to arbitrary dimension d , would widen the range of applicability of Thiémond's algorithm.) Despite its limitations, Thiémond's algorithm is a helpful tool in moderate dimensions, as was reported, e.g., in [33, 128, 130] or [106], see also Section 5.1.

For more specific details we refer to [110, 128, 130]. For a modification of Thiémond's approach to approximate L_∞ -extreme discrepancies see [50, Sect. 2.2].

3.4.2 An Approach Based on Integer Linear Programming

Since the large scale enumeration problem (17) is infeasible in high dimensions, a number of algorithms have been developed that are based on heuristic approaches. One such approach was suggested by E. Thiémond in [131] (a more detailed description of his algorithm can be found in his PhD thesis [129]). Thiémond's algorithm is based on *integer linear programming*, a concept that we shall describe below. His approach is interesting in that, despite being based on heuristics in the initialization phase, it allows for an arbitrarily good approximation of the star discrepancy of any

given point set. Furthermore, the user can decide on the fly which approximation error he is willing to tolerate. This is possible because the algorithm outputs, during the optimization of the star discrepancy approximation, upper and lower bounds for the exact $d_\infty^*(X)$ -value. The user can abort the optimization procedure once the difference between the lower and upper bound are small enough for his needs, or he may wait until the optimization procedure is finished, and the exact star discrepancy value $d_\infty^*(X)$ is computed.

Before we describe a few details of the algorithm, let us mention that numerical tests in [130, 131] suggest that the algorithm from [131] outperforms the one from [130] (see Section 3.4.1 for a description of the latter algorithm). In particular, instances that are infeasible for the algorithm from [130] can be solved using the integer linear programming approach described below, see also the discussion in Section 5.1.

The basic idea of Thiéard's algorithm is to split optimization problem (17) into $2n$ optimization problems similarly as done in equation (21), and to transform these problems into integer linear programs. To be more precise, he considers for each value $k \in \{0, 1, 2, \dots, n\}$ the volume of the smallest and the largest box containing exactly k points of X . These values are denoted V_{\min}^k and V_{\max}^k , respectively. It is easily verified, using similar arguments as in Lemma 1, that these values (if they exist) are obtained by the grid points of X , i.e., there exist grid points $y_{\min}^k \in \Gamma(X)$, $y_{\max}^k \in \bar{\Gamma}(X)$ such that

$$\begin{aligned} \bar{A}(y_{\min}^k, X) &= k \text{ and } V_{y_{\min}^k} = V_{\min}^k, \\ A(y_{\max}^k, X) &= k \text{ and } V_{y_{\max}^k} = V_{\max}^k. \end{aligned}$$

It follows from (21) that

$$d_\infty^*(X) = \max_{k \in \{0, 1, 2, \dots, n\}} \max \left\{ \frac{k}{n} - V_{\min}^k, V_{\max}^k - \frac{k}{n} \right\}. \quad (33)$$

As noted in [55], boxes containing *exactly* k points may not exist, even if the n points are pairwise different. However, they do exist if for at least one dimension $j \in \{1, \dots, d\}$ the coordinates $(x_j)_{x \in X}$ are pairwise different. If they are pairwise different for all $j \in \{1, \dots, d\}$, in addition we have

$$V_{\min}^1 \leq \dots \leq V_{\min}^n \quad \text{and} \quad V_{\max}^0 \leq \dots \leq V_{\max}^{n-1}.$$

Note that we obviously have $V_{\min}^0 = 0$ and $V_{\max}^n = 1$. We may therefore disregard these two values in equation (33).

As mentioned in Section 3.3.1 (cf. Theorem 5 and the text thereafter), already the related problem “Is $V_{\max}^0 \geq \varepsilon$?” is an NP-hard one. By adding “dummy points” at or close to the origin $(0, \dots, 0)$ one can easily generalize this result to all questions of the type “Is $V_{\max}^k \geq \varepsilon$?”. Likewise, it is shown in [55] that the following decision problem is NP-hard.

Decision Problem V_{\min}^k -BOX

Instance: Natural numbers $n, d \in \mathbb{N}$, $k \in \{0, 1, \dots, n\}$, sequence $X = (x^{(i)})_{i=1}^n$ in $[0, 1]^d$, $\varepsilon \in (0, 1]$

Question: Is there a point $y \in \Gamma(X)$ such that $\bar{A}(y, X) \geq k$ and $V_y \leq \varepsilon$?

This suggests that the V_{\max}^k - and V_{\min}^k -problems are difficult to solve to optimality. As we shall see below, in Thiémarc's integer linear programming ansatz, we will not have to solve all $2n$ optimization problems in (33) to optimality. Instead it turns out that for most practical applications of his algorithms only very few of them need to be solved exactly, whereas for most of the problems it suffices to find good upper and lower bounds. We shall discuss this in more detail below.

For each of the $2n$ subproblems of computing V_{\min}^k and V_{\max}^k , respectively, Thiémarc formulates, by taking the logarithm of the volumes, an integer linear program with $n + d(n - k)$ binary variables. The size of the linear program is linear in the size nd of the input X . We present here the integer linear program (ILP) for the V_{\min}^k -problems. The ones for V_{\max}^k -problems are similar. However, before we are ready to formulate the ILPs, we need to fix some notation.

For every $n \in \mathbb{N}$ we abbreviate by S_n the set of permutations of $\{1, \dots, n\}$. For all $j \in \{1, \dots, d\}$ put $x_j^{(n+1)} := 1$ and let $\sigma_j \in S_{n+1}$ such that

$$x_j^{(\sigma_j(1))} \leq \dots \leq x_j^{(\sigma_j(n))} \leq x_j^{(\sigma_j(n+1))} = 1.$$

With σ_j at hand, we can define, for every index $\delta = (\delta_1, \dots, \delta_d) \in \{1, \dots, n+1\}^d$ the closed and half-open boxes *induced by δ* ,

$$[0, \delta] := \prod_{j=1}^d [0, x_j^{(\sigma_j(\delta_j))}] \quad \text{and} \quad [0, \delta) = \prod_{j=1}^d [0, x_j^{(\sigma_j(\delta_j))}).$$

One of the crucial observations for the formulation of the ILPs is the fact that $x^{(i)} \in [0, \delta]$ (resp. $x^{(i)} \in [0, \delta)$), if and only if for all $j \in \{1, \dots, d\}$ it holds that $\sigma_j^{-1}(i) \leq \delta_j$ (resp. $\sigma_j^{-1}(i) \leq \delta_j - 1$). We set

$$z_j^i(\delta) := \begin{cases} 1, & \text{if } \sigma_j^{-1}(i) \leq \delta_j, \\ 0, & \text{otherwise.} \end{cases}$$

Every δ induces exactly one sequence $z = ((z_j^i(\delta)))_{j=1}^d_{i=1}^n$ in $(\{0, 1\}^d)^n$, and, likewise, for every *feasible* sequence z there is exactly one $\delta(z) \in \{1, \dots, n+1\}^d$ with $z = z(\delta(z))$. In the following linear program formulation we introduce also the variables $y^{(1)}, \dots, y^{(n)}$, and we shall have $y^{(i)} = 1$ if and only if $x^{(i)} \in [0, \delta(z)]$.

The integer linear program for the V_{\min}^k -problem can now be defined as follows.

$$\ln(V_{\min}^k) = \min \sum_{j=1}^d [\ln(x_j^{(\sigma_j(1))}) + \sum_{i=2}^n z_j^{(\sigma_j(i))} (\ln(x_j^{(\sigma_j(i))}) - \ln(x_j^{(\sigma_j(i-1))}))] \quad (34)$$

subject to

$$\begin{aligned}
\text{(i)} \quad & 1 = z_j^{(\sigma_j(1))} = \dots = z_j^{(\sigma_j(k))} \geq \dots \geq z_j^{(\sigma_j(n))} \quad \forall j \in \{1, \dots, d\} \\
\text{(ii)} \quad & z_j^{(\sigma_j(i))} = z_j^{(\sigma_j(i+1))} \quad \forall j \in \{1, \dots, d\} \forall i \in \{1, \dots, n\} : \\
& \quad \quad \quad x_j^{(\sigma_j(i))} = x_j^{(\sigma_j(i+1))} \\
\text{(iii)} \quad & y^{(i)} \leq z_j^{(i)} \quad \forall j \in \{1, \dots, d\} \forall i \in \{1, \dots, n\} \\
\text{(iv)} \quad & y^{(i)} \geq 1 - d + \sum_{j=1}^d z_j^{(i)} \quad \forall i \in \{1, \dots, n\} \\
\text{(v)} \quad & \sum_{i=1}^n y^{(i)} \geq k \\
\text{(vi)} \quad & y^{(i)} \in \{0, 1\} \quad \forall i \in \{1, \dots, n\} \\
\text{(vii)} \quad & z_j^{(i)} \in \{0, 1\} \quad \forall j \in \{1, \dots, d\} \forall i \in \{1, \dots, n\}
\end{aligned}$$

We briefly discuss the constraints of the integer linear program (34).

- Since we request at least k points to lie in the box $[0, \delta(z)]$, the inequality $x_j^{(\sigma_j(\delta(z)_j))} \geq x_j^{(\sigma_j(k))}$ must hold for all $j \in \{1, \dots, d\}$. We may thus fix the values $1 = z_j^{(\sigma_j(1))} = \dots = z_j^{(\sigma_j(k))}$.
- The second constraint expresses that for two points with the same coordinate $x_j^{(\sigma_j(i))} = x_j^{(\sigma_j(i+1))}$ in the j th dimension, we must satisfy $z_j^{(\sigma_j(i))} = z_j^{(\sigma_j(i+1))}$.
- The third and fourth condition say that $y^{(i)} = 1$ if and only if $x^{(i)} \in [0, \delta(z)]$. For $x^{(i)} \in [0, \delta(z)]$ we have $\sigma_j^{-1}(i) \leq \delta(z)_j$ and thus $z_j^{(i)} = 1$, $j \in \{1, \dots, d\}$. According to condition (iv) this implies $y^{(i)} \geq 1 - d + \sum_{j=1}^d z_j^{(i)} = 1$, and thus $y^{(i)} = 1$. If, on the other hand, $x^{(i)} \notin [0, \delta(z)]$, there exists a coordinate $j \in \{1, \dots, d\}$ with $x_j^{(i)} > \delta(z)_j$. Thus, $z_j^{(i)} = 0$ and condition (iii) implies $y^{(i)} \leq z_j^{(i)} = 0$.
- Condition (v) ensures the existence of at least k points inside $[0, \delta(z)]$.
- Conditions (vi) and (vii) are called the *integer* or *binary constraints*. Since only integer (binary) values are allowed, the linear program (34) is called a (*binary*) *integer linear program*. We shall see below that by changing these conditions to $y^{(i)} \in [0, 1]$ and $z_j^{(i)} \in [0, 1]$, we get the *linear relaxation* of the integer linear program (34). The solution of this linear relaxation is a lower bound for the true V_{\min}^k -solution.

Using the V_{\min}^k - and V_{\max}^k -integer linear programs, Thiémond computes the star discrepancy of a set X in a sequence of optimization steps, each of which possibly deals with a different k -box problem. Before the optimization phase kicks in, there is an *initialization phase*, in which for each k an upper bound \overline{V}_{\min}^k for V_{\min}^k and a lower bound \underline{V}_{\max}^k for V_{\max}^k is computed. This is done by a simple greedy strategy followed by a local optimization procedure that helps to improve the initial value of the greedy strategy. Thiémond reports that the estimates obtained for the V_{\max}^k -problems are usually quite good already, whereas the estimates for the V_{\min}^k -problems are usually too pessimistic. A lower bound \underline{V}_{\min}^k for the V_{\min}^k -problem is also computed, using the simple observation that for each dimension $j \in \{1, \dots, d\}$, the j th coordinate of the smallest V_{\min}^k -box must be at least as large as the k th smallest coordinate in

$(x_j)_{j \in X}$. That is, we have $V_{\min}^k \geq \prod_{j=1}^d x_j^{(\sigma_j(k))}$. We initialize the lower bound \underline{V}_{\min}^k by setting it equal to this expression. Similarly, $\prod_{j=1}^d x_j^{(\sigma_j(k+1))}$ is a lower bound for the V_{\max}^k -problem, but this bound is usually much worse than the one provided by the heuristics. For an initial upper bound of V_{\max}^k , Thiémond observes that the V_{\max}^{n-1} -problem can be solved easily. In fact, we have $V_{\max}^{n-1} = \max\{x_j^{(\sigma_j(n))} \mid j \in \{1, \dots, d\}\}$. As mentioned in the introduction to this section, if for all $j \in \{1, \dots, d\}$ the j th coordinates $x_j^{(1)}, \dots, x_j^{(n)}$ of the points in X are pairwise different, we have $V_{\max}^0 \leq \dots \leq V_{\max}^{n-1}$. Thus, in this case, we may initialize \overline{V}_{\max}^k , $k = 1, \dots, n-1$, by V_{\max}^{n-1} .

From these values (we neglect a few minor steps in Thiémond's computation) we compute the following estimates

$$\begin{aligned} \underline{D}_{\min}^k(X) &:= \frac{k}{n} - \overline{V}_{\min}^k \text{ and } \overline{D}_{\min}^k(X) := \frac{k}{n} - \underline{V}_{\min}^k, \\ \underline{D}_{\max}^k(X) &:= \underline{V}_{\max}^k - \frac{k}{n} \text{ and } \overline{D}_{\max}^k(X) := \overline{V}_{\max}^k - \frac{k}{n}. \end{aligned}$$

Clearly, $\underline{D}_{\min}^k(X) \leq \frac{k}{n} - V_{\min}^k \leq \overline{D}_{\min}^k(X)$ and $\underline{D}_{\max}^k(X) \leq V_{\max}^k - \frac{k}{n} \leq \overline{D}_{\max}^k(X)$.

After this initialization phase, the *optimization phase* begins. It proceeds in rounds. In each round, the k -box problem yielding the largest estimate

$$\overline{D}_{\infty}^*(X) := \max \left\{ \max_{k \in \{1, 2, \dots, n\}} \overline{D}_{\min}^k(X), \max_{k \in \{0, 1, \dots, n-1\}} \overline{D}_{\max}^k(X) \right\}$$

is investigated further.

If we consider a V_{\min}^k - or a V_{\max}^k -problem for the first time, we regard the *linear relaxation* of the integer linear program for $\ln(V_{\min}^k)$ or $\ln(V_{\max}^k)$, respectively. That is—cf. the comments below the formulation of the ILP for V_{\min}^k -problem above—instead of requiring the variables $y^{(i)}$ and $z_j^{(i)}$ to be either 0 or 1, we only require them to be in the interval $[0, 1]$. This turns the integer linear program into a linear program. Although it may seem that this relaxation does not change much, linear programs are known to be polynomial time solvable, and many fast readily available solving procedures, e.g., commercial tools such as CPLEX, are available. Integer linear programs and binary integer linear programs such as ours, on the other hand, are known to be NP-hard in general, and are usually solvable only with considerable computational effort. Relaxing the binary constraints (vi) and (vii) in (34) can thus be seen as a heuristic to get an initial approximation of the V_{\min}^k - and V_{\max}^k -problems, respectively.

In case of a V_{\min}^k -problem the value of this relaxed program is a lower bound for $\ln(V_{\min}^k)$ —we thus obtain new estimates for the two values \underline{V}_{\min}^k and \overline{D}_{\min}^k . If, on the other hand, we regard a V_{\max}^k -problem, the solution of the relaxed linear program establishes an upper bound for $\ln(V_{\max}^k)$; and we thus get new estimates for \overline{V}_{\max}^k and \overline{D}_{\max}^k . We may be lucky that we get an integral solution, in which case we have

determined V_{\min}^k or V_{\max}^k , respectively, and do not need to consider this problem in any further iteration of the algorithm.

If we consider a V_{\min}^k - or V_{\max}^k -problem for the second time, we solve the integer linear program itself, using a standard *branch and bound* technique. Branch and bound resembles a divide and conquer approach: the problem is divided into smaller subproblems, for each of which upper and lower bounds are computed.

Let us assume that we are, for now, considering a fixed V_{\min}^k -problem (V_{\max}^k -problems are treated the same way). As mentioned above, we divide this problem into several subproblems, and we compute upper and lower bounds for these subproblems. We then investigate the most “promising” subproblems (i.e., the ones with the largest upper bound and smallest lower bound for the value of V_{\min}^k) further, until the original V_{\min}^k -problem at hand has been solved to optimality or until the bounds for V_{\min}^k are good enough to infer that this k -box problem does not cause the maximal discrepancy value in (33).

A key success factor of the branch and bound step is a further strengthening of the integer linear program at hand. Thiémond introduces further constraints to the ILP, some of which are based on straightforward combinatorial properties of the k -box problems and others which are based on more sophisticated techniques such as *cutting planes* and *variable forcing* (cf. Thiémond’s PhD thesis [129] for details). These additional constraints and techniques strengthen the ILP in the sense that the solution to the linear relaxation is closer to that of the integer program. Thiémond provides some numerical results indicating that these methods frequently yield solutions based on which we can exclude the k -box problem at hand from our considerations for optimizing (33). That is, only few of the $2n$ many k -box problems in (33) need to be solved to optimality, cf. [131] for the details.

As explained above, Thiémond’s approach computes upper and lower bounds for the star discrepancy of a given point set at the same time. Numerical experiments indicate that the lower bounds are usually quite strong from the beginning, whereas the initial upper bounds are typically too large, and decrease only slowly during the optimization phase, cf. [131, Section 4.2] for a representative graph of the convergence behavior. Typical running times of the algorithms can be found in [131] and in [56]. The latter report contains also a comparison to the alternative approach described in the next section.

3.4.3 Approaches Based on Threshold-Accepting

In the next two sections we describe three heuristic approaches to compute lower bounds for the star discrepancy of a given point set X . All three algorithms are based on randomized local search heuristics; two of them on a so-called *threshold accepting approach*, see this section, and one of them on a *genetic algorithm*, see Section 3.4.4.

Randomized local search heuristics are problem-independent algorithms that can be used as frameworks for the optimization of inherently difficult problems, such as combinatorial problems, graph problems, etc. We distinguish between Monte-

Carlo algorithms and Las Vegas algorithms. Las Vegas algorithms are known to converge to the optimal solution, but their exact running time cannot be determined in advance. Monte-Carlo algorithms, on the other hand, have a fixed running time (usually measured by the number of iterations or the number of function evaluations performed), but we usually do not know the quality of the final output. The two threshold accepting algorithms presented next are Monte-Carlo algorithms for which the user may specify the number of iterations he is willing to invest for a good approximation of the star discrepancy value. The genetic algorithm presented in Section 3.4.4, on the other hand, is a Monte-Carlo algorithm with unpredictable running time (as we shall see below, in this algorithm, unconventionally, the computation is aborted when no improvement has happened for some t iterations in a row).

This said, it is clear that the lower bounds computed by both the threshold accepting algorithms as well as the one computed by the genetic algorithm may be arbitrarily bad. However, as all reported numerical experiments suggest, they are usually quite good approximations of the true discrepancy value—in almost all cases for which the correct discrepancy value can be computed the same value was also reported by the improved threshold accepting heuristic [56] described below. We note that these heuristic approaches allow the computation of lower bounds for the star discrepancy also in those settings where the running time of exact algorithms like the one of Dobkin, Eppstein, and Mitchell described in Section 3.2 are not feasible.

Algorithm 1: Simplified scheme of a threshold accepting algorithm for the computation of star discrepancy values. I is the runtime of the algorithm (number of iterations), and T is the threshold value for the acceptance of a new candidate solution z .

```

1 Initialization: Select  $y \in \overline{F}(X)$  uniformly at random and compute  $d_{\infty}^*(y, X)$ ;
2 for  $i = 1, 2, \dots, I$  do
3   Mutation Step: Select a random neighbor  $z$  of  $y$  and compute  $d_{\infty}^*(z, X)$ ;
4   Selection step: if  $d_{\infty}^*(z, X) - d_{\infty}^*(y, X) \geq T$  then  $y \leftarrow z$ ;
5 Output  $d_{\infty}^*(y, X)$ ;
```

Threshold accepting is based on a similar idea as the well-known simulated annealing algorithm [80]. In fact, it can be seen as a simulated annealing algorithm in which the selection step is derandomized, cf. Algorithm 1 for the general scheme of a threshold accepting algorithm. In our application of computing star discrepancy values, we accept a new candidate solution z if its local discrepancy $d_{\infty}^*(z, X)$ is not much worse than that of the previous step, and we discard z otherwise. More precisely, we accept z if and only if the difference $d_{\infty}^*(z, X) - d_{\infty}^*(y, X)$ is at least as large as some *threshold value* T . The threshold value is a parameter to be specified by the user. We typically have $T < 0$. $T < 0$ is a reasonable choice as it prevents the algorithm from getting stuck in some local maximum of the local discrepancy func-

tion. In the two threshold accepting algorithms presented below, T will be updated frequently during the run of the algorithm (details follow).

The first to apply threshold accepting to the computation of star discrepancies were P. Winker and K.-T. Fang [139]. Their algorithm was later improved in [56]. In this section, we briefly present the original algorithm from [139], followed by a short discussion of the modifications made in [56].

The algorithm of Winker and Fang uses the grid structure $\overline{\Gamma}(X)$. As in line 1 of Algorithm 1, they initialize the algorithm by selecting a grid point $y \in \overline{\Gamma}(X)$ uniformly at random. In the mutation step (line 3), a point z is sampled uniformly at random from the neighborhood $\mathcal{N}_k^{mc}(y)$ of y . For the definition of $\mathcal{N}_k^{mc}(y)$ let us first introduce the functions φ_j , $j \in \{1, \dots, d\}$, which order the elements in $\overline{\Gamma}_j(X)$; i.e., for $n_j := |\overline{\Gamma}_j(X)|$ the function φ_j is a permutation of $\{1, \dots, n_j\}$ with $x_j^{(\varphi_j(1))} \leq \dots \leq x_j^{(\varphi_j(n_j))} = 1$. For sampling a neighbor z of y we first draw mc coordinates j_1, \dots, j_{mc} from $\{1, \dots, d\}$ uniformly at random. We then select, independently and uniformly at random, for each j_i , $i = 1, \dots, mc$, a value $k_i \in \{-k, \dots, -1, 0, 1, \dots, k\}$. Finally, we let $z = (z_1, \dots, z_d)$ with

$$z_j := \begin{cases} y_j, & \text{for } j \notin \{j_1, \dots, j_{mc}\}, \\ y_j + k_j, & \text{for } j \in \{j_1, \dots, j_{mc}\}. \end{cases}$$

Both the values $mc \in \{1, \dots, d\}$ and $k \in \{1, \dots, n/2\}$ are inputs of the algorithm to be specified by the user. For example, if we choose $mc = 3$ and $k = 50$, then in the mutation step we change up to three coordinates of y , and for each such coordinate we allow to do up to 50 steps on the grid $\overline{\Gamma}_j(X)$, either to the “right” or to the “left”.

In the selection step (line 4), the search point z is accepted if its discrepancy value is better than that of y or if it is at least not worse than $d_\infty^*(y, X) + T$, for some threshold $T \leq 0$ that is determined in a precomputation step of the algorithm. Winker and Fang decided to keep the same threshold value for \sqrt{T} iterations, and to replace it every \sqrt{T} iterations with a new value $0 \geq T' > T$. The increasing sequence of threshold values guarantees that the algorithm has enough flexibility in the beginning to explore the search space, and enough stability towards its end so that it finally converges to a local maximum. This is achieved by letting T be very close to zero towards the end of the algorithm.

The algorithm by Winker and Fang performs well in numerical tests on rank-1 lattice rules, and it frequently computes the correct star discrepancy values in cases where this can be checked. However, as pointed out in [56], their algorithm does not perform very well in dimensions 10 and larger. For this reason, a number of modifications have been introduced in [56]. These modification also improve the performance of the algorithm in small dimensions.

The main differences of the algorithm presented in [56] include a refined neighborhood structure that takes into account the topological structure of the point set X and the usage of the concept of critical boxes as introduced in Definition 1. Besides this, there are few minor changes such as a variable size of the neighborhood structures and splitting the optimization process of $d_\infty^*(\cdot, X)$ into two separate processes

for $\delta(\cdot, X)$ and $\bar{\delta}(\cdot, X)$, respectively. Extensive numerical experiments are presented in [56]. As mentioned above, in particular for large dimension this refined algorithm seems to compute better lower bounds for $d_\infty^*(\cdot, X)$ than the basic one from [139].

We briefly describe the refined neighborhoods used in [56]. To this end, we first note that the neighborhoods used in Winker and Fang's algorithm do not take into account the absolute size of the gaps $x_j^{(\varphi(i+1))} - x_j^{(\varphi(i))}$ between two successive coordinates of grid points. This is unsatisfactory since large gaps usually indicate large differences in the local discrepancy function. Furthermore, for a grid cell $[y, z]$ in $\bar{\Gamma}(X)$ (i.e., $y, z \in \bar{\Gamma}(X)$ and $(z_j = x_j^{(\varphi_j(i_j))}) \Rightarrow (y_j = x_j^{(\varphi_j(i_j+1))})$ for all $j \in [d]$) with large volume, we would expect that $\bar{\delta}(y, X)$ or $\delta(z, X)$ are also rather large. For this reason, the following continuous neighborhood is considered. As in the algorithm by Winker and Fang we sample mc coordinates j_1, \dots, j_{mc} from $\{1, \dots, d\}$ uniformly at random. The neighborhood of y is the set $\mathcal{N}_k^{mc}(y) := [\ell_1, u_1] \times \dots \times [\ell_d, u_d]$ with

$$[\ell_j, u_j] := \begin{cases} \{y_j\}, & \text{for } j \notin \{j_1, \dots, j_{mc}\}, \\ [x^{(\varphi_j(\varphi_j^{-1}(y_j) - k \vee 1))}, x^{(\varphi_j(\varphi_j^{-1}(y_j) + k \wedge n_j))}], & \text{for } j \in \{j_1, \dots, j_{mc}\}, \end{cases}$$

where we abbreviate $\varphi_j^{-1}(y_j) - k \vee 1 := \max\{\varphi_j^{-1}(y_j) - k, 1\}$ and, likewise, $\varphi_j^{-1}(y_j) + k \wedge n_j := \min\{\varphi_j^{-1}(y_j) + k, n_j\}$. That is, for each of the coordinates $j \in \{j_1, \dots, j_{mc}\}$ we do k steps to the “left” and k steps to the “right”. We sample a point $\tilde{z} \in \mathcal{N}_k^{mc}(y)$ (not uniformly, but according to some probability function described below) and we round \tilde{z} once up and once down to the nearest critical box. For both these points \tilde{z}^- and \tilde{z}^+ we compute the local discrepancy value, and we set as neighbor of y the point $z \in \arg \max\{\bar{\delta}(\tilde{z}^-, X), \delta(\tilde{z}^+, X)\}$. The rounded grid points \tilde{z}^+ and \tilde{z}^- are obtained by the *snapping procedure* described in [56, Section 4.1]. We omit the details but mention briefly that rounding down to \tilde{z}^- can be done deterministically (\tilde{z}^- is unique), whereas for the upward rounding to \tilde{z}^+ there are several choices. The strategy proposed in [56] is based on a randomized greedy approach.

We owe the reader the explanation of how to sample the point \tilde{z} . To this end, we need to define the functions

$$\Psi_j : [\ell_j, u_j] \rightarrow [0, 1], r \mapsto \frac{r^d - (\ell_j)^d}{(u_j)^d - (\ell_j)^d}, j \in \{j_1, \dots, j_{mc}\}$$

whose inverse functions are

$$\Psi_j^{-1} : [0, 1] \rightarrow [\ell_j, u_j], s \mapsto \left(((u_j)^d - (\ell_j)^d)s + (\ell_j)^d \right)^{1/d}.$$

To sample \tilde{z} , we first sample values $s_1, \dots, s_{mc} \in [0, 1]$ independently and uniformly at random. We set $\tilde{z}_j := \Psi_j^{-1}(s_j)$ for $j \in \{j_1, \dots, j_{mc}\}$ and we set $\tilde{z}_j := y_j$ for $j \notin \{j_1, \dots, j_{mc}\}$. The intuition behind this probability measure is the fact that it favors larger coordinates than the uniform distribution. To make this precise, observe that in the case where $mc = d$, the probability measure on $\mathcal{N}_k^{mc}(y)$ is induced by the affine transformation from $\mathcal{N}_k^{mc}(y)$ to $[0, 1]^d$ and the polynomial product measure

$$\pi^d(dx) = \otimes_{j=1}^d f(x_j) \lambda(dx_j) \text{ with density function } f: [0, 1] \rightarrow \mathbb{R}, r \mapsto dr^{d-1}$$

on $[0, 1]^d$. The expected value of a point selected according to π^d is $d/(d+1)$, whereas the expected value of a point selected according to the uniform measure (which implicitly is the one employed by Winker and Fang) has expected value $1/2$. Some theoretical and experimental justifications for the choice of this probability measure are given in [56, Section 5.1]. The paper also contains numerical results for the computation of rank-1 lattice rules, Sobol' sequences, Faure sequences, and Halton sequences up to dimension 50. The new algorithm based on threshold accepting outperforms all other algorithms that we are aware of. For more recent applications of this algorithm we refer the reader to Section 5.2, where we present one example that indicates the future potential of this algorithm.

3.4.4 An Approach Based on Genetic Algorithms

A different randomized algorithm to calculate lower bounds for the star discrepancy of a given point set was proposed by M. Shah in [114]. His algorithm is a *genetic algorithm*. Genetic algorithms are a class of local search heuristics that have been introduced in the sixties and seventies of the last century, cf. [72] for the seminal work on evolutionary and genetic algorithms. In the context of geometric discrepancies, genetic algorithms have also been successfully applied to the design of low-discrepancy sequences (cf. Section 5.2 for more details). In this section, we provide a very brief introduction into this class of algorithms, and we outline its future potential in the analysis of discrepancies.

While threshold accepting algorithms take their inspiration from physics, genetic algorithms are inspired by biology. Unlike the algorithms presented in the previous section, in genetic algorithms, we typically do not keep only one solution candidate at a time, but we maintain a whole set of candidate solutions instead. This set is referred to as a *population* in the genetic algorithms literature. Algorithm 2 provides a high-level pseudo-code for genetic algorithms, adjusted again to the problem of computing lower bounds for the star discrepancy of a given point configuration. More precisely, this algorithm is a so-called $(\mu + \lambda)$ evolutionary algorithm (with $\lambda = C + M$ in this case). Evolutionary Algorithms are genetic algorithms that are based on Darwinian evolution principles. We discuss the features of such algorithms further below.

As mentioned above, the nomenclature used in the genetic algorithms literature deviates from the standard one used in introductory books to algorithms. We briefly name a few differences. In a high-level overview, a genetic algorithm runs in several *generations* (steps, iterations), in which the solution candidates (*individuals*) from the current population are being *recombined* and *mutated*.

We initialize such an algorithm by selecting μ individuals at random. They form the *parent population* (line 1). To this population we first apply a series of *crossover* steps (line 5), through which two (or more, depending on the implementation) individuals from the parent population are recombined. A very popular recombina-

Algorithm 2: Simplified scheme of a $(\mu + \lambda)$ evolutionary algorithm for the computation of star discrepancy values. I is the runtime of the algorithm (i.e., the number of iterations), C is the number of crossover steps per generation, and M is the number of mutation steps.

```

1 Initialization: Select  $y^{(1)}, \dots, y^{(\mu)} \in \overline{\Gamma}(X)$  uniformly at random and compute
    $d_{\infty}^*(y^{(1)}, X), \dots, d_{\infty}^*(y^{(\mu)}, X)$ ;
2 for  $i = 1, 2, \dots, I$  do
3   Crossover Step: for  $j = 1, 2, \dots, C$  do
4     Select two individuals  $y, y' \in \{y^{(1)}, \dots, y^{(\mu)}\}$  at random and create from  $y$  and  $y'$  a
     new individual  $z^{(j)}$  by recombination;
5     Compute  $d_{\infty}^*(z^{(j)}, X)$ ;
6   Mutation Steps: for  $j = 1, 2, \dots, M$  do
7     Select an individual  $y \in \{y^{(1)}, \dots, y^{(\mu)}, z^{(1)}, \dots, z^{(C)}\}$  at random;
8     Sample a neighbor  $n^{(j)}$  from  $y$  and compute  $d_{\infty}^*(n^{(j)}, X)$ ;
9   Selection step:
10    From  $\{y^{(1)}, \dots, y^{(\mu)}, z^{(1)}, \dots, z^{(C)}, n^{(1)}, \dots, n^{(M)}\}$  select—based on their local
    discrepancy values  $d_{\infty}^*(\cdot, X)$ —a subset of size  $\mu$ ;
11    Rename these individuals  $y^{(1)}, \dots, y^{(\mu)}$ ;
12 Output  $d_{\infty}^*(y, X)$ ;

```

tion operator is the so-called *uniform crossover* through which two search points $y, y' \in \overline{\Gamma}(X)$ are recombined to some search point z by setting $z_j := y_j$ with probability $1/2$, and by setting $z_j = y'_j$ otherwise. Several other recombination operators exist, and they are often adjusted to the problem at hand. The random choice of the parents to be recombined must not be uniform, and it may very well depend on the local discrepancy values $d_{\infty}^*(y^{(1)}, X), \dots, d_{\infty}^*(y^{(\mu)}, X)$, which are also referred to as the *fitness* of these individuals.

Once C such recombined individuals $z^{(1)}, \dots, z^{(C)}$ have been created and evaluated, we enter the *mutation* step, in which we compute for a number M of search points one neighbor each, cf. line 8. Similarly as in the threshold accepting algorithm, Algorithm 1, it is crucial here to find a meaningful notion of neighborhood. This again depends on the particular application. For our problem of computing lower bounds for the star discrepancy value of a given point configuration, we have presented two possible neighborhood definitions in Section 3.4.3. The newly sampled search points are evaluated, and from the set of old and new search points a new population is selected in the *selection* step, line 9. The selection typically depends again on the fitness values of the individuals. If always the μ search points of largest local discrepancy value are selected, we speak of an *elitist selection scheme*. This is the most commonly used selection operator in practice. However, to maintain more diversity in the population, it may also be reasonable to use other selection schemes, or to randomize the decision.

In the scheme of Algorithm 2, the algorithm runs for a fixed number I of iterations. However, as we mentioned in the beginning of Section 3.4.3, Shah's algorithm

works slightly different. His algorithm stops when no improvement has happened for some t iterations in a row, where t is a parameter to be set by the user.

The details of Shah's implementation can be found in [114]. His algorithm was used in [106] for the approximation of the star discrepancy value of 10-dimensional permuted Halton sequences. Further numerical results are presented in [114]. The results reported in [114], however, are not demanding enough to make a proper comparison between his algorithm and the ones presented in the previous section. On the few instances where a comparison seems meaningful, the results based on the threshold accepting algorithms outperform the ones of the genetic algorithm, cf. [56, Section 6.4] for the numerical results. Nevertheless, it seems that the computation of star discrepancy values with genetic and evolutionary algorithms is a promising direction and further research would be of interest.

Notes

In the literature one can find some attempts to compute for L_∞ -discrepancies the smallest possible discrepancy value of all n -point configurations. For the star discrepancy B. White determined in [138] the smallest possible discrepancy values for $n = 1, 2, \dots, 6$ in dimension $d = 2$, and T. Pillards, B. Vandewoestyne, and R. Cools in [111] for $n = 1$ in arbitrary dimension d . G. Larcher and F. Pillichshammer provided in [85] for the star and the extreme discrepancy the smallest discrepancy values for $n = 2$ in arbitrary dimension d . Furthermore, they derived for the isotrope discrepancy the smallest value for $n = 3$ in dimension $d = 2$ and presented good bounds for the smallest value for $n = d + 1$ in arbitrary dimension $d \geq 3$. (Note that the isotrope discrepancy of $n < d + 1$ points in dimension d is necessarily the worst possible discrepancy 1.)

4 Calculation of L_p -Discrepancies for $p \notin \{2, \infty\}$

This section is the shortest section in this book chapter. The reason for this is not that the computation of L_p -discrepancies, $p \notin \{2, \infty\}$, is an easy task which is quickly explained, but rather that not much work has been done so far and that therefore, unfortunately, not much is known to date. We present here a generalization of Warnock's formula for even p .

Let $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_d \geq 0$, and let $(\gamma_u)_{u \subseteq \{1, \dots, d\}}$ be the corresponding product weights; i.e., $\gamma_u = \prod_{j \in u} \gamma_j$ for all u . For this type of weights G. Leobacher and F. Pillichshammer derived a formula for the weighted L_p -star discrepancy $d_{p,\gamma}^*$ for arbitrary even positive integers p that generalizes the formula (12):

$$(d_{p,\gamma}^*(X))^p = \sum_{\ell=0}^p \binom{p}{\ell} \left(-\frac{1}{n}\right)^\ell \sum_{(i_1, \dots, i_\ell) \in \{1, \dots, n\}^\ell} \prod_{j=1}^d \left(1 + \gamma_j \frac{1 - \max_{1 \leq k \leq \ell} (x_j^{(i_k)})^{p-\ell+1}}{p-\ell+1}\right), \quad (35)$$

see [89, Thm. 2.1] (notice that in their definition of the weighted discrepancy they replaced the weights γ_u appearing in our definition (8) by $\gamma_u^{p/2}$). Recall that in the special case where $\gamma_1 = \gamma_2 = \dots = \gamma_d = 1$, the weighted L_p -star discrepancy $d_{p,\gamma}^*(X)$ coincides with Hickernell's modified L_p -discrepancy. Using the approach from [89] one may derive analogous formulas for the L_p -star and L_p -extreme discrepancy. The formula (35) can be evaluated directly at cost $O(dn^p)$, where the implicit constant in the big- O -notation depends on p . Obviously, the computational burden will become infeasible even for moderate values of n if p is very large.

Apart from the result in [89] we are not aware of any further results that are helpful for the calculation or approximation of weighted L_p -discrepancies.

Notes

The calculation of *average L_p -discrepancies* of Monte Carlo point sets attracted reasonable attention in the literature. One reason for this is that L_p -discrepancies such as, e.g., the L_p -star or L_p -extreme discrepancy, converge to the corresponding L_∞ -discrepancy if p tends to infinity, see, e.g., [49, 61]. Thus the L_p -discrepancies can be used to derive results for the corresponding L_∞ -discrepancy. In the literature one can find explicit representations of average L_p -discrepancies in terms of sums involving Stirling numbers of the first and second kind as well as upper bounds and formulas for their asymptotic behavior, see, e.g., [49, 61, 70, 89, 123].

5 Some Applications

We present some applications of algorithms that approximate discrepancy measures. The aim is to show here some more recent examples of how the algorithms are used in practice, what typical instances are, and what kind of problems occur. Of course, the selection of topics reflects the interest of the authors and is far from being complete. Further applications can, e.g., be found in the design of computer experiments ("experimental design"), see [38, 88], the generation of pseudo-random numbers, see [82, 87, 99, 126], or in computer graphics, see [27].

5.1 Quality Testing of Point Sets

A rather obvious application of discrepancy approximation algorithms is to estimate the quality of low-discrepancy point sets or, more generally, deterministic or randomized quasi-Monte Carlo point configurations.

Thiérmard, e.g., used his algorithm from [130], which we described in Section 3.4.1, to provide upper and lower bounds for the star discrepancy of Faure $(0, m, s)$ -nets [40] with sample sizes varying from 1,048,576 points in the smallest dimension $d = 2$ to 101 points in the largest dimension 100 (where, not very surprisingly, the resulting discrepancy is almost 1). He also uses his algorithm to compare the performance of two sequences of pseudo-random numbers, generated by `Rand()` and `MRG32k3a` [86], and Faure, Halton [57], and Sobol' [120] sequences by calculating bounds for their star discrepancy for sample sizes between 30 and 250 points in dimension 7.

For the same instances Thiérmard was able to calculate the exact star discrepancy of the Faure, Halton and Sobol' sequences by using his algorithm from [131], which we described in Section 3.4.2, see [131, Sect. 4.3].

In the same paper he provided the exact star discrepancy of Faure $(0, m, s)$ -nets ranging from sample sizes of 625 points in dimension 4 to 169 points in dimension 12. These results complement the computational results he achieved for (less demanding) instances in [128] with the help of the algorithm presented there.

Algorithms to approximate discrepancy measures were also used to judge the quality of different types of generalized Halton sequences. Since these sequences are also important for our explanation in Section 5.2, we give a definition here.

Halton sequences are a generalization of the one-dimensional van der Corput sequences. For a prime base p and a positive integer $i \in \mathbb{N}$ let $i = d_k d_{k-1} \dots d_2 d_1$ be the digital expansion of i in base p . That is, let d_1, \dots, d_k be such that $i = \sum_{\ell=1}^k d_\ell p^{\ell-1}$. Define the *radical inverse function* ϕ_p in base p by

$$\phi_p(i) := \sum_{\ell=1}^k d_\ell p^{-\ell}. \quad (36)$$

Let p_j denote the j th prime number. The i th element of the d -dimensional *Halton sequence* is defined as

$$x^{(i)} := (\phi_{p_1}(i), \dots, \phi_{p_d}(i)).$$

The Halton sequence is a low-discrepancy sequence, i.e., its first n points $X = (x^{(i)})_{i=1}^n$ in dimension d satisfy the star discrepancy bound

$$d_\infty^*(X) = O(n^{-1} \ln(n)^d). \quad (37)$$

In fact, the Halton sequence was the first construction for which (37) was verified for any dimension d [57], and up to now there is no sequence known that exhibits a better asymptotical behavior than (37).

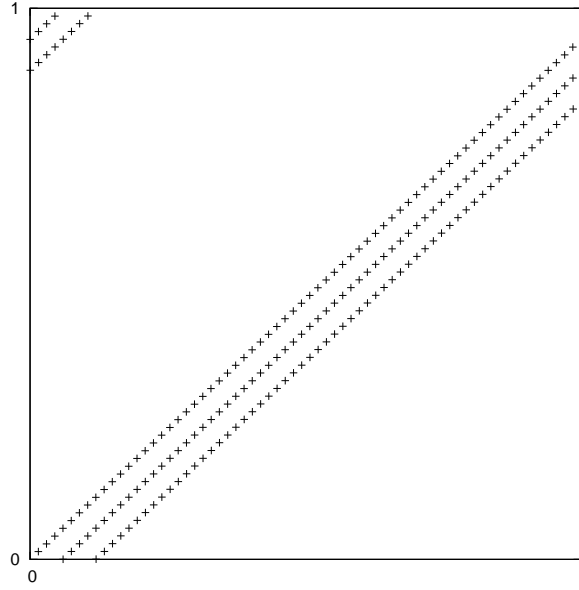


Fig. 2 The first 200 points of the 20-dimensional Halton sequence, projected to dimensions 19 and 20.

Nevertheless, for higher dimension d the use of larger prime p_j bases leads to some irregularity phenomenon, which is often referred to as *high correlation between higher bases* [9]. This phenomenon can easily be visualized by looking at two-dimensional projections of Halton sequences over higher coordinates, see Figure 2.

To reduce these undesirable effects, Braaten and Weller [9] suggested to use generalized Halton sequences. To obtain such a sequence, one applies digit permutations to the Halton sequence: For a permutation π_p of $\{0, 1, \dots, p-1\}$ with fixpoint $\pi_p(0) = 0$ define in analogy to (36) the *scrambled radical inverse function* ϕ_p^π by

$$\phi_p^\pi(i) := \sum_{\ell=1}^k \pi_p(d_\ell) p^{-\ell}.$$

The i th element of the *generalized* (or *scrambled*) *Halton sequence* in d dimensions is then defined by

$$x^{(i)}(\Pi) := (\phi_{p_1}^{\pi_{p_1}}(i), \dots, \phi_{p_d}^{\pi_{p_d}}(i)), \quad (38)$$

where we abbreviate $\Pi := (\pi_{p_1}, \dots, \pi_{p_d})$.

In several publications different permutations were proposed to shuffle the digits of the Halton sequence, see, e.g., [2, 18, 23, 41, 106, 133, 134] and the literature mentioned therein. Many authors tried to compare the quality of some of these

permutations by considering theoretical or numerical discrepancy bounds or other numerical tests.

Vandewoestyne and Cools [133], e.g., calculated the L_2 -star and L_2 -extreme discrepancy of several generalized Halton sequences by using Warnock's formula (9) and the corresponding modification (11), respectively. The instances they studied ranged from 10,000 sample points in dimension 8 to 1,000 points in dimension 64. They reported that those generalized Halton sequences performed best which are induced by the simple *reverse permutations* $\pi_p(0) = 0$ and $\pi_p(j) = p - j$ for $j = 1, \dots, p - 1$. These sequences showed usually a smaller L_2 -star and L_2 -extreme discrepancy as, e.g., the original Halton sequences, the generalized Halton sequences proposed in [2, 9] or the randomized Halton sequences from [134]; for more details see [133].

Ökten, Shah, and Goncharov [106] tried to compare different generalized Halton sequences by calculating bounds for their L_∞ -star discrepancy. For the calculations of the upper bounds for the star discrepancy they used Thiémarc's algorithm presented in [130]. For the calculation of the lower bounds they used Shah's algorithm [114], which gave consistently better lower bounds than Thiémarc's algorithm. They did the calculation for instances of 100 points in dimension 5 and 10 and studied different cases of prime bases. So they considered, e.g., different generalized Halton sequences in $d = 5$ where the prime bases are the 46th to 50th prime numbers p_{46}, \dots, p_{50} (which corresponds to the study of the projections over the last 5 coordinates of 50-dimensional sequences induced by the first 50 primes). Ökten et al. found that, apart from the original Halton sequence, the reverse permutations led to the highest discrepancy bounds, in contrast to their very low L_2 -star and L_2 -extreme discrepancy values. (This indicates again that the conventional L_2 -star and L_2 -extreme discrepancy are of limited use for judging the uniformity of point sets, cf. also the notes at the end of Section 2.) Furthermore, they reported that the average star discrepancy bounds of generalized Halton sequences induced by random permutations were rather low. For more details we refer to [106].

Algorithms to approximate star discrepancies were used in [33] to compare the quality of small samples of classical low-discrepancy points and Monte Carlo points with new point sets generated in a randomized and in a deterministic fashion. Here, "small samples" has to be understood as point sets whose size is small compared to the dimension, say, bounded by a constant times d or $d^{3/2}$. Notice that for this sample sizes asymptotic bounds like (37) do not provide any helpful information. The classical point sets were Halton-Hammersley points [24, 57, 99], Sobol' points [77, 120, 121], and Faure points shuffled by a Gray code [40, 126, 127]. The algorithms that generated the new point sets rely on certain random experiments based on randomized rounding with hard constraints [28, 122] and large deviation bounds that guarantee small discrepancies with high probability. The deterministic versions of these algorithms make use of derandomization techniques from [28, 34]. The concrete algorithms are randomized versions of the component-by-component (CBC) construction proposed in [30] and implemented in [32] and a randomized and derandomized version of the algorithm proposed in [29, 33]. The CBC construction has the advantage that it is faster and can be used to extend given low-discrepancy

point sets in the dimension, but its theoretical error bound is worse than the one for the latter algorithm.

In the numerical experiments different methods to approximate the discrepancy were used, depending on the instance sizes. For instances of 145 – 155 points in dimension 7 and 85 – 95 points in dimension 9 exact discrepancy calculations were performed by using the method of Bundschuh and Zhu [11]. For larger instances this was infeasible, so for instances ranging from 145 – 155 points in dimension 9 to 65 – 75 points in dimension 12 a variant of the algorithm of Dobkin, Eppstein, and Mitchell [27] was used that gained speed by allowing for an imprecision of order d/n . It should be noted that since the publication of these experiments, the implementation of this algorithm has been improved; a version that attains the same speed without making the d/n -order sacrifice of precision is available at the third author's homepage at <http://www.mpi-inf.mpg.de/~wahl/>. For the final set of instances, ranging from 145 – 155 points in dimension 12 to 95 – 105 points in dimension 21, the authors relied on upper bounds from Thiémarc's algorithm from [130] and on lower bounds from a randomized algorithm based on threshold accepting from [140], which is a precursor of the algorithm from [56]. Similarly as in [106], the randomized algorithm led consistently to better lower bounds than Thiémarc's algorithm. For the final instances of 95 – 105 points in dimension 21 the gaps between the upper and lower discrepancy bounds were roughly of the size 0.3, thus these results can only be taken as very coarse indicators for the quality of the considered point sets.

As expected, the experiments indicate that for small dimension and relatively large sample sizes the Faure and Sobol' points are superior, but the derandomized algorithms (and also their randomized versions) performed better than Monte Carlo, Halton-Hammersley, and Faure points in higher dimension for relatively small sample sizes. From the classical low-discrepancy point sets the Sobol' points performed best and were competitive for all instances. For more details see [33, Sect. 4].

5.2 *Generating Low-Discrepancy Points via an Optimization Approach*

In Section 3.4.4 we have seen that genetic algorithms can be used for an approximation of the star discrepancy values of a given point configuration. Here in this section we present another interesting application of biology-inspired algorithms in the field of geometric discrepancies.

Whereas the works presented in Sections 3.4 focus mainly on the computation of star discrepancy values, the authors of [23], F.-M. de Rainville, C. Gagné, O. Teytaud, and D. Laurendeau, apply evolutionary algorithms (cf. Section 3.4.4 for a brief introduction) to generate *low discrepancy point configurations*. Since the fastest known algorithms to compute the exact star discrepancy values have running time exponential in the dimension (cf. Section 3.2 for a discussion of this algorithm and Section 3.3 for related complexity-theoretic results), and the authors were—

naturally—interested in a fast computation of the results (cf. [23, Page 3]), the point configurations considered in [23] are optimized for Hickernell’s modified L_2 -star discrepancies, see equation (6) in the introduction to this chapter. As mentioned in Section 2, L_2 -star discrepancies can be computed efficiently via Warnock’s formula, cf. equation (9). Similarly, Hickernell’s modified L_2 -discrepancy can be computed efficiently with $O(dn^2)$ arithmetic operations.

The point sets generated by the algorithm of de Rainville, Gagné, Teytaud, and Laurendeau are generalized Halton sequences. As explained in Section 5.1, generalized Halton sequences are digit-permuted versions of the Halton sequence, cf. equation (38). The aim of their work is thus to find permutations $\pi_{p_1}, \dots, \pi_{p_d}$ such that the induced generalized Halton sequence has small modified L_2 -discrepancy. They present numerical results for sequences with 2,500 points in dimensions 20, 50, and 100 and they compare the result of their evolutionary algorithm with that of standard algorithms from the literature. Both for the modified L_2 -discrepancy as well as for the L_2 -star discrepancy the results indicate that the evolutionary algorithm is at least comparable, if not superior, to classic approaches.

The evolutionary algorithm of de Rainville, Gagné, Teytaud, and Laurendeau uses both the concept of crossover and mutation (as mentioned above, see Section 3.4.4 for a brief introduction into the notations of genetic and evolutionary algorithms). The search points are permutations, or, to be more precise configurations of permutations. The mutation operator shuffles the values of the permutation by deciding independently for each value in the domain of the permutation if its current value shall be swapped, and if so, with which value to swap it. The crossover operator recombines two permutations by iteratively swapping pairs of values in them.

We recall that in the definition of the generalized Halton sequence, equation (38), we need a vector (a *configuration*) $\Pi = (\pi_{p_1}, \dots, \pi_{p_d})$ of permutations of different length. The length of permutation π_{p_i} is determined by the i th prime number p_i . These configurations are extended component-by-component. That is, the evolutionary algorithm first computes an optimal permutation π_2 on $\{0, 1\}$ and sets $\Pi := (\pi_2)$. It then adds to Π , one after the other, the optimal permutation π_3 for $\{0, 1, 2\}$, the one π_5 on $\{0, 1, 2, 3, 4\}$, and so on. Here, “optimal” is measured with respect to the fitness function, which is simply the modified (Hickernell) L_2 -discrepancy of the point set induced by Π . We recall from equation (38) that a vector Π of permutations fully determines the resulting generalized Halton sequence.

The iterative construction of Π allows the user to use, for any two positive integers $d < D$, the same permutations $\pi_{p_1}, \dots, \pi_{p_d}$ in the first d dimensions of the two sequences in dimensions d and D , respectively.

The population sizes vary from 500 individuals for the 20-dimensional point sets to 750 individuals for the 50- and 100-dimensional point configurations. Similarly, the number of generations that the author allow the algorithm for finding the best suitable permutation ranges from 500 in the 20-dimensional case to 750 for the 50-dimensional one, and 1,000 for the 100-dimensional configuration.

It seems evident that combining the evolutionary algorithm of de Rainville, Gagné, Teytaud, and Laurendeau with the approximation algorithms presented in

Section 3.4 is a promising idea to generate low star-discrepancy point sequences. This is ongoing work of Doerr and de Rainville [35]. In their work, Doerr and de Rainville use the threshold accepting algorithm from [56] (cf. Section 3.4.3) for the intermediate evaluation of the candidate permutations. Only the final evaluation of the resulting point set is done by the exact algorithm of Dobkin, Eppstein, and Mitchell [27] (cf. Section 3.2). This allows the computation of low star-discrepancy sequences in dimensions where an exact evaluation of the intermediate permutations of the generalized Halton sequences would be far too costly. Using this approach, one can, at the moment, not hope to get results for such dimensions where the algorithm of Dobkin, Eppstein, and Mitchell does not allow for a final evaluation of the point configurations. However, due to the running time savings during the optimization process, one may hope to get good results for moderate dimensions for up to, say, 12 or 13. Furthermore, it seems possible to get a reasonable indication of good generating permutations in (38) for dimensions much beyond this, if one uses only the threshold accepting algorithm to guide the search. As is the case for all applications of the approximation algorithms presented in Section 3.4, in such dimensions, however, we do not have a *proof* that the computed lower bounds are close to the exact discrepancy values.

We conclude this section by emphasizing that, in contrast to the previous subsection, where the discrepancy approximation algorithms were only used to *compare* the quality of different point sets, here in this application the calculation of the discrepancy is an integral part of the optimization process to *generate* good low discrepancy point sets.

Notes

Also other researchers used algorithms to approximate discrepancy measures in combination with optimization approaches to generate low discrepancy point sets. A common approach described in [27] is to apply a multi-dimensional optimization algorithm repeatedly to randomly jiggled versions of low discrepancy point sets to search for smaller local minima of the discrepancy function. In [38, 90] the authors used the optimization heuristic threshold accepting to generate experimental designs with small discrepancy. The discrepancy measures considered in [38] are the L_∞ - and the L_2 -star discrepancy, as well as the centered, the symmetric, and the modified L_2 -discrepancy. The discrepancy measure considered in [90] is the central composite discrepancy for a flexible region as defined in [19].

5.3 Scenario Reduction in Stochastic Programming

Another interesting application of discrepancy theory is in the area of *scenario reduction*. We briefly describe the underlying problem, and we illustrate the importance of discrepancy computation in this context.

Many real-world optimization problems, e.g., of financial nature, are subject to stochastic uncertainty. *Stochastic programming* is a suitable tool to model such problems. That is, in stochastic programming we aim at minimizing or maximizing the expected value of a random process, typically taking into account several constraints. These problems, of both continuous and discrete nature, are often infeasible to solve optimally. Hence, to deal with such problems in practice, one often resorts to approximating the underlying (continuous or discrete) probability distribution by a discrete one of much smaller support. In the literature, this approach is often referred to as *scenario reduction*, cf. [64, 112] and the numerous references therein.

For most real-world situations, the size of the support of the approximating probability distribution has a very strong influence on (i.e., “determines”) the complexity of solving the resulting stochastic program. On the other hand, it is also true for all relevant similarity measures between the original and the approximating probability distribution that a larger support of the approximating distribution allows for a better approximation of the original one. Therefore, we have a fundamental trade-off between the running time of the stochastic program and the quality of the approximation. This trade-off has to be taken care of by the practitioner, and his decision typically depends on his time constraints and the availability of computational resources.

To explain the scenario reduction problem more formally, we need to define a suitable measure of similarity between two distributions. For simplicity, we regard here only distance measures for two discrete distributions whose support is a subset of $[0, 1]^d$. Unsurprisingly, a measure often regarded in the scenario reduction literature is based on discrepancies, cf. again [64, 112] and references therein. Formally, let P and Q be two discrete Borel distributions on $[0, 1]^d$ and let \mathcal{B} be a system of Borel sets of $[0, 1]^d$. The \mathcal{B} -discrepancy between P and Q is

$$\text{disc}_\infty(\mathcal{B}; P, Q) = \sup_{B \in \mathcal{B}} |P(B) - Q(B)|.$$

The right choice of the set \mathcal{B} of test sets depends on the particular application. Common choices for \mathcal{B} are

- \mathcal{C}_d , the class of all axis-parallel half-open boxes,
- \mathcal{R}_d , the class of all half-open axis-parallel boxes,
- $\mathcal{P}_{d,k}$, the class of all polyhedra having at most k vertices, and
- \mathcal{I}_d , the set of all closed, convex subsets,

which were introduced in Section 1.

In the scenario reduction literature, the discrepancy measure associated with \mathcal{C}_d is referred to as *star discrepancy*, *uniform*, or *Kolmogorov metric*; the one associated with \mathcal{I}_d is called *isotrope discrepancy*, whereas the distance measure induced by \mathcal{R}_d is simply called the *rectangle discrepancy* measure, and the one induced by $\mathcal{P}_{d,k}$ as *polyhedral discrepancy*.

With these distance measures at hand, we can now describe the resulting approximation problem. For a given distribution P of support $\{x^{(1)}, \dots, x^{(N)}\}$ we aim

at finding, for a given integer $n < N$, a distribution Q such that (i) the support $\{y^{(1)}, \dots, y^{(n)}\}$ of Q is a subset of the support of P of size n and that (ii) Q minimizes the distance between P and Q .

Letting $\delta(z)$ denote the Dirac measure placing mass one at point z , $p^{(i)} := P(x^{(i)})$, $1 \leq i \leq N$, and $q^{(i)} := Q(y^{(i)})$, $1 \leq i \leq n$, we can formulate this minimization problem as

$$\begin{aligned} \text{minimize} \quad & \text{disc}_\infty(\mathcal{B}; P, Q) = \text{disc}_\infty(\mathcal{B}; \sum_{i=1}^N p^{(i)} \delta(x^{(i)}), \sum_{i=1}^n q^{(i)} \delta(y^{(i)})) \quad (39) \\ \text{subject to} \quad & \{y^{(1)}, \dots, y^{(n)}\} \subseteq \{x^{(1)}, \dots, x^{(N)}\}, \\ & q^{(i)} > 0, \quad 1 \leq i \leq n, \\ & \sum_{i=1}^n q^{(i)} = 1. \end{aligned}$$

This optimization problem can be decomposed into an *outer* optimization problem of finding the support $\{y^{(1)}, \dots, y^{(n)}\}$ and an *inner* optimization problem of finding—for fixed support $\{y^{(1)}, \dots, y^{(n)}\}$ —the optimal probabilities $q^{(1)}, \dots, q^{(n)}$.

Some heuristics for solving both the inner and the outer optimization problems have been suggested in [64]. In that paper, Henrion, Küchler, and Römisch mainly regard the star discrepancy measure, but results for other distance measures are provided as well; see also the paper [63] by the same set of authors for results on minimizing the distance with respect to polyhedral discrepancies.

We present here a few details about the algorithms that Henrion, Küchler, and Römisch developed for the optimization problem (39) with respect to the star discrepancy. A more detailed description can be found in their paper [64].

Two simple heuristics for the outer optimization problem are *forward* and *backward selection*. In forward selection we start with an empty support set Y , and we add to Y , one after the other, the element from $X = \{x^{(1)}, \dots, x^{(N)}\}$ that minimizes the star discrepancy between P and Q , for an optimal allocation Q of probabilities $q^{(1)}, \dots, q^{(|Y|)}$ to the points in Y . We stop this forward selection when Y has reached its desired size, i.e., when $|Y| = n$.

Backward selection follows an orthogonal idea. We start with the full support set $Y = X$ and we remove from Y , one after the other, the element such that an optimal probability distribution Q on the remaining points of Y minimizes the star discrepancy $\text{disc}_\infty(\mathcal{E}_d; P, Q)$. Again we stop once $|Y| = n$. It seems natural that forward selection is favorable for values n that are much smaller than N , and, likewise, backward selection is more efficient when the difference $N - n$ is small.

For the inner optimization problem of determining the probability distribution Q for a fixed support Y , Henrion, Küchler, and Römisch formulate a linear optimization problem. Interestingly, independently of the discrepancy community, the authors develop to this end the concept of *supporting* boxes—a concept that coincides with the critical boxes introduced in Definition 1, Section 3.1. Using these supporting boxes, they obtain a linear program that has much less constraints than the natural straightforward formulation resulting from problem (39). However, the

authors remark that not the solution to the reduced linear program itself is computationally challenging, but the computation of the supporting (i.e., critical) boxes. Thus, despite significantly reducing the size of the original problem (39), introducing the concept of critical boxes alone is not sufficient to considerably reduce the computational effort required to solve problem (39). The problems considered in [64] are thus only of moderate dimension and moderate values of N and n . More precisely, results for four and eight dimensions with $N = 100$, $N = 200$, and $N = 300$ points are computed. The reduced scenarios have $n = 10$, 20, and 30 points.

Since most of the running time is caused by the computation of the star discrepancy values, it seems thus promising to follow a similar approach as in Section 5.2 and to use one of the heuristic approaches for star discrepancy estimation presented in Section 3.4 for an intermediate evaluation of candidate probability distributions Q . This would allow us to compute the exact distance of P and Q only for the resulting approximative distribution Q .

We conclude this section by mentioning that similar approaches could be useful also for the other discrepancy measures, e.g., the rectangle or the isotrope discrepancy of P and Q . However, for this to materialize, more research is needed to develop good approximations of such discrepancy values. (This is particularly the case for the isotrope discrepancy—see the comment regarding the isotrope discrepancy at the beginning of Section 3.)

Acknowledgements The authors would like to thank Sergei Kucherenko, Shu Tezuka, Tony Warnock, Greg Wasilkowski, Peter Winker, and an anonymous referee for their valuable comments.

Carola Doerr is supported by a Feodor Lynen postdoctoral research fellowship of the Alexander von Humboldt Foundation and by the Agence Nationale de la Recherche under the project ANR-09-JCJC-0067-01.

The work of Michael Gnewuch was supported by the German Science Foundation DFG under grant GN-91/3 and the Australian Research Council ARC.

The work of Magnus Wahlström was supported by the German Science Foundation DFG via its priority program “SPP 1307: Algorithm Engineering” under grant DO 749/4-1.

References

1. A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Massachusetts, 1974.
2. E. I. ATANASSOV, *On the discrepancy of the Halton sequences*, Math. Balkanica (N. S.), 18 (2004), pp. 15–32.
3. J. BECK, *Some upper bounds in the theory of irregularities of distribution*, Acta. Arith., 43 (1984), pp. 115–130.
4. ———, *On the discrepancy of convex plane sets*, Monatsh. Math., (1988), pp. 91–106.
5. J. BECK AND W. W. L. CHEN, *Irregularities of Distribution*, Cambridge University Press, Cambridge, 1987.
6. J. L. BENTLEY, *Algorithms for Klee’s rectangle problem*. Unpublished notes, Dept. of Computer Science, Carnegie Mellon University, 1977.

7. D. BILYK, *Roth's orthogonal function method in discrepancy theory and some new connections*, in A Panorama of Discrepancy Theory, W. W. L. Chen, A. Srivastav, and G. Travaglino, eds., Berlin, 2012, Springer.
8. D. BILYK, M. T. LACEY, I. PARISSIS, AND A. VAGHARSHAKYAN, *Exponential squared integrability of the discrepancy function in two dimensions*, *Mathematika*, 55 (2009), pp. 1–27.
9. E. BRAATEN AND G. WELLER, *An improved low-discrepancy sequence for multidimensional quasi-Monte Carlo integration*, *J. Comput. Physics*, 33 (1979), pp. 249–258.
10. L. BRANDOLINI, C. CHOIRAT, L. COLZANI, G. GIGANTE, R. SERI, AND G. TRAVAGLINI, *Quadrature rules and distribution of points on manifolds*, 2011. To appear in: *Ann. Scuola Norm. Sup. Pisa Cl. Sci.*
11. P. BUNDSCHUH AND Y. C. ZHU, *A method for exact calculation of the discrepancy of low-dimensional point sets I*, *Abh. Math. Sem. Univ. Hamburg*, 63 (1993), pp. 115–133.
12. T. M. CHAN, *A (slightly) faster algorithm for Klee's measure problem*, *Comput. Geom.*, 43 (2010), pp. 243–250.
13. B. CHAZELLE, *The Discrepancy Method*, Cambridge University Press, Cambridge, 2000.
14. J. CHEN, B. CHOR, M. FELLOWS, X. HUANG, D. W. JUEDES, I. A. KANJ, AND G. XIA, *Tight lower bounds for certain parameterized NP-hard problems*, *Inf. Comput.*, 201 (2005), pp. 216–231.
15. J. CHEN, X. HUANG, I. A. KANJ, AND G. XIA, *Strong computational lower bounds via parameterized complexity*, *J. Comput. Syst. Sci.*, 72 (2006), pp. 1346–1367.
16. W. W. L. CHEN AND M. M. SKRIGANOV, *Upper bounds in irregularities of point distribution*, in A Panorama of Discrepancy Theory, W. W. L. Chen, A. Srivastav, and G. Travaglino, eds., Berlin, 2012, Springer.
17. W. W. L. CHEN AND G. TRAVAGLINI, *Discrepancy with respect to convex polygons*, *J. Complexity*, 23 (2007), pp. 662–672.
18. H. CHI, M. MASCAGNI, AND T. WARNOCK, *On the optimal Halton sequence*, *Math. Comput. Simul.*, 70 (2005), pp. 9–21.
19. S. C. CHUANG AND Y. C. HUNG, *Uniform design over general input domains with applications to target region estimation in computer experiments*, *Comput. Statist. Data Anal.*, 54 (2010), pp. 219–232.
20. T. H. CORMEN, C. E. LEISERSON, R. L. RIVEST, AND C. STEIN, *Introduction to Algorithms (3. ed.)*, MIT Press, 2009.
21. M. DE BERG, O. CHEONG, M. VAN KREVELD, AND M. OVERMARS, *Computational Geometry: Algorithms and Applications*, Springer-Verlag, Berlin, 2008.
22. L. DE CLERCK, *A method for exact calculation of the star-discrepancy of plane sets applied to the sequence of Hammersley*, *Monatsh. Math.*, 101 (1986), pp. 261–278.
23. F.-M. DE RAINVILLE, C. GAGNÉ, O. TEYTAUD, AND D. LAURENDEAU, *Evolutionary optimization of low-discrepancy sequences*, *ACM Trans. Model. Comput. Simul.*, 22 (2012). Article 9.
24. J. DICK AND F. PILLICHSHAMMER, *Digital Nets and Sequences*, Cambridge University Press, Cambridge, 2010.
25. ———, *Discrepancy theory and quasi-Monte Carlo integration*, in A Panorama of Discrepancy Theory, W. W. L. Chen, A. Srivastav, and G. Travaglino, eds., Berlin, 2012, Springer.
26. J. DICK, I. H. SLOAN, X. WANG, AND H. WOŹNIAKOWSKI, *Liberating the weights*, *J. Complexity*, 20 (2004), pp. 593–623.
27. D. P. DOBKIN, D. EPPSTEIN, AND D. P. MITCHELL, *Computing the discrepancy with applications to supersampling patterns*, *ACM Trans. Graph.*, 15 (1996), pp. 354–376.
28. B. DOERR, *Generating randomized roundings with cardinality constraints and derandomizations*, in Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS'06), B. Durand and W. Thomas, eds., LNCS, vol. 3884, Berlin Heidelberg, 2006, Springer-Verlag, pp. 571–583.
29. B. DOERR AND M. GNEWUCH, *Construction of low-discrepancy point sets of small size by bracketing covers and dependent randomized rounding*, in Monte Carlo and Quasi-Monte Carlo Methods 2006, A. Keller, S. Heinrich, and H. Niederreiter, eds., Berlin Heidelberg, 2008, Springer-Verlag, pp. 299–312.

30. B. DOERR, M. GNEWUCH, P. KRITZER, AND F. PILLICHSHAMMER, *Component-by-component construction of low-discrepancy point sets of small size*, Monte Carlo Methods Appl., 14 (2008), pp. 129–149.
31. B. DOERR, M. GNEWUCH, AND A. SRIVASTAV, *Bounds and constructions for the star discrepancy via δ -covers*, J. Complexity, 21 (2005), pp. 691–709.
32. B. DOERR, M. GNEWUCH, AND M. WAHLSTRÖM, *Implementation of a component-by-component algorithm to generate low-discrepancy samples*, in Monte Carlo and Quasi-Monte Carlo Methods 2008, P. L’Ecuyer and A. B. Owen, eds., Berlin Heidelberg, 2009, Springer.
33. ———, *Algorithmic construction of low-discrepancy point sets via dependent randomized rounding*, J. Complexity, 26 (2010), pp. 490–507.
34. B. DOERR AND M. WAHLSTRÖM, *Randomized rounding in the presence of a cardinality constraint*, in Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX 2009), I. Finocchi and J. Hershberger, eds., SIAM, 2009. 162–174.
35. C. DOERR AND F.-M. DE RAINVILLE, *Constructing low star discrepancy point sets with genetic algorithms*, in Proc. of Genetic and Evolutionary Computation Conference (GECCO’13), ACM, 2013. To appear.
36. R. G. DOWNEY AND M. R. FELLOWS, *Parameterized Complexity*, Springer-Verlag, 1999.
37. M. DRMOTA AND R. F. TICHY, *Sequences, Discrepancies and Applications*, vol. 1651 of Lecture Notes in Mathematics, Springer, Berlin and Heidelberg, 1997.
38. K.-T. FANG, D. K. J. LIN, P. WINKER, AND Y. ZHANG, *Uniform design: theory and application*, Technometrics, 42 (2000), pp. 237–248.
39. H. FAURE, *Discrépance de suites associées à un système de numération (en dimension un)*, Bull. Soc. Math. France, 109 (1981), pp. 143–182.
40. ———, *Discrépance de suites associées à un système de numération (en dimension s)*, Acta Arith., 41 (1982), pp. 338–351.
41. H. FAURE AND C. LEMIEUX, *Generalized Halton sequences in 2008: A comparative study*, ACM Trans. Model. Comput. Simul., 19 (2009), pp. 15: 1–31.
42. H. FLUM AND M. GROHE, *Parameterized Complexity Theory*, Springer-Verlag New York, Inc., 2006.
43. K. FRANK AND S. HEINRICH, *Computing discrepancies of Smolyak quadrature rules*, J. Complexity, 12 (1996), pp. 287–314.
44. M. L. FREDMAN AND B. W. WEIDE, *On the complexity of computing the measure of $\bigcup [a_i, b_i]$* , Commun. ACM, 21 (1978), pp. 540–544.
45. H. GABAI, *On the discrepancy of certain sequences mod 1*, Indag. Math., 25 (1963), pp. 603–605.
46. M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco, 1979.
47. T. GERSTNER AND M. GRIEBEL, *Numerical integration using sparse grids*, Numer. Algorithms, (1998), pp. 209–232.
48. P. GIANNOPOULUS, C. KNAUER, M. WAHLSTRÖM, AND D. WERNER, *Hardness of discrepancy computation and epsilon-net verification in high dimensions*, J. Complexity, 28 (2012), pp. 162–176.
49. M. GNEWUCH, *Bounds for the average L^p -extreme and the L^∞ -extreme discrepancy*, Electron. J. Combin., (2005), pp. 1–11. Research Paper 54.
50. ———, *Bracketing numbers for axis-parallel boxes and applications to geometric discrepancy*, J. Complexity, 24 (2008), pp. 154–172.
51. ———, *Construction of minimal bracketing covers for rectangles*, Electron. J. Combin., 15 (2008). Research Paper 95.
52. ———, *Entropy, randomization, derandomization, and discrepancy*, in Monte Carlo and Quasi-Monte Carlo Methods 2010, L. Plaskota and H. Woźniakowski, eds., Berlin Heidelberg, 2012, Springer-Verlag, pp. 43–78.
53. ———, *Weighted geometric discrepancies and numerical integration on reproducing kernel Hilbert spaces*, J. Complexity, 28 (2012), pp. 2–17.

54. M. GNEWUCH AND A. ROŚCA, *On G -discrepancy and mixed Monte Carlo and quasi-Monte Carlo sequences*, Acta Univ. Apul. Math. Inform., 18 (2009), pp. 97–110.
55. M. GNEWUCH, A. SRIVASTAV, AND C. WINZEN, *Finding optimal volume subintervals with k points and calculating the star discrepancy are NP-hard problems*, J. Complexity, 25 (2009), pp. 115–127.
56. M. GNEWUCH, M. WAHLSTRÖM, AND C. WINZEN, *A new randomized algorithm to approximate the star discrepancy based on threshold accepting*, SIAM J. Numer. Anal., 50 (2012), pp. 781–807.
57. J. H. HALTON, *On the efficiency of certain quasi-random sequences of points in evaluating multidimensional integrals*, Numer. Math., 2 (1960), pp. 84–90.
58. J. H. HALTON AND S. K. ZAREMBA, *The extreme and L^2 -discrepancies of some plane sets*, Monatshefte Math., 73 (1969), pp. 316–328.
59. J. M. HAMMERSLEY, *Monte Carlo methods for solving multivariate problems*, Ann. New York Acad. Sci., 86 (1960), pp. 844–874.
60. S. HEINRICH, *Efficient algorithms for computing the L_2 discrepancy*, Math. Comp., 65 (1996), pp. 1621–1633.
61. S. HEINRICH, E. NOVAK, G. W. WASILKOWSKI, AND H. WOŹNIAKOWSKI, *The inverse of the star-discrepancy depends linearly on the dimension*, Acta Arith., 96 (2001), pp. 279–302.
62. P. HELLEKALEK AND H. LEEB, *Dyadic diaphony*, Acta Arith., 80 (1997), pp. 187–196.
63. R. HENRION, C. KÜCHLER, AND W. RÖMISCH, *Discrepancy distances and scenario reduction in two-stage stochastic mixed-integer programming*, J. Ind. Manag. Optim., 4 (2008), pp. 363–384.
64. ———, *Scenario reduction in stochastic programming with respect to discrepancy distances*, Comput. Optim. Appl., 43 (2009), pp. 67–93.
65. F. J. HICKERNELL, *A generalized discrepancy and quadrature error bound*, Math. Comp., 67 (1998), pp. 299–322.
66. F. J. HICKERNELL, I. H. SLOAN, AND G. W. WASILKOWSKI, *On tractability of weighted integration over bounded and unbounded regions in \mathbb{R}^s* , Math. Comp., 73 (2004), pp. 1885–1901.
67. F. J. HICKERNELL AND X. WANG, *The error bounds and tractability of quasi-Monte Carlo algorithms in infinite dimensions*, Math. Comp., 71 (2001), pp. 1641–1661.
68. A. HINRICHS, *Discrepancy of Hammersley points in Besov spaces of dominated mixed smoothness*, Math. Nachr., 283 (2010), pp. 477–483.
69. ———, *Discrepancy, integration and tractability*, in Monte Carlo and Quasi-Monte Carlo Methods 2012, J. Dick, F. Y. Kuo, G. Peters, and I. H. Sloan, eds., Springer, 2013. To appear.
70. A. HINRICHS AND H. WEYHAUSEN, *Asymptotic behavior of average L_p -discrepancies*, J. Complexity, 28 (2012), pp. 425–439.
71. E. HLAWKA, *Discrepancy and uniform distribution of sequences*, Compositio Math., 16 (1964), pp. 83–91.
72. J. H. HOLLAND, *Adaptation In Natural And Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
73. J. HRONKOVIĆ, *Algorithms for Hard Problems*, Springer, Berlin Heidelberg, 2003. 2nd edition.
74. R. IMPAGLIAZZO AND R. PATURI, *The complexity of k -SAT*, in Proc. 14th IEEE Conf. on Computational Complexity, 1999, pp. 237–240.
75. R. IMPAGLIAZZO, R. PATURI, AND F. ZANE, *Which problems have strongly exponential complexity?*, Journal of Computer and System Sciences, 63 (2001), pp. 512–530.
76. S. JOE, *An intermediate bound on the star discrepancy*, in Monte Carlo and Quasi-Monte Carlo Methods 2010, L. Plaskota and H. Woźniakowski, eds., Berlin Heidelberg, 2012, Springer-Verlag, pp. 251–269.
77. S. JOE AND F. Y. KUO, *Constructing Sobol’ sequences with better two-dimensional projections*, SIAM J. Sci. Comput., 30 (2008), pp. 2635–2654.
78. S. JOE AND I. H. SLOAN, *On computing the lattice rule criterion R* , Math. Comp., 59 (1992), pp. 557–568.

79. W. KAHAN, *Further remarks on reducing truncation errors*, Communications of the ACM, 8 (1965), p. 40.
80. S. KIRCKPATRICK, C. GELATT, AND M. VECCHI, *Optimization by simulated annealing*, Science, 20 (1983), pp. 671–680.
81. V. KLEE, *Can the measure of $\bigcup_1^m [a_i, b_i]$ be computed in less than $O(n \log n)$ steps?*, The American Mathematical Monthly, 84 (1977), pp. 284–285.
82. D. E. KNUTH, *The Art of Computer Programming. Vol. 2. Seminumerical Algorithms*, Addison-Wesley Series in Computer Science and Information Processing, Addison-Wesley, Reading, Mass., 3rd ed., 1997.
83. L. KUIPERS AND H. NIEDERREITER, *Uniform distribution of sequences*, Wiley-Interscience [John Wiley & Sons], New York, 1974. Pure and Applied Mathematics.
84. F. Y. KUO, *Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces*, J. Complexity, 19 (2003), pp. 301–320.
85. G. LARCHER AND F. PILLICHSHAMMER, *A note on optimal point distributions in $[0, 1]^s$* , J. Comput. Appl. Math., 206 (2007), pp. 977–985.
86. P. L'ECUYER, *Good parameter sets for combined multiple recursive random number generators*, Oper. Res., 47 (1999), pp. 159–164.
87. P. L'ECUYER AND P. HELLEKALEK, *Random number generators: selection criteria and testing*, in Random and quasi-random point sets, vol. 138 of Lecture Notes in Statist., Springer, New York, 1998, pp. 223–265.
88. C. LEMIEUX, *Monte Carlo and Quasi-Monte Carlo Sampling*, Springer, New York, 2009.
89. G. LEOBACHER AND F. PILLICHSHAMMER, *Bounds for the weighted L_p discrepancy and tractability of integration*, J. Complexity, (2003), p. 529547.
90. D. K. J. LIN, C. SHARPE, AND P. WINKER, *Optimized U -type designs on flexible regions*, Comput. Statist. Data Anal., 54 (2010), pp. 1505–1515.
91. L. MARKHASIN, *Quasi-Monte Carlo methods for integration of functions with dominating mixed smoothness in arbitrary dimension*. Preprint, ArXiv 1201.2311v3, 2012. To appear in J. Complexity.
92. ———, *Discrepancy of generalized Hammersley type point sets in Besov spaces with dominating mixed smoothness*, Uniform Distribution Theory, 8 (2013), pp. 135–164.
93. J. MATOUŠEK, *On the L_2 -discrepancy for anchored boxes*, J. Complexity, 14 (1998), pp. 527–556.
94. ———, *Geometric Discrepancy*, Springer, Berlin, 2nd ed., 2010.
95. K. MEHLHORN, *Multi-dimensional Searching and Computational Geometry, Data Structures and Algorithms 3*, Springer-Verlag, Berlin/New York, 1984.
96. W. MOROKOFF AND R. CAFLISCH, *Quasi-random sequences and their discrepancies*, SIAM J. Sci. Comput., 15 (1994), pp. 1251–1279.
97. H. NIEDERREITER, *Discrepancy and convex programming*, Ann. Mat. Pura Appl., 93 (1972), pp. 89–97.
98. ———, *Methods for estimating discrepancy*, in Applications of Number Theory to Numerical Analysis, S. K. Zaremba, ed., New York, 1972, Academic Press, pp. 203–236.
99. ———, *Random Number Generation and Quasi-Monte Carlo Methods*, vol. 63 of SIAM CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, 1992.
100. E. NOVAK AND K. RITTER, *High dimensional integration of smooth functions over cubes*, Numer. Math., 75 (1996), pp. 79–97.
101. E. NOVAK AND H. WOŹNIAKOWSKI, *Tractability of Multivariate Problems. Vol. 1, Linear Information*, EMS Tracts in Mathematics, European Mathematical Society (EMS), Zürich, 2008.
102. ———, *L_2 discrepancy and multivariate integration*, in Analytic number theory. Essays in honour of Klaus Roth., W. W. L. Chen, W. T. Gowers, H. Halberstam, W. M. Schmidt, and R. C. Vaughan, eds., Cambridge, 2009, Cambridge Univ. Press, pp. 359–388.
103. ———, *Tractability of Multivariate Problems. Vol. 2, Standard Information for Functionals*, EMS Tracts in Mathematics, European Mathematical Society (EMS), Zürich, 2010.

104. D. NUYENS AND R. COOLS, *Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces*, Math. Comp., 75 (2006), pp. 903–920.
105. G. ÖKTEN, *Error reduction techniques in quasi-Monte Carlo integration*, Math. Comput. Modelling, 30 (1999), pp. 61–69.
106. G. ÖKTEN, M. SHAH, AND Y. GONCHAROV, *Random and deterministic digit permutations of the Halton sequence*, in Monte Carlo and Quasi-Monte Carlo Methods 2010, L. Plaskota and H. Woźniakowski, eds., Berlin Heidelberg, 2012, Springer-Verlag, pp. 609–622.
107. M. H. OVERMARS AND C.-K. YAP, *New upper bounds in Klee’s measure problem*, SIAM J. Comput., 20 (1991), pp. 1034–1045.
108. S. H. PASKOV, *Average case complexity of multivariate integration for smooth functions*, J. Complexity, 9 (1993), pp. 291–312.
109. P. PEART, *The dispersion of the Hammersley sequence in the unit square*, Monatshefte Math., 94 (1982), pp. 249–261.
110. T. PILLARDS AND R. COOLS, *A note on E. Thiémond’s algorithm to compute bounds for the star discrepancy*, J. Complexity, 21 (2005), pp. 320–323.
111. T. PILLARDS, B. VANDEWOESTYNE, AND R. COOLS, *Minimizing the L_2 and L_∞ star discrepancies of a single point in the unit hypercube*, J. Comput. Appl. Math., 197 (2006), pp. 282–285.
112. W. RÖMISCH, *Scenario reduction techniques in stochastic programming*, in SAGA 2009, O. Watanabe and T. Zeugmann, eds., vol. 43 of LNCS, Berlin-Heidelberg, 2009, Springer-Verlag, pp. 1–14.
113. W. M. SCHMIDT, *Irregularities of distribution IX*, Acta Arith., (1975), pp. 385–396.
114. M. SHAH, *A genetic algorithm approach to estimate lower bounds of the star discrepancy*, Monte Carlo Methods Appl., 16 (2010), pp. 379–398.
115. I. H. SLOAN AND S. JOE, *Lattice Methods for Multiple Integration*, Clarendon Press, Oxford, 1994.
116. I. H. SLOAN, F. Y. KUO, AND S. JOE, *On the step-by-step construction of quasi-monte carlo integration rules that achieve strong tractability error bounds in weighted sobolev spaces*, Math. Comp., 71 (2002), pp. 1609–1640.
117. I. H. SLOAN AND A. V. REZTSOV, *Component-by-component construction of good lattice rules*, Math. Comp., 71 (2002), pp. 263–273.
118. I. H. SLOAN AND H. WOŹNIAKOWSKI, *When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?*, J. Complexity, 14 (1998), pp. 1–33.
119. S. A. SMOLYAK, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Dokl. Akad. Nauk. SSSR, 4 (1963), pp. 1042–1045.
120. I. M. SOBOL’, *The distribution of points in a cube and the approximate evaluation of integrals*, Zh. Vychisl. Mat. i. Mat. Fiz., 7 (1967), pp. 784–802.
121. I. M. SOBOL’, D. ASOTSKY, A. KREININ, AND S. KUCHERENKO, *Construction and comparison of high-dimensional Sobol’ generators*, Wilmott Magazine, Nov. 2011 (2011), pp. 64–79.
122. A. SRINIVASAN, *Distributions on level-sets with applications to approximation algorithms*, in Proceedings of FOCS’01, 2001, pp. 588–597.
123. S. STEINERBERGER, *The asymptotic behavior of the average L_p -discrepancies and a randomized discrepancy*, Electron. J. Combin., (2010), pp. 1–18. Research Paper 106.
124. W. STUTE, *Convergence rates for the isotrope discrepancy*, Ann. Probability, (1977), pp. 707–723.
125. V. N. TEMPLYAKOV, *On a way of obtaining lower estimates for the errors of quadrature formulas*, Math. USSR Sbornik, 71 (1992), pp. 247–257.
126. S. TEZUKA, *Uniform Random Numbers: Theory and Practice*, Kluwer Academic Publishers, Boston, 1995.
127. E. THIÉMOND, *Economic generation of low-discrepancy sequences with a b-ary gray code*, EPFL-DMA-ROSO, RO981201, <http://rosowww.epfl.ch/papers/grayfaure/>, 1998.
128. ———, *Computing bounds for the star discrepancy*, Computing, 65 (2000), pp. 169–186.

129. ———, *Sur le calcul et la majoration de la discr pance   l'origine*, PhD thesis  cole polytechnique f d rale de Lausanne EPFL, nbr 2259, 2000. Available from <http://infoscience.epfl.ch/record/32735>.
130. ———, *An algorithm to compute bounds for the star discrepancy*, J. Complexity, 17 (2001), pp. 850–880.
131. ———, *Optimal volume subintervals with k points and star discrepancy via integer programming*, Math. Meth. Oper. Res., 54 (2001), pp. 21–45.
132. H. TRIEBEL, *Bases in Function Spaces, Sampling, Discrepancy, Numerical Integration*, EMS Tracts in Mathematics 11, European Mathematical Society, Z rich, 2010.
133. B. VANDEWOESTYNE AND R. COOLS, *Good permutations for deterministic scrambled Halton sequences in terms of L_2 -discrepancy*, J. Comput. Appl. Math., 189 (2006), pp. 341–361.
134. X. WANG AND F. J. HICKERNELL, *Randomized Halton sequences*, Math. Comput. Modelling, 32 (2000), pp. 887–899.
135. T. T. WARNOCK, *Computational investigations of low-discrepancy point sets*, in Applications of number theory to numerical analysis, S. K. Zaremba, ed., New York, 1972, Academic Press, pp. 319–343.
136. ———, 2013. Personal communication.
137. G. W. WASILKOWSKI AND H. WOŻNIAKOWSKI, *Explicit cost bounds of algorithms for multivariate tensor product problems*, J. Complexity, 11 (1995), pp. 1–56.
138. B. E. WHITE, *On optimal extreme-discrepancy point sets in the square*, Numer. Math., 27 (1976/77), p. 157164.
139. P. WINKER AND K. T. FANG, *Applications of threshold-accepting to the evaluation of the discrepancy of a set of points*, SIAM J. Numer. Anal., 34 (1997), pp. 2028–2042.
140. C. WINZEN, *Approximative Berechnung der Sterndiskrepanz*. Diplomarbeit, 2007.
141. H. WOŻNIAKOWSKI, *Average case complexity of multivariate integration*, Bull. Amer. Math. Soc. (N. S.), 24 (1991), pp. 185–191.
142. S. K. ZAREMBA, *Some applications of multidimensional integration by parts*, Ann. Polon. Math., 21 (1968), pp. 85–96.
143. ———, *La discr pance isotrope et l'int gration num rique*, Ann. Mat. Pura Appl., 37 (1970), pp. 125–136.
144. C. ZENGER, *Sparse grids*, in Parallel Algorithms for Partial Differential Equations, W. Hackbusch, ed., Braunschweig, 1991, Vieweg, pp. 241–251.
145. P. ZINTERHOF, * ber einige Absch tzungen bei der Approximation von Funktionen mit Gleichverteilungsmethoden*, Sitzungsber.  sterr. Akad. Wiss. Math.-Naturwiss. Kl. II, 185 (1976), pp. 121–132.