AUSTRIAN ACADEMY OF SCIENCES
RICAM – JOHANN RADON INSTITUTE FOR COMPUTATIONAL AND APPLIED MATHEMATICS

A note on the CBC-DBD construction of lattice rules with general positive weights

P. Kritzer

RICAM-Report 2022-14

WWW.RICAM.OEAW.AC.AT
ALTENBERGER STRASSE 69 | 4040 LINZ, AUSTRIA
T: +43 732 2468-5222 | F: +43 732 2468-5212 | OFFICE@RICAM.OEAW.AC.AT
IBAN: AT541100000262650519 | BIC: BKAUATWW | VAT: ATU37116303

RICAM

A note on the CBC-DBD construction of lattice rules with general positive weights

Peter Kritzer

August 29, 2022

Abstract

Lattice rules are among the most prominently studied quasi-Monte Carlo methods to approximate multivariate integrals. A rank-1 lattice rule to approximate an s-dimensional integral is fully specified by its *generating vector* $\boldsymbol{z} \in \mathbb{Z}^s$ and its number of points N. While there are many results on the existence of "good" rank-1 lattice rules, there are no explicit constructions of good generating vectors for dimensions $s \geq 3$. This is why one usually resorts to computer search algorithms. In the paper [5], we showed a component-by-component digit-by-digit (CBC-DBD) construction for good generating vectors of rank-1 lattice rules for integration of functions in weighted Korobov classes. However, the result in that paper was limited to product weights. In the present paper, we shall generalize this result to arbitrary positive weights, thereby answering an open question posed in [5].

Keywords: Numerical integration; lattice points; quasi-Monte Carlo methods; weighted function spaces; digit-by-digit construction; component-by-component construction; fast construction.
2020 MSC: 65D30, 65D32, 41A55, 41A63.

1 Introduction

In high-dimensional numerical integration, one frequently uses *quasi-Monte Carlo (QMC)* rules $Q_{N,s}$ to efficiently approximate integrals I_s,

$$I_s(f) := \int_{[0,1]^s} f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \quad \approx \quad Q_{N,s}(f) := \frac{1}{N} \sum_{k=0}^{N-1} f(\boldsymbol{x}_k)$$

for a suitably chosen integrand f (usually, we assume that f is an element of a Hilbert or Banach space, see below). I.e., a QMC rule is an equal-weight quadrature rule, and it is—opposed to *Monte Carlo* rules—based on deterministically chosen integration nodes $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1} \in [0,1)^s$. A non-trivial question in this field is how the nodes of a QMC rule can be chosen in order to guarantee a low integration error. Depending on the properties of the integrand f under consideration, two classes of integration node sets have gained most attention in the past decades, namely *digital nets and sequences*, introduced by Sobol', Faure, and Niederreiter (see, e.g., [4, 9]), and *lattice point sets*, introduced by Korobov and Hlawka (see, e.g., [2, 13]). In this paper, we focus on an instance of the latter, namely so called *rank-1 lattice point sets* yielding *lattice rules* when used in QMC rules. These are sets of integration nodes with N points

$$\boldsymbol{x}_k := \left(\left\{ \frac{k z_1}{N} \right\}, \ldots, \left\{ \frac{k z_s}{N} \right\} \right) \in [0,1)^s, \qquad \text{for } k \in \{0, 1, \ldots, N-1\},$$

and where $\{x\} = x - \lfloor x \rfloor$ denotes the fractional part of a real x. Note that, given N and s, the lattice rule is completely determined by the choice of the *generating vector* $\boldsymbol{z} = (z_1, \ldots, z_s) \in \mathbb{Z}_N^s$, where $\mathbb{Z}_N := \{0, \ldots, N-1\}$. We remark that it is sufficient to consider the choice of z_j modulo N since $\{kz_j/N\} = (kz_j \bmod N)/N$ for integer k, N, and z_j. Obviously, not every choice of a generating vector $\boldsymbol{z}$ also yields a lattice rule of good quality for approximating the integral. For dimensions $s \leq 2$, explicit constructions of good generating vectors are available, see, e.g., [2,9,13], but there are no explicit constructions of good generating vectors known for $s > 2$. Therefore, one usually studies search algorithms for generating vectors of good lattice rules, which are designed to make a certain error criterion sufficiently small. Korobov [7], and later Sloan and his collaborators [14], introduced a *component-by-component (CBC) construction*, which is a greedy algorithm constructing the components $z_1, \ldots, z_s$ of $\boldsymbol{z} \in \mathbb{Z}_N^s$ successively, choosing one z_j at a time, and keeping previous components fixed. It was shown in [8] for prime N and in [1] for composite N that the CBC construction yields generating vectors with essentially optimal convergence rates for particular spaces of integrands. For suitable choices of function spaces, there exist fast implementations of CBC constructions, which have a run-time of order $\mathcal{O}(sN \log N)$, see [11, 12].

In this short paper, we consider an alternative to the common CBC search algorithm, namely a so-called CBC-DBD algorithm, which is based on the idea that the single components of the generating vector $\boldsymbol{z}$ are chosen one after another, and each component is constructed *digit-by-digit (DBD)*, i.e., the base-2 digits of the components are chosen in a greedy fashion, starting with the least significant digit. The principle idea of this construction is due to Korobov (see [7]), and it was shown to work for a modern function space setting in the recent paper [5]. However, due to technical difficulties, the main result in [5] does not hold in full generality, but only with certain restrictions on the *weight* parameters of the function space involved. In the present paper, we close this gap (see below for a more detailed explanation).

In order to make use of a construction method like the CBC or the CBC-DBD construction, one needs to define the search criterion which the algorithm is based on. Usually, this criterion is related to the class of integrands under consideration. In [5], we consider a Banach space (called *Korobov space* or *Korobov class*) of functions.

We consider integrands f with absolutely converging Fourier series,

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{m} \in \mathbb{Z}^s} \hat{f}(\boldsymbol{m})\, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{m} \cdot \boldsymbol{x}} \qquad \text{with} \qquad \hat{f}(\boldsymbol{m}) := \int_{[0,1]^s} f(\boldsymbol{x})\, \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{m} \cdot \boldsymbol{x}} \,\mathrm{d}\boldsymbol{x},$$

i.e., $\hat{f}(\boldsymbol{m})$ is the $\boldsymbol{m}$-th Fourier coefficient of f, where $\boldsymbol{m} \cdot \boldsymbol{x} := \sum_{j=1}^s m_j x_j$ is the vector dot product. Since the Fourier series are absolutely summable, the Fourier series are pointwise convergent, 1-periodic, and continuous. Here, we will consider Banach spaces which are based on assuming sufficient decay of the Fourier coefficients of its elements to guarantee certain smoothness. These spaces will be denoted by $E_{s,\boldsymbol{\gamma}}^{\alpha}$, where s denotes the number of variables the functions depend on, $\alpha > 1$ is a real number frequently referred to as the smoothness parameter, and $\boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u} \subset \mathbb{N}}$, is a set of strictly positive *weights* to model the importance of different subsets of components. In this context, by the notation "$\mathfrak{u} \subset \mathbb{N}$" we mean all finite subsets $\mathfrak{u}$ of the positive integers. We extend the range of $\mathfrak{u}$ to all finite subsets of $\mathbb{N}$ since we also would like to include results that hold asymptotically when s tends to infinity.

Intuitively, a large $\gamma_{\mathfrak{u}}$ corresponds to a high influence of the variables x_j with $j \in \mathfrak{u}$, while a small $\gamma_{\mathfrak{u}}$ means low influence. The idea of weights goes back to Sloan and Woźniakowski [15], and will be made more precise by incorporating the weights in the norm of the space $E_{s,\boldsymbol{\gamma}}^{\alpha}$ below. We are interested in conditions on the weights such that we can bound the integration error independently of s. This is called *strong polynomial tractability*, see, e.g., [10], for a general reference.

In the literature on QMC methods, the weights $\boldsymbol{\gamma}$ are not always chosen as fully general. Indeed, a very common variant is to work with an infinite sequence $(\gamma_j)_{j \geq 1}$ and to put $\gamma_{\mathfrak{u}} := \prod_{j \in \mathfrak{u}} \gamma_j$, which is the case of *product weights*. Other variants are, e.g., finite-order weights or product-and-order-dependent weights, and we refer the reader to [3] for a more detailed discussion. The main result in [5] is shown for the special case of product weights, due to one step in the proof that we were not able to carry out for more general weights. In the present paper, we close this gap, and show a corresponding result for general weights $\boldsymbol{\gamma}$. The only assumption we make is that all $\gamma_{\mathfrak{u}}$ are positive, which is to avoid too much technical notation. Presumably, a similar result also holds for weights that are allowed to be zero.

Now, for a given smoothness parameter $\alpha > 1$ and strictly positive weights $\{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$, we define, for any $\boldsymbol{m} = (m_1, \ldots, m_s) \in \mathbb{Z}^s$,

$$r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{m}) := \gamma_{\mathrm{supp}(\boldsymbol{m})}^{-1} \prod_{j \in \mathrm{supp}(\boldsymbol{m})} |m_j|^{\alpha} \,,$$

where $\mathrm{supp}(\boldsymbol{m}) := \{j \in \{1, \ldots, s\} : m_j \neq 0\}$ is the support of $\boldsymbol{m}$. We set $\gamma_{\emptyset} = 1$, so $r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{0}) = 1$. Using this notation, we define the norm of our Banach space $E^{\alpha}_{s,\boldsymbol{\gamma}}$,

$$\|f\|_{E^{\alpha}_{s,\boldsymbol{\gamma}}} := \sup_{\boldsymbol{m}\in\mathbb{Z}^s} |\hat{f}(\boldsymbol{m})|\, r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{m}), \quad \text{for} \quad f \in E^{\alpha}_{s,\boldsymbol{\gamma}}, \tag{1}$$

eq:norm

and our weighted function space by

$$E^{\alpha}_{s,\boldsymbol{\gamma}} := \left\{ f \in L^2([0,1]^s) \colon \|f\|_{E^{\alpha}_{s,\boldsymbol{\gamma}}} < \infty \right\}.$$

The criterion by which we assess the quality of a given rank-1 lattice rule is the *worst-case error*, which is defined as

$$e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) := \sup_{\substack{f \in E^{\alpha}_{s,\boldsymbol{\gamma}} \\ \|f\|_{E^{\alpha}_{s,\boldsymbol{\gamma}}} \leq 1}} |I_s(f) - Q_{N,s}(f, \boldsymbol{z})| \,,$$

where we stress the dependence on $\boldsymbol{z}$ in our notation of the error and of the QMC rule $Q_{N,s}$.

Remark 1. Note that since $\alpha > 1$, the membership of f to the space $E^{\alpha}_{s,\boldsymbol{\gamma}}$ implies the absolute convergence of its Fourier series, which in turn entails that f is continuous and 1-periodic with respect to each variable. In addition, if $f \in E^{\alpha}_{s,\boldsymbol{\gamma}}$, f has 1-periodic continuous mixed partial derivatives $f^{(\boldsymbol{\tau})}$ for any $\boldsymbol{\tau} \in \mathbb{N}_0^s$ with all $\tau_j < \alpha - 1$. Furthermore, it is known that the optimal convergence rate of the worst-case error in the function space is of order $\mathcal{O}(N^{-\alpha})$. We also remark that, by slightly modifying the definition of the norm in (1), one could define a function space that is similar to $E^{\alpha}_{s,\boldsymbol{\gamma}}$, but a Hilbert space. The latter is frequently studied in the literature on lattice rules (see, e.g., [2]), and the worst-case error in $E^{\alpha}_{s,\boldsymbol{\gamma}}$ is exactly the square of the worst-case error in the Hilbert case. For consistency with [5], we stay with $E^{\alpha}_{s,\boldsymbol{\gamma}}$ in this note, and refer to that paper for further information on the properties of the function space, as well as for further references.

We write $\mathbb{N} := \{1, 2, \ldots\}$ for the set of natural numbers and $\mathbb{N}_0 := \{0, 1, 2, \ldots\}$, $\mathbb{Z}$ for the set of integers and $\mathbb{Z}_N := \{0, \ldots, N-1\}$. To denote sets of components we use fraktur font, e.g., $\mathfrak{u} \subset \mathbb{N}$. As a shorthand we write $\{k_1 : k_2\}$ for the set $\{k_1, k_1 + 1, \ldots, k_2\}$, for two integers k_1, k_2 with $k_1 \leq k_2$. To denote the projection of a vector $\boldsymbol{x} \in [0,1)^s$ or $\boldsymbol{m} \in \mathbb{Z}^s$ onto the components in a set $\mathfrak{u} \subseteq \{1{:}s\}$ we write $\boldsymbol{x}_{\mathfrak{u}} := (x_j)_{j\in\mathfrak{u}}$ or $\boldsymbol{m}_{\mathfrak{u}} := (m_j)_{j\in\mathfrak{u}}$, respectively.

2 The CBC-DBD construction yields optimal convergence rates for general weights

As outlined above, the paper [5] shows a result for lattice rules with generating vectors obtained by a CBC-DBD algorithm, which implies that the lattice rules achieve a convergence order in

$E^{\alpha}_{s,\boldsymbol{\gamma}}$ that can be arbitrarily close to the optimal rate $N^{-\alpha}$. Furthermore, the error bound can be made independent of the dimension if the weights $\boldsymbol{\gamma}$ satisfy suitable conditions. While the results in [5] are limited to product weights, we show their generalization to arbitrary positive weights here. To be more precise, several auxiliary results in our previous paper hold for general weights, and only some of the proofs there require the assumption of product weights. Here, we will mostly highlight those passages where there is a significant difference to what is discussed in [5]. For consistency, we shall use notation that is as similar as possible to that in [5], with only few minor adaptions.

A keystone in the paper [5] was the following proposition, which implies that in analyzing the worst-case error $e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ it is sufficient to consider only a truncated variant of the error expression, namely

$$T_{\alpha,\boldsymbol{\gamma}}(N,\boldsymbol{z}) := \sum_{\substack{\boldsymbol{m}\in M_{N,s}\setminus\{\boldsymbol{0}\} \\ \boldsymbol{m}\cdot\boldsymbol{z}\equiv 0 \pmod{N}}} \frac{1}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{m})}, \quad \text{where} \quad M_{N,s} := \{-(N-1),\dots,N-1\}^s. \tag{2}$$ eq:def_Ta

Indeed, we have

p:trunc_error **Proposition 1.** *[5, Proposition 1] Let $\boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ be positive weights and let $\boldsymbol{z} = (z_1,\dots,z_s) \in \mathbb{Z}^s$ with $\gcd(z_j,N) = 1$ for all $j \in \{1,\dots,s\}$. Then, for $\alpha > 1$, we have that*

$$e_{N,s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) - T_{\alpha,\boldsymbol{\gamma}}(N,\boldsymbol{z}) \le \frac{1}{N^\alpha} \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \, (4\zeta(\alpha))^{|\mathfrak{u}|},$$

where $\zeta(\cdot)$ denotes the Riemann zeta function.

Proof. We refer to [5] for a proof. □

Note that one can also define the quantity $T_{\alpha,\boldsymbol{\gamma}}(N,\boldsymbol{z})$ by replacing α by 1 in (2) and making the obvious adaptions. Regarding $T_{1,\boldsymbol{\gamma}}(N,\boldsymbol{z})$, the following estimate was shown in [5] for the case when N is a power of 2.

target_CBCDBD **Theorem 1.** *[5, Theorem 2] Let $N = 2^n$, with $n \ge 1$, and let $\boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ be positive weights. Furthermore, let $\boldsymbol{z} = (z_1,\dots,z_s) \in \{1,\dots,N-1\}^s$ with $\gcd(z_j,N) = 1$ for $1 \le j \le s$. Then,*

$$\begin{aligned} T_{1,\boldsymbol{\gamma}}(N,\boldsymbol{z}) \quad &\le \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \frac{\gamma_{\mathfrak{u}}}{N} (\log 4 + 2(1+\log N))^{|\mathfrak{u}|} - \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \, (\log 4)^{|\mathfrak{u}|} \\ &\quad + \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \frac{\gamma_{\mathfrak{u}}}{N} \, 2|\mathfrak{u}| \, (1+2\log N)^{|\mathfrak{u}|} \, (1+\log N) + \frac{1}{N} H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z}), \end{aligned}$$

where

$$H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z}) := \sum_{k=1}^{2^n-1} \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi k z_j/2^n)}\right).$$

Based on the quantity $H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z})$, an averaging argument was used in [5] to obtain a quality function which serves as the relevant criterion in the CBC-DBD algorithm. We state its definition here for completeness.

def:h_rv **Definition 1** (Digit-wise quality function)**.** Let $x \in \mathbb{N}$ be an odd integer, let $n, s \in \mathbb{N}$ be positive integers, and let $\boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ be positive weights. For $1 \le v \le n$ and $1 \le r \le s$, and odd integers $z_1,\dots,z_{r-1} \in \mathbb{Z}$, we define the quality function $h_{r,n,v,\boldsymbol{\gamma}} : \mathbb{Z} \to \mathbb{R}$ as

$$h_{r,n,v,\boldsymbol{\gamma}}(x) := \sum_{t=v}^{n} \frac{1}{2^{t-v}} \sum_{\substack{k=1 \\ k\equiv 1 \pmod{2}}}^{2^t-1} \left[\sum_{\emptyset\neq\mathfrak{u}\subseteq\{1:r-1\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi k z_j/2^t)}\right) \right.$$

$$+\sum_{\mathfrak{u}\subseteq\{1:r-1\}}\gamma_{\mathfrak{u}\cup\{r\}}\log\left(\frac{1}{\sin^2(\pi k x/2^v)}\right)\prod_{j\in\mathfrak{u}}\log\left(\frac{1}{\sin^2(\pi k z_j/2^t)}\right)\Bigg].$$

Note that while the quantity $h_{r,n,v,\boldsymbol{\gamma}}$ depends on the integers $z_1,\dots,z_{r-1}$, this dependency is not explicitly visible in our notation. Nevertheless, in the following these integers will always be the components of the generating vector which have been selected in the previous steps of our algorithm. Based on $h_{r,n,v,\boldsymbol{\gamma}}$ the component-by-component digit-by-digit (CBC-DBD) algorithm is formulated as follows in [5].

alg:cbc-dbd

Algorithm 1 Component-by-component digit-by-digit construction, [5, Algorithm 1]

Input: Integer $n\in\mathbb{N}$, dimension s, and positive weights $\boldsymbol{\gamma}=\{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$.

Set $z_{1,n}=1$ and $z_{2,1}=\cdots=z_{s,1}=1$.

for $r=2$ **to** s **do**
  for $v=2$ **to** n **do**
    $z^*=\underset{z\in\{0,1\}}{\operatorname{argmin}}\, h_{r,n,v,\boldsymbol{\gamma}}(z_{r,v-1}+z\,2^{v-1})$
    $z_{r,v}=z_{r,v-1}+z^*\,2^{v-1}$
  end for
  Set $z_r:=z_{r,n}$.
end for

Set $\boldsymbol{z}=(z_1,\dots,z_s)$.

Return: Generating vector $\boldsymbol{z}=(z_1,\dots,z_s)$ for $N=2^n$.

In the analysis of the worst-case error of rank-1 lattice rules generated by vectors $\boldsymbol{z}$ obtained from Algorithm 1,the following theorem was shown in [5].

m:H-induction

Theorem 2. *[5, Theorem 3] Let $n,s\in\mathbb{N}$, $N=2^n$, and let $\boldsymbol{\gamma}=\{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ be positive weights with $\gamma_\emptyset=1$. Furthermore, let the generating vector $\boldsymbol{z}\in\mathbb{Z}_N^s$ be constructed by Algorithm 1. Then,*

$$H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z})\le H_{s-1,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s-1\}})+\log 4\left[\gamma_{\{s\}}N+H_{s-1,n,\boldsymbol{\gamma}\cup\{s\}}(\boldsymbol{z}_{\{1:s-1\}})\right], \qquad (3)$$ eq:estima

where $\boldsymbol{\gamma}\cup\mathfrak{v}$ is the collection of weights $\{\gamma_{\mathfrak{u}\cup\mathfrak{v}}\}_{\mathfrak{u}\subset\mathbb{N}}$ for a finite $\mathfrak{v}\subset\mathbb{N}$.

Remark 2. Note that, with the notation introduced in Theorem 2, we have $(\boldsymbol{\gamma}\cup\mathfrak{v})_{\mathfrak{u}}=\gamma_{\mathfrak{u}\cup\mathfrak{v}}$ for finite $\mathfrak{u},\mathfrak{v}\subset\mathbb{N}$.

Moreover, the following estimate was derived for the special case of product weights in [5]. This is the point from which onwards the analysis in [5] is only done for the case of product weights.

upper_bound_H

Theorem 3. *[5, Theorem 4] Let $n,s\in\mathbb{N}$, $N=2^n$, and let $\boldsymbol{\gamma}=\{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$, with $\gamma_{\mathfrak{u}}=\prod_{j\in\mathfrak{u}}\gamma_j$ be positive product weights. Furthermore, let the generating vector $\boldsymbol{z}\in\mathbb{Z}_N^s$ be constructed by Algorithm 1. Then for $H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z})$ the following upper bound holds,*

$$H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z})\le N\left[-1+\prod_{j=1}^{s}(1+\gamma_j\log 4)\right].$$

We will now show a generalization of Theorem 3 to general positive weights $\boldsymbol{\gamma}=\{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}\subset\mathbb{N}}$ without having to make any further assumptions on their structure. Indeed, we have the following theorem.

r_bound_H_gen **Theorem 4.** *Let $n, s \in \mathbb{N}$, $N = 2^n$, and let $\boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u} \subset \mathbb{N}}$ be positive weights. Furthermore, let the generating vector $\boldsymbol{z} \in \mathbb{Z}_N^s$ be constructed by Algorithm 1. Then for $H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z})$ the following upper bound holds:*

$$H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z}) \le N \sum_{\emptyset \neq \mathfrak{v} \subseteq \{1:s\}} (\log 4)^{|\mathfrak{v}|} \gamma_{\mathfrak{v}}.$$

Proof. Due to the formulation of Algorithm 1, it is such that it is extensible in the dimension, and for every $r \in \{2, \ldots, s\}$ we have, similarly to Theorem 2,

$$H_{r,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:r\}}) \le H_{r-1,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:r-1\}}) + (\log 4)[\gamma_{\{r\}} N + H_{r-1,n,\boldsymbol{\gamma} \cup \{r\}}(\boldsymbol{z}_{\{1:r-1\}})]. \tag{4}$$ eq:recurs

Thus, we obtain

$$\begin{aligned}
H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s\}}) &\le H_{s-1,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s-1\}}) + (\log 4)[\gamma_{\{s\}} N + H_{s-1,n,\boldsymbol{\gamma} \cup \{s\}}(\boldsymbol{z}_{\{1:s-1\}})] \\
&\le H_{s-2,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s-2\}}) + (\log 4)[\gamma_{\{s-1\}} N + H_{s-2,n,\boldsymbol{\gamma} \cup \{s-1\}}(\boldsymbol{z}_{\{1:s-2\}})] \\
&\quad + (\log 4)[\gamma_{\{s\}} N + H_{s-1,n,\boldsymbol{\gamma} \cup \{s\}}(\boldsymbol{z}_{\{1:s-1\}})] \\
&\le H_{s-2,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s-2\}}) + (\log 4)\gamma_{\{s-1\}} N + (\log 4)\gamma_{\{s\}} N \\
&\quad + (\log 4) H_{s-2,n,\boldsymbol{\gamma} \cup \{s-1\}}(\boldsymbol{z}_{\{1:s-2\}}) \\
&\quad + (\log 4)\left[H_{s-2,n,\boldsymbol{\gamma} \cup \{s\}}(\boldsymbol{z}_{\{1:s-2\}}) + (\log 4)\gamma_{\{s-1,s\}} N \right. \\
&\quad \left. + (\log 4) H_{s-2,n,\boldsymbol{\gamma} \cup \{s-1,s\}}(\boldsymbol{z}_{\{1:s-2\}})\right] \\
&= H_{s-2,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s-2\}}) \\
&\quad + (\log 4) H_{s-2,n,\boldsymbol{\gamma} \cup \{s-1\}}(\boldsymbol{z}_{\{1:s-2\}}) + (\log 4) H_{s-2,n,\boldsymbol{\gamma} \cup \{s\}}(\boldsymbol{z}_{\{1:s-2\}}) \\
&\quad + (\log 4)^2 H_{s-2,n,\boldsymbol{\gamma} \cup \{s-1,s\}}(\boldsymbol{z}_{\{1:s-2\}}) \\
&\quad + (\log 4)\gamma_{\{s-1\}} N + (\log 4)\gamma_{\{s\}} N + (\log 4)^2 \gamma_{\{s-1,s\}} N.
\end{aligned}$$

Consequently, we have shown

$$H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s\}}) \le \sum_{\mathfrak{v} \subseteq \{s-1:s\}} (\log 4)^{|\mathfrak{v}|} H_{s-2,n,\boldsymbol{\gamma} \cup \mathfrak{v}}(\boldsymbol{z}_{\{1:s-2\}}) + N \sum_{\emptyset \neq \mathfrak{v} \subseteq \{s-1:s\}} (\log 4)^{|\mathfrak{v}|} \gamma_{\mathfrak{v}}.$$

This yields, by using (4) again,

$$\begin{aligned}
H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s\}}) &\le \sum_{\mathfrak{v} \subseteq \{s-1:s\}} (\log 4)^{|\mathfrak{v}|} \left[H_{s-3,n,\boldsymbol{\gamma} \cup \mathfrak{v}}(\boldsymbol{z}_{\{1:s-3\}}) + (\log 4)(\boldsymbol{\gamma} \cup \mathfrak{v})_{\{s-2\}} N \right. \\
&\quad \left. + (\log 4) H_{s-3,n,\boldsymbol{\gamma} \cup \mathfrak{v} \cup \{s-2\}}(\boldsymbol{z}_{\{1:s-3\}})\right] + N \sum_{\emptyset \neq \mathfrak{v} \subseteq \{s-1:s\}} (\log 4)^{|\mathfrak{v}|} \gamma_{\mathfrak{v}} \\
&= \sum_{\mathfrak{v} \subseteq \{s-2:s\}} (\log 4)^{|\mathfrak{v}|} H_{s-3,n,\boldsymbol{\gamma} \cup \mathfrak{v}}(\boldsymbol{z}_{\{1:s-3\}}) + N \sum_{\emptyset \neq \mathfrak{v} \subseteq \{s-2:s\}} (\log 4)^{|\mathfrak{v}|} \gamma_{\mathfrak{v}}.
\end{aligned}$$

We can use this principle recursively, until we arrive at

$$H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s\}}) \le \sum_{\mathfrak{v} \subseteq \{2:s\}} (\log 4)^{|\mathfrak{v}|} H_{1,n,\boldsymbol{\gamma} \cup \mathfrak{v}}(\boldsymbol{z}_{\{1\}}) + N \sum_{\emptyset \neq \mathfrak{v} \subseteq \{2:s\}} (\log 4)^{|\mathfrak{v}|} \gamma_{\mathfrak{v}}.$$

Furthermore, it can be shown in complete analogy to [5, Proof of Theorem 4] that

$$H_{1,n,\boldsymbol{\gamma} \cup \mathfrak{v}}(\boldsymbol{z}_{\{1\}}) = (\log 4)\gamma_{\{1\} \cup \mathfrak{v}}(N - n - 1) \le (\log 4)\gamma_{\{1\} \cup \mathfrak{v}} N,$$

which yields

$$H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z}_{\{1:s\}}) \quad \le \quad N \sum_{\mathfrak{v} \subseteq \{2:s\}} (\log 4)^{|\mathfrak{v}|+1} \gamma_{\{1\} \cup \mathfrak{v}} + N \sum_{\emptyset \neq \mathfrak{v} \subseteq \{2:s\}} (\log 4)^{|\mathfrak{v}|} \gamma_{\mathfrak{v}}$$

$$= \quad N \sum_{\emptyset \neq \mathfrak{v} \subseteq \{1:s\}} (\log 4)^{|\mathfrak{v}|} \gamma_{\mathfrak{v}},$$

as claimed. $\square$

We can now use the general estimate in Theorem 4 to also show a more general version of [5, Theorem 5].

in_result_gen **Theorem 5.** *Let $N = 2^n$, with $n \in \mathbb{N}$, and let $\boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u} \subset \mathbb{N}}$ be general positive weights. Furthermore, denote by $\boldsymbol{z} = (z_1, \ldots, z_s)$ the corresponding generating vector constructed by Algorithm 1. Then there exists a constant $a > 0$, which is independent of s and N, such that*

$$T_{1,\boldsymbol{\gamma}}(N, \boldsymbol{z}) \leq (1 + \log N) \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \frac{\gamma_{\mathfrak{u}}}{N} (a \log N)^{|\mathfrak{u}|}. \tag{5}$$ eq:optcoe

Moreover, if the weights satisfy

$$\sum_{j \geq 1} \tilde{\gamma}_j < \infty, \quad \textit{where} \quad \tilde{\gamma}_j := \max_{\mathfrak{v} \subseteq \{1:j-1\}} \frac{\gamma_{\mathfrak{v} \cup \{j\}}}{\gamma_{\mathfrak{v}}},$$

then, for any $\delta > 0$ there exists a constant $C_\delta > 0$, which is again independent of s and N, such that

$$T_{1,\boldsymbol{\gamma}}(N, \boldsymbol{z}) \leq \frac{C_\delta}{N^{1-\delta}}.$$

Proof. We use the bound in Theorem 1 combined with the bound on $H_{s,n,\boldsymbol{\gamma}}(\boldsymbol{z})$ in Theorem 4 to obtain

$$\begin{aligned}
T_{1,\boldsymbol{\gamma}}(N, \boldsymbol{z}) \quad &\leq \quad \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \frac{\gamma_{\mathfrak{u}}}{N} (\log 4 + 2(1 + \log N))^{|\mathfrak{u}|} \\
&\quad + \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \frac{\gamma_{\mathfrak{u}}}{N} \, 2|\mathfrak{u}| \, (1 + 2 \log N)^{|\mathfrak{u}|} \, (1 + \log N) \\
&\leq \quad \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \frac{\gamma_{\mathfrak{u}}}{N} (\log 4 + 2(1 + \log N))^{|\mathfrak{u}|} \\
&\quad + \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \frac{\gamma_{\mathfrak{u}}}{N} \, 2^{|\mathfrak{u}|+1} \, (1 + 2 \log N)^{|\mathfrak{u}|} \, (1 + \log N),
\end{aligned}$$

where we used that $|\mathfrak{u}| \leq 2^{|\mathfrak{u}|}$ for non-empty $\mathfrak{u}$. Thus, there exists a constant $a > 0$ such that

$$T_{1,\boldsymbol{\gamma}}(N, \boldsymbol{z}) \leq (1 + \log N) \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \frac{\gamma_{\mathfrak{u}}}{N} (a \log N)^{|\mathfrak{u}|},$$

which is the bound in (5). The second claim in the theorem follows immediately by using [5, Lemma 4]. $\square$

Next, we use Theorem 5 to show a more general version of [5, Corollary 1]. We formulate this result for weights denoted by $\boldsymbol{\eta}$, and then describe two special cases in a remark below.

esult-dbd_gen **Corollary 1.** *Let $N = 2^n$, with $n \in \mathbb{N}$, and let $\boldsymbol{\eta} = \{\eta_{\mathfrak{u}}\}_{\mathfrak{u} \subset \mathbb{N}}$ be general positive weights, satisfying*

$$\sum_{j \geq 1} \tilde{\eta}_j < \infty, \quad \textit{where} \quad \tilde{\eta}_j := \max_{\mathfrak{v} \subseteq \{1:j-1\}} \frac{\eta_{\mathfrak{v} \cup \{j\}}}{\eta_{\mathfrak{v}}}. \tag{6}$$ eq:summab

Denote by $\boldsymbol{z} = (z_1, \ldots, z_s)$ the generating vector constructed by Algorithm 1 for the weights $\boldsymbol{\eta}$. Then, for any $\delta > 0$ and each $\alpha > 1$, the worst-case error $e_{N,s,\alpha,\boldsymbol{\eta}^\alpha}(\boldsymbol{z})$ in $E^\alpha_{s,\boldsymbol{\eta}^\alpha}$ satisfies

$$e_{N,s,\alpha,\boldsymbol{\eta}^\alpha}(\boldsymbol{z}) \leq \frac{C}{N^\alpha} + \frac{C_\delta^\alpha}{N^{\alpha(1-\delta)}}$$

with constants $C_\alpha, C_\delta > 0$, where C_α may depend on α, and C_δ may depend on δ, but C_α and C_δ are independent of s and N.

Proof. Since $N = 2^n$ and by the formulation of Algorithm 1 all components of $\boldsymbol{z}$ are odd, we have in particular that $\gcd(z_j, N) = 1$ for all $j \in \{1, \ldots, s\}$. Therefore, by Proposition 1, the worst-case error $e_{N,s,\alpha,\boldsymbol{\eta}^\alpha}(\boldsymbol{z})$ satisfies

$$e_{N,s,\alpha,\boldsymbol{\eta}^\alpha}(\boldsymbol{z}) \leq \frac{1}{N^\alpha} \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \eta_{\mathfrak{u}}^\alpha \left(4\zeta(\alpha)\right)^{|\mathfrak{u}|} + T_{\alpha,\boldsymbol{\eta}^\alpha}(N, \boldsymbol{z}).$$

Next, we use that for $\alpha \geq 1$ we have $\sum_i x_i^\alpha \leq (\sum_i x_i)^\alpha$ for $x_i \geq 0$, and we also use Theorem 5. This yields that for any $\delta > 0$ there exists a constant $C_\delta > 0$, which is independent of s and N, such that

$$T_{\alpha,\boldsymbol{\eta}^\alpha}(N, \boldsymbol{z}) = \sum_{\substack{\boldsymbol{0} \neq \boldsymbol{m} \in M_{N,s} \\ \boldsymbol{m}\cdot\boldsymbol{z} \equiv 0 \pmod{N}}} \frac{1}{r_{\alpha,\boldsymbol{\eta}^\alpha}(\boldsymbol{m})} \leq \left(\sum_{\substack{\boldsymbol{0} \neq \boldsymbol{m} \in M_{N,s} \\ \boldsymbol{m}\cdot\boldsymbol{z} \equiv 0 \pmod{N}}} \frac{1}{r_{1,\boldsymbol{\eta}}(\boldsymbol{m})} \right)^\alpha = (T_{1,\boldsymbol{\eta}}(N, \boldsymbol{z}))^\alpha \leq \frac{C_\delta^\alpha}{N^{\alpha(1-\delta)}}.$$

Using Assumption (6) and standard arguments (or a similar reasoning as in the proof of [5, Lemma 4]), we also see that

$$\sum_{\emptyset \neq \mathfrak{u} \subseteq \{1:s\}} \eta_{\mathfrak{u}} \left(4\zeta(\alpha)\right)^{|\mathfrak{u}|} < C_\alpha$$

for some constant $C_\alpha > 0$. This yields the result. □

Remark 3. If we choose $\boldsymbol{\eta} = \boldsymbol{\gamma} = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u} \subset \mathbb{N}}$ for positive weights $\boldsymbol{\gamma}$ in Corollary 1, then the algorithm can be run independently of α, and we get an error bound of almost optimal convergence order for the space $E^\alpha_{s,\boldsymbol{\gamma}^\alpha}$ for any $\alpha > 1$. In this sense, Algorithm 1 can be said to be universal with respect to α.

If, on the other hand, we would like to have an error bound of almost optimal convergence order for the space $E^\alpha_{s,\boldsymbol{\gamma}}$, then we need to choose $\boldsymbol{\eta} = \boldsymbol{\gamma}^{1/\alpha} = \{\gamma_{\mathfrak{u}}^{1/\alpha}\}_{\mathfrak{u} \subset \mathbb{N}}$ in Corollary 1, and also run Algorithm 1 for $\boldsymbol{\gamma}^{1/\alpha}$, i.e., in this case Algorithm 1 is not universal.

Acknowledgments

The author is supported by the Austrian Science Fund, Project F5506, which is part of the Special Research Program "Quasi-Monte Carlo Methods: Theory and Applications", and Project P34808. For the purpose of open access, the author has applied a CC BY public copyright licence to any author accepted manuscript version arising from this submission.

References

[1] J. Dick. On the convergence rate of the component-by-component construction of good lattice rules. J. Complexity 20, 493–522, 2004.

[2] J. Dick, P. Kritzer, F. Pillichshammer. *Lattice Rules.* Springer, Cham, 2022.

[3] J. Dick, F.Y. Kuo, I.H. Sloan. High-dimensional integration—the quasi-Monte Carlo way. Acta Numer. 22, 133–288, 2013.

[4] J. Dick, F. Pillichshammer. *Digital Nets and Sequences.* Cambridge University Press, Cambridge, 2010.

[5] A. Ebert, P. Kritzer, O. Osisiogu, D. Nuyens. Digit-by-digit and component-by-component constructions of lattice rules for periodic functions with unknown smoothness. J. Complexity 66, 101555, 2021.

[6] F.J. Hickernell, H. Niederreiter. The existence of good extensible rank-1 lattice. J. Complexity, 19, 286-300, 2003.

[7] N.M. Korobov. *Number-theoretic methods in approximate analysis.* Goz. Izdat. Fiz.-Math., 1963. In Russian.

[8] 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, 301–320, 2003.

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

[10] E. Novak, H. Woźniakowski. Tractability of Multivariate Problems. Volume I: Linear Information. EMS, Zurich, 2008.

[11] D. Nuyens, R. Cools. Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces. Math. Comp. 75, 903–920, 2006.

[12] D. Nuyens, R. Cools. Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points. J. Complexity 22, 4–28, 2006.

[13] I.H. Sloan, S. Joe. *Lattice Methods for Multiple Integration.* Clarendon Press, Oxford, 1994.

[14] I.H. Sloan, V.A. Reztsov. Component-by-component construction of good lattice rules. Math. Comp. 71, 263–273, 2002.

[15] I.H. Sloan, H. Woźniakowski. When are quasi Monte Carlo algorithms efficient for high-dimensional problems? J. Complexity 14, 1–33, 1998.

Author's address:

Peter Kritzer
Johann Radon Institute for Computational and Applied Mathematics (RICAM)
Austrian Academy of Sciences
Altenbergerstr. 69, 4040 Linz, Austria.
`peter.kritzer@oeaw.ac.at`