sciendo

uniform distribution theory

DOI: 10.2478/UDT-2023-0007
Unif. Distrib. Theory **18** (2023), no.1, 97–140

ON A REDUCED COMPONENT-BY-COMPONENT DIGIT-BY-DIGIT CONSTRUCTION OF LATTICE POINT SETS

Peter Kritzer[1] — Onyekachi Osisiogu[1,2]

[1]RICAM, Austrian Academy of Sciences, AUSTRIA

[2]Illinois Institute of Technology, USA

ABSTRACT. In this paper, we study an efficient algorithm for constructing point sets underlying quasi-Monte Carlo integration rules for weighted Korobov classes. The algorithm presented is a reduced fast component-by-component digit-by-digit (CBC-DBD) algorithm, which is useful for situations where the weights in the function space show a sufficiently fast decay. The advantage of the algorithm presented here is that the computational effort can be independent of the dimension of the integration problem to be treated if suitable assumptions on the integrand are met. By considering a reduced digit-by-digit construction, we allow an integration algorithm to be less precise with respect to the number of bits in those components of the problem that are considered less important. The new reduced CBC-DBD algorithm is designed to work for the construction of lattice point sets, and the corresponding integration rules (so-called lattice rules) can be used to treat functions in different kinds of function spaces. We show that the integration rules constructed by our algorithm satisfy error bounds of almost optimal convergence order. Furthermore, we give details on an efficient implementation such that we obtain a considerable speed-up of a previously known CBC-DBD algorithm that has been studied in the paper *Digit-by-digit and*

2020 Mathematics Subject Classification: 65D30, 65D32, 41A55, 41A63.
Keywords: Numerical integration; lattice points; quasi-Monte Carlo methods; component-by-component construction; digit-by-digit construction; weighted function spaces; fast implementation.
Supported by the Austrian Science Fund, Project F5506, which is part of the Special Research Program "Quasi-Monte Carlo Methods: Theory and Applications" (Kritzer and Osisiogu), and Project P34808 (Kritzer).

component-by-component constructions of lattice rules for periodic functions with unknown smoothness by Ebert, Kritzer, Nuyens, and Osisiogu, published in the Journal of Complexity in 2021. This improvement is illustrated by numerical results.

Communicated by Josef Dick

1. Introduction

In this paper we study numerical integration of functions defined over the d-dimensional unit cube. For an integrable function $f : [0,1)^d \to \mathbb{R}$, we denote the integral of f by

$$I_d(f) := \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},$$

and we study the efficient construction of high-dimensional *quadrature rules* of the form

$$Q_N(f,(\boldsymbol{x}_k)_{k=0}^{N-1}) := \frac{1}{N}\sum_{k=0}^{N-1} f(\boldsymbol{x}_k)$$

for numerically approximating I_d. We assume that the integrand f lies in a Banach space $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$, and that the integration nodes $\boldsymbol{x}_0,\dots,\boldsymbol{x}_{N-1} \in [0,1)^d$ are to be chosen deterministically. Here Q_N is called a *quasi-Monte Carlo (QMC)* rule, which is a special case of an equal-weight integration rule. One way to measure the quality of a QMC integration rule Q_N is to consider the *worst-case error* in the unit ball of the space $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$, i.e.,

$$e_{N,d}(Q_N,\mathcal{F}) := \sup_{\substack{f\in\mathcal{F}\\ \|f\|_{\mathcal{F}}\le 1}} \left| I_d(f) - Q_N(f,(\boldsymbol{x}_k)_{k=0}^{N-1}) \right|.$$

In general, it is highly non-trivial to choose the set of integration nodes such that the resulting rule has a low worst-case error. There are two main types of QMC point sets $\mathcal{P} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_{N-1}\}$, namely *lattice point sets* and *digital nets.* In this paper we consider lattice point sets, which are obtained by considering discrete subsets of $\mathbb{R}^d$ that are closed under both addition and subtraction and contain $\mathbb{Z}^d$ as a subset; by intersecting such a discrete set with the d-dimensional unit cube $[0,1)^d$, we then obtain a lattice point set. We are interested in a special kind of lattice point sets that is essentially based on one *generating vector* $\boldsymbol{z}$. These lattice point sets were first introduced by Korobov [14] and independently by Hlawka [12] and are among the most prominently studied QMC point sets to approximate multivariate integrals (see standard textbooks such as [5,13,20,23], and also [6]). For a natural number $N \in \mathbb{N}$ and an integer generating vector $\boldsymbol{z} \in \{1, 2, \dots, N-1\}^d$, a point set with points of the form

$$\boldsymbol{x}_k := \left\{ \frac{k}{N} \boldsymbol{z} \right\} = \left(\left\{ \frac{k z_1}{N} \right\}, \dots, \left\{ \frac{k z_d}{N} \right\} \right) \in [0,1)^d, \qquad \text{for } k = 0, 1, \dots, N-1,$$

is called a *rank-1 lattice point set*, which we shall sometimes denote by $P(\boldsymbol{z}, N)$. For vectors $\boldsymbol{x} \in \mathbb{R}^d$ we apply $\{\cdot\}$ component-wise, where $\{x\} = x - \lfloor x \rfloor$ denotes the fractional part of x. A QMC rule using such a point set as integration nodes is called a *(rank-1) lattice rule* with generating vector $\boldsymbol{z}$. We remark that, given N and d, a rank-1 lattice rule is completely determined by the choice of the generating vector

$$\boldsymbol{z} = (z_1, \dots, z_d) \in \mathbb{Z}_N^d, \quad \text{where we write} \quad \mathbb{Z}_N := \{0, \dots, N-1\}.$$

However, it should be obvious that not every choice of a generating vector $\boldsymbol{z}$ also yields a rank-1 lattice rule with good quality for approximating the integral, and it is usually necessary to tailor the choice of the integration nodes to the function space $\mathcal{F}$ under consideration. For lattice rules we consider certain Banach spaces which are based on assuming sufficient decay of the Fourier coefficients of their elements to guarantee certain smoothness properties. These spaces are called *weighted Korobov classes*, which we will denote by $E^{\alpha}_{d,\boldsymbol{\gamma}}$, where d denotes the number of variables the functions depend on, $\alpha > 1$ is a real number frequently referred to as the smoothness parameter, and $\boldsymbol{\gamma} = (\gamma_j)_{j \geq 1}$ is a sequence of strictly positive reals, which model the importance of different coordinates. The idea of these additional parameters γ_j, which we refer to as weights in the definition of the function spaces under consideration, goes back to Sloan and Woźniakowski [24], see also [10, 25], and this will be made more precise by incorporating the weights in the norm of the space $E^{\alpha}_{d,\boldsymbol{\gamma}}$ in Section 2. If we denote the variables in the integration problem by $\boldsymbol{x} = (x_1, \dots, x_d)$, a small value of γ_j corresponds to a low influence of the variable x_j, while a large γ_j means high influence of x_j. Regarding the role of these weights in integration problems, we favor a situation in which the weights decay sufficiently fast for coordinates with increasing indices, which helps in vanquishing the curse of dimensionality that is inherent to many high-dimensional problems. Indeed, under certain summability conditions on the weights, it is even possible to obtain bounds on the integration error that do not depend on the dimension of the problem at all. Such a situation is called *strong polynomial tractability*, see, e.g., [21], for a general reference.

Regarding the construction of generating vectors, for dimensions $d \leq 2$ explicit constructions of good generating vectors are available, see, e.g., [20, 23], but there are no explicit constructions of good generating vectors known for $d > 2$;

how to find a generating vector $\boldsymbol{z}$ that guarantees a low worst-case error of integration in a given function space is a crucial question regarding rank-1 lattice rules. One way to find good generating vectors is the component-by-component (CBC) construction, which is based on a greedy algorithm choosing one component of the generating vector at a time. It was shown in [19] for prime N (see also [2] for the case of composite N) that it is possible to find generating vectors yielding essentially optimal results for certain spaces of d-variate functions by the CBC construction. Furthermore, it was shown in [22] that a fast CBC construction can reduce the computational cost of these algorithms to only $\mathcal{O}(dN \log N)$, by making use of a clever ordering of the points of a rank-1 lattice, and the Fast Fourier Transform.

In the paper [4] the weights $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ in a given function space were incorporated in the definition of the search spaces in the CBC construction of lattice rules. To be more precise, depending on the weight γ_j, $j \in \mathbb{N}$, the search space for the jth component z_j of the generating vector $\boldsymbol{z}$ can be shrinked as compared to the usual CBC construction, and this reduction is the motivation for the modified CBC algorithm to be called the "reduced" CBC construction. Under suitable assumptions on the integrands considered, it was shown in [4] that the lattice point sets obtained by the reduced CBC construction can still yield excellent results when used as integration nodes in a QMC rule, while the construction cost can be made independent of the dimension d of the integration problem. It was also shown in [4] that the fast CBC construction principle of Nuyens and Cools, which was mentioned above, can be adapted to the reduced CBC construction.

A different construction algorithm for generating vectors of good lattice rules inspired by articles of Korobov (see [15] and [16], and [17] for an English translation) was dealt with in the recent paper [8], where a so-called component-by--component digit-by-digit (CBC-DBD) algorithm was presented. In this approach, one constructs the generating vector $\boldsymbol{z}$ in a component-by-component (CBC) fashion, in which each component z_j is assembled digit-by-digit (DBD). That is, for a number $N = 2^m$ of points we greedily construct the components z_j bit-by-bit starting from the least significant bit. It was shown that the rules constructed by the CBC-DBD algorithm yield a convergence rate that is arbitrarily close to the optimal rate. It is also possible to have a fast implementation of the CBC-DBD algorithm which has a computational cost of $\mathcal{O}(dN \log N)$, and also numerical results on the performance of the CBC-DBD algorithm are presented in [8]. We remark that the error analysis for the CBC-DBD algorithm is such that no prior knowledge of the smoothness parameter α is required to construct the generating vector. The resulting generating vector will still deliver the near optimal rate of convergence, for arbitrary smoothness parameters $\alpha > 1$, and this result can be stated independently of the dimension, assuming that the weights satisfy certain summability conditions.

In the present paper, we would like to study a combination of the reduced CBC algorithm from [4] and the CBC-DBD construction algorithm approach from [8], and present a *reduced (fast)* CBC-DBD *algorithm* for generating vectors of good lattice rules with a large number of lattice points in high dimensions, assuming sufficiently fast decaying weights. This new algorithm will work by constructing the generating vector $\boldsymbol{z}$ in a component-by-component (CBC) fashion in which each component z_j is assembled digit-by-digit (DBD). However, the search space for each component will be reduced in comparison to what is shown in [8], thus speeding up the construction method. Our new method is a natural extension of the aforementioned papers, as it corresponds to allowing less digit precision in representing those components of the integration rule that are modeled as less important than others. Hence, the combination of the reduced CBC construction and the CBC-DBD is intuitively obvious. We will show that for suitable choices of weights the construction cost of the reduced fast CBC-DBD algorithm can be made independent of the dimension. The main aim of this paper is to show that one can reduce the construction cost of the lattice rule by making the search space for later components smaller than the earlier ones, while still achieving (strong) polynomial tractability. We present our method for the case of *product weights*, and also the non-reduced CBC-DBD construction in [8] was originally presented for this case, due to technical limitations. In the mean time, however, it has been shown in the paper [18] that the non-reduced CBC-DBD construction can also be used for more general classes of weights, including POD weights. We assume that a similar generalization of the reduced CBC-DBD construction would be technically challenging, and leave this problem open for future research.

The structure of this paper is as follows. In the subsequent section (Section 2), we outline the general setting, and then our construction algorithm as well as the main result are stated and proved in Section 3. Section 4 contains remarks on how to efficiently implement our newly found algorithm and some numerical results.

Regarding notation, let $\mathbb{N} := \{1, 2, \ldots\}$ be the set of natural numbers, let $\mathbb{N}_0 := \{0, 1, 2, \ldots\}$, and let $\mathbb{Z}$ be the set of integers. Additionally, let the set of non-zero integers be denoted by $\mathbb{Z}_* := \mathbb{Z} \setminus \{0\}$. For sets of components we use fraktur font, e.g., $\mathfrak{u} \subset \mathbb{N}$. To denote the projection of a vector $\boldsymbol{x} \in [0,1)^d$ or $\boldsymbol{\ell} \in \mathbb{Z}^d$ to the components in a set $\mathfrak{u} \subseteq [d] := \{1, \ldots, d\}$, we write

$$\boldsymbol{x}_{\mathfrak{u}} := (x_j)_{j \in \mathfrak{u}} \quad \text{or} \quad \boldsymbol{\ell}_{\mathfrak{u}} := (\ell_j)_{j \in \mathfrak{u}},$$

respectively.

2. Setting and overview

We consider one-periodic real-valued L_2 functions defined on $[0,1]^d$ with absolutely convergent Fourier series

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

where $\hat{f}(\boldsymbol{\ell})$ are the Fourier coefficients of f and

$$\boldsymbol{\ell}\cdot\boldsymbol{x} := \sum_{j=1}^{d} \ell_j x_j = \ell_1 x_1 + \cdots + \ell_d x_d$$

is the vector dot product.

It is known that rank-1 lattice point sets, as introduced in Section 1, have a property commonly referred to as the *character property*. Indeed, for a rank-1 lattice point set with generating vector $\boldsymbol{z}\in\mathbb{Z}^d$, it is true that

$$\frac{1}{N}\sum_{k=0}^{N-1} \mathrm{e}^{2\pi\mathrm{i}(\boldsymbol{\ell}\cdot\boldsymbol{z})\,k/N} = \begin{cases} 1 & \text{if } \boldsymbol{\ell}\cdot\boldsymbol{z}\equiv 0 \pmod{N}, \\ 0 & \text{otherwise.}\end{cases}$$

We introduce the dual of a rank-1 lattice, which is of great importance in representing the error of approximating an integral by a lattice rule Q_N.

Definition 2.1 (Dual lattice). Let $N\geq 2$ be the number of points and let $\boldsymbol{z}\in\{1,\ldots,N-1\}^d$ be the generating vector of a rank-1 lattice point set $P(\boldsymbol{z},N)$. The dual lattice $\Lambda^{\top}(\boldsymbol{z},N)$ of a rank-1 lattice $P(\boldsymbol{z},N)$ is given by

$$\mathcal{D} = \Lambda^{\top}(\boldsymbol{z},N) := \{\boldsymbol{\ell}\in\mathbb{Z}^d \mid \boldsymbol{\ell}\cdot\boldsymbol{z}\equiv 0 \ (\mathrm{mod}\ \ N)\}.$$

For a non-empty set $\mathfrak{u}\subseteq[d]$, we can further define

$$\mathcal{D}_{\mathfrak{u}} = \mathcal{D}_{\mathfrak{u}}(\boldsymbol{z}_{\mathfrak{u}}) := \{\boldsymbol{\ell}_{\mathfrak{u}}\in\mathbb{Z}_*^{|\mathfrak{u}|} \mid \boldsymbol{\ell}_{\mathfrak{u}}\cdot\boldsymbol{z}_{\mathfrak{u}}\equiv 0 \ (\mathrm{mod}\ \ N)\}.$$

Then, to obtain the integration error for a given function f in terms of its Fourier coefficients, we interchange the order of summation and use the character property of the lattice points, and get

$$\begin{aligned} Q_N(f,\boldsymbol{z}) - I_d(f) &= \sum_{\boldsymbol{0}\neq\boldsymbol{\ell}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{\ell}) \left[\frac{1}{N}\sum_{k=0}^{N-1}\mathrm{e}^{2\pi\mathrm{i}(\boldsymbol{\ell}\cdot\boldsymbol{z})\,k/N}\right] \\ &= \sum_{\substack{\boldsymbol{0}\neq\boldsymbol{\ell}\in\mathbb{Z}^d \\ \boldsymbol{\ell}\cdot\boldsymbol{z}\equiv 0 \ (\mathrm{mod}\ N)}} \hat{f}(\boldsymbol{\ell}) = \sum_{\boldsymbol{0}\neq\boldsymbol{\ell}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{\ell})\,\delta_N(\boldsymbol{\ell}\cdot\boldsymbol{z}), \end{aligned}$$

where, for $a \in \mathbb{Z}$,

$$\delta_N(a) := \frac{1}{N}\sum_{k=0}^{N-1} \mathrm{e}^{2\pi \mathrm{i} ak/N} = \begin{cases} 1 & \text{if } a \equiv 0 \pmod{N}, \\ 0 & \text{otherwise.} \end{cases}$$

For a vector $\boldsymbol{\ell} \in \mathbb{Z}^d$, smoothness parameter $\alpha > 1$, and strictly positive weights $\boldsymbol{\gamma} = (\gamma_j)_{j \geq 1}$, we define

$$r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{\ell}) := \prod_{j \in \operatorname{supp}(\boldsymbol{\ell})} \gamma_j \, |\ell_j|^\alpha , \tag{1}$$

where $\operatorname{supp}(\boldsymbol{\ell}) := \{j \in [d] : \ell_j \neq 0\}$ is the support of $\boldsymbol{\ell}$. As usual, we define the empty product as one such that $r_{\alpha,\boldsymbol{\gamma}}(\mathbf{0}) = 1$. Formally, we will also use a function $r_{1,\boldsymbol{\gamma}}$ below, which is obtained by replacing α by 1 in (1).

We then define our function space $E^\alpha_{d,\boldsymbol{\gamma}}$ of one-periodic real-valued L_2 functions defined on $[0,1]^d$ with absolutely convergent Fourier series. The norm in this space is given as

$$\|f\|_{E^\alpha_{d,\boldsymbol{\gamma}}} := \sup_{\boldsymbol{\ell} \in \mathbb{Z}^d} |\hat{f}(\boldsymbol{\ell})| \, r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{\ell}),$$

and for $\alpha > 1$, dimension $d \in \mathbb{N}$, and positive weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j \geq 1}$, our weighted function space is

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

It is known (see, for example, [5]) that the worst-case error of an N-point rank-1 lattice rule generated by $\boldsymbol{z} \in \mathbb{Z}^d$ in the space $E^\alpha_{d,\boldsymbol{\gamma}}$ is given by

$$e_{N,d,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) = \sum_{\mathbf{0} \neq \boldsymbol{\ell} \in \mathcal{D}} r^{-1}_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{\ell}) = \sum_{\mathbf{0} \neq \boldsymbol{\ell} \in \mathbb{Z}^d} \frac{\delta_N(\boldsymbol{\ell} \cdot \boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{\ell})}. \tag{3}$$

We remark that the worst-case error (3) is sometimes referred to as P_α in the literature on lattice rules. As pointed out above, it is the main goal of the present paper to state a new effective construction method for good instances of $\boldsymbol{z}$ such that $e_{N,d,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$ is reasonably small.

It is well known, see, e.g., [23], that the optimal convergence rate of the worst-case integration error in the space $E^\alpha_{d,\boldsymbol{\gamma}}$ is of order $\mathcal{O}(N^{-\alpha})$. Furthermore, we would like to remark that in the literature on lattice rules often a slightly different, but related, function space than $E^\alpha_{d,\boldsymbol{\gamma}}$ is studied. Indeed, by modifying the norm in the space to $\left(\sum_{\boldsymbol{\ell} \in \mathbb{Z}^d} |\hat{f}(\boldsymbol{\ell})|^2 \, r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{\ell})\right)^{1/2}$, one obtains a reproducing kernel Hilbert space, which is often referred to as "Korobov space". It is known (see, e.g., [6]) that the worst-case error in that space corresponds to the square root of the worst-case error in $E^\alpha_{d,\boldsymbol{\gamma}}$, and so all results shown in the present paper can directly be carried over to related results for the Korobov space. Note also that, by using embedding results, any results on the error in $E_{s,\boldsymbol{\gamma}}^{\alpha}$ immediately yield results on the worst-case error of “tent-transformed” lattice rules in certain Sobolev spaces of functions whose mixed partial derivatives of order 1 and 2 in each variable are square integrable (with $\alpha = 2$ and $\alpha = 4$ in our notation, respectively), see [1, 5, 7, 9] for further details.

2.1. The quality criterion used in this paper

In this section and the following we shall always assume that the number N of points equals 2^m, where $m \in \mathbb{N}$. Furthermore, we assume that we are given a sequence of non-negative integers $(w_j)_{j\geq 1}$ ordered in a non-decreasing fashion, to be more precise, $w_1, w_2, \ldots \in \mathbb{N}_0$ with $0 = w_1 \leq w_2 \leq \cdots$, and we put $Y_j := 2^{w_j}$ for $j \in [d]$. The numbers w_j for $j \geq 1$ are also called *reduction indices*, as they will help to reduce the computational cost of the CBC-DBD algorithm studied in this paper. In what follows, we set d^* as the largest j such that $w_j < m$, i.e., $d^* := \max\{j \in \mathbb{N} : w_j < m\}$. Moreover, we will usually (unless stated otherwise) assume that the generating vector $\boldsymbol{z}$ of a lattice rule under consideration takes the form $\boldsymbol{z} = (Y_1 z_1, \ldots, Y_d z_d) \in \mathbb{Z}^d$, where $z_j \in \{1, 3, 5, \ldots, 2^{m-w_j} - 1\}$ for $j \in [d^*]$, and z_j is odd if $j > d^*$ (in fact, we will often choose $z_j = 1$ for $j > d^*$, but any choice of odd numbers z_j for those j would yield the same results for our lattices rules).

We define the following quality measure $T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w})$ for $N \in \mathbb{N}$ and $\boldsymbol{w} = (w_1, \ldots, w_d) \in \mathbb{N}_0^d$ with $0 = w_1 \leq w_2 \leq \cdots$, by

$$T_{\gamma}(N, \boldsymbol{z}, \boldsymbol{w}) := \sum_{\boldsymbol{0} \neq \boldsymbol{\ell} \in M_{N,d,\boldsymbol{w}}} \frac{\delta_N(\boldsymbol{\ell} \cdot \boldsymbol{z})}{r_{1,\boldsymbol{\gamma}}(\boldsymbol{\ell})}, \tag{4}$$

where the set $M_{N,d,\boldsymbol{w}}$ is defined as follows. Let

$$M_{N,j,w_j} := \{-(2^{\max(0,m-w_j)} - 1), \ldots, 2^{\max(0,m-w_j)} - 1\} \qquad \text{for } j \in [d].$$

Furthermore, we write

$$M_{N,j,w_j}^* := \big(\{-(2^{\max(0,m-w_j)} - 1), \ldots, 2^{\max(0,m-w_j)} - 1\} \setminus \{0\}\big).$$

Note that $M_{N,j,w_j}^* = \emptyset$ for $j > d^*$. Furthermore, let

$$M_{N,d,\boldsymbol{w}} = M_{N,1,w_1} \times \cdots \times M_{N,d,w_d},$$

and

$$M_{N,d,\boldsymbol{w}}^* = M_{N,1,w_1}^* \times \cdots \times M_{N,d,w_d}^*.$$

Then, for $\emptyset \neq \mathfrak{u} \subseteq [d]$, by $M_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}}$ and $M_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}}^*$, we denote the projections of $M_{N,d,\boldsymbol{w}}$ and $M_{N,d,\boldsymbol{w}}^*$ onto the components with indices in $\mathfrak{u}$, respectively.

Likewise we define the auxiliary quantity

$$T_{\alpha,\boldsymbol{\gamma}}(N,\boldsymbol{z},\boldsymbol{w}) := \sum_{\boldsymbol{0}\neq\boldsymbol{\ell}\in M_{N,d,\boldsymbol{w}}} \frac{\delta_N(\boldsymbol{\ell}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{\ell})}.$$

Let us denote the Riemann zeta function by $\zeta(\alpha) := \sum_{n=1}^{\infty} n^{-\alpha}$ for $\alpha > 1$. Moreover, we write, for a non-empty set $\mathfrak{u} \subseteq [d]$,

$$\gamma_{\mathfrak{u}} := \prod_{j\in\mathfrak{u}} \gamma_j,$$

and we put $\gamma_{\emptyset} = 1$.

The following proposition shows that we can use $T_{\alpha,\boldsymbol{\gamma}}(N,\boldsymbol{z},\boldsymbol{w})$ as a suitable approximation to the worst-case error of a rank-1 lattice rule.

Proposition 2.2. *Let $N = 2^m$, let $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ be a sequence of positive weights, let $\boldsymbol{z} = (Y_1 z_1, \ldots, Y_d z_d) \in \mathbb{Z}^d$, and let $\boldsymbol{w} = (w_j)_{j\geq 1}$ be a sequence in $\mathbb{N}_0$, with $0 = w_1 \leq w_2 \leq \cdots$. Furthermore, assume that all z_j, $j \in [d]$, are odd, and that $z_j \in \{1,3,5,\ldots,2^{m-w_j}-1\}$ for $j \in [d^*]$. Then, for $\alpha > 1$, we have that*

$$\begin{aligned} e_{N,d,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) - T_{\alpha,\boldsymbol{\gamma}}(N,\boldsymbol{z},\boldsymbol{w}) &= \sum_{\boldsymbol{0}\neq\boldsymbol{\ell}\in\mathbb{Z}^d} \frac{\delta_N(\boldsymbol{\ell}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{\ell})} \\ &\quad - \sum_{\boldsymbol{0}\neq\boldsymbol{\ell}\in M_{N,d,\boldsymbol{w}}} \frac{\delta_N(\boldsymbol{\ell}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{\ell})} \\ &\leq \sum_{\emptyset\neq\mathfrak{u}\subseteq[d]} \gamma_{\mathfrak{u}} \frac{(4\zeta(\alpha))^{|\mathfrak{u}|}}{2^{\alpha\max(0,m-\max_{j\in\mathfrak{u}} w_j)}}. \end{aligned}$$

Proof. We can rewrite the difference $e_{N,d,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) - T_{\alpha,\boldsymbol{\gamma}}(N,\boldsymbol{z},\boldsymbol{w})$ as

$$\sum_{\emptyset\neq\mathfrak{u}\subseteq[d]} \left(\sum_{\boldsymbol{\ell}_{\mathfrak{u}}\in\mathbb{Z}_*^{|\mathfrak{u}|}} \frac{\delta_N(\boldsymbol{\ell}_{\mathfrak{u}}\cdot\boldsymbol{z}_{\mathfrak{u}})}{r_{\alpha,\gamma_{\mathfrak{u}}}(\boldsymbol{\ell}_{\mathfrak{u}})} - \sum_{\boldsymbol{\ell}_{\mathfrak{u}}\in M^*_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}}} \frac{\delta_N(\boldsymbol{\ell}_{\mathfrak{u}}\cdot\boldsymbol{z}_{\mathfrak{u}})}{r_{\alpha,\gamma_{\mathfrak{u}}}(\boldsymbol{\ell}_{\mathfrak{u}})} \right),$$

motivating us to define, for $\mathfrak{u} \neq \emptyset$, $\mathfrak{u} \subseteq [d]$,

$$T_{\mathfrak{u},\boldsymbol{w}_{\mathfrak{u}}} := \sum_{\boldsymbol{\ell}_{\mathfrak{u}}\in\mathbb{Z}_*^{|\mathfrak{u}|}} \frac{\delta_N(\boldsymbol{\ell}_{\mathfrak{u}}\cdot\boldsymbol{z}_{\mathfrak{u}})}{r_{\alpha,\gamma_{\mathfrak{u}}}(\boldsymbol{\ell}_{\mathfrak{u}})} - \sum_{\boldsymbol{\ell}_{\mathfrak{u}}\in M^*_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}}} \frac{\delta_N(\boldsymbol{\ell}_{\mathfrak{u}}\cdot\boldsymbol{z}_{\mathfrak{u}})}{r_{\alpha,\gamma_{\mathfrak{u}}}(\boldsymbol{\ell}_{\mathfrak{u}})}.$$

In the following we distinguish two cases for $\emptyset \neq \mathfrak{u} \subseteq [d]$, depending on whether $|\mathfrak{u}| = 1$ or not.

Case 1. Suppose that $|\mathfrak{u}| = 1$, i.e., $\mathfrak{u} = \{j\}$ for some $j \in [d]$. Then we have

$$\begin{aligned} T_{\mathfrak{u},\boldsymbol{w}_{\mathfrak{u}}} = T_{\{j\},w_j} &= \sum_{\ell_j \in \mathbb{Z}_*} \frac{\delta_N(\ell_j Y_j z_j)}{r_{\alpha,\gamma_{\{j\}}}(\ell_j)} - \sum_{\ell_j \in M^*_{N,\{j\},w_j}} \frac{\delta_N(\ell_j Y_j z_j)}{r_{\alpha,\gamma_{\{j\}}}(\ell_j)} \\ &= \sum_{|\ell_j| \geq 2^{\max(0,m-w_j)}} \gamma_j \frac{\delta_N(\ell_j Y_j z_j)}{|\ell_j|^\alpha} \\ &= 2\gamma_j \sum_{t=1}^{\infty} \frac{1}{(t 2^{\max(0,m-w_j)})^\alpha} \\ &= \frac{2\zeta(\alpha)}{2^{\alpha \max(0,m-w_j)}} \gamma_j, \qquad (5) \end{aligned}$$

which follows from the fact that if $\ell_j Y_j z_j \equiv 0 \pmod N$, this is equivalent to $\ell_j \equiv 0 \pmod{2^{\max(0,m-w_j)}}$.

Case 2. For $\mathfrak{u} \neq \emptyset$, $\mathfrak{u} \subseteq [d]$ with $|\mathfrak{u}| > 1$, and $i \in \mathfrak{u}$, we write, for short, $\boldsymbol{\ell}_{\mathfrak{u}\setminus\{i\}}, \boldsymbol{z}_{\mathfrak{u}\setminus\{i\}} \in \mathbb{Z}^{|\mathfrak{u}|-1}$ to denote the projections of $\boldsymbol{\ell}$ and $\boldsymbol{z}$, respectively, onto those components with indices in $\mathfrak{u} \setminus \{i\}$.

In this case, we estimate

$$T_{\mathfrak{u},\boldsymbol{w}_{\mathfrak{u}}} \leq \sum_{i \in \mathfrak{u}} \sum_{\boldsymbol{\ell}_{\mathfrak{u}\setminus\{i\}} \in \mathbb{Z}_*^{|\mathfrak{u}|-1}} \sum_{|\ell_i| \geq 2^{\max(0,m-w_i)}} \frac{\delta_N(\ell_i Y_i z_i + \boldsymbol{\ell}_{\mathfrak{u}\setminus\{i\}} \cdot \boldsymbol{z}_{\mathfrak{u}\setminus\{i\}})}{r_{\alpha,\gamma_{\mathfrak{u}}}(\boldsymbol{\ell}_{\mathfrak{u}})}.$$

For $i \in \mathfrak{u}$ and $\boldsymbol{\ell}_{\mathfrak{u}\setminus\{i\}} \in \mathbb{Z}_*^{|\mathfrak{u}|-1}$, we write $b := \boldsymbol{\ell}_{\mathfrak{u}\setminus\{i\}} \cdot \boldsymbol{z}_{\mathfrak{u}\setminus\{i\}}$, and consider the expression

$$\begin{aligned} \sum_{|\ell_i| \geq 2^{\max(0,m-w_i)}} \frac{\delta_N(\ell_i Y_i z_i + b)}{r_{\alpha,\gamma_{\mathfrak{u}}}(\boldsymbol{\ell}_{\mathfrak{u}})} &= \gamma_{\mathfrak{u}} \sum_{|\ell_i| \geq 2^{\max(0,m-w_i)}} \frac{\delta_N(\ell_i Y_i z_i + b)}{\prod_{j \in \mathfrak{u}} |\ell_j|^\alpha} \\ = \gamma_{\mathfrak{u}} &\left(\prod_{\substack{j \in \mathfrak{u} \\ j \neq i}} |\ell_j|^{-\alpha} \right) \sum_{|\ell_i| \geq 2^{\max(0,m-w_i)}} \frac{\delta_N(\ell_i Y_i z_i + b)}{|\ell_i|^\alpha} \\ = \gamma_{\mathfrak{u}} &\left(\prod_{\substack{j \in \mathfrak{u} \\ j \neq i}} |\ell_j|^{-\alpha} \right) \sum_{t=1}^{\infty} \sum_{\ell_i = t 2^{\max(0,m-w_i)}}^{(t+1)2^{\max(0,m-w_i)}-1} \left[\frac{\delta_N(\ell_i Y_i z_i + b)}{|\ell_i|^\alpha} + \frac{\delta_N(\ell_i Y_i z_i - b)}{|\ell_i|^\alpha} \right] \end{aligned}$$

$$\begin{aligned}
&\le \gamma_{\mathfrak{u}} \left(\prod_{\substack{j \in \mathfrak{u} \\ j \neq i}} |\ell_j|^{-\alpha} \right) \sum_{t=1}^{\infty} \frac{1}{(t 2^{\max(0,m-w_i)})^{\alpha}} \\
&\quad \times \sum_{\ell_i = t2^{\max(0,m-w_i)}}^{(t+1)2^{\max(0,m-w_i)}-1} (\delta_N(\ell_i Y_i z_i + b) + \delta_N(\ell_i Y_i z_i - b)) \\
&= \gamma_{\mathfrak{u}} \left(\prod_{\substack{j \in \mathfrak{u} \\ j \neq i}} |\ell_j|^{-\alpha} \right) \frac{1}{2^{\alpha \max(0,m-w_i)}} \sum_{t=1}^{\infty} \frac{1}{t^{\alpha}} \\
&\quad \times \sum_{\ell_i = t2^{\max(0,m-w_i)}}^{(t+1)2^{\max(0,m-w_i)}-1} \big(\delta_N(\ell_i Y_i z_i + b) + \delta_N(\ell_i Y_i z_i - b)\big) \\
&= \gamma_{\mathfrak{u}} \left(\prod_{\substack{j \in \mathfrak{u} \\ j \neq i}} |\ell_j|^{-\alpha} \right) \frac{2\zeta(\alpha)}{2^{\alpha \max(0,m-w_i)}},
\end{aligned}$$

where the last equality follows from the fact that

$$\sum_{\ell_i = t2^{\max(0,m-w_i)}}^{(t+1)2^{\max(0,m-w_i)}-1} \delta_N(\ell_i Y_i z_i + b) = 1$$

holds since the congruence $\ell_i Y_i z_i + b = \ell_i 2^{w_i} z_i + b \equiv 0 \pmod{N}$ is equivalent to $\ell_i 2^{w_i} \equiv -z_i^{-1} b \pmod{N}$, and then the latter congruence can have at most one solution ℓ_i in $\{t2^{\max(0,m-w_i)}, \ldots, (t+1)2^{\max(0,m-w_i)} - 1\}$, see also [15, Corollary of Proposition 1], and a similar argument can be used to handle the terms $\delta_N(\ell_i Y_i z_i - b)$. Therefore, we can estimate $T_{\mathfrak{u},\boldsymbol{w}_{\mathfrak{u}}}$ for $|u| > 1$, by

$$\begin{aligned}
T_{\mathfrak{u},\boldsymbol{w}_{\mathfrak{u}}} &\le \gamma_{\mathfrak{u}} 2\zeta(\alpha) \sum_{i \in \mathfrak{u}} \frac{1}{2^{\alpha \max(0,m-w_i)}} \sum_{\boldsymbol{\ell}_{\mathfrak{u} \setminus \{i\}} \in \mathbb{Z}_*^{|\mathfrak{u}|-1}} \prod_{\substack{j \in \mathfrak{u} \\ j \neq i}} |\ell_j|^{-\alpha} \\
&= \gamma_{\mathfrak{u}} 2\zeta(\alpha) \sum_{i \in \mathfrak{u}} \frac{1}{2^{\alpha \max(0,m-w_i)}} \left(2 \sum_{\ell=1}^{\infty} \frac{1}{\ell^{\alpha}} \right)^{|\mathfrak{u}|-1} \\
&= (2\zeta(\alpha))^{|\mathfrak{u}|} \gamma_{\mathfrak{u}} \sum_{i \in \mathfrak{u}} \frac{1}{2^{\alpha \max(0,m-w_i)}}
\end{aligned}$$

$$\leq \frac{(2\zeta(\alpha))^{|\mathfrak{u}|}\gamma_{\mathfrak{u}}\,|\mathfrak{u}|}{2^{\alpha\max(0,m-\max_{i\in\mathfrak{u}} w_i)}} \leq \gamma_{\mathfrak{u}} \frac{(4\zeta(\alpha))^{|\mathfrak{u}|}}{2^{\alpha\max(0,m-\max_{i\in\mathfrak{u}} w_i)}}. \tag{6}$$

In summary we obtain the claim in the proposition by combining the estimates in (5) and (6). □

We have obtained a key ingredient for the worst-case error analysis from the result on the truncation error in Proposition 2.2.

3. The construction method for rank-1 lattice rules

In this section, we will introduce, formulate, and analyze a method for the construction of lattice rules. Firstly, we want to be able to estimate efficiently the quantity $T_{\boldsymbol{\gamma}}(N,\boldsymbol{z},\boldsymbol{w})$ in (4), which is needed for our construction method, and to do this we summarize some auxiliary statements which will be needed in the following analysis.

3.1. Preliminary results

The following lemma (see [8] for a proof) shows that the function $\log\bigl(\sin^{-2}(\pi x)\bigr)$ can be written in terms of its truncated Fourier series with uniformly bounded remainder term. We also bear in mind that it cannot be evaluated in $x=0$ and $x=1$.

Lemma 3.1. *Let $N\in\mathbb{N}$, then for any $x\in(0,1)$ there exists a $\tau(x)\in\mathbb{R}$ with $|\tau(x)|\leq 1$ such that*

$$\log\left(\frac{1}{\sin^2(\pi x)}\right) = \log 4 + \sum_{\substack{\ell=-(N-1)\\ \ell\neq 0}}^{N-1} \frac{\mathrm{e}^{2\pi\mathrm{i}\ell x}}{|\ell|} + \frac{\tau(x)}{N\|x\|} = \sum_{\ell=-(N-1)}^{N-1} \frac{\mathrm{e}^{2\pi\mathrm{i}\ell x}}{b(\ell)} + \frac{\tau(x)}{N\|x\|},$$

with coefficients

$$b(\ell) := \begin{cases} |\ell| & \text{for } \ell\neq 0,\\ 1/\log 4 & \text{for } \ell = 0, \end{cases} \tag{7}$$

where $\|x\|$ denotes the distance to the nearest integer of x, i.e.,

$$\|x\| := \min\bigl\{\{x\}, 1-\{x\}\bigr\}.$$

What is more, we will make use of the following lemma, which was also proved in [8].

Lemma 3.2. *For $j \in [d]$, let $u_j, v_j, \overline{u}_j$, and r_j be real numbers which satisfy*

$$(a) \quad u_j = v_j + r_j, \quad (b) \quad |u_j| \le \bar{u}_j, \quad (c) \quad \bar{u}_j \ge 1.$$

Then, for any subset $\emptyset \neq \mathfrak{u} \subseteq [d]$ there exists a $\theta_{\mathfrak{u}}$ with $|\theta_{\mathfrak{u}}| \le 1$ such that the following identity holds,

$$\prod_{j\in\mathfrak{u}} u_j = \prod_{j\in\mathfrak{u}} v_j + \theta_{\mathfrak{u}} \left(\prod_{j\in\mathfrak{u}} (\bar{u}_j + |r_j|) \right) \sum_{j\in\mathfrak{u}} |r_j|. \tag{8}$$

3.2. The reduced CBC-DBD construction

We are now ready to study the reduced component-by-component digit-by--digit (CBC-DBD) construction for lattice rules. To this end, we assume again throughout the section that N is of the form 2^m for some positive integer m; this choice of N is natural as the components of the generating vector will be constructed digit by digit, i.e., bit by bit as we consider N to be a power of 2. Indeed, we will define below a function $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{\omega}}$ closely related to $T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w})$, which will depend on the s-th component of the generating vector when the first $s-1$ components have already been chosen. The core of Algorithm 1 is the following double loop, after having chosen suitable initial values:

Algorithm 0 (Double loop in Algorithm 1)**.**

```
for $s = 2$ to $\min\{d, d^*\}$ do
  for $v = 2$ to $m - w_s$ do
    $z^* = \operatorname{argmin}_{z\in\{0,1\}} h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}(z_{s,v-1} + z\,2^{v-1})$
    $z_{s,v} = z_{s,v-1} + z^*\,2^{v-1}$
  end for
  $z_s = z_{s,m-w_s}$
end for
```

I.e., the outer loop in Algorithm 0 goes through the components of the generating vector, up to $\min\{d, d^*\}$, whereas the inner loop recursively constructs each component digit-by-digit, where for component s we end after $m - w_s$ digits. We will give the full version in Algorithm 1 below, after we have identified the target function $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{\omega}}$.

We first show the following estimate on the quantity $T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w})$.

Theorem 3.3. *Let $N = 2^m$, with $m > 3$, let $\boldsymbol{\gamma} = (\gamma_j)_{j\ge1}$ be positive weights, and let $\boldsymbol{w} = (w_j)_{j\ge1}$ be a sequence of reduction indices in $\mathbb{N}_0$ with*

$$0 = w_1 \le w_2 \le \cdots$$

Furthermore, let $Y_j := 2^{w_j}$ *for* $j \in [d]$, *and let*

$$\boldsymbol{z} = (Y_1 z_1, \dots, Y_d z_d) \in \{1, \dots, N-1\}^d,$$

where we assume that all z_j, $j \in [d]$, *are odd, and that* $z_j \in \{1, 3, \dots, 2^{m-w_j} - 1\}$ *for* $j \in [d^*]$. *Then the following estimate holds,*

$$\begin{aligned} T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w}) &\le \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \frac{\gamma_{\mathfrak{u}}}{N} 2^{w_{j_{|\mathfrak{u}|}}+1} (6 \log N)^{|\mathfrak{u}|} (1 + \log N) \\ &\quad + \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \frac{\gamma_{\mathfrak{u}}}{N} H_{N,\mathfrak{u},\boldsymbol{w}} - \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} (\log 4)^{|\mathfrak{u}|}, \end{aligned}$$

where $j_{|\mathfrak{u}|}$ *denotes the largest element of* $\mathfrak{u}$ *for* $\emptyset \neq \mathfrak{u} \subseteq [d^*]$, *where* $d^* := \max\{j \in \mathbb{N} : w_j < m\}$, *and where*

$$H_{N,\mathfrak{u},\boldsymbol{w}}(\boldsymbol{z}) = H_{N,\mathfrak{u},\boldsymbol{w}} := \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} \prod_{j \in \mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k / 2^{m-w_j})} \right).$$

Proof. We have

$$T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w}) = \sum_{\boldsymbol{0} \neq \boldsymbol{\ell} \in M_{N,d,\boldsymbol{w}}} \frac{\delta_N(\boldsymbol{\ell} \cdot \boldsymbol{z})}{r_{1,\boldsymbol{\gamma}}(\boldsymbol{\ell})} = \sum_{\emptyset \neq \mathfrak{u} \subseteq [d]} \gamma_{\mathfrak{u}} \sum_{\boldsymbol{\ell}_{\mathfrak{u}} \in M^*_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}}} \frac{\delta_N(\boldsymbol{\ell}_{\mathfrak{u}} \cdot \boldsymbol{z}_{\mathfrak{u}})}{\prod_{j \in \mathfrak{u}} |\ell_j|}.$$

Note that

$$M^*_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}} = \emptyset \quad \text{if } \mathfrak{u} \cap \{d^*+1, d^*+2, \dots, d\} \neq \emptyset.$$

Therefore, recalling the definition of $b(\ell)$ in (7),

$$\begin{aligned} T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w}) &= \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} \sum_{\boldsymbol{\ell}_{\mathfrak{u}} \in M^*_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}}} \frac{\delta_N(\boldsymbol{\ell}_{\mathfrak{u}} \cdot \boldsymbol{z}_{\mathfrak{u}})}{\prod_{j \in \mathfrak{u}} b(\ell_j)} \\ &\le \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} \sum_{\boldsymbol{0} \neq \boldsymbol{\ell}_{\mathfrak{u}} \in M_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}}} \frac{\delta_N(\boldsymbol{\ell}_{\mathfrak{u}} \cdot \boldsymbol{z}_{\mathfrak{u}})}{\prod_{j \in \mathfrak{u}} b(\ell_j)} \\ &= \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \frac{\gamma_{\mathfrak{u}}}{N} \sum_{k=0}^{N-1} \left(\sum_{\boldsymbol{\ell}_{\mathfrak{u}} \in M_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}}} \frac{e^{2\pi \mathrm{i} k \boldsymbol{\ell}_{\mathfrak{u}} \cdot \boldsymbol{z}_{\mathfrak{u}}/N}}{\prod_{j \in \mathfrak{u}} b(\ell_j)} - (\log 4)^{|\mathfrak{u}|} \right) \\ &= \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \frac{\gamma_{\mathfrak{u}}}{N} \sum_{k=0}^{N-1} \sum_{\boldsymbol{\ell}_{\mathfrak{u}} \in M_{N,|\mathfrak{u}|,\boldsymbol{w}_{\mathfrak{u}}}} \frac{e^{2\pi \mathrm{i} k \boldsymbol{\ell}_{\mathfrak{u}} \cdot \boldsymbol{z}_{\mathfrak{u}}/N}}{\prod_{j \in \mathfrak{u}} b(\ell_j)} - \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} (\log 4)^{|\mathfrak{u}|} \end{aligned}$$

$$\begin{aligned}
&= \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \frac{\gamma_{\mathfrak{u}}}{N} \sum_{k=0}^{N-1} \prod_{j \in \mathfrak{u}} \left(\log 4 + \sum_{\ell_j \in M^*_{N,j,w_j}} \frac{e^{2\pi \mathrm{i} k \ell_j 2^{w_j} z_j / 2^m}}{|\ell_j|} \right) \\
&\quad - \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} (\log 4)^{|\mathfrak{u}|} \\
&= \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \frac{\gamma_{\mathfrak{u}}}{N} \sum_{k=0}^{N-1} \prod_{j \in \mathfrak{u}} \left(\log 4 + \sum_{\ell_j \in M^*_{N,j,w_j}} \frac{e^{2\pi \mathrm{i} k \ell_j z_j / 2^{m-w_j}}}{|\ell_j|} \right) \\
&\quad - \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} (\log 4)^{|\mathfrak{u}|}. \qquad (9)
\end{aligned}$$

We can analyze the first term in (9) as follows. We shall assume that $\mathfrak{u} = \{j_1, j_2, \dots, j_{|\mathfrak{u}|}\}$, such that $1 \leq j_1 < j_2 < \cdots < j_{|\mathfrak{u}|}$. This implies

$$m \geq m - w_{j_1} \geq m - w_{j_2} \geq \cdots \geq m - w_{j_{|\mathfrak{u}|}}.$$

Suppose now that $k \in \{0, 1, \dots, N-1\}$ is such that $k \not\equiv 0 \pmod{2^{m-w_{j_{|\mathfrak{u}|}}}}$. This also implies

$$k \not\equiv 0 \pmod{2^{m-w_{j_i}}} \quad \forall i \in \{1, \dots, |\mathfrak{u}|\}.$$

Therefore, for fixed $\emptyset \neq \mathfrak{u} \subseteq [d^*]$, we can rewrite the inner sum over k of the first term in (9) as

$$\begin{aligned}
&\sum_{k=0}^{N-1} \prod_{j \in \mathfrak{u}} \left(\log 4 + \sum_{\ell_j \in M^*_{N,j,w_j}} \frac{e^{2\pi \mathrm{i} k \ell_j z_j / 2^{m-w_j}}}{|\ell_j|} \right) \\
&\quad = \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} \left(\prod_{j \in \mathfrak{u}} \left(\log 4 + \sum_{\ell_j \in M^*_{N,j,w_j}} \frac{e^{2\pi \mathrm{i} k \ell_j z_j / 2^{m-w_j}}}{|\ell_j|} \right) \right) \\
&\qquad + \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \left(\prod_{j \in \mathfrak{u}} \left(\log 4 + \sum_{\ell_j \in M^*_{N,j,w_j}} \frac{e^{2\pi \mathrm{i} t 2^{m-w_{j_{|\mathfrak{u}|}}} \ell_j z_j / 2^{m-w_j}}}{|\ell_j|} \right) \right) \\
&\quad =: \Sigma_{\mathfrak{u}}^{(1)} + \Sigma_{\mathfrak{u}}^{(2)}. \qquad (10)
\end{aligned}$$

Now we estimate the second summand $\Sigma_{\mathfrak{u}}^{(2)}$ in (10) as follows,

$$\begin{aligned}\Sigma_{\mathfrak{u}}^{(2)} \le \left|\Sigma_{\mathfrak{u}}^{(2)}\right| &\le \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \prod_{j\in\mathfrak{u}} \left(\log 4 + \sum_{\ell_j \in M_{N,j,w_j}^{*}} \frac{1}{|\ell_j|} \right) \\ &= \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \prod_{j\in\mathfrak{u}} \left(\log 4 + 2 \sum_{\ell=1}^{2^{m-w_j}-1} \frac{1}{\ell} \right) \\ &\le \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \prod_{j\in\mathfrak{u}} \left(\log 4 + 2 \sum_{\ell=1}^{N-1} \frac{1}{\ell} \right) \\ &\le \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \prod_{j\in\mathfrak{u}} (\log 4 + 4 \log N) \\ &\le \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} (6\log N)^{|\mathfrak{u}|} \;=\; 2^{w_{j_{|\mathfrak{u}|}}} (6 \log N)^{|\mathfrak{u}|},\end{aligned}$$

where we implicitly assumed that $N \ge 3$ (which is no significant restriction). Let us now analyze $\Sigma_{\mathfrak{u}}^{(1)}$ in (10), where we proceed similarly to [8]. For $j \in \mathfrak{u}$ we get

$$\begin{aligned}\Sigma_{\mathfrak{u}}^{(1)} &= \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} \left(\prod_{j\in\mathfrak{u}} v_j(k) - \prod_{j\in\mathfrak{u}} u_j(k) + \prod_{j\in\mathfrak{u}} u_j(k) \right) \\ &= \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} \prod_{j\in\mathfrak{u}} u_j(k) \\ &\quad + \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} \left(\prod_{j\in\mathfrak{u}} v_j(k) - \prod_{j\in\mathfrak{u}} u_j(k) \right) \\ &= \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} \prod_{j\in\mathfrak{u}} u_j(k) \\ &\quad + \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} (-\theta_{\mathfrak{u}}(k)) \left(\prod_{j\in\mathfrak{u}} (\overline{u}_j(k) + |r_j(k)|) \right) \sum_{j\in\mathfrak{u}} |r_j(k)| , \quad (11)\end{aligned}$$

where in (11) we used Lemma 3.2 with

$$u_j = u_j(k) := \log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})}\right), \qquad \overline{u}_j = \overline{u}_j(k) := 2\log N,$$

$$v_j = v_j(k) := \log 4 + \sum_{\ell_j \in M^*_{N,j,w_j}} \frac{e^{2\pi \mathrm{i} k \ell_j z_j/2^{m-w_j}}}{|\ell_j|},$$

$$r_j = r_j(k) := \frac{\tau_j(k)}{2^{m-w_j}\,\|z_j k/2^{m-w_j}\|},$$

where the terms $\theta_{\mathfrak{u}}(k)$ are defined analogously to Lemma 3.2, and satisfy $|\theta_{\mathfrak{u}}(k)| \le 1$ and the $\tau_j(k)$ are analogous to Lemma 3.1, and also satisfy $|\tau_j(k)| \le 1$. It can be checked easily that the conditions in the lemmas are fulfilled. Indeed, Condition (a) in Lemma 3.2 is satisfied due to Lemma 3.1. Furthermore, as in [8, Proof of Theorem 2], we see that

$$\sin^2\left(\pi \frac{z_j k}{2^{m-w_j}}\right) \ge \left(\frac{1}{2^{m-w_j}}\right)^2.$$

Therefore,

$$|u_j(k)| = \log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})}\right) \le \log\left(\left(2^{m-w_j}\right)^2\right) \le 2\log N = \overline{u}_j(k).$$

Furthermore, $\overline{u}_j(k) \ge 1$, as long as $N \ge 2$, which again is not a real restriction. Next we show how to bound the second summand in (11) independently of the choice of $\boldsymbol{z}$ as follows,

$$\begin{aligned}
&\sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} (-\theta_{\mathfrak{u}}(k)) \left(\prod_{j\in\mathfrak{u}} (\overline{u}_j(k) + |r_j(k)|)\right) \sum_{j\in\mathfrak{u}} |r_j(k)| \\
&\le \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} |\theta_{\mathfrak{u}}(k)| \left(\prod_{j\in\mathfrak{u}} \left(2\log N + \frac{|\tau_j(k)|}{2^{m-w_j}\,\|z_j k/2^{m-w_j}\|}\right)\right) \\
&\quad \times \sum_{j\in\mathfrak{u}} \frac{|\tau_j(k)|}{2^{m-w_j}\,\|z_j k/2^{m-w_j}\|} \\
&\le \sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1} \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1} |\theta_{\mathfrak{u}}(k)| \left(\prod_{j\in\mathfrak{u}} (1 + 2\log N)\right) \\
&\quad \times \sum_{j\in\mathfrak{u}} \frac{|\tau_j(k)|}{2^{m-w_j}\,\|z_j k/2^{m-w_j}\|}
\end{aligned}$$

$$\begin{aligned}
&= \left(\prod_{j\in\mathfrak{u}}(1+2\log N)\right)\sum_{j\in\mathfrak{u}}\sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1}\sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1}\frac{|\theta_{\mathfrak{u}}(k)|\,|\tau_j(k)|}{2^{m-w_j}\,\|z_jk/2^{m-w_j}\|}\\
&\le (1+2\log N)^{|\mathfrak{u}|}\sum_{j\in\mathfrak{u}}\sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1}\sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1}\frac{1}{2^{m-w_j}\,\|z_jk/2^{m-w_j}\|}\\
&\le (1+2\log N)^{|\mathfrak{u}|}\sum_{j\in\mathfrak{u}}\sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1}\sum_{k=1}^{2^{m-w_j}-1}\frac{1}{2^{m-w_j}\,\|z_jk/2^{m-w_j}\|}\\
&\le (1+2\log N)^{|\mathfrak{u}|}2\,|\mathfrak{u}|\,2^{w_{j_{|\mathfrak{u}|}}}(1+\log N)\\
&\le (1+2\log N)^{|\mathfrak{u}|}2^{|\mathfrak{u}|}2^{w_{j_{|\mathfrak{u}|}}}(1+\log N)\\
&\le (6\log N)^{|\mathfrak{u}|}2^{w_{j_{|\mathfrak{u}|}}}(1+\log N),
\end{aligned}$$

where we used that $|\theta_{\mathfrak{u}}(k)|\le 1$ and $|\tau_j(k)|\le 1$, and that

$$2^{m-w_j}\left\|\frac{z_jk}{2^{m-w_j}}\right\|\ge 2^{m-w_j}\left\|\frac{1}{2^{m-w_j}}\right\|=1,$$

that $2\,|\mathfrak{u}|\le 2^{|\mathfrak{u}|}$, and that $2^{m-w_j}\ge 2^{m-w_{j_{|\mathfrak{u}|}}}$ for $j\in\mathfrak{u}$, and hence, if k runs through the integers $\{t2^{m-w_{j_{|\mathfrak{u}|}}}+1,\dots,(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1\}$, the values of $\|z_jk/2^{m-w_j}\|$ are a subset of the values of $\|z_jk/2^{m-w_j}\|$ when k runs through the integers $\{1,\dots,2^{m-w_j}-1\}$. Then, due to [15, Corollary of Proposition 4], we know that

$$\begin{aligned}
&\sum_{k=1}^{2^{m-w_j}-1}\frac{1}{2^{m-w_j}\,\|z_jk/2^{m-w_j}\|}\\
&\le 2(1+\log 2^{m-w_j})\le 2(1+\log N).
\end{aligned}$$

Combining the estimates for $\Sigma_{\mathfrak{u}}^{(1)}$ and $\Sigma_{\mathfrak{u}}^{(2)}$, this yields

$$\begin{aligned}
\Sigma_{\mathfrak{u}}^{(1)}+\Sigma_{\mathfrak{u}}^{(2)} &= 2^{w_{j_{|\mathfrak{u}|}}}(6\log N)^{|\mathfrak{u}|}+H_{N,\mathfrak{u},\boldsymbol{w}}+(6\log N)^{|\mathfrak{u}|}2^{w_{j_{|\mathfrak{u}|}}}(1+\log N)\\
&\le 2^{w_{j_{|\mathfrak{u}|}}+1}(6\log N)^{|\mathfrak{u}|}(1+\log N)+H_{N,\mathfrak{u},\boldsymbol{w}}, \qquad (12)
\end{aligned}$$

where

$$H_{N,\mathfrak{u},\boldsymbol{w}}=\sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1}\sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1}\prod_{j\in\mathfrak{u}}u_j(k).$$

We can now plug the bound in (12) into (10) and insert the resulting estimate into (9), which then yields the result. □

In the following, we write

$$H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}) = H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}} := \sum_{\emptyset \neq \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}} H_{N,\mathfrak{u},\boldsymbol{w}}.$$

The next lemma motivates the choice of our quality function for the algorithm. In particular, assuming we have already fixed the first (i.e., the least significant) $v-1$ bits of z_s for $s \in [d^*]$, we would like to find out how good a specific choice for the vth bit is, in terms of $H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}$. To this end, we consider the average over all remaining $m - w_s - v$ bits.

For the base 2-digit representation of a z_s with $1 \leq s \leq d$, we write

$$z_s = z_s^{(0)} + z_s^{(1)} 2 + z_s^{(2)} 2^2 + \cdots$$

Lemma 3.4. *For an integer $v \in \{1, \ldots, m - w_s - 1\}$, with $m \in \mathbb{N}$ and $s \in [d^*]$, let $z \in \{0,1\}$ and $\boldsymbol{z} = (z_1, \ldots, z_s) \in \mathbb{Z}^s$, and where the first $v-1$ bits of z_s have been selected. We write $z_{s,v-1} := z_s^{(0)} + z_s^{(1)} 2 + \cdots + z_s^{(v-2)} 2^{v-2}$, and $\widetilde{z}_s := z_s^{(0)} + z_s^{(1)} 2 + \cdots + z_s^{(v-2)} 2^{v-2} + z 2^{v-1}$. Then the average of $H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}$ over the next $m - w_s - v$ bit choices for z_s is given by*

$$\frac{1}{2^{m-w_s-v}} \sum_{\overline{z} \in \mathbb{Z}_{2^{m-w_s-v}}} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(z_1, \ldots, z_{s-1}, \widetilde{z}_s + \overline{z} 2^v)$$

$$= \sum_{t=v}^{m-w_s} \frac{1}{2^{t-v}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{\substack{k=\ell 2^t+1 \\ k \equiv 1 \ (\mathrm{mod}\ 2)}}^{(\ell+1)2^t-1} \sum_{\substack{\emptyset \neq \mathfrak{u} \subseteq [s] \\ s \in \mathfrak{u}}} \gamma_{\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi k \widetilde{z}_s / 2^v)}\right)$$

$$\times \prod_{\substack{j \in \mathfrak{u} \\ j \neq s}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t} \frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right) + S_{N,v,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}), \tag{13}$$

where the term $S_{N,v,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z})$, which is independent of z and $\bar{z}$, is given by

$$S_{N,v,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}) = \sum_{r=1}^{s-1} \sum_{\substack{k=0 \\ k \not\equiv 0 \ (\mathrm{mod}\ 2^{m-w_r})}}^{N-1} \sum_{\substack{\emptyset \neq \mathfrak{u} \subseteq [r] \\ r \in \mathfrak{u}}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k / 2^{m-w_j})}\right)$$

$$+ \sum_{\ell=0}^{2^{w_s}-1} \sum_{t=1}^{v-1} \sum_{\substack{k=\ell 2^t+1 \\ k \equiv 1 \ (\mathrm{mod}\ 2)}}^{(\ell+1)2^t-1} \sum_{\substack{\emptyset \neq \mathfrak{u} \subseteq [s] \\ s \in \mathfrak{u}}} \gamma_{\mathfrak{u}} \left(\prod_{\substack{j \in \mathfrak{u} \\ j \neq s}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t} \frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right)\right)$$

$$\times \log\left(\frac{1}{\sin^2(\pi z_{s,v-1} k / 2^t)}\right)$$

$$+\sum_{\ell=0}^{2^{w_s}-1}\sum_{t=v}^{m-w_s}\frac{2^{t-v}-1}{2^{t-v}}(\log 4)$$
$$\times\sum_{\substack{k=\ell 2^t+1\\ k\equiv 1 \pmod 2}}^{(\ell+1)2^t-1}\sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}}\gamma_{\mathfrak{u}}\prod_{\substack{j\in\mathfrak{u}\\ j\neq s}}\log\left(\frac{1}{\sin^2\left(\pi z_j\frac{k}{2^t}\frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right).$$

Proof. For arbitrary $\boldsymbol{z}=(z_1,\dots,z_s)$, we can rewrite the quantity $H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}$ as follows,

$$\begin{aligned}
H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}} &= \sum_{\emptyset\neq\mathfrak{u}\subseteq[s]}\gamma_{\mathfrak{u}}\sum_{t=0}^{2^{w_{j_{|\mathfrak{u}|}}}-1}\ \sum_{k=t2^{m-w_{j_{|\mathfrak{u}|}}}+1}^{(t+1)2^{m-w_{j_{|\mathfrak{u}|}}}-1}\prod_{j\in\mathfrak{u}}\log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})}\right)\\
&= \sum_{\emptyset\neq\mathfrak{u}\subseteq[s]}\gamma_{\mathfrak{u}}\sum_{\substack{k=0\\ k\not\equiv 0 \pmod{2^{m-w_{j_{|\mathfrak{u}|}}}}}}^{N-1}\prod_{j\in\mathfrak{u}}\log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})}\right)\\
&= \sum_{r=1}^{s}\sum_{\substack{k=0\\ k\not\equiv 0 \pmod{2^{m-w_r}}}}^{N-1}\sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[r]\\ r\in\mathfrak{u}}}\gamma_{\mathfrak{u}}\prod_{j\in\mathfrak{u}}\log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})}\right). \qquad (14)
\end{aligned}$$

Hence, we would like to estimate

$$\frac{1}{2^{m-w_s-v}}\sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}}\sum_{r=1}^{s}\sum_{\substack{k=0\\ k\not\equiv 0 \pmod{2^{m-w_r}}}}^{N-1}\sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[r]\\ r\in\mathfrak{u}}}\gamma_{\mathfrak{u}}\prod_{j\in\mathfrak{u}}\log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})}\right), \qquad (15)$$

where we now assume that $z_s=z_{s,v-1}+z2^{v-1}+\overline{z}2^v=\widetilde{z}_s+\overline{z}2^v$. Observe that all terms in the sum over $r\in\{1,2,\dots,s\}$ for which $1\le r\le s-1$ are independent of $\overline{z}$, and from this we obtain the first sum in the definition of $S_{m,v,\boldsymbol{\gamma},\boldsymbol{w}}$. Therefore we need to analyze the remaining part of (15), which equals

$$\frac{1}{2^{m-w_s-v}}\sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}}\sum_{\substack{k=0\\ k\not\equiv 0 \pmod{2^{m-w_s}}}}^{N-1}\sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}}\gamma_{\mathfrak{u}}\prod_{j\in\mathfrak{u}}\log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})}\right). \qquad (16)$$

Now, we are going to use the general identity

$$\sum_{k=1}^{2^p-1} f(k/2^p) = \sum_{t=1}^{p} \sum_{\substack{k=1 \\ k\equiv 1 \pmod 2}}^{2^t-1} f(k/2^t). \tag{17}$$

Then, we can write the term in (16) as

$$\begin{aligned}
&\frac{1}{2^{m-w_s-v}} \sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{k=\ell\, 2^{m-w_s}+1}^{(\ell+1)2^{m-w_s}-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s] \\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \\
&\qquad \times \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})}\right) \\
&= \frac{1}{2^{m-w_s-v}} \sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{k=1}^{2^{m-w_s}-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s] \\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \\
&\qquad \times \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j (k+\ell\, 2^{m-w_s})/2^{m-w_j})}\right) \\
&= \frac{1}{2^{m-w_s-v}} \sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{k=1}^{2^{m-w_s}-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s] \\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \\
&\qquad \times \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j \ell\, 2^{w_j-w_s} + \pi z_j k/2^{m-w_j})}\right) \\
&= \frac{1}{2^{m-w_s-v}} \sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{k=1}^{2^{m-w_s}-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s] \\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \\
&\qquad \times \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j \ell\, 2^{w_j-w_s} + (\pi z_j k/2^{m-w_s}) 2^{w_j-w_s})}\right) \\
&= \frac{1}{2^{m-w_s-v}} \sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{t=1}^{m-w_s} \sum_{\substack{k=1 \\ k\equiv 1 \pmod 2}}^{2^t-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s] \\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \\
&\qquad \times \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j \ell\, 2^{w_j-w_s} + (\pi z_j k/2^{t}) 2^{w_j-w_s})}\right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2^{m-w_s-v}} \sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{t=1}^{m-w_s} \sum_{\substack{k=1\\ k\equiv 1 \ (\mathrm{mod}\ 2)}}^{2^t-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \\
&\quad \times \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{\ell 2^t+k}{2^t}\, \frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right) \\
&= \frac{1}{2^{m-w_s-v}} \sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{t=1}^{m-w_s} \sum_{\substack{k=\ell\, 2^t+1\\ k\equiv 1 \ (\mathrm{mod}\ 2)}}^{(\ell+1)2^t-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \\
&\quad \times \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t}\, \frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right) \\
&= \sum_{\ell=0}^{2^{w_s}-1} \sum_{t=1}^{m-w_s} \sum_{\substack{k=\ell 2^t+1\\ k\equiv 1 \ (\mathrm{mod}\ 2)}}^{(\ell+1)2^t-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \left(\prod_{\substack{j\in\mathfrak{u}\\ j\neq s}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t}\, \frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right)\right) \\
&\quad \times \frac{1}{2^{m-w_s-v}} \sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}} \log\left(\frac{1}{\sin^2(\pi(\widetilde{z}_s+\overline{z}2^v)k/2^t)}\right),
\end{aligned}$$

where we remind the reader that $\widetilde{z}_s = z_s^{(0)} + z_s^{(1)}2 + \cdots + z_s^{(v-2)}2^{v-2} + z2^{v-1}$. Therefore, in the above summation, if $t \in \{1, 2, \ldots, v-1\}$, then

$$\sin^2(\pi(\widetilde{z}_s+\overline{z}2^v)k/2^t) = \sin^2(\pi(z_s^{(0)} + \cdots + z_s^{(v-2)}2^{v-2})k/2^t).$$

Consequently, we obtain

$$\begin{aligned}
&\frac{1}{2^{m-w_s-v}} \sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}} \sum_{\substack{k=0\\ k\not\equiv 0 \ (\mathrm{mod}\ 2^{m-w_s})}}^{N-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})}\right) \\
&\quad = \sum_{\ell=0}^{2^{w_s}-1} \sum_{t=1}^{v-1} \sum_{\substack{k=\ell 2^t+1\\ k\equiv 1 \ (\mathrm{mod}\ 2)}}^{(\ell+1)2^t-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \left(\prod_{\substack{j\in\mathfrak{u}\\ j\neq s}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t}\, \frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right)\right) \\
&\qquad \times \log\left(\frac{1}{\sin^2(\pi(z_s^{(0)} + z_s^{(1)}2 + \cdots + z_s^{(v-2)}2^{v-2})k/2^t)}\right)
\end{aligned}$$

$$+\sum_{\ell=0}^{2^{w_s}-1}\sum_{t=v}^{m-w_s}\sum_{\substack{k=\ell 2^t+1\\ k\equiv 1 \pmod 2}}^{(\ell+1)2^t-1}\sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}}\gamma_{\mathfrak{u}}\left(\prod_{\substack{j\in\mathfrak{u}\\ j\neq s}}\log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t}\,\frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right)\right)$$
$$\times\frac{1}{2^{m-w_s-v}}\sum_{\overline{z}\in\mathbb{Z}_{2^{m-w_s-v}}}\log\left(\frac{1}{\sin^2(\pi(\widetilde{z}_s+\overline{z}2^v)k/2^t)}\right). \tag{18}$$

For the second of the two summands in (18), we argue as in the proof of Lemma 6 in [8] to see that it equals, for $t\in\{v,\dots,m-w_s\}$,

$$\sum_{\ell=0}^{2^{w_s}-1}\sum_{t=v}^{m-w_s}\sum_{\substack{k=\ell 2^t+1\\ k\equiv 1 \pmod 2}}^{(\ell+1)2^t-1}\sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}}\gamma_{\mathfrak{u}}\left(\prod_{\substack{j\in\mathfrak{u}\\ j\neq s}}\log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t}\,\frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right)\right)$$
$$\times\frac{2^{m-w_s-t}}{2^{m-w_s-v}}\sum_{\overline{z}\in\mathbb{Z}_{2^{t-v}}}\log\left(\frac{1}{\sin^2(\pi(\widetilde{z}_s+\overline{z}2^v)k/2^t)}\right)$$
$$=\sum_{\ell=0}^{2^{w_s}-1}\sum_{t=v}^{m-w_s}\sum_{\substack{k=\ell 2^t+1\\ k\equiv 1 \pmod 2}}^{(\ell+1)2^t-1}\sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s]\\ s\in\mathfrak{u}}}\gamma_{\mathfrak{u}}\left(\prod_{\substack{j\in\mathfrak{u}\\ j\neq s}}\log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t}\,\frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right)\right)$$
$$\times\left((1-2^{v-t})\log 4+2^{v-t}\log\left(\frac{1}{\sin^2(\pi\widetilde{z}_s k/2^v)}\right)\right),$$

where in the last step we proceeded exactly as in the proof of Lemma 6 in [8]. This, together with the previous identity yields the claim. □

We observe that in Lemma 3.4 only the first term depends on the vth bit z of z_s, while $S_{N,v,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z})$ is independent of this bit. This now leads to the introduction of the following digit-wise quality function for the reduced CBC-DBD algorithm which is based on the first term in (13). Note that the quality function is not exactly equal to the first term in (13), but we add further terms that, though independent of the argument of the function, facilitate fast implementation (see Section 4).

Definition 3.5. Let $x\in\mathbb{N}$ be an odd integer, let $m,d^*\in\mathbb{N}$, let $\boldsymbol{w}=(w_j)_{j\geq 1}$ be a sequence of reduction indices in $\mathbb{N}_0$ with $0=w_1\leq w_2\leq\cdots$, and let $\boldsymbol{\gamma}=(\gamma_j)_{j\geq 1}$ be a sequence of positive weights. For $1\leq s\leq d^*$ and $1\leq v\leq m-w_s$

we define the quality function $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}} : \mathbb{Z} \to \mathbb{R}$ as

$$h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}(x) := \sum_{t=v}^{m-w_s} \frac{1}{2^{t-v}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{\substack{k=\ell 2^t+1 \\ k \equiv 1 \pmod 2}}^{(\ell+1)2^t-1} \sum_{\substack{\mathfrak{u} \subseteq [s] \\ s \in \mathfrak{u}}} \gamma_{\mathfrak{u} \setminus \{s\}}$$

$$\times \left(\prod_{\substack{j \in \mathfrak{u} \\ j \neq s}} \log \left(\frac{1}{\sin^2 \left(\pi z_j \frac{k}{2^t} \frac{2^{m-w_s}}{2^{m-w_j}} \right)} \right) \right) \left(1 + \gamma_s \log \left(\frac{1}{\sin^2(\pi k x / 2^v)} \right) \right),$$

where we assume that $z_j \in \{1, 3, \ldots, 2^{m-w_j} - 1\}$ is odd for $j \in [s-1]$.

Based on the quality function $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}$, we formulate the following reduced component-by-component digit-by-digit (CBC-DBD) algorithm.

Algorithm 1 (Reduced component-by-component digit-by-digit construction)**.**

Input: Integer $m \in \mathbb{N}$, dimension $d, \boldsymbol{w} = (w_j)_{j \geq 1}$ with $0 = w_1 \leq w_2 \leq \cdots$ and $Y_j = 2^{w_j}$ for $j \in \{1, \ldots, d\}$, and positive weights $\boldsymbol{\gamma} = (\gamma_j)_{j \geq 1}$.

```
Set $z_1 = z_{1,m} = 1$ and $z_{1,1} = z_{2,1} = \cdots = z_{d,1} = 1$.
If $d > d^*$, set $z_{d^*+1} = \cdots = z_d = 1$.

for $s = 2$ to $\min\{d, d^*\}$ do
  for $v = 2$ to $m - w_s$ do
    $z^* = \underset{z \in \{0,1\}}{\operatorname{argmin}}\, h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}(z_{s,v-1} + z\, 2^{v-1})$
    $z_{s,v} = z_{s,v-1} + z^*\, 2^{v-1}$
  end for
  $z_s = z_{s,m-w_s}$
end for

Set $\boldsymbol{z} = (Y_1 z_1, \ldots, Y_d z_d)$.
```

Return: Generating vector $\boldsymbol{z}$ for $N = 2^m$.

3.3. Error convergence behaviour of the constructed lattice rules

In the following, we study the worst-case error behaviour of the constructed lattice rules, i.e, we want to show that under certain suitable conditions on the weights $\boldsymbol{\gamma}$, Algorithm 1 can construct generating vectors which yield lattice point sets with very good properties if they are used as integration nodes in a QMC rule. For $\boldsymbol{z} = (Y_1 z_1, \ldots, Y_d z_d)$, we write $\boldsymbol{z}_{[s]}$ to denote the vector $(z_1, \ldots, z_s)$ for $s \in [d]$.

Theorem 3.6. *Let $m \in \mathbb{N}$, $N = 2^m$, $\boldsymbol{w} = (w_j)_{j\geq 1}$ in $\mathbb{N}_0$ with $0 = w_1 \leq w_2 \leq \cdots$, and let $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ be positive product weights. Furthermore, let the generating vector $\boldsymbol{z} = (Y_1 z_1, \ldots, Y_d z_d) \in \mathbb{Z}^d$ be constructed by Algorithm 1. Denote by $\boldsymbol{z}_{[s]}$ the vector $(z_1, \ldots, z_s)$ for $s \in [d]$. Then the following estimate holds for $s \in [d^*]$,*

$$H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s]}) \leq (1 + \gamma_s \log 4) H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]}) + \gamma_s (\log 4)(2^m - 2^{w_s}).$$

Proof. We will prove the stated estimate via an inductive argument over the selection of the $m - w_s$ bits of the component z_s for $s \in [d^*]$. We first observe that according to the formulation of Algorithm 1, the vth bit of z_s, with $v \in \{2, \ldots, m - w_s\}$, has been selected by minimizing $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}(z_{s,v-1} + z2^{v-1})$ with respect to the choices $z \in \{0,1\}$, and that we have chosen $z_{s,v-1}$ by the same algorithm. By Lemma 3.4 and Definition 3.5 this is equivalent to minimizing

$$\frac{1}{2^{m-w_s-v}} \sum_{\overline{z} \in \mathbb{Z}_{2^{m-w_s-v}}} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(z_1, \ldots, z_{s-1}, z_{s,v-1} + z2^{v-1} + \overline{z}2^v)$$

with respect to $z \in \{0,1\}$. By the standard averaging argument, this yields

$$\begin{aligned} &\operatorname*{argmin}_{z\in\{0,1\}} \frac{1}{2^{m-w_s-v}} \sum_{\overline{z} \in \mathbb{Z}_{2^{m-w_s-v}}} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(z_1, \ldots, z_{s-1}, z_{s,v-1} + z2^{v-1} + \overline{z}2^v) \\ \leq\ & \frac{1}{2} \frac{1}{2^{m-w_s-v}} \sum_{z\in\mathbb{Z}_2} \sum_{\overline{z} \in \mathbb{Z}_{2^{m-w_s-v}}} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(z_1, \ldots, z_{s-1}, z_{s,v-1} + z2^{v-1} + \overline{z}2^v) \qquad (19) \\ =\ & \frac{1}{2^{m-w_s-v+1}} \sum_{\overline{z} \in \mathbb{Z}_{2^{m-w_s-v+1}}} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}\Bigg(z_1, \ldots, z_{s-1}, \underbrace{z_{s,v-2} + \tilde{z}\, 2^{v-2}}_{=z_{s,v-1}} + \overline{z}\, 2^{v-1}\Bigg), \end{aligned}$$

where we split up $z_{s,v-1}$ according to Algorithm 1 such that $\tilde{z}$ is the $(v-1)$th bit of z_s, selected in the previous step of the algorithm. Noting that the inequality in (19) holds for any $v \in \{2, \ldots, m - w_s\}$, we can inductively use this estimate for $v = m - w_s, m - w_s - 1, \ldots, 2$ to obtain

$$\begin{aligned} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s]}) &= \operatorname*{argmin}_{z\in\{0,1\}} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(z_1, \ldots, z_{s-1}, z_{s,m-w_s-1} + z2^{m-w_s-1}) \\ &\leq \frac{1}{2^{m-w_s-1}} \sum_{\overline{z} \in \mathbb{Z}_{2^{m-w_s-1}}} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(z_1, \ldots, z_{s-1}, 1 + \overline{z}2), \qquad (20) \end{aligned}$$

where we used that $z_{s,1} = 1$. Now setting $v = 1$ in our expression in Lemma 3.4

to equate the right-hand side term in (20), we get

$$\begin{aligned} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s]}) \leq & \sum_{t=1}^{m-w_s} \frac{1}{2^{t-1}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{\substack{k=\ell 2^t+1 \\ k\equiv 1 \pmod 2}}^{(\ell+1)2^t-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s] \\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \\ & \times \left(\prod_{\substack{j\in\mathfrak{u} \\ j\neq s}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t} \frac{2^{m-w_s}}{2^{m-w_j}}\right)} \right) \right) \log\left(\frac{1}{\sin^2(\pi k/2)} \right) \\ & + \sum_{r=1}^{s-1} \sum_{\substack{k=0 \\ k\not\equiv 0 \pmod{2^{m-w_r}}}}^{N-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[r] \\ r\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})} \right) \\ & + \sum_{\ell=0}^{2^{w_s}-1} \sum_{t=1}^{m-w_s} \frac{2^{t-1}-1}{2^{t-1}} (\log 4) \sum_{\substack{k=\ell 2^t+1 \\ k\equiv 1 \pmod 2}}^{(\ell+1)2^t-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s] \\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \\ & \times \prod_{\substack{j\in\mathfrak{u} \\ j\neq s}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t} \frac{2^{m-w_s}}{2^{m-w_j}}\right)} \right). \end{aligned}$$

Note that for odd k we have $\log\bigl(\sin^{-2}(\pi k/2)\bigr) = \log 1 = 0$, so we obtain, using (14) and (17),

$$\begin{aligned} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s]}) \leq & \sum_{r=1}^{s-1} \sum_{\substack{k=0 \\ k\not\equiv 0 \pmod{2^{m-w_r}}}}^{N-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[r] \\ r\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k/2^{m-w_j})} \right) \\ & + \sum_{\ell=0}^{2^{w_s}-1} \sum_{t=1}^{m-w_s} (\log 4) \sum_{\substack{k=\ell 2^t+1 \\ k\equiv 1 \pmod 2}}^{(\ell+1)2^t-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s] \\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \prod_{\substack{j\in\mathfrak{u} \\ j\neq s}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t} \frac{2^{m-w_s}}{2^{m-w_j}}\right)} \right) \\ = \quad & H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]}) \\ & + (\log 4) \sum_{\ell=0}^{2^{w_s}-1} \sum_{t=1}^{m-w_s} \sum_{\substack{k=\ell 2^t+1 \\ k\equiv 1 \pmod 2}}^{(\ell+1)2^t-1} \sum_{\substack{\emptyset\neq\mathfrak{u}\subseteq[s] \\ s\in\mathfrak{u}}} \gamma_{\mathfrak{u}} \prod_{\substack{j\in\mathfrak{u} \\ j\neq s}} \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t} \frac{2^{m-w_s}}{2^{m-w_j}}\right)} \right) \end{aligned}$$

$$
\begin{aligned}
&= H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]}) \\
&\quad + (\log 4) \sum_{\ell=0}^{2^{w_s}-1} \sum_{k=\ell 2^{m-w_s}+1}^{(\ell+1)2^{m-w_s}-1} \sum_{\substack{\emptyset \neq \mathfrak{u} \subseteq [s] \\ s \in \mathfrak{u}}} \gamma_{\mathfrak{u}} \prod_{\substack{j \in \mathfrak{u} \\ j \neq s}} \log\left(\frac{1}{\sin^2(\pi z_j k / 2^{m-w_j})}\right) \\
&= H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]}) \\
&\quad + (\log 4)\, \gamma_s \sum_{\substack{k=0 \\ k \not\equiv 0 \pmod{2^{m-w_s}}}}^{N-1} \sum_{\mathfrak{u} \subseteq [s-1]} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k / 2^{m-w_j})}\right) \\
&= H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]}) + (\log 4)\, \gamma_s \Bigg[\sum_{\substack{k=0 \\ k \not\equiv 0 \pmod{2^{m-w_s}}}}^{N-1} 1 \\
&\quad + \sum_{\substack{k=0 \\ k \not\equiv 0 \pmod{2^{m-w_s}}}}^{N-1} \sum_{\emptyset \neq \mathfrak{u} \subseteq [s-1]} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k / 2^{m-w_j})}\right) \Bigg] \\
&= H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]}) + (\log 4)\, \gamma_s \Bigg[(2^m - 2^{w_s}) \\
&\quad + \sum_{r=1}^{s-1} \sum_{\substack{k=0 \\ k \not\equiv 0 \pmod{2^{m-w_s}}}}^{N-1} \sum_{\substack{\emptyset \neq \mathfrak{u} \subseteq [r-1] \\ r \in \mathfrak{u}}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k / 2^{m-w_j})}\right) \Bigg] \\
&\leq H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]}) + (\log 4)\, \gamma_s \Bigg[(2^m - 2^{w_s}) \\
&\quad + \sum_{r=1}^{s-1} \sum_{\substack{k=0 \\ k \not\equiv 0 \pmod{2^{m-w_r}}}}^{N-1} \sum_{\substack{\emptyset \neq \mathfrak{u} \subseteq [r-1] \\ r \in \mathfrak{u}}} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \log\left(\frac{1}{\sin^2(\pi z_j k / 2^{m-w_j})}\right) \Bigg] \\
&= H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]}) + (\log 4) \left[\gamma_s (2^m - 2^{w_s}) + \gamma_s H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]})\right].
\end{aligned}
$$

Hence we have the claimed result. □

Based on the result in Theorem 3.6 we can use an inductive argument to show that the quantity $H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}$ is sufficiently small if $\boldsymbol{z}_{[s]}$ has been constructed by Algorithm 1. We obtain the following estimate.

Theorem 3.7. *Let $m \in \mathbb{N}$, $N = 2^m$, $\boldsymbol{w} = (w_j)_{j\geq 1}$ in $\mathbb{N}_0$ with $0 = w_1 \leq w_2 \leq \cdots$, and let $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ be positive product weights. Furthermore, let the generating vector $\boldsymbol{z} = (Y_1 z_1, \dots, Y_d z_d) \in \mathbb{Z}^d$ be constructed by Algorithm 1. Denote by $\boldsymbol{z}_{[s]}$ the vector $(z_1, \dots, z_s)$ for $s \in [d]$. Then the following upper bound on $H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s]})$ holds for $s \in [d^*]$,*

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

Proof. Observe that due to Theorem 3.6,

$$H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s]}) \leq (1 + \gamma_s \log 4) H_{s-1,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-1]}) + \gamma_s (\log 4)(2^m - 2^{w_s}) \qquad (21)$$

holds for $2 \leq s \leq d^*$. We can apply this estimate inductively to obtain the following,

$$\begin{aligned}
& H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s]}) \\
\leq\ & (1 + \gamma_s \log 4) \left[(1 + \gamma_{s-1} \log 4) H_{s-2,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-2]}) \right. \\
& \left. + \gamma_{s-1} (\log 4)(2^m - 2^{w_{s-1}}) \right] + \gamma_s (\log 4)(2^m - 2^{w_s}) \\
=\ & H_{s-2,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-2]}) \prod_{j=s-1}^{s} (1 + \gamma_j \log 4) + \gamma_{s-1} (\log 4)(2^m - 2^{w_{s-1}}) \\
& + \gamma_s \gamma_{s-1} (\log 4)^2 (2^m - 2^{w_{s-1}}) + \gamma_s (\log 4)(2^m - 2^{w_s}) \\
\leq\ & \left(\prod_{j=s-1}^{s} (1 + \gamma_j \log 4) \right) \left[(1 + \gamma_{s-2} \log 4) H_{s-3,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-3]}) \right. \\
& \left. + \gamma_{s-2} (\log 4)(2^m - 2^{w_{s-2}}) \right] + \gamma_{s-1} (\log 4)(2^m - 2^{w_{s-1}}) \\
& + \gamma_s \gamma_{s-1} (\log 4)^2 (2^m - 2^{w_{s-1}}) + \gamma_s (\log 4)(2^m - 2^{w_s}) \\
=\ & H_{s-3,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s-3]}) \prod_{j=s-2}^{s} (1 + \gamma_j \log 4) + \gamma_{s-2} (\log 4)(2^m - 2^{w_{s-2}}) \\
& + \gamma_s \gamma_{s-2} (\log 4)^2 (2^m - 2^{w_{s-2}}) + \gamma_{s-1} \gamma_{s-2} (\log 4)^2 (2^m - 2^{w_{s-2}}) \\
& + \gamma_s \gamma_{s-1} \gamma_{s-2} (\log 4)^3 (2^m - 2^{w_{s-2}}) + \gamma_{s-1} (\log 4)(2^m - 2^{w_{s-1}}) \\
& + \gamma_s \gamma_{s-1} (\log 4)^2 (2^m - 2^{w_{s-1}}) + \gamma_s (\log 4)(2^m - 2^{w_s}).
\end{aligned}$$

Repeating this argument inductively, we finally arrive at

$$\begin{aligned} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s]}) &\leq H_{1,N,\boldsymbol{\gamma},\boldsymbol{w}}(z_1)\prod_{j=2}^{s}(1+\gamma_j\log 4) \\ &\quad + \sum_{\emptyset\neq\mathfrak{u}\subseteq\{2:s\}}\gamma_{\mathfrak{u}}(\log 4)^{|\mathfrak{u}|}(2^m-2^{\min_{j\in\mathfrak{u}} w_j}) \\ &\leq H_{1,N,\boldsymbol{\gamma},\boldsymbol{w}}(z_1)\prod_{j=2}^{s}(1+\gamma_j\log 4) \\ &\quad + \sum_{\emptyset\neq\mathfrak{u}\subseteq\{2:s\}}\gamma_{\mathfrak{u}}(\log 4)^{|\mathfrak{u}|}2^m. \qquad (22) \end{aligned}$$

However, recall that we have, by definition,

$$\begin{aligned} H_{1,N,\boldsymbol{\gamma},\boldsymbol{w}}(z_1) &= H_{1,N,\boldsymbol{\gamma},\boldsymbol{w}}(1) = \sum_{k=1}^{N-1}\gamma_1 u_1(k) \\ &= \gamma_1\sum_{k=1}^{N-1}\log\left(\frac{1}{\sin^2(\pi k/2^m)}\right) \\ &= -2\gamma_1\sum_{k=1}^{N-1}\log\left(\sin\left(\frac{\pi k}{N}\right)\right) = \gamma_1(N-m-1)\log 4 \\ &\leq \gamma_1 N\log 4-\gamma_1\log 4\leq\gamma_1 N\log 4, \end{aligned}$$

where we used the identity

$$\prod_{k=1}^{N-1}\left(2\sin\left(\frac{\pi k}{N}\right)\right)=N.$$

Consequently, combining this bound with (22), expanding the expression, and using that $\gamma_{\mathfrak{u}}=\prod_{j\in\mathfrak{u}}\gamma_j$, finally gives

$$\begin{aligned} H_{s,N,\boldsymbol{\gamma},\boldsymbol{w}}(\boldsymbol{z}_{[s]}) &\leq N\gamma_1(\log 4)\left(\prod_{j=2}^{s}(1+\gamma_j\log 4)\right)+N\sum_{\emptyset\neq\mathfrak{u}\subseteq\{2:s\}}\gamma_{\mathfrak{u}}(\log 4)^{|\mathfrak{u}|} \\ &= N\left(\gamma_1(\log 4)\prod_{j=2}^{s}(1+\gamma_j\log 4)+(-1)+\prod_{j=2}^{s}(1+\gamma_j\log 4)\right) \\ &= N\prod_{j=1}^{s}(1+\gamma_j\log 4)-N=N\left[-1+\prod_{j=1}^{s}(1+\gamma_j\log 4)\right], \end{aligned}$$

which is the claim. □

We are now able to show the main result regarding the reduced component-by-component digit-by-digit construction.

Theorem 3.8. *Let $N = 2^m$, with $m \in \mathbb{N}$, let $\boldsymbol{w} = (w_j)_{j\geq 1}$ be a sequence in $\mathbb{N}_0$ with $0 = w_1 \leq w_2 \leq \cdots$, and let $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ be positive product weights. Furthermore, denote by $\boldsymbol{z} = (Y_1 z_1, \dots, Y_d z_d)$ the corresponding generating vector constructed by Algorithm 1. Then the following estimate holds,*

$$T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w}) \leq \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} \frac{2(6 \log N)^{|\mathfrak{u}|+1}}{2^{m - \max_{j \in \mathfrak{u}} w_j}}. \tag{23}$$

Proof. Using the result in Theorem 3.3, the structure of product weights, and the estimate for $H_{d^*,N,\boldsymbol{\gamma},\boldsymbol{w}}$ in Theorem 3.7, we obtain

$$\begin{aligned}
T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w}) &\leq \frac{2(1 + \log N)}{N} \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} 2^{w_{j_{|\mathfrak{u}|}}} (6 \log N)^{|\mathfrak{u}|} \\
&\quad + \frac{1}{N} N \left(-1 + \prod_{j=1}^{d^*} (1 + \gamma_j \log 4) \right) - \prod_{j=1}^{d^*} (1 + \gamma_j \log 4) + 1 \\
&= \frac{2(1 + \log N)}{N} \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} 2^{w_{j_{|\mathfrak{u}|}}} (6 \log N)^{|\mathfrak{u}|} \\
&\leq \sum_{\emptyset \neq \mathfrak{u} \subseteq [d^*]} \gamma_{\mathfrak{u}} \frac{2(6 \log N)^{|\mathfrak{u}|+1}}{2^{m - \max_{j \in \mathfrak{u}} w_j}},
\end{aligned}$$

where we used that for a non-empty set $\mathfrak{u} \subseteq [d^*]$ we have $w_{j_{|\mathfrak{u}|}} = \max_{j \in \mathfrak{u}} w_j$. This is the claimed result. □

We now have the following corollary.

Corollary 3.9. *Let $N = 2^m$, with $m \in \mathbb{N}$, let $\boldsymbol{w} = (w_j)_{j\geq 1}$ be a sequence in $\mathbb{N}_0$ with $0 = w_1 \leq w_2 \leq \cdots$, and let $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ be positive product weights satisfying*

$$\sum_{j \geq 1} \gamma_j 2^{w_j} < \infty.$$

Furthermore, denote by $\boldsymbol{z} = (Y_1 z_1, \dots, Y_d z_d)$ the generating vector constructed by Algorithm 1 run for the weights $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$. Then, for any $\delta > 0$ and each $\alpha > 1$, the generating vector $\boldsymbol{z}$ satisfies

$$e_{N,d,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{z}) \leq \frac{1}{N^\alpha} \big(C_1(\boldsymbol{\gamma}^\alpha) + C_2(\boldsymbol{\gamma}, \delta) N^{\alpha\delta} \big)$$

with weight sequence $\boldsymbol{\gamma}^\alpha = (\gamma_j^\alpha)_{j\geq 1}$ and positive constants $C_1(\boldsymbol{\gamma}^\alpha)$ and $C_2(\boldsymbol{\gamma}, \delta)$, which are independent of d and N. Additionally, if Algorithm 1 is run for weights $\boldsymbol{\gamma}^{1/\alpha} = (\gamma_j^{1/\alpha})_{j\geq 1}$ with $\alpha > 1$, satisfying

$$\sum_{j\geq 1} \gamma_j^{1/\alpha} 2^{w_j} < \infty,$$

then, for any $\delta > 0$, the resulting generating vector $\widetilde{\boldsymbol{z}} = (Y_1\widetilde{z}_1, \dots, Y_d\widetilde{z}_d)$ satisfies the error bound

$$e_{N,d,\alpha,\boldsymbol{\gamma}}(\widetilde{\boldsymbol{z}}) \leq \frac{1}{N^\alpha}\left(F_1(\boldsymbol{\gamma}) + F_2(\boldsymbol{\gamma}^{1/\alpha}, \delta) N^{\alpha\delta}\right),$$

with positive constants $F_1(\boldsymbol{\gamma})$ and $F_2(\boldsymbol{\gamma}^{1/\alpha}, \delta)$, which are independent of d and N.

Proof. Taking into account that we have product weights $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$, the estimate in Proposition 2.2 yields for the worst-case error $e_{N,d,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{z})$,

$$\begin{aligned} e_{N,d,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{z}) &\leq \frac{1}{N^\alpha} \sum_{\emptyset\neq\mathfrak{u}\subseteq[d]} \left(\prod_{j\in\mathfrak{u}} \left(\gamma_j^\alpha\, 4\zeta(\alpha)\right)\right) 2^{\alpha \max_{j\in\mathfrak{u}} w_j} + T_{\alpha,\boldsymbol{\gamma}^\alpha}(N, \boldsymbol{z}, \boldsymbol{w}) \\ &\leq \frac{1}{N^\alpha} \prod_{j=1}^d \left(1 + \gamma_j^\alpha\, 4\zeta(\alpha) 2^{\alpha w_j}\right) + T_{\alpha,\boldsymbol{\gamma}^\alpha}(N, \boldsymbol{z}, \boldsymbol{w}), \end{aligned}$$

where we used that for $\emptyset \neq \mathfrak{u} \subseteq [d]$ we have $\max_{j\in\mathfrak{u}} w_j \leq \sum_{j\in\mathfrak{u}} w_j$. Since $\alpha > 1$, we can use an inequality sometimes called Jensen's inequality, which states that

$$\sum_{i=1}^n a_i \leq \left(\sum_{i=1}^n a_i^p\right)^{1/p}$$

for non-negative $a_1, \dots, a_n$ and $0 \leq p \leq 1$, and thus we have

$$\begin{aligned} T_{\alpha,\boldsymbol{\gamma}^\alpha}(N, \boldsymbol{z}, \boldsymbol{w}) &= \sum_{\boldsymbol{0}\neq\boldsymbol{\ell}\in M_{N,d,\boldsymbol{w}}} \frac{\delta_N(\boldsymbol{\ell}\cdot\boldsymbol{z})}{r_{\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{\ell})} \\ &\leq \left(\sum_{\boldsymbol{0}\neq\boldsymbol{\ell}\in M_{N,d,\boldsymbol{w}}} \frac{\delta_N(\boldsymbol{\ell}\cdot\boldsymbol{z})}{r_{1,\boldsymbol{\gamma}}(\boldsymbol{\ell})}\right)^\alpha = \left(T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w})\right)^\alpha, \end{aligned}$$

and by Theorem 3.8 we know that the $\boldsymbol{z}$ generated by Algorithm 1 is such that

$$T_{\boldsymbol{\gamma}}(N, \boldsymbol{z}, \boldsymbol{w}) \leq \sum_{\emptyset\neq\mathfrak{u}\subseteq[d^*]} \gamma_{\mathfrak{u}} \frac{2(6\log N)^{|\mathfrak{u}|+1}}{2^{m-\max_{j\in\mathfrak{u}} w_j}}.$$

From this, we deduce, using $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$, that we have

$$\begin{aligned}
2^m T_{\boldsymbol{\gamma}}(N,\boldsymbol{z},\boldsymbol{w}) &\leq 2(6\log N) \sum_{\emptyset\neq\mathfrak{u}\subseteq[d^*]} \gamma_{\mathfrak{u}} (6\log N)^{|\mathfrak{u}|} 2^{\max_{j\in\mathfrak{u}} w_j} \\
&\leq 2(6\log N) \sum_{\emptyset\neq\mathfrak{u}\subseteq[d^*]} \left(\prod_{j\in\mathfrak{u}} (\gamma_j\, 6\log N) \right) 2^{\sum_{j\in\mathfrak{u}} w_j} \\
&\leq \big(1+2(6\log N)\big) \prod_{j=1}^{d^*} \big(1+\gamma_j\,(6\log N)\,2^{w_j}\big) \\
&\leq \widetilde{C}(\delta/2) 2^{m\delta/2} \prod_{j=1}^{\infty} \big(1+\gamma_j\,(6\log N)\,2^{w_j}\big), \qquad (24)
\end{aligned}$$

for arbitrary $\delta > 0$, where $\widetilde{C}(\delta/2)$ is a constant depending only on δ, and where we again used for $\emptyset \neq \mathfrak{u} \subseteq [d^*]$ that we have $\max_{j\in\mathfrak{u}} w_j \leq \sum_{j\in\mathfrak{u}} w_j$. Due to the given condition imposed on the weights, i.e., $\sum_{j\geq 1} \gamma_j 2^{w_j} < \infty$, we can use the result in [11, Lemma 3] to see that the product in (24) can be bounded by $\hat{C}(\boldsymbol{\gamma},\delta)2^{m\delta}$, where $\hat{C}(\boldsymbol{\gamma},\delta)$ may depend on the weights $\boldsymbol{\gamma}$ and δ, but is independent of the dimension. This yields

$$\big(T_{\boldsymbol{\gamma}}(N,\boldsymbol{z},\boldsymbol{w})\big)^{\alpha} \leq \frac{1}{N^{\alpha}} \left(\widetilde{C}(\delta/2)\right)^{\alpha} \left(\hat{C}(\boldsymbol{\gamma},\delta)\right)^{\alpha} N^{\alpha\delta}.$$

Setting then

$$C_1(\boldsymbol{\gamma}^{\alpha}) = \prod_{j=1}^{\infty} \big(1+\gamma_j^{\alpha} 4\zeta(\alpha) 2^{\alpha w_j}\big),$$

which is finite due to our assumption on the weights, and

$$C_2(\boldsymbol{\gamma},\delta) = \left(\widetilde{C}(\delta/2)\right)^{\alpha} \left(\hat{C}(\boldsymbol{\gamma},\delta)\right)^{\alpha}$$

we get the claimed estimate.

Similarly, for weights $\boldsymbol{\gamma}^{1/\alpha}$ and due to the condition imposed on the weights, i.e., $\sum_{j\geq 1} \gamma_j^{1/\alpha} 2^{w_j} < \infty$, and by using the result in [11, Lemma 3] we have

$$\big(T_{\boldsymbol{\gamma}^{1/\alpha}}(N,\widetilde{\boldsymbol{z}},\boldsymbol{w})\big)^{\alpha} \leq \frac{1}{N^{\alpha}} \left(\widetilde{C}(\delta/2)\right)^{\alpha} \left(\hat{C}(\boldsymbol{\gamma}^{1/\alpha},\delta)\right)^{\alpha} N^{\alpha\delta}.$$

Then setting

$$F_1(\boldsymbol{\gamma}) = \prod_{j=1}^{\infty}\left(1+\gamma_j 4\zeta(\alpha)2^{w_j}\right)$$

and

$$F_2(\boldsymbol{\gamma}^{1/\alpha},\delta) = \left(\widetilde{C}(\delta/2)\right)^{\alpha}\left(\hat{C}(\boldsymbol{\gamma}^{1/\alpha},\delta)\right)^{\alpha}$$

we get the claimed estimate. □

The result in Corollary 3.9 involves two cases regarding the worst-case error behavior of the generating vectors constructed by Algorithm 1. To be more precise, we can run the algorithm with weights $\boldsymbol{\gamma}$, and hence it does not depend on the parameter α, then the algorithm produces a generating vector for which bounds on the worst-case errors in the spaces $E^{\alpha}_{d,\boldsymbol{\gamma}^{\alpha}}$ hold simultaneously for all $\alpha > 1$. On the other hand, when we run Algorithm 1 with weights $\gamma^{1/\alpha}$, we have a dependence on the parameter α, and the algorithm yields error bounds for the worst-case error in the spaces $E^{\alpha}_{d,\boldsymbol{\gamma}}$.

4. Fast implementation of the construction scheme

In this section, we discuss the efficient implementation of the introduced reduced CBC-DBD algorithm and analyze its complexity. To be more precise, we will discuss an implementation of the algorithms for the case of product weights, in which we can compute the values of the quality function recursively with respect to the indices of the components. This facilitates a fast implementation.

We recall the definition of the quality function in Definition 3.5, and we see that for product weights $h_{d,v,m,\boldsymbol{\gamma},\boldsymbol{w}}$ can be rewritten as follows. Let $x \in \mathbb{N}$ be an odd integer, let $m, d \in \mathbb{N}$, and let $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ be positive product weights. For $1 \leq s \leq d$, $1 \leq v \leq m - w_s$, and odd integers $z_1, \ldots, z_{s-1}$ the quality function $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}$ reads

$$h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}(x) := \sum_{t=v}^{m-w_s} \frac{1}{2^{t-v}} \sum_{\ell=0}^{2^{w_s}-1} \sum_{\substack{k=\ell 2^t+1 \\ k\equiv 1 \pmod 2}}^{(\ell+1)2^t-1} \left(1+\gamma_s \log\left(\frac{1}{\sin^2(\pi k x/2^v)}\right)\right)$$
$$\times \prod_{j=1}^{s-1}\left[1+\gamma_j \log\left(\frac{1}{\sin^2\left(\pi z_j \frac{k}{2^t}\,\frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right)\right],$$

where the components $z_1, \ldots, z_{s-1}$ have been determined in the previous steps of the algorithm. We are interested in the cost of a single evaluation of the function $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}$, which is crucial for the total cost of Algorithm 1, hence we will discuss an efficient evaluation procedure in the following paragraph.

For integers $t\in\{2,\dots,m-w_s\}$ (note that we always have $v\geq 2$ in Algorithm 1, so we do not need to consider the case $t=1$) and odd

$$k\in\{\ell 2^t+1,\dots,(\ell+1)2^t-1\}\quad\text{for}\ \ \ell\in\{0,\dots,2^{w_s}-1\},$$

we define $r(s,t,k)$ as

$$r(s,t,k):=\prod_{j=1}^{s}\left(1+\gamma_j\log\left(\frac{1}{\sin^2\left(\pi z_j\frac{k}{2^t}\frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right)\right),$$

and observe that for the evaluation of $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}(x)$ we can compute and store the terms $r(s-1,t,k)$ for suitable values of t since they are independent of v and x. This way we can rewrite $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}(x)$ as

$$\sum_{t=v}^{m-w_s}\frac{1}{2^{t-v}}\sum_{\ell=0}^{2^{w_s}-1}\sum_{\substack{k=\ell 2^t+1\\ k\equiv 1\ (\mathrm{mod}\ 2)}}^{(\ell+1)2^t-1} r(s-1,t-w_{s-1}+w_s,k)\left(1+\gamma_s\log\left(\frac{1}{\sin^2(\pi kx/2^v)}\right)\right). \tag{25}$$

Note that if $v\leq t\leq m-w_s$, then

$$0\leq v-w_{s-1}+w_s\leq t-w_{s-1}+w_s\leq m-w_{s-1}.$$

In Algorithm 1, after having determined z_s, the values of $r(s,v,k)$ for odd integers $k\in\{\ell 2^v+1,\dots,(\ell+1)2^v-1\}$ with $\ell\in\{0,\dots,2^{w_s}-1\}$ are computed via the recurrence relation

$$r(s,t,k)=r(s-1,t-w_{s-1}+w_s,k)\left(1+\gamma_s\log\left(\frac{1}{\sin^2(\pi kz_s/2^t)}\right)\right).$$

Now, for an algorithmic implementation, we introduce the vector

$$\boldsymbol{u}=\big(u(1),\dots,u(2^m-1)\big)\in\mathbb{R}^{2^m-1},$$

whose components, for the current $s\in\{1,\dots,d^*\}$, are given by

$$u(k\,2^{m-w_s-t})=\prod_{j=1}^{s}\left(1+\gamma_j\log\left(\frac{1}{\sin^2\left(\pi z_j\frac{k}{2^t}\frac{2^{m-w_s}}{2^{m-w_j}}\right)}\right)\right)$$

for each $t\in\{1,\dots,m-w_s\}$ and corresponding odd index

$$k\in\{\ell 2^t+1,\dots,(\ell+1)2^t-1\}\quad\text{for}\quad \ell\in\{0,\dots,2^{w_s}-1\}.$$

Note that $k\,2^{m-w_s-t}$ runs through the whole range $\{1,\dots,2^m-1\}$ when t, k, and ℓ are chosen as stated. Furthermore, note that the quantity $u(k\,2^{m-w_s-t})$ corresponds to $r(s,t,k)$ for $t \geq 2$, and that for the evaluation of $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}$ we do not require the values of $r(s,t,k)$ for $t = 2,\dots,v-1$. Combining these findings leads to the following fast implementation of Algorithm 1.

Algorithm 2 (Fast reduced component-by-component digit-by-digit algorithm)**.**

Input: Integers $m,d \in \mathbb{N}$, positive product weights $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$, and integer reduction indices $(w_j)_{j\geq 1}$ with $0 = w_1 \leq w_2 \leq \cdots$.

```
for $t = 2$ to $m = m - w_1$ do
  for $k = 1$ to $2^t - 1$ in steps of $2$ do
    $u(k2^{m-w_1-t}) = \left(1 + \gamma_1 \log\left(\frac{1}{\sin^2(\pi k/2^t)}\right)\right)$
  end for
end for

Let $d^*$ be the largest integer such that $w_{d^*} < m$.
Set $z_{1,1} = \cdots = z_{d,1} = 1$.

for $s = 2$ to $\min\{d, d^*\}$ do
  for $v = 2$ to $m - w_s$ do
    $z^* = \operatorname*{argmin}_{z\in\{0,1\}} h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}(z_{s,v-1} + z\,2^{v-1})$ with $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}$ evaluated using (25)
    $z_{s,v} = z_{s,v-1} + z^* 2^{v-1}$
    for $\ell = 0$ to $2^{w_s} - 1$ do
      for $k = \ell 2^v + 1$ to $(\ell+1)2^v - 1$ in steps of $2$ do
        $u(k2^{m-w_s-v}) = u(k2^{m-w_{s-1}-v})\left(1 + \gamma_s \log\left(\frac{1}{\sin^2(\pi k z_{s,v}/2^v)}\right)\right)$
      end for
    end for
  end for
  Set $z_s := z_{s,m-w_s}$.
end for
if $d > d^*$ then
  set $z_{d^*+1} = \cdots = z_d = 1$
end if

Set $\boldsymbol{z} = (Y_1 z_1, \dots, Y_d z_d)$.
```

Return: Generating vector $\boldsymbol{z} = (Y_1 z_1, \dots, Y_d z_d)$ for $N = 2^m$.

The computational complexity of Algorithm 2 is then summarized in the following theorem. Note that the computational cost is of order

$$\mathcal{O}(\min\{d, d^*\} N \log N),$$

which is the same order of magnitude as for the reduced CBC construction in [4]. For quickly decreasing weights γ_j, the reduction indices w_j can be chosen to increase quickly without losing significantly much quality of the integration rule, and in such cases d^* can be considerably smaller than d. A detailed discussion on chosing reduction indices can be found in [4].

Theorem 4.1. *Let $N = 2^m$ with $m \in \mathbb{N}$, let $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ be a given sequence of positive weights, and let integer reduction indices w_j with $0 = w_1 \leq w_2 \leq \cdots$ be given. Moreover, denote by d^* the largest integer such that $w_{d^*} < m$. Then Algorithm 2 constructs a generating vector $\boldsymbol{z} = (z_1, \ldots, z_d)$ using*

$$\mathcal{O}\left(\min\{d, d^*\}\, 2^m + \sum_{s=1}^{\min\{d,d^*\}} (m - w_s) 2^m\right)$$

operations and requiring $\mathcal{O}(2^m)$ memory.

Proof. Due to the relation in (25), the cost of evaluating $h_{s,v,m,\boldsymbol{\gamma},\boldsymbol{w}}(x)$ for one fixed v can be reduced to $\mathcal{O}(\sum_{t=v}^{m-w_s} 2^{w_s+t-1})$ operations. Moreover, updating the values of u needs $\mathcal{O}(2^{w_s+v-1})$ operations, so the computational cost for one fixed v in the inner loop over $v = 2, \ldots, m - w_s$ of Algorithm 2 is of order

$$\mathcal{O}\left(\sum_{t=v}^{m-w_s} 2^{w_s+t-1}\right).$$

Thus, the number of calculations in the inner loop over $v = 2, \ldots, m - w_s$ of Algorithm 2 is of order

$$\begin{aligned}\mathcal{O}\left(\sum_{v=2}^{m-w_s} 2 \sum_{t=v}^{m-w_s} 2^{w_s+t-1}\right) &= \mathcal{O}\left(\sum_{v=2}^{m-w_s} \sum_{t=v}^{m-w_s} 2^{w_s+t}\right)\\ &= \mathcal{O}\big((m - w_s)\, 2^{m+1} - 2(2^{m+1} - 2^{w_s+1})\big)\\ &= \mathcal{O}\big((m - w_s)\, 2^m\big).\end{aligned}$$

Hence, the outer loop over $s = 1, \dots, \min\{d, d^*\}$, which is the main cost of Algorithm 2, can be executed in

$$\mathcal{O}\left(\sum_{s=1}^{\min\{d,d^*\}} (m - w_s)\, 2^m \right)$$

operations. Furthermore, we observe that initialization and updating of the vector $\boldsymbol{u} \in \mathbb{R}^{2^m-1}$ can both be executed in $\mathcal{O}(\min\{d, d^*\}2^m)$ operations. Additionally, storing the vector $\boldsymbol{u}$ requires $\mathcal{O}(2^m)$ of memory. □

The run-time of Algorithm 2 can also be reduced further by precomputing and storing the 2^m values

$$\log\left(\frac{1}{\sin^2(\pi k/2^m)}\right) \quad \text{for} \quad k = 1, \dots, 2^m - 1.$$

The derivation leading to the fast implementation in Algorithm 2 is using arguments that were also used in [8], where a non-reduced component-by-component digit-by-digit construction for lattice rules in weighted Korobov spaces has been studied.

5. Numerical results

In this section, we illustrate the error convergence behaviour of the lattice rules constructed by the reduced fast CBC-DBD algorithm and display the computational complexity of the construction using numerical experiments. As in the previous section, we consider lattice rules in the weighted Korobov space $E^{\alpha}_{s,\boldsymbol{\gamma}}$ of smoothness $\alpha > 1$, and we assume product weights $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$ given in terms of positive reals $(\gamma_j)_{j\geq 1}$. For $\boldsymbol{z} = (z_1, \dots, z_d)$, the worst-case error is then given by

$$e_{N,d,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z}) = -1 + \frac{1}{2^m} \sum_{k=0}^{2^m-1} \prod_{j=1}^{d} \left(1 + \gamma_j \sum_{\ell\in\mathbb{Z}_*} \frac{e^{2\pi i k \ell z_j/2^m}}{|\ell|^{\alpha}} \right).$$

To demonstrate the performance of the algorithm, we compare the worst-case errors of the constructed lattice rules as well as the algorithm's computation times with the corresponding quantities obtained by the non-reduced component--by-component digit-by-digit algorithm, see [8]. Both constructions deliver lattice rules for the space $E^{\alpha}_{s,\boldsymbol{\gamma}}$ consisting of 2^m cubature points. The different algorithms have been implemented in double-precision and arbitrary-precision floating-point arithmetic, with the latter provided by the multi-precision Python library mpmath.

5.1. Error convergence behaviour

Let $m, d \in \mathbb{N}$, $\alpha > 1$, a sequence of positive weights $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$, and reduction indices $(w_j)_{j\geq 1}$ with $0 = w_1 \leq w_2 \leq \cdots$ be given. In particular, we consider the convergence rate of the worst-case error $e_{2^m,d,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{z})$ for different weight sequences $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ of the form $\gamma_j = c^j$ with $c \in (0,1)$ or $\gamma_j = 1/j^q$ with $q > 1$ and reduction indices of the form $w_j = \lfloor p \log_2 j \rfloor$ with $p > 0$. We display the computational results for dimension $d = 100$ for different sequences of product weights, different values of m, and reduction indices w_j. We stress that the almost optimal error rates of $\mathcal{O}(N^{-\alpha+\delta})$, as guaranteed by Corollary 3.9, may not always be visible for the weights, reduction indices, and ranges of N considered in our numerical experiments. The graphs shown are therefore to be understood as illustrations of the pre-asymptotic behaviour of the worst-case error. Figures 1 and 2 show numerical results using different choices of weights $(\gamma_j)_{j\geq 1}$ for reduction indices of the form $w_j = \lfloor 2 \log_2 j \rfloor$ and $w_j = \lfloor \frac{7}{2} \log_2 j \rfloor$, respectively. The generating vectors $\boldsymbol{z}$ are obtained by the reduced fast CBC-DBD algorithm and the non-reduced fast CBC-DBD algorithm, respectively.

The results in Figures 1 and 2 show that the reduced fast CBC-DBD algorithm constructs generating vectors of good lattice rules which have worst-case errors that are essentially comparable to those of lattice rules obtained by the non-reduced fast CBC-DBD algorithm. We observe asymptotic error rates for both algorithms considered. Only in Part (a) of Figures 1 and 2, respectively, the errors of the lattice rules obtained by the reduced construction seem to be significantly higher than those of the non-reduced case. This error behaviour can be explained as follows. As illustrated in Figures 1 and 2, throughout we use $b = 2$ and $\alpha = 2$. Corollary 3.9 assures independence of the dimension d whenever the chosen weights γ_j satisfy

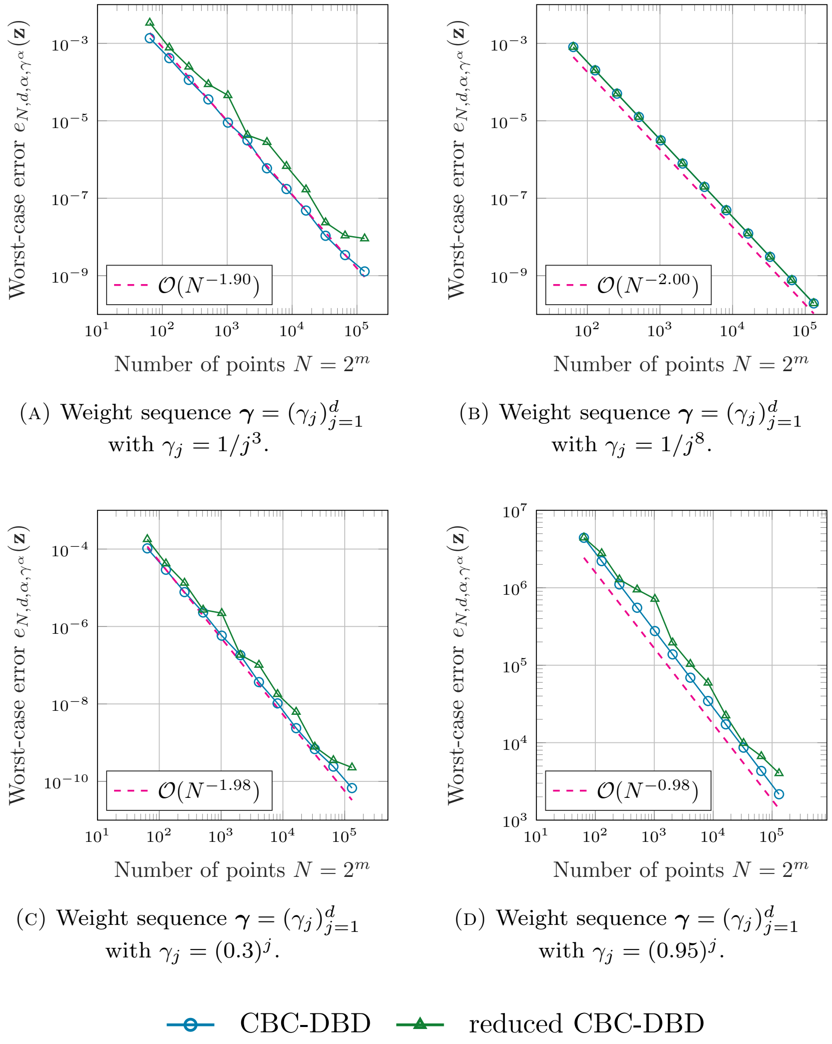
$$\sum_{j\geq 1} \gamma_j b^{w_j} = \sum_{j\geq 1} \gamma_j 2^{w_j} < \infty. \tag{26}$$

However, the sequence of weights $\gamma_j = 1/j^3$ does not always satisfy Condition (26) if we choose the reduction indices as $w_j = \lfloor p \log_2 j \rfloor$, since

$$\sum_{j\geq 1} \gamma_j 2^{w_j} = \sum_{j\geq 1} j^{-3} 2^{\lfloor p \log_2 j \rfloor} \geq \sum_{j\geq 1} j^{-3} 2^{p \log_2 j - 1} \geq \frac{1}{2} \sum_{j\geq 1} j^{p-3}.$$

The latter series only convergence if $3 - p > 1$, and this is not satisfied for our choices of p made in Figures 1 and 2, which is $p = 2$ and $p = 7/2$, respectively. Thus, this gives rise to the difference in the errors obtained by the reduced CBC-DBD and non-reduced CBC-DBD algorithms. This phenomenon is to be expected and shows that the reduction indices and the weights must be balanced carefully, and that in general the reduced algorithm works better for situations where the weights γ_j decay fast.

Error convergence in the space $E_{d,\boldsymbol{\gamma}}^{\alpha}$ with $d\!=\!100, \alpha\!=\!2,$ $w_j = \lfloor 2\log_2 j \rfloor$.



(A) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^d$ with $\gamma_j = 1/j^3$.

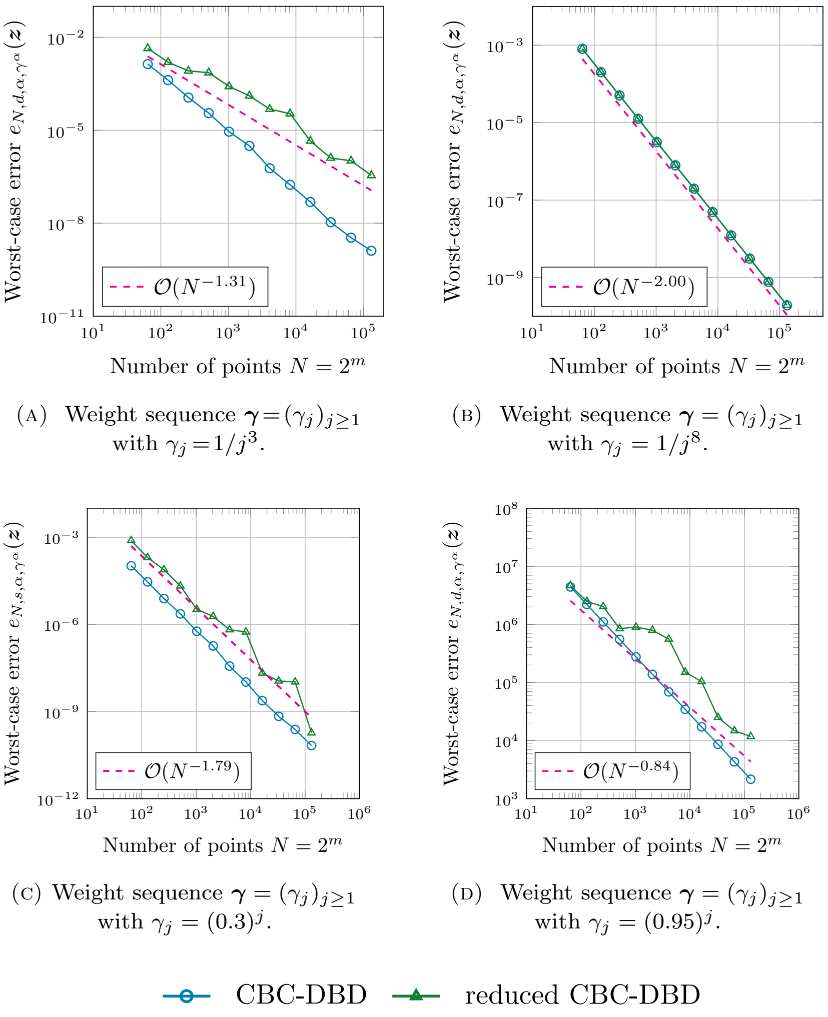
(B) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^d$ with $\gamma_j = 1/j^8$.

(C) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^d$ with $\gamma_j = (0.3)^j$.

(D) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^d$ with $\gamma_j = (0.95)^j$.

FIGURE 1. Convergence of the worst-case errors $e_{N,d,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{z})$ in the weighted space $E_{d,\boldsymbol{\gamma}}^{\alpha}$ for smoothness parameter $\alpha = 2$ with dimension $d = 100$ and reduction indices $w_j = \lfloor 2\log_2 j \rfloor$. The generating vector $\boldsymbol{z}$ is constructed via the reduced CBC-DBD construction and the non-reduced CBC-DBD construction for $N = 2^m$, respectively.

Error convergence in the space $E^{\alpha}_{d,\boldsymbol{\gamma}}$ with $d\!=\!100, \alpha\!=\!2,$ $w_j\!=\!\lfloor\frac{7}{2}\log_2 j\rfloor$.



(A) Weight sequence $\boldsymbol{\gamma}\!=\!(\gamma_j)_{j\geq 1}$ with $\gamma_j\!=\!1/j^3$.

(B) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ with $\gamma_j = 1/j^8$.

(C) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ with $\gamma_j = (0.3)^j$.

(D) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ with $\gamma_j = (0.95)^j$.

FIGURE 2. Convergence of the worst-case errors $e_{N,d,\alpha,\boldsymbol{\gamma}^\alpha}(\boldsymbol{z})$ in the weighted space $E^{\alpha}_{d,\boldsymbol{\gamma}}$ for smoothness parameter $\alpha = 2$ with dimension $d\!=\!100$ and reduction indices $w_j = \lfloor\frac{7}{2}\log_2 j\rfloor$. The generating vector $\boldsymbol{z}$ is constructed via the reduced CBC-DBD construction and the non-reduced CBC-DBD construction for $N\!=\!2^m$, respectively.

5.2. Computational complexity

Here, we illustrate the computational complexity of the reduced fast CBC-DBD construction in Algorithm 2 which was proved in Theorem 4.1. For this purpose, let $N = 2^m$, let the weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ be given by $\gamma_j = (0.95)^j$, and let the reduction indices be given by $w_j = \lfloor \frac{3}{2} \log_2 j \rfloor$ for $j \geq 1$. We measure and compare the computation times of implementations of Algorithm 2 and the non-reduced fast CBC-DBD algorithm for lattice rules (see [8] for details on the implementation of the latter). Note that the chosen weight sequence does not affect the computation times. The timings were performed on an Intel Core i5 CPU with 2.3 GHz using Python 3.6.3.

Table 1. Computation times (in seconds) for constructing the generating vector $\boldsymbol{z}$ of a lattice rule with $N = 2^m$ points in d dimensions using the reduced CBC-DBD algorithm (**bold font**) and the non-reduced CBC-DBD construction (normal font) with $\alpha = 2$, $\gamma_j = (0.95)^j$, and $w_j = \lfloor \frac{3}{2} \log_2 j \rfloor$.

	$d=50$	$d=100$	$d=500$	$d=1000$	$d=2000$
$m=10$	0.077	0.154	0.799	1.571	3.182
	0.017	**0.018**	**0.02**	**0.021**	**0.025**
$m=12$	0.128	0.252	1.224	2.424	4.908
	0.035	**0.046**	**0.052**	**0.054**	**0.057**
$m=14$	0.211	0.415	2.044	4.049	8.256
	0.066	**0.089**	**0.138**	**0.136**	**0.141**
$m=16$	0.43	0.874	4.363	8.796	17.631
	0.103	**0.152**	**0.299**	**0.354**	**0.359**
$m=18$	1.467	2.982	14.924	30.545	59.967
	0.195	**0.272**	**0.577**	**0.761**	**0.904**
$m=20$	7.222	14.538	73.21	147.759	294.616
	0.5	**0.623**	**1.146**	**1.516**	**1.931**

In Table 1, we display the computation times for the construction of the generating vector $\boldsymbol{z}$ via the two considered algorithms, where for the reduction indices we use $w_j = \lfloor \frac{3}{2} \log_2 j \rfloor$. We emphasize that the used algorithms solely construct the generating vector $\boldsymbol{z}$ but do not calculate the worst-case error $e_{N,d,\alpha,\boldsymbol{\gamma}}(\boldsymbol{z})$, which allows for an unbiased comparison between the considered algorithms.

Table 1 illustrates a dramatic difference in the computational cost between the non-reduced fast CBC-DBD construction and the reduced fast CBC-DBD construction. The extent of the speed-up depends on the chosen reduction indices w_j. Note, however, that the reduction indices have to be chosen such that they are balanced with the weights, in order to guarantee useful error convergence.

6. Conclusion

In this paper, we have presented a combination of the CBC-DBD algorithm introduced in [8] and the reduced construction method in [4] for constructing good lattice rules for numerical integration in weighted Korobov spaces. In particular, we have aimed to gain from the reduced construction method to shrink the computational cost as compared to the non-reduced CBC-DBD algorithm. We showed that the reduced CBC-DBD construction with quality measure independent of the smoothness parameter α, similarly to [8] also formulated for product weights, yields lattice rules which admit error convergence rates that are arbitrarily close to the optimal convergence order. We remark that there has recently been considerable interest in finding algorithms that guarantee some degree of universality with respect to the smoothness parameter and/or the weights (see, e.g., [3]), and also our result can be seen as a step in this direction. Furthermore, the errors can be bounded independently of the dimension if the weights satisfy suitable summability conditions. In addition to these theoretical results, we have derived a fast implementation of the considered algorithm. Numerical tests confirm our main findings.

Acknowledgement. The authors would like to thank an anonymous referee for their valuable suggestions that helped to improve the presentation of the results.

P. Kritzer 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 authors have applied a CC BY public copyright licence to any author accepted manuscript version arising from this submission. O. Osisiogu is partially supported by the Austrian Science Fund, Project F5506, which is part of the Special Research Program "Quasi-Monte Carlo Methods: Theory and Applications".

REFERENCES

[1] COOLS, R.—KUO, F. Y.—NUYENS, D.—SURYANARAYANA, G.: *Tent-transformed lattice rules for integration and approximation of multivariate non-periodic functions*, J. Complexity **36** (2016), 166–181.

[2] DICK, J.: *On the convergence rate of the component-by-component construction of good lattice rules*, J. Complexity **20** (2004), 493–522.

[3] DICK, J.—GODA, T.: *Stability of lattice rules and polynomial lattice rules constructed by the component-by-component algorithm*, J. Comput. Appl. Math. **382** (2021), 113062.

[4] DICK, J.—KRITZER, P.—LEOBACHER, G.—PILLICHSHAMMER, F.: *A reduced fast component-by-component construction of lattice points for integration in weighted spaces with fast decreasing weights*, J. Comput. Appl. Math. **276** (2015), 1–15.

[5] DICK, J.—KRITZER, P.—PILLICHSHAMMER, F.: *Lattice Rules*. Springer, Cham, 2022.

[6] DICK, J.—KUO, F. Y.—SLOAN, I. H.: *High-dimensional integration—the quasi-Monte Carlo way*, Acta Numer. **22** (2013), 133–288.

[7] DICK, J.—NUYENS, D.—PILLICHSHAMMER, F.: *Lattice rules for nonperiodic smooth integrands*, Numer. Math. **126** (2014), 259–291.

[8] EBERT, A.—KRITZER, P.—NUYENS, D.—OSISIOGU, O.: *Digit-by-digit and component-by-component constructions of lattice rules for periodic functions with unknown smoothness*, J. Complexity **66** (2021) 101555.

[9] GODA, T.—SUZUKI, K.—YOSHIKI, T.: *Lattice rules in non-periodic subspaces of Sobolev spaces*, Numer. Math. **141** (2019), 399–427.

[10] HICKERNELL F. J.: *A generalized discrepancy and quadrature error bound*, Math. Comp. **67** (1998), 299–322.

[11] HICKERNELL, F. J.—NIEDERREITER, H.: *The existence of good extensible rank-*1 *lattices*, J. Complexity **19** (2003), 286–300.

[12] HLAWKA, E.: *Zur angenäherten Berechnung mehrfacher Integrale*, Monatsh. Math. **66** (1962), 140–151.

[13] HUA, L.K.—WANG, Y.: *Applications of Number Theory to Numerical Analysis*. Springer-Verlag, Berlin, 1981.

[14] KOROBOV, N. M.: *Approximate evaluation of repeated integrals*, Dokl. Akad. Nauk SSSR **124** (1959), 1207–1210. In Russian. (English translation of the theorems in [26].)

[15] KOROBOV, N. M.: *Number-theoretic methods in approximate analysis*. Goz. Izdat. Fiz.--Math., 1963. In Russian. (English translation of results on optimal coefficients in [26].)

[16] KOROBOV, N. M.: *On the computation of optimal coefficients*, Dokl. Akad. Nauk SSSR **267** (1982), 289–292. In Russian. (English translation see [17].)

[17] KOROBOV, N. M.: *On the computation of optimal coefficients*, Dokl. Akad. Nauk SSSR **26** (1982), 590–593.

[18] KRITZER, P.: *A note on the CBC-DBD construction of lattice rules with general positive weights*, J. Complexity **76** (2023), 101721.

[19] KUO, F. Y.: *Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces*, J. Complexity **19** (2003), 301–320.

[20] NIEDERREITER, H.: *Random Number Generation and Quasi-Monte Carlo Methods.* SIAM, Philadelphia, 1992.

[21] NOVAK, E.—WOŹNIAKOWSKI, H.: *Tractability of Multivariate Problems. Volume I: Linear Information.* EMS, Zurich, 2008.

[22] NUYENS, D.—COOLS, R.: *Fast algorithms for component-by-component construction of rank-*1 *lattice rules in shift-invariant reproducing kernel Hilbert spaces*, Math. Comp. **75** (2006), 903–920.

[23] SLOAN, I. H.—JOE, S.: *Lattice Methods for Multiple Integration.* Clarendon Press, Oxford, 1994.

[24] SLOAN, I. H.—WOŹNIAKOWSKI, H.: *When are quasi Monte Carlo algorithms efficient for high-dimensional problems?*, J. Complexity **14** (1998), 1–33.

[25] SLOAN, I. H.—WOŹNIAKOWSKI, H.: *Tractability of multivariate integration for weighted Korobov classes*, J. Complexity **17** (2001), 697–721.

[26] STROUD, A. H.: *Approximate Calculation of Multiple Integrals.* Prentice-Hall, Englewood Cliffs, 1971.

Received November 22, 2022
Accepted May 11, 2023

Peter Kritzer
RICAM
Austrian Academy of Sciences
Altenbergerstr. 69
4040 Linz
AUSTRIA
E-mail: peter.kritzer@ricam.oeaw.ac.at

Onyekachi Osisiogu
Department of Applied Mathematics
Illinois Institute of Technology
10 West 35th Street
Chicago, IL 60616
USA
E-mail: oosisiogu@iit.edu