On the Computational Complexity of Linear Discrepancy

Lily Li
Department of Computer Science, University of Toronto, Canada
xinyuan@cs.toronto.edu

Aleksandar Nikolov
Department of Computer Science, University of Toronto, Canada
anikolov@cs.toronto.edu

──── **Abstract** ────

Many problems in computer science and applied mathematics require rounding a vector $\mathbf{w}$ of fractional values lying in the interval $[0, 1]$ to a binary vector $\mathbf{x}$ so that, for a given matrix $\mathbf{A}$, $\mathbf{Ax}$ is as close to $\mathbf{Aw}$ as possible. For example, this problem arises in LP rounding algorithms used to approximate NP-hard optimization problems and in the design of uniformly distributed point sets for numerical integration. For a given matrix $\mathbf{A}$, the worst-case error over all choices of $\mathbf{w}$ incurred by the best possible rounding is measured by the linear discrepancy of $\mathbf{A}$, a quantity studied in discrepancy theory, and introduced by Lovasz, Spencer, and Vesztergombi (EJC, 1986).

We initiate the study of the computational complexity of linear discrepancy. Our investigation proceeds in two directions: (1) proving hardness results and (2) finding both exact and approximate algorithms to evaluate the linear discrepancy of certain matrices. For (1), we show that linear discrepancy is NP-hard. Thus we do not expect to find an efficient exact algorithm for the general case. Restricting our attention to matrices with a constant number of rows, we present a poly-time exact algorithm for matrices consisting of a single row and matrices with a constant number of rows and entries of bounded magnitude. We also present an exponential-time approximation algorithm for general matrices, and an algorithm that approximates linear discrepancy to within an exponential factor.

1 Notation

Below bold-face capital letters such as $\mathbf{A}$ denote matrices; $A_{i,j}$ is the entry in the ith row and j^{th} column of the matrix $\mathbf{A}$. Bold-face lowercase letters such as $\mathbf{x}$ denote vectors; x_i is the ith entry of the vector $\mathbf{x}$. We denote the all ones vector by $\mathbb{1}$. Whenever we write $i \in [n]$, we mean $i \in \{1, 2, ..., n\}$. For simplicity of exposition, assume that $\mathbf{A} \in \mathbb{Q}^{m \times n}$ in the following sections. We will explicitly indicate other uses of $\mathbf{A}$.

2 Introduction

A number of questions in mathematics and computer science can be reduced to the following basic rounding question: given a vector $\mathbf{w} \in [0, 1]^n$, and an $m \times n$ matrix $\mathbf{A}$, find an integer vector $\mathbf{x} \in [0, 1]^n$ such that $\mathbf{Ax}$ is as close as possible to $\mathbf{Aw}$. For example, many NP-hard optimization problems can be modeled as an integer program

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \in \{0,1\}^n \end{aligned}$$

This integer program can be relaxed to a linear program by replacing the integer variables $\mathbf{x} \in \{0,1\}^n$ with real-valued variables $\mathbf{w} \in [0,1]^n$. A powerful method in approximation algorithms is to solve this linear programming relaxation to get an optimal $\mathbf{w}$, and then round $\mathbf{w}$ to an integer solution $\mathbf{x}$ which is feasible (i.e., $\mathbf{Ax} \geq \mathbf{b}$), and has objective value not much bigger than $\mathbf{c}^\top \mathbf{w}$. Often, a useful intermediate step is to guarantee that $\mathbf{x}$ is approximately optimal, i.e., that the coordinates of $\mathbf{b} - \mathbf{Ax}$ are bounded from above. This approximately feasible solution can then, hopefully, be turned into a truly feasible one with a small loss in the objective value. This method was used, for example, by Rothvoss [18], and Rothvoss and Hoberg [12] to give the best known approximation algorithm for the bin packing problem.

Another example is provided by the problem of constructing uniformly distributed points, or, more generally, points that are well-distributed with respect to some measure. Variants of this problem date back to work by Weyl, van der Corput, van Aardenne-Ehrenfest, and Roth, and have important applications to such fields as numerical integration; see the book of Matoušek [15] for references and an introduction to the area. In the classical setting, the problem is to find, for any positive integer n, a set of n points P in $[0,1]^d$, so as to minimize the quantity

$$\sup_{R \in \mathcal{R}_d} ||R \cap P| - n\lambda_d(R)|,$$

where $\mathcal{R}_d$ is the set of all axis-aligned boxes contained in $[0,1]^d$, and λ_d is the Lebesgue measure on $\mathbb{R}^d$. The quantity above is known as the (unnormalized) discrepancy of P. Note that if we sample a random point uniformly from P, then it would land in R with probability $\frac{|R \cap P|}{n}$; on the other hand, if we sample a random point uniformly from $[0,1]^d$, then it would land in R with probability $\lambda_d(R)$. The problem of minimizing the discrepancy of P is then equivalent to finding a distribution that is uniform over n points that "looks the same" as the continuous uniform distribution to all boxes R.

The discrepancy minimization problem can be modeled by the rounding problem with which we started our discussion. To that end, we can discretize the domain $[0,1]^d$ to a finite set X of size N, and let $\mathbf{A}$ be the incidence matrix of sets induced by axis-aligned boxes, i.e., each row of $\mathbf{A}$ is associated with a box R, and equals the indicator vector of $R \cap X$. If we also let $\mathbf{w} = \frac{n}{N}\mathbb{1}$, where $\mathbb{1}$ is the all-ones vector, then, for a sufficiently fine discretization, each coordinate of $\mathbf{Aw}$ is a close approximation of $n\lambda_d(R)$. The problem of finding an n-point set P of minimum discrepancy then becomes essentially equivalent to minimizing $\|\mathbf{Ax} - \mathbf{Aw}\|_\infty$ over $\mathbf{x} \in \{0,1\}^N$, where $\|\cdot\|_\infty$ is the standard ℓ_∞ norm. In particular, given $\mathbf{x}$, we can take P to consist of the points in X for which the corresponding coordinate in $\mathbf{x}$ is set to 1. Then, since $[0,1]^d \in \mathcal{R}_d$, we have $||P| - n| \leq \|\mathbf{Ax} - \mathbf{Aw}\|_\infty$, and we can remove or add at most $\|\mathbf{Ax} - \mathbf{Aw}\|_\infty$ to P to make it exactly of size n. The discrepancy of P is then bounded by $2\|\mathbf{Ax} - \mathbf{Aw}\|_\infty$ plus the additional error incurred by the discretization of $[0,1]^d$.

These two examples motivate the definition of *linear discrepancy*, initially introduced by Lovász, Spencer, and Vesztergombi [14]. The smallest possible error for rounding $\mathbf{w}$ with respect to $\mathbf{A}$ is

$$\text{lindisc}(\mathbf{A}, \mathbf{w}) = \min_{\mathbf{x} \in \{0,1\}^n} \|\mathbf{A}(\mathbf{w} - \mathbf{x})\|_\infty.$$

This is the linear discrepancy of $\mathbf{A}$ with respect to $\mathbf{w}$. The linear discrepancy of $\mathbf{A}$ is defined as as the worse case over all $\mathbf{w} \in [0,1]^n$, i.e.,

$$\operatorname{lindisc}(\mathbf{A}) = \max_{\mathbf{w} \in [0,1]^n} \operatorname{lindisc}(\mathbf{A}, \mathbf{w}). \tag{1}$$

It will be useful to consider a maximizer of equation (1) i.e. $\mathbf{w}^* \in [0,1]^n$ such that $\operatorname{lindisc}(\mathbf{A}, \mathbf{w}^*) = \operatorname{lindisc}(\mathbf{A})$. We call $\mathbf{w}^*$ a *deep-hole* of $\mathbf{A}$. Every $\mathbf{A}$ has at least one deep-hole, since linear discrepancy is a continuous function over the compact set $[0,1]^n$.

The special case of $\operatorname{lindisc}(\mathbf{A}, \mathbf{w})$ when $\mathbf{w} = \frac{1}{2}\mathbb{1}$ is especially well studied. When $\mathbf{A}$ is the indicator matrix of a collection $\mathcal{S}$ of m subsets of a universe X, $\operatorname{lindisc}(\mathbf{A}, \frac{1}{2}\mathbb{1})$ measures to what extent it is possible to choose a subset S of X that contains approximately half the elements of each set. This is a rescaling of the well-known combinatorial discrepancy of $\mathcal{S}$, defined as

$$\operatorname{disc}(\mathcal{S}) = \min_{\chi: X \to \{-1,+1\}} \max_{S \in \mathcal{S}} \left| \sum_{s \in S} \chi(s) \right|$$

It is straightforward to check that, by a change of variables, $\operatorname{disc}(\mathcal{S}) = 2 \cdot \operatorname{lindisc}(\mathbf{A}, \frac{1}{2}\mathbb{1})$ where, again, $\mathbf{A}$ is the incidence matrix of $\mathcal{S}$. This definition can be extended to arbitrary matrices $\mathbf{A}$ as $\operatorname{disc}(\mathbf{A}) = 2 \cdot \operatorname{lindisc}(\mathbf{A}, \frac{1}{2}\mathbb{1})$. Combinatorial discrepancy has been widely studied in combinatorics and computer science, see [4, 8, 15].

Sometimes $\operatorname{disc}(\mathbf{A})$ can be small "by accident", thus it is useful to define a more robust discrepancy variant.[1] The hereditary discrepancy of $\mathbf{A}$ is the maximum discrepancy over all sub-matrices, i.e.,

$$\operatorname{herdisc}(\mathbf{A}) = \max_{\mathbf{B}} \operatorname{disc}(\mathbf{B}), \tag{2}$$

where $\mathbf{B}$ ranges over submatrices of $\mathbf{A}$.

A fundamental theorem by Lovász, Spencer, and Vesztergombi shows that linear discrepancy can be bounded above by twice the hereditary discrepancy.

▶ **Theorem 1** (Lovász et al. 1986, [14]). $\operatorname{lindisc}(\mathbf{A}) \leq 2 \cdot \operatorname{herdisc}(\mathbf{A})$.

A number of the applications of combinatorial discrepancy use this basic theorem. In particular, it is common to give an upper bound on the hereditary discrepancy, and from that deduce an upper bound on the linear discrepancy. For example, this strategy was used to give approximation algorithms for bin packing [18, 12], and broadcast scheduling [3], and to design point sets well distributed with respect to arbitrary Borel measures [17, 1]. However, linear discrepancy can be much smaller (by a factor of at least 2^n) than hereditary discrepancy,[2] so hereditary discrepancy lower bounds do not translate to linear discrepancy, and, in general, linear discrepancy lower bounds appear to be challenging. Arguably, a better understanding of linear discrepancy itself would allow proving more and tighter results, in comparison with going through hereditary discrepancy. For example, it is likely that new

---

[1] Consider any matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ and let $\mathbf{A} \in \mathbb{R}^{m \times 2n}$ be the concatenation of two copies of $\mathbf{B}$ side by side. Regardless of the discrepancy of $\mathbf{B}$, $\operatorname{disc}(\mathbf{A}) = 0$ since there exists $\mathbf{x} \in \{-1,1\}^n$ such that $\|\mathbf{A}\mathbf{x}\|_\infty = 0$, namely

$$\mathbf{x}^\intercal = [\underbrace{-1, ..., -1}_{n}, \underbrace{1, ..., 1}_{n}].$$

[2] Whether this remains true if the matrix $\mathbf{A}$ has bounded entries is a tantalizing open question.

analytic tools to estimate linear discrepancy would allow progress on questions in geometric discrepancy theory, as well as questions about the integrality gaps of linear programming relaxations of important optimization problems, such as the bin packing problem.

A sequence of recent works has shed light on the computational complexity of combinatorial and hereditary discrepancy. It is now known that combinatorial discrepancy does not allow efficient approximation algorithms, even in a weak sense (assuming $P \neq NP$)[6], while hereditary discrepancy is NP-hard to approximate better than a factor of 2 [2], and can be approximated within poly-logarithmic factors [16]. Despite being the tool most directly relevant to many applications of discrepancy, however, essentially nothing is known about the computational complexity of linear discrepancy itself. In this paper, we initiate the study of linear discrepancy from a computational viewpoint, and give both the first hardness results, as well as the first exact and approximate algorithms for it.

Before stating our results, it is worth mentioning that linear discrepancy can also be seen as an analogue of the covering radius in lattice theory. Let $\Lambda \subset \mathbb{R}^n$ be a lattice, i.e. discrete additive subgroup of $\mathbb{R}^n$, and let us choose $\mathbf{b}_1, \ldots, \mathbf{b}_n$ to be a basis of Λ. Let $\mathbf{B}$ be a matrix with the $\mathbf{b_i}$ as its columns. The covering radius of Λ in the ℓ_p-norm is defined as

$$\rho(\Lambda) = \max_{\mathbf{y} \in \mathbb{R}^n} \min_{\mathbf{z} \in \Lambda} \|\mathbf{y} - \mathbf{z}\|_p = \max_{\mathbf{w} \in \mathbb{R}^n} \min_{\mathbf{x} \in \mathbb{Z}^n} \|\mathbf{B} \cdot (\mathbf{w} - \mathbf{x})\|_p = \max_{\mathbf{w} \in [0,1]^n} \min_{\mathbf{x} \in \mathbb{Z}^n} \|\mathbf{B} \cdot (\mathbf{w} - \mathbf{x})\|_p, \quad (3)$$

and is independent of the basis. This definition is equivalent to the the definition of $\mathrm{lindisc}(\mathbf{A})$, except that the minimum is over $\mathbb{Z}^n$ rather than $\{0, 1\}^n$. Haviv and Regev showed that the covering radius problem (CRP) in the ℓ_p-norm is Π_2-hard to approximate within some fixed constant for all large enough p [11], and Guruswami, Micciancio, and Regev showed it can be approximated within a factor of $2^{O(n \log n / \log \log n)}$ for the case of $p = 2$ [10].

2.1   Our Results

Let us start with the simple observation that, when $\mathbf{A}$ is a single row matrix, deciding $\mathrm{lindisc}(\mathbf{A}, t\mathbb{1}) = 0$ is the NP-hard Subset Sum problem with target sum $t \sum_{j=1}^{n} A_{1,j}$, and is, therefore, NP-hard. This does not show, however, that computing $\mathrm{lindisc}(\mathbf{A})$ is NP-hard. In this work we show the following hardness result for linear discrepancy.

▶ **Theorem 2.** *The Linear Discrepancy problem of deciding, given an $m \times n$ matrix $\mathbf{A}$ with rational entries, and a rational number t, whether $\mathrm{lindisc}(\mathbf{A}) \leq t$, is NP-hard and is contained in the class Π_2.*

We present algorithms for computing linear discrepancy exactly when the matrix $\mathbf{A}$ has a constant number of rows. We start with a result for a single row matrix.

▶ **Theorem 3.** *For any matrix $\mathbf{A} \in \mathbb{Q}^{1 \times n}$, $\mathrm{lindisc}(\mathbf{A})$ can be computed in time $O(n \log n)$.*

Note that this stands in contrast to the observation above that computing $\mathrm{lindisc}(\mathbf{A}, \mathbf{w})$ is hard even for a single-row matrix $\mathbf{A}$. In addition to the theorem above, we also give a corresponding rounding result, showing that any $\mathbf{w} \in \mathbb{Q}^n$ can be efficiently rounded to within error bounded by the linear discrepancy in the case of single row matrices.

▶ **Theorem 4.** *For any matrix $\mathbf{A} \in \mathbb{Q}^{1 \times n}$ and any $\mathbf{w} \in ([0, 1] \cap \mathbb{Q})^n$, we can find an $\mathbf{x} \in \{0, 1\}^n$ such that $\|\mathbf{A}(\mathbf{w} - \mathbf{x})\|_\infty \leq \mathrm{lindisc}(\mathbf{A})$ in time $O(n \log n)$.*

This result stands in contrast with the hardness of the subset sum problem, which easily implies that it is NP-hard to round $\mathbf{w}$ to within error $\mathrm{lindisc}(\mathbf{A}, \mathbf{w})$ even when $\mathbf{A}$ is a single row matrix.

We can extend Theorem 3 to the case of matrices with a bounded number of rows, with the additional assumption that the entries of $\mathbf{A}$ are bounded. Removing this additional assumption is a fascinating open question.

▶ **Theorem 5.** *For any matrix* $\mathbf{A} \in \mathbb{Z}^{d \times n}$ *where d is some fixed constant and* $\max_{i,j} |A_{i,j}| \leq \delta$, $\mathrm{lindisc}(\mathbf{A})$ *can be computed in time* $O\left(d(n\delta)^{d^2+d}\right)$.

We further present an approximation algorithm for linear discrepancy.

▶ **Theorem 6.** *For any matrix* $\mathbf{A} \in \mathbb{Q}^{m \times n}$, $\mathrm{lindisc}(\mathbf{A})$ *can be approximated in polynomial time within a factor of* 2^{n+1}.

3 Hardness Result

In this section, we show that linear discrepancy (LDS) is NP-Hard by reducing from monotone not-all-equal 3-SAT (MNAE3SAT) [9] to each. The decision problem version of linear discrepancy we consider is defined below.

[MNAE3SAT] **Monotone Not-All-Equal 3-SAT**
Let U be a collection of variables $\{u_1, ..., u_n\}$ and $\mathcal{C}$ be a 3-CNF with clauses $\{C_1, ..., C_m\}$ such that $C_i = t_{i,1} \vee t_{i,2} \vee t_{i,3}$ for positive literals $t_{i,j}$.
Question: Does there exist a truth assignment $\tau : U \to \{\mathsf{T}, \mathsf{F}\}$ such that $\mathcal{C}$ is satisfied and each clause has at least one true and one false literal?

[LDS] **Linear Discrepancy**
Let $\mathbf{A} \in \mathbb{Q}^{m \times n}$ be a matrix and $t \geq 0$ a rational value.
Question: Is $\mathrm{lindisc}(\mathbf{A}) \leq t$?

3.1 Linear Discrepancy

Before we show that linear discrepancy is hard, we will show that the value of $\mathrm{lindisc}(\mathbf{A})$ can be expressed using a polynomial number of bits in the bit complexity of a matrix for rational matrices. Due to space constraints, the proof can be found in Section A.1.

▶ **Lemma 7.** *For any matrix* $\mathbf{A} \in \mathbb{Q}^{m \times n}$, *there exists a deep hole for $\mathbf{A}$ with bit complexity polynomial in n and the bit complexity of $\mathbf{A}$, and, therefore, $\mathrm{lindisc}(\mathbf{A})$ can be written in number of bits polynimial in n and the bit complexity of $\mathbf{A}$.*

Proof of Theorem 2. Note first that the fact that LDS is contained in Π_2 is a straightforward consequence of Lemma 7: the "for-all" quantifier is over potential deep holes $\mathbf{w} \in [0,1]^n$ of the appropriate polynomially bounded bit complexity, and the "exists" quantifier is over $\mathbf{x} \in \{0,1\}^n$.

Next we prove hardness. Let 3-CNF $\mathcal{C}$ be a MNAE3SAT instance as described above. The corresponding LDS instance will be the incidence matrix $\mathbf{A} \in \{0,1\}^{m \times n}$ of $\mathcal{C}$: column $\mathbf{a}_j$ of $\mathbf{A}$ corresponds to variable u_j and row $\mathbf{r}_i$ of $\mathbf{A}$ corresponds to clause C_i, and $A_{i,j} = 1$ if and only if variable u_j appears in clause C_i. Let the target t in the LDS problem be $\frac{3}{2} - \epsilon$ for $\epsilon > 0$ to be determined later.

Consider first that case that $\mathcal{C}$ is a NO-instance of MNAE3SAT i.e. for every truth assignments τ, there exists a clause C_i whose literals all get the same truth assignment. Each $\mathbf{x} \in \{0,1\}^n$ corresponds to a truth assignment. If $x_i = 1$ (resp. $x_i = 0$) then u_i is true (resp. u_i is false). Let C_j be the clause whose literals have the same truth value. Then

$$\text{lindisc}(\mathbf{A}) \geq \text{lindisc}(\mathbf{A}, (1/2) \cdot \mathbb{1}) \geq \left| \mathbf{r}_j \left(\frac{1}{2} \cdot \mathbb{1} - \mathbf{x} \right) \right| = \frac{3}{2} > \frac{3}{2} - \epsilon,$$

so $\mathbf{A}$ is a NO-instance of LDS.

Consider next the case that $\mathcal{C}$ is a YES-instance of MNAE3SAT, and let τ be a satisfying assignment. Suppose $\mathbf{w}^* \in [0,1]^n$ is a deep-hole of $\mathbf{A}$. If $w_i^* = \frac{1}{2}$ for all $i \in [n]$ then

$$\text{lindisc}(\mathbf{A}) = \text{lindisc}(\mathbf{A}, (1/2) \cdot \mathbb{1}) = \text{disc}(\mathbf{A}) \leq \left\| \mathbf{A} \left(\frac{1}{2} \cdot \mathbb{1} - \mathbf{x}^* \right) \right\|_\infty = \frac{1}{2}$$

since every clause has exactly two elements with the same truth value. Thus $\mathbf{A}$ is a YES-instance of LDS as long as we choose $\epsilon \leq 1$. Suppose then that $\mathbf{w}^* \neq \frac{1}{2}\mathbb{1}$, and let ϵ be a lower bound on the smallest non-zero gap between w_i^* and $1/2$ i.e. for all $w_i^* \neq \frac{1}{2}$,

$$\left| w_i^* - \frac{1}{2} \right| \geq \epsilon.$$

By Lemma 7, which implies that $\mathbf{w}^*$ has polynomial bit complexity, we know that we can choose such an ϵ of polynomial bit complexity. We will show that $\text{lindisc}(\mathbf{A}, \mathbf{w}^*) \leq \frac{3}{2} - \epsilon$ by constructing a colouring $\mathbf{x}^*$. Let

$$x_i^* = \begin{cases} \text{rd}(w_i^*) & \text{if } w_i^* \neq \frac{1}{2} \\ \tau(u_i) & \text{otherwise} \end{cases}$$

where $\text{rd}(w_i^*)$ is w_i^* rounded to the closest integer and u_i is the variable corresponding to column i. Let $\mathbf{r}$ be a row of matrix $\mathbf{A}$ with non-zero entries in columns i, j, and k. We bound the discrepancy of row $\mathbf{r}$ based on the number of rounded variables R_v among $\{x_i, x_j, x_k\}$.

$R_v = 0$: Since none of the variables are rounded, $w_i^* = w_j^* = w_k^* = \frac{1}{2}$ and

$$|\mathbf{r}(\mathbf{x}^* - \mathbf{w}^*)| = \left| \left(x_i^* - \frac{1}{2} \right) + \left(x_j^* - \frac{1}{2} \right) + \left(x_k^* - \frac{1}{2} \right) \right| = \frac{1}{2}$$

since τ is a satisfying assignment.

$R_v = 1$: W.l.o.g assume that that x_i^* is the rounded value and $w_j^* = w_k^* = \frac{1}{2}$. Then

$$|\mathbf{r}(\mathbf{x}^* - \mathbf{w}^*)| = \left| (x_i^* - w_i^*) + \left(x_j^* - \frac{1}{2} \right) + \left(x_k^* - \frac{1}{2} \right) \right| \leq \left(\frac{1}{2} - \epsilon \right) + 1 = \frac{3}{2} - \epsilon.$$

$R_v = 2$: W.l.o.g assume that x_i^* and x_j^* are the rounded values and $w_k^* = \frac{1}{2}$. Then

$$|\mathbf{r}(\mathbf{x}^* - \mathbf{w}^*)| = \left| (x_i^* - w_i^*) + (x_j^* - w_j^*) + \left(x_k^* - \frac{1}{2} \right) \right| \leq 2 \cdot \left(\frac{1}{2} - \epsilon \right) + \frac{1}{2} = \frac{3}{2} - 2\epsilon.$$

$R_v = 3$: All three values are rounded so

$$|\mathbf{r}(\mathbf{x}^* - \mathbf{w}^*)| = \left| (x_i^* - w_i^*) + (x_j^* - w_j^*) + (x_k^* - w_k^*) \right| \leq 3 \cdot \left(\frac{1}{2} - \epsilon \right) = \frac{3}{2} - 3\epsilon.$$

Since $\mathbf{r}$ was an arbitrary row of $\mathbf{A}$, $\text{lindisc}(\mathbf{A}) = \text{lindisc}(\mathbf{A}, \mathbf{w}^*) \leq \frac{3}{2} - \epsilon$ as required. This completes the reduction.   ◄

4   Algorithms for Linear Discrepancy

In the following we consider restrictions and variants of linear discrepancy for which we are able to give poly-time algorithms. The first subsection considers matrices with a single row. The second subsection considers matrices $\mathbf{A} \in \mathbb{Z}^{d \times n}$ with constant d and entry of largest magnitude δ. In that case, we compute $\mathrm{lindisc}(\mathbf{A})$ in time $O\left(d(2n\delta)^{d^2}\right)$. The third subsection presents a poly-time 2^n approximation to $\mathrm{lindisc}(\mathbf{A})$ for $\mathbf{A} \in \mathbb{Q}^{m \times n}$.

4.1   Linear Discrepancy of a Row Matrix

We begin by developing some intuition for the linear discrepancy of a one-row matrix, $\mathbf{A} = [a_1, ..., a_n]$. For now, let us make the simplifying assumption that the entries of $\mathbf{A}$ are non-negative and sorted in decreasing order. Define the *subset sums* of $\mathbf{A}$ to be the multi-set $\mathcal{S}(\mathbf{A}) = \{s_1, ..., s_{2^n}\}$ where each $s_i = \mathbf{A}\mathbf{x}$ for exactly one $\mathbf{x} \in \{0,1\}^n$. Enumerate the element of $\mathcal{S}(\mathbf{A})$ in non-decreasing order, i.e. $s_i \le s_{i+1}$. If $\ell_A = 2 \cdot \mathrm{lindisc}(\mathbf{A})$, then ℓ_A is the width of the largest gap between consecutive entries in $\mathcal{S}(\mathbf{A})$.

Suppose $\mathbf{A}_i = [a_1, ..., a_i]$. Let us consider how $\mathcal{S}(\mathbf{A}_i)$ and $\mathrm{lindisc}(A_i)$ change for the first couple of values of i. Clearly, $\mathcal{S}(\mathbf{A}_1) = [0, a_1]$ and $\mathrm{lindisc}(\mathbf{A}_1) = \frac{a_1}{2}$. $\mathcal{S}(\mathbf{A}_2)$ is the disjoint union of $\mathcal{S}(\mathbf{A}_1)$ and $\mathcal{S}(\mathbf{A}_1)$ shifted to the right by a_2. Since $a_1 \ge a_2$, $\mathcal{S}(\mathbf{A}_2) = [0, a_2, a_1, a_1+a_2]$ where the largest gap is of size $\max(a_2, a_1 - a_2)$. See Figure 1. In general, the entries of $\mathcal{S}(\mathbf{A}_i)$ consists of two copies of $\mathcal{S}(\mathbf{A}_{i-1})$ with one shifted to the right by a_i. The gaps in $\mathcal{S}(\mathbf{A}_i)$ are gaps previously in $\mathcal{S}(\mathbf{A}_{i-1})$ or between an element of $\mathcal{S}(\mathbf{A}_{i-1})$ and one in $\{a_i + s : s \in \mathcal{S}(\mathbf{A}_{i-1})\}$.
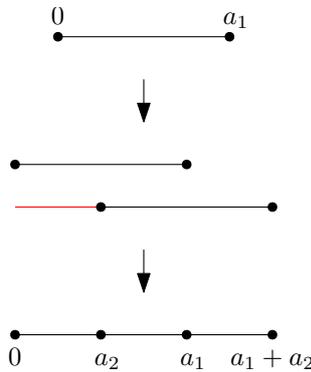

Figure 1 Obtaining $\mathcal{S}(\mathbf{A}_2)$ from $\mathcal{S}(\mathbf{A}_1)$ when $a_1 \ge a_2$.

A similar structure occurs for general matrices with real valued entries with two caveats: (1) the previous interval is shifted left or right depending on the sign of the current entry (negative and positive respectively) and (2) the smallest entry of $\mathcal{S}(\mathbf{A})$ is not zero but the sum of the negative entries in $\mathbf{A}$.

▶ **Lemma 8.** *Suppose $\mathbf{A}_{k-1} = [a_1, ..., a_{k-1}]$ with entries in $\mathbb{R}$ and $|a_i| \ge |a_{i+1}|$. Let the largest gap in $\mathcal{S}(\mathbf{A}_{k-1})$ be of size ℓ_{k-1}. Then, for $\mathbf{A}_k = [a_1, ..., a_{k-1}, a_k]$ where $|a_k| \le |a_i|$ for all $i \in [k-1]$, the largest gap in $\mathcal{S}(\mathbf{A}_k)$ is of size $\max(|a_k|, \ell_{k-1} - |a_k|)$.*

Proof. Again, it is important to remember that the entries of $\mathcal{S}(\mathbf{A}_k)$ are exactly those in $\mathcal{S}(\mathbf{A}_{k-1})$ along with those in $\{a_k + s : s \in \mathcal{S}(\mathbf{A}_{k-1})\}$. Let $\ell = \max(|a_k|, \ell_{k-1} - |a_k|)$.

We first show that $2 \cdot \mathrm{lindisc}(\mathbf{A}_k) \le \ell$ by showing that gaps between consecutive entries in $\mathcal{S}(\mathbf{A}_k)$ have size at most ℓ. If (s_j, s_{j+1}) is a consecutive pair in $\mathcal{S}(\mathbf{A}_{k-1})$ such that $s_{j+1} - s_j > \ell$, then s_j and s_{j+1} are no longer consecutive in $\mathcal{S}(\mathbf{A}_k)$, since $s_j \le s_j + a_k \le s_{j+1}$

if $a_k > 0$ and $s_j \leq s_{j+1} + a_k \leq s_{j+1}$ if $a_k < 0$. See Figure 2. Then, the gap given by any such pair gets split into gaps of size at most $\max\{|a_k|, s_{j+1} - s_j - |a_k|\} \leq \ell$, where the inequality holds because $s_{j+1} - s_j \leq \ell_{k-1}$. It follows that the size of each gap in $\mathcal{S}(\mathbf{A}_k)$ is at most ℓ.
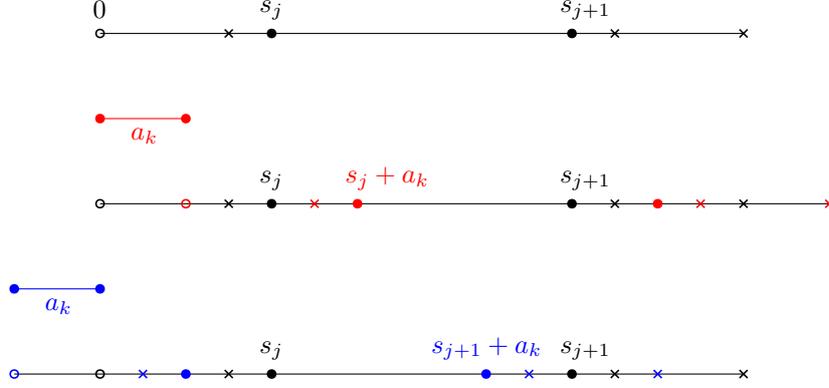

Figure 2 All consecutive pairs in $\mathcal{S}(\mathbf{A}_{k-1})$ of size greater than $|a_k|$ will be divided into two or more consecutive pairs in $\mathcal{S}(\mathbf{A}_k)$. The red interval indicates what happens when $a_k > 0$. The blue interval indicates what happens when $a_k < 0$.

Next we will show that $2 \cdot \operatorname{lindisc}(\mathbf{A}_k) \geq \ell$ by producing a pair of consecutive entries in $\mathcal{S}(\mathbf{A}_k)$ which achieves gap ℓ. Suppose $\ell = |a_k|$. Recall that s_0 is the smallest subset sum of all entries in $\mathbf{A}_k$, which equals the sum of all negative entries in $\mathbf{A_k}$. Then it is easy to check that s_1 equals $s_0 + |a_k|$, where we recall that a_k is the entry in $\mathbf{A}_k$ with minimum absolute value. Therefore, $(s_0, s_0 + |a_k|)$ is a consecutive pair in $\mathcal{S}(\mathbf{A}_k)$. This means that if $\ell = |a_k|$, then we are done, as we have produced a pair with gap ℓ.

When $\ell = \ell_{k-1} - |a_k| > |a_k|$, we split our analysis into two cases: (1) $a_k > 0$ and (2) $a_k < 0$.

In the former case, let (s_{j^*}, s_{j^*+1}) be a consecutive pair in $\mathcal{S}(\mathbf{A}_{k-1})$ that achieves gap ℓ_{k-1} and suppose, towards a contradiction, that $s_{j^*} + a_k$ and s_{j^*+1} do not appear consecutively in $\mathcal{S}(\mathbf{A}_k)$. Then there must be some $s \in \mathcal{S}(\mathbf{A}_k)$ such that $s_{j^*} + a_k < s < s_{j^*+1}$. Note that s cannot be an element of $\mathcal{S}(\mathbf{A}_{k-1})$ since s_{j^*} and s_{j^*+1} are consecutive in $\mathcal{S}(\mathbf{A}_{k-1})$, so $s - a_k$ must be an element of $\mathcal{S}(\mathbf{A}_{k-1})$. However, since $s > s_{j^*} + a_k$, we have $s - a_k > s_{j^*}$. This is a contradiction since s_{j^*} and s_{j^*+1} are consecutive entries in $\mathcal{S}(\mathbf{A}_{k-1})$. See Figure 3. Thus $(s_{j^*} + a_k, s_{j^*+1})$ must be a consecutive pair in $\mathcal{S}(\mathbf{A}_k)$.
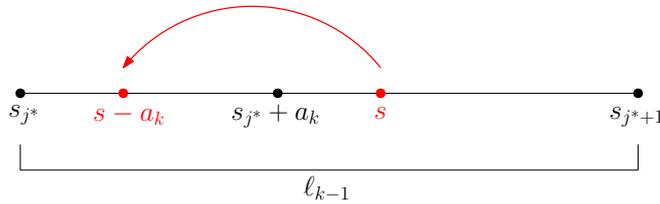
Figure 3 Suppose $a_k < \ell_{k-1} - a_k$ and there exists $s \in \mathcal{S}(\mathbf{A}_k)$ such that $s_{j^*} + a_k < s < s_{j^*+1}$.

The latter case, when $a_k < 0$, is similar. Again there exists a pair of consecutive entries (s_{j^*}, s_{j^*+1}) in $\mathcal{S}(\mathbf{A}_{k-1})$ which achieves gap ℓ_{k-1}. Suppose, towards contradiction, that s_{j^*} and $s_{j^*+1} - |a_k|$ do not appear consecutively in $\mathcal{S}(\mathbf{A}_k)$. Then there must be some $s \in \mathcal{S}(\mathbf{A}_k)$ such that $s_{j^*} < s < s_{j^*+1} - |a_k|$. Again, s cannot be an element of $\mathcal{S}(\mathbf{A}_{k-1})$ since s_{j^*} and s_{j^*+1} are consecutive in $\mathcal{S}(\mathbf{A}_{k-1})$, so $s + |a_k|$ must be an element of $\mathcal{S}(\mathbf{A}_{k-1})$. However since $s < s_{j^*+1} - |a_k|$, we have $s + |a_k| < s_{j^*+1}$. This is a contradiction since s_{j^*} and s_{j^*+1} are consecutive entries in $\mathcal{S}(\mathbf{A}_{k-1})$. ◄

Lemma 8 has the following curious corollary.

▶ **Corollary 9.** *Let* $\mathbf{A} = [a_1, ..., a_n]$ *and* $\mathbf{A}' = [|a_1|, ..., |a_n|]$. *Then* $\mathrm{lindisc}(\mathbf{A}) = \mathrm{lindisc}(\mathbf{A}')$.

Lemma 8 and Corollary 9 suggest an algorithm: replace the entries of $\mathbf{A}$ by their magnitudes. Sort $\mathbf{A}$. Consider each entry in turn and update the largest gap accordingly. See Algorithm 1.

Proof of Theorem 3. By Corollary 9 it is sufficient to consider row matrices with non-negative entries. Suppose that $\mathbf{A} = [a_1, ..., a_n]$ is such a matrix with entries sorted in decreasing order. Algorithm 1 correctly outputs the linear discrepancy for matrices with a single entry. Let $\mathbf{A}_i = [a_1, ..., a_i]$. Lemma 8 gives us a recursive method for computing the largest gap in $\mathcal{S}(\mathbf{A}_{i+1})$ from the largest gap in $\mathcal{S}(\mathbf{A}_i)$. Since $\mathrm{lindisc}(\mathbf{A})$ is half the size of the largest gap in $\mathcal{S}(\mathbf{A})$, Algorithm 8 output $\mathrm{lindisc}(\mathbf{A})$ as required. ◀

■ **Algorithm 1** Linear discrepancy of row matrix.

    Input: Matrix $\mathbf{A} \in \mathbb{Q}^{1 \times n}$.
    Output: $\mathrm{lindisc}(\mathbf{A})$.
1 for i *from* 1 *to* n **do**
2 $\mathbf{A}[i] \leftarrow |a_i|$
3 sort $\mathbf{A}$ in decreasing order
4 $\ell \leftarrow a_1$
5 for i *from* 2 *to* n **do**
6 $\ell \leftarrow \max(a_i, \ell - a_i)$
7 return $\frac{\ell}{2}$

Thus, for any row matrix $\mathbf{A}$ with n elements, we can find $\mathrm{lindisc}(\mathbf{A})$ in time $O(n \log n)$.

4.1.1 One Row Linear Discrepancy Rounding

Let $\mathrm{lindisc}(\mathbf{A}) = \ell$. By the definition of linear discrepancy, for every $\mathbf{w} \in [0, 1]^n$ there exists an $\mathbf{x} \in \{0, 1\}^n$ such that $\|\mathbf{A}(\mathbf{w} - \mathbf{x})\|_\infty \leq \ell$. In-fact, if $\mathbf{w}$ is not a deep-hole, there exists an $\mathbf{x}$ which satisfies $\|\mathbf{A}(\mathbf{w} - \mathbf{x})\|_\infty < \ell$. However it is not obvious that finding such an $\mathbf{x}$ can be done efficiently i.e. in polynomial time with respect to the bit complexity of $\mathbf{A}$ and n. By reducing from the subset-sum problem, we observe that it is difficult to compute $\mathrm{lindisc}(\mathbf{A}, \mathbf{w})$ let alone find an $\mathbf{x}$ which minimizes $\|\mathbf{A}(\mathbf{w} - \mathbf{x})\|_\infty \leq \ell$.

Proof of Theorem 4. To begin, let $\mathbf{A} = [a_1, ..., a_n]$ for positive a_i in non-increasing order. We will consider $\mathbf{A}$ with arbitrary entries at the end. Let $w = \mathbf{A}\mathbf{w}$. As before, let $\mathcal{S}(\mathbf{A}) = [s_0, ..., s_{2^n - 1}]$ be the subset-sums of $\mathbf{A}$ where each $s_i = \mathbf{A}\mathbf{x}$ for an $\mathbf{x} \in \{0, 1\}^n$ and $s_i \leq s_{i+1}$ for all i. Recall that $2 \cdot \mathrm{lindisc}(\mathbf{A})$ is the largest gap between any two consecutive entries in $\mathcal{S}(\mathbf{A})$. Our algorithm will find a pair of subset sums containing w. If we can show that the size of the interval between these two subset sums is no more than the gap between some two consecutive entries in $\mathcal{S}(\mathbf{A})$, then the closest subset sum to w among these two will be within $\mathrm{lindisc}(\mathbf{A})$ of w.

Just as in Algorithm 1, we refine the interval between two subset sums containing w by incrementally adding the entries of $\mathbf{A}$ in decreasing order. Initially our interval is $g_0 = [0, \sum_{i=1}^n a_i]$. We maintain the invariants: (1) $w \in g_i$ for all i, and (2) the end-points of g_i are subset sums.

Suppose $w \in g_i = [u, v]$ and we are considering a_i. If $u + a_i > w$ then set $v \leftarrow \min(v, u + a_i)$. Otherwise let $u \leftarrow u + a_i$. Algorithm 2 computes this interval and the associated vectors $\mathbf{u}$ and $\mathbf{v}$ representing its endpoints.

Consider the values of u and v at the end of the algorithm. We claim that the final interval $[u, v]$ is at most the width of some gap between two consecutive terms in $\mathcal{S}(\mathbf{A})$, the array of all subset sums of $\mathbf{A}$. Notice $u = a_1 u_1 + \cdots + a_n u_n$ where $\mathbf{u} = [u_1, ..., u_n]$ is an endpoint of the interval once Algorithm 2 completes.

We partition $\mathbf{u}$ into maximal blocks where all entries in the same block have the same value i.e. $[u_1, u_2, ..., u_{\ell_1}], ..., [u_{\ell_r+1}, u_{\ell_r+2}, ..., u_n]$ such that $u_{\ell_i+1} = u_{\ell_i+2} = \cdots = u_{\ell_{i+1}}$ for $i = 0, 1, ..., r - 1$ where $\ell_0 = 0$.

We claim that Algorithm 2 outputs an interval containing w whose width is at most the distance between some two consecutive entries in $\mathcal{S}(\mathbf{A})$. The proof is by induction on r, the number of blocks. In the base case, $r = 1$ and there is only one block. Thus $u = 0$ or $u = \sum a_i$. In the case where $u = 0$, we must have $a_i > w$ for all $i \in [n]$, and $v = a_n$. Thus $w \in [0, a_n]$ with consecutive elements 0 and a_n of $\mathcal{S}(\mathbf{A})$. In the latter case when $u = \sum a_i$, we can output $\mathbf{w}$ since it is already a subset sum.

Suppose next that the claim holds for all matrices where the algorithm outputs a vector $\mathbf{u}$ with k blocks, and we will show that it still holds for a matrix $\mathbf{A}$ whose output $\mathbf{u}$ has $k + 1$ blocks. Let $\mathbf{u}' = [u_1, ..., u_{\ell_{k+1}}]$ and $\mathbf{v}' = [v_1, ..., v_{\ell_{k+1}}]$ be the final vectors after running the algorithm on $\mathbf{A}' = [a_1, ..., a_{\ell_{k+1}}]$. Further let $u' = \sum_{i=1}^{\ell_{k+1}} a_i u_i$ and $v' = \sum_{i=1}^{\ell_{k+1}} a_i v_i$. By the induction hypothesis, the width of $[u', v']$ is at most the distance between some two consecutive elements in the list of subset sums of $\mathbf{A}'$. The last block of $\mathbf{u}$ is $[u_{\ell_{k+1}+1}, ..., u_n]$. The entries of this block are either all zeros or all ones. Consider each case in-turn.

First suppose $u_{\ell_{k+1}+1} = \cdots = u_n = 0$. Since none of the a_i for $i = \ell_{k+1} + 1, ..., n$ were added to u, it must be the case that $u' + a_i > w$ for all such i. Thus the interval $[u, v] = [u', \min(v', u' + a_n)]$ has width at most a_n. Since 0 and a_n are consecutive in $\mathcal{S}(\mathbf{A})$, as $|a_n|$ is the entry with the smallest magnitude in $\mathbf{A}$, the output interval satisfies our requirements.

Next suppose $u_{\ell_{k+1}+1} = \cdots = u_n = 1$. It must be the case that $u = u' + a_{\ell_{k+1}+1} + \cdots + a_n \leq w$. Observe that $a_{\ell_{k+1}}$ is in the kth block and so $u_{\ell_{k+1}} = 0$. Let $[u'', v'']$ be our interval after processing the $k - 1^{\text{st}}$ block i.e. $u'' = \sum_{i=1}^{\ell_k} a_i u_i$ and $v'' = \sum_{i=1}^{\ell_k} a_i v_i$. Notice that since none of the entries in the kth block were added to u'', we must have $u'' + a_i > w$ for all $i = \ell_k + 1, ..., \ell_{k+1}$. In such cases, we always update $v'' \leftarrow \min(v'', u'' + a_i)$ after each such i, thus the interval $[u', v']$ has width at most $a_{\ell_{k+1}}$. Thus it suffices to show that $a_{\ell_{k+1}+1} + \cdots + a_n$ and $a_{\ell_{k+1}}$ are consecutive in $\mathcal{S}(\mathbf{A})$. First note that $a_{\ell_{k+1}+1} + \cdots + a_n \leq a_{\ell_{k+1}}$ since $u' + a_{\ell_{k+1}+1} + \cdots + a_n \leq w \leq v' \leq u' + a_{\ell_{k+1}}$. The two subset sums then are also consecutive, since $a_i > a_{\ell_{k+1}}$ for all $i < \ell_{k+1}$.

Now consider the case where $\mathbf{A}$ can have both positive and negative entries. Without loss of generality we can assume that none of the entries are zero. Let $A_- = \{a_i \in \mathbf{A} : a_i < 0\}$ and $A_+ = \{a_i \in \mathbf{A} : a_i > 0\}$. It suffices to set $u_0 = \sum_{a \in A_-} a$ and $v_0 = \sum_{a \in A_+} a$ and let $\mathbf{u}$ and $\mathbf{v}$ be the indicator vectors of $\mathbf{A}_-$ and $\mathbf{A}_+$ respectively. The remainder of the algorithm is identical except that the matrix should be sorted in decreasing order of *magnitude* and every time an element $a_i \in \mathbf{A}_-$ is added to u, its entry in $\mathbf{u}$ should be set to zero.   ◀

4.2   Constant Rows with Bounded Matrix Entries

Let $\mathbf{A} \in \mathbb{Z}^{d \times n}$ with $\max_{i,j} |A_{i,j}| \leq \delta$. Let $Z = \mathbf{A}[0, 1]^d$ be the zonotope of $\mathbf{A}$ and let $T = [-n\delta, n\delta]^d \cap \mathbb{Z}^d$ be the set of all integer lattice points of Z. The following algorithm computes lindisc$(\mathbf{A})$ in polynomial time with respect to n for fixed d and δ. The algorithm makes use of Lemma 11, which is proved in the Appendix.

■ **Algorithm 2** Finding a close subset sum to **Aw**.

Input: A vector $\mathbf{w} \in [0,1]^n$ and a row matrix $\mathbf{A} = [a_1, ..., a_n]$ of positive integers
         sorted in increasing order.
Output: A vector $\mathbf{x} \in \{0,1\}^n$ such that $\|\mathbf{A}(\mathbf{w} - \mathbf{x})\|_\infty \leq \operatorname{lindisc}(\mathbf{A})$.

1 $\mathbf{A} \leftarrow \text{SORT-DECREASING}(\mathbf{A})$
2 $\mathbf{u} \leftarrow \text{ZEROS}(n)$
3 $\mathbf{v} \leftarrow \text{ONES}(n)$
4 $w \leftarrow \mathbf{Aw}$, $u \leftarrow \mathbf{Au}$, $v \leftarrow \mathbf{Av}$
5 **return v** if $w == v$
6 **for** $k = 1..n$ **do**
7   **if** $u + a_k > w$ **then**
8 $v \leftarrow \min\left(v, u + a_k\right)$
9   **if** $v == u + a_k$ **then**
10 $\mathbf{v} \leftarrow \text{COPY}(\mathbf{u})$
11 $\mathbf{v}[k] \leftarrow 1$
12   **else**
13 $u \leftarrow u + a_k$
14 $\mathbf{u}[k] \leftarrow 1$

15 **return u** *if u is closer to w else* **v**

Proof of Theorem 5. For every one of the $(2n\delta+1)^d$ integral points $\mathbf{b} \in T$, compute whether $\mathbf{Ax} = \mathbf{b}$ for some $\mathbf{x} \in \{0,1\}^n$ using dynamic programming. This procedure generalizes dynamic programming algorithms for knapsack and subset sum and will be outlined in the following. Let $\mathbf{a}_1, ..., \mathbf{a}_n$ be the columns of $\mathbf{A}$. Construct a matrix $\mathbf{M}$ with dimensions $[-n\delta, n\delta]^d \times n$. Cell $(\mathbf{v}, i)$ of $\mathbf{M}$ contains the indicator $[\mathbf{M}(\mathbf{v} - \mathbf{a}_i, i - 1) \vee \mathbf{M}(\mathbf{v}, i - 1)]$; this corresponds to a linear combination of the first $i - 1$ columns of $\mathbf{A}$ which adds up to $\mathbf{v} - \mathbf{a}_i$ or a linear combination of the first $i - 1$ columns which adds up to $\mathbf{v}$. The first column of $\mathbf{M}$ is the indicator vector for $\{\mathbf{a}_1\}$. Computing the entries of $\mathbf{M}$ takes time $O(2n\delta)^{d+1}$. $\mathbf{M}(\mathbf{b}, n)$ indicates the feasibility of $\mathbf{Ax} = \mathbf{b}$. Computing this for all $\mathbf{b}$ takes time $O(2n\delta)^{d+1}$. Let $S \subseteq T$ be the set of points $\mathbf{b}$ in Z such that $\mathbf{Ax} = \mathbf{b}$ for some $\mathbf{x} \in \{0,1\}^n$, and set $|S| = N$.

Apply Lemma 11 to the points of S in ℓ_∞-norm. The output is some radius r and point $\mathbf{x}^*$ such that the ℓ_∞-ball centered at $\mathbf{x}^*$ with radius r is the largest such ball with center inside the convex hull of S not containing any points of S. Note that r is in-fact the linear discrepancy of $\mathbf{A}$. Since r and $\mathbf{x}^*$ can be computed in time $O(N^d)$, $\operatorname{lindisc}(\mathbf{A})$ can be computed in time $O(2n\delta)^{d^2+d}$. ◄

4.3 Poly-time Approximation Algorithm

Next, we prove Theorem 6 present a 2^n-approximation algorithm for linear discrepancy. Recall that $\operatorname{rd}(\mathbf{w})$ is the function which rounds each coordinate of $\mathbf{w}$ to its nearest integer (with ties broken arbitrarily). Let the operator norms of a matrix $\mathbf{A}$ be:

$$\|\mathbf{A}\|_{p \to q} = \max_{\mathbf{x} \in \mathbb{R}^n \setminus \{0\}} \frac{\|\mathbf{Ax}\|_q}{\|\mathbf{x}\|_p}.$$

Note that

$$\text{lindisc}(\mathbf{A}) \leq \max_{\mathbf{w} \in [0,1]^n} \|\mathbf{A}(\mathbf{w} - \text{rd}(\mathbf{w}))\|_\infty \leq \frac{1}{2} \max_{\mathbf{z} \in [-1,1]^n} \|\mathbf{A}\mathbf{z}\|_\infty = \frac{1}{2} \|\mathbf{A}\|_{\infty \to \infty}.$$

To bound $\text{lindisc}(\mathbf{A})$ from below, we show that $\|\mathbf{A}\|_{\infty \to \infty} \leq 2^{n+1} \cdot \text{lindisc}(\mathbf{A})$. This completes the proof of the theorem, since $\|\mathbf{A}\|_{\infty \to \infty}$ equals the largest ℓ_1 norm of any row of $\mathbf{A}$, and can be computed in polynomial time.

Let us try to interpret the statement $\|\mathbf{A}\|_{\infty \to \infty} \leq 2^{n+1} \cdot \text{lindisc}(\mathbf{A})$. Note that $\|\mathbf{A}\mathbf{z}\|_\infty$ is equal to the Minkowski $\mathcal{P}$-norm $\|\mathbf{z}\|_\mathcal{P}$ for $\mathcal{P} = \{\mathbf{x} : \|\mathbf{A}\mathbf{x}\|_\infty \leq 1\}$ i.e. $\|\mathbf{z}\|_\mathcal{P} = \inf\{t \geq 0 : \mathbf{z} \in t\mathcal{P}\}$ so

$$\|\mathbf{A}\|_{\infty \to \infty} = \max_{\mathbf{z} \in [-1,1]^n} \|\mathbf{A}\mathbf{z}\|_\infty = \max_{\mathbf{z} \in [-1,1]^n} \|\mathbf{z}\|_\mathcal{P}.$$

By interpreting $\mathbf{z}$ as the difference of two vectors $\mathbf{x}, \mathbf{x}' \in [0,1]^n$ we have that

$$\|\mathbf{A}\|_{\infty \to \infty} = \max_{\mathbf{z} \in [-1,1]^n} \|\mathbf{z}\|_\mathcal{P} = \max_{\mathbf{x},\mathbf{x}' \in [0,1]^n} \|\mathbf{x} - \mathbf{x}'\|_\mathcal{P}.$$

It is an easy, and well-known fact that $\text{lindisc}(\mathbf{A})$ is the smallest t such that $[0,1]^n \subseteq \bigcup_{\mathbf{x} \in \{0,1\}^n} (\mathbf{x} + \mathcal{P})$; see [15]. We then just need to show that the diameter of the unit hypercube with respect to the Minkowski $\mathcal{P}$-norm is no more than this scale-factor t times $O(2^n)$. We prove the following more general statement.

▶ **Lemma 10.** *Let $\mathcal{K}$ be a convex symmetric polytope and $S \subset \mathbb{R}^n$ be convex. Suppose there exist N elements $x_1, ..., x_N \in S$ such that*

$$S \subseteq \bigcup_{x_i} x_i + t\mathcal{K}.$$

Then $\max_{x,x' \in S} \|x - x'\|_\mathcal{K} \leq 2tN$.

Proof. Fix any two points x and x' in S. Let $\mathcal{P}_i$ be the polytope $x_i + t\mathcal{K}$. Since S is convex, the line segment $\lambda x + (1 - \lambda)x'$ for $\lambda \in [0,1]$ is in S. Therefore $\lambda x + (1 - \lambda)x'$ intersects a sequence of polytopes $\mathcal{P}_{k_1}, ..., \mathcal{P}_{k_r}$ with centres $x_{k_1}, ..., x_{k_r}$, such that any two consequtive polytopes in the sequence intersect. Since the polytopes are convex, we can assume that they appear in the sequence at most once, so $r \leq N$. By the triangle inequality we have

$$\begin{aligned}
\|x - x'\|_K &= \|(x - x_{k_1}) + (x_{k_1} - x_{k_2}) + \cdots + (x_{k_r} - x')\|_K \\
&\leq \|x - x_{k_1}\|_K + \|x_{k_1} - x_{k_2}\|_K + \cdots + \|x_{k_r} - x'\|_K \\
&\leq t + 2t(N - 1) + t = 2tN
\end{aligned}$$

where the last inequality follows as $x \in \mathcal{P}_{k_1}$, $x' \in \mathcal{P}_{k_r}$, and $\|x_{k_i} - x_{k_{i+1}}\|_K \leq 2t$.   ◀

Proof of Theorem 6. In Lemma 10, set $\mathcal{K}$ to be the parallelepiped defined by $\mathbf{A}$, $S = [0,1]^n$, $t = \text{lindisc}(\mathbf{A})$, and $\{x_1, ..., x_N\} = \{0,1\}^n$.   ◀

5   Open Problems

Because of the similarity between the closest vector problem and linear discrepancy, we suspect that linear discrepancy is also Π_2-complete, and the hardness result of Theorem 2 is, in this sense, not tight. Further, Haviv and Regev also showed that CRP is Π_2-hard to approximate to with-in a factor of $\frac{3}{2}$, and we conjecture that a similar hardness of approximation result should hold for linear discrepancy.

We suspect that the algorithm used to prove Theorem 3 can be generalized to matrices $\mathbf{A} \in \mathbb{Q}^{d \times n}$ with running time $\tilde{O}(n^d)$. This would be a substantial improvement on the $O\left(d(n\delta)^{d^2}\right)$ running time algorithm used to prove Theorem 5, and would be independent of the magnitude of the largest entry of $\mathbf{A}$.

It is also interesting to extend the largest empty ball algorithm from Lemma 11 to other ℓ_p norms, or even arbitrary norms, given appropriate access to the norm ball. Currently, this seems rather difficult as Voronoi diagrams with respect to the ℓ_p-norm for $p \in (2, \infty)$ are poorly behaved. For the standard ℓ_2-norm Voronoi diagram in $\mathbb{R}^d$, it is the case that $d + 1$ affinely independent vertices are equidistant to exactly one point. This is no longer the case even in $\mathbb{R}^3$ for ℓ_4-norm [13]. In particular, there exists a set of four vertices such that the intersection of their pair-wise bisectors has size three. The situation is even worse for general strictly convex norms. There exists such norms where the pair-wise bisectors of a set of four points in $\mathbb{R}^3$ can have arbitrarily many intersections.

We currently also have no evidence that the approximation factor in Theorem 6 is tight. One possibility is that there exists an approximation preserving reduction from the closest vector problem in lattices to linear discrepancy. This would show that one cannot expect a significant improvement to Theorem 6 without also improving the best polynomial time approximation to the covering radius, which is currently also exponential in the dimension n. On the other hand, we also conjecture that the approximation factor in Theorem 6 can be taken to be a function of $\min\{m, n\}$, or even of the rank of the matrix $\mathbf{A}$.

────── **References** ──────

1   Christoph Aistleitner, Dmitriy Bilyk, and Aleksandar Nikolov. Tusnády's problem, the transference principle, and non-uniform qmc sampling. In Art B. Owen and Peter W. Glynn, editors, *Monte Carlo and Quasi-Monte Carlo Methods*, pages 169–180. Springer International Publishing, 2016.

2   Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\varepsilon)$-sat is np-hard. *SIAM Journal on Computing*, 46(5):1554–1573, 2017.

3   Nikhil Bansal, Ravishankar Krishnaswamy, and Viswanath Nagarajan. Better scalable algorithms for broadcast scheduling. *ACM Trans. Algorithms*, 11(1):3:1–3:24, 2014.

4   József Beck and Vera T Sós. Discrepancy theory. In *Handbook of combinatorics (vol. 2)*, pages 1405–1446. MIT Press, 1996.

5   Jean-Daniel Boissonnat, Micha Sharir, Boaz Tagansky, and Mariette Yvinec. Voronoi diagrams in higher dimensions under certain polyhedral distance functions. *Discrete & Computational Geometry*, 19(4):485–519, 1998. `doi:10.1007/PL00009366`.

6   Moses Charikar, Alantha Newman, and Aleksandar Nikolov. Tight hardness results for minimizing discrepancy. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1607–1614. Society for Industrial and Applied Mathematics, 2011.

7   Bernard Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, 10(4):377–409, 1993.

8   Bernard Chazelle. *The discrepancy method: randomness and complexity*. Cambridge University Press, 2001.

9   E Mark Gold. Complexity of automaton identification from given data. *Information and control*, 37(3):302–320, 1978.

10   Venkatesan Guruswami, Daniele Micciancio, and Oded Regev. The complexity of the covering radius problem. *Computational Complexity*, 14(2):90–121, 2005.

11   Ishay Haviv and Oded Regev. Hardness of the covering radius problem on lattices. In *Computational Complexity, 2006. CCC 2006. Twenty-First Annual IEEE Conference on*, pages 14–pp. IEEE, 2006.

12   Rebecca Hoberg and Thomas Rothvoss. A logarithmic additive integrality gap for bin packing. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2616–2625. SIAM, 2017.

13   Christian Icking, Rolf Klein, Ngoc-Minh Lé, and Lihong Ma. Convex distance functions in 3-space are different. *Fundamenta Informaticae*, 22(4):331–352, 1995.

14   László Lovász, Joel Spencer, and Katalin Vesztergombi. Discrepancy of set-systems and matrices. *European Journal of Combinatorics*, 7(2):151–160, 1986.

15   Jiri Matousek. *Geometric discrepancy: An illustrated guide.* Springer, 1999.

16   Jiří Matoušek, Aleksandar Nikolov, and Kunal Talwar. Factorization Norms and Hereditary Discrepancy. *International Mathematics Research Notices*, 2020(3):751–780, March 2018. `doi:10.1093/imrn/rny033`.

17   A. Nikolov. Tighter bounds for the discrepancy of boxes and polytopes. *Mathematika*, 63(3):1091–1113, 2017. `doi:10.1112/S0025579317000250`.

18   Thomas Rothvoß. Approximating bin packing within o (log opt* log log opt) bins. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 20–29. IEEE, 2013.

19   Alexander Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1998.

20   Godfried T Toussaint. Computing largest empty circles with location constraints. *International journal of computer & information sciences*, 12(5):347–358, 1983.

A   Appendix

A.1   Bit Complexity of Linear Discrepancy

Proof of Lemma 7. Let $\mathbf{r}_i$ for $i \in [m]$ be the rows of $\mathbf{A}$, $\text{lindisc}(\mathbf{A}) = \lambda_A$, and $\mathbf{w}^*$ be a deep-hole of $\mathbf{A}$. For every $\mathbf{x} \in \{0,1\}^n$ there exists an $i \in [m]$ and $\sigma \in \{-1,1\}$ such that $\sigma\mathbf{r}_i(\mathbf{w}^* - \mathbf{x}) \geq \lambda_A$. Let $\mathbf{b}_x = \sigma\mathbf{r}_i$ and consider the following linear program over the variables $\mathbf{w} \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$:

Maximize: λ

Subject to: $\mathbf{b_x}(\mathbf{w} - \mathbf{x}) \geq \lambda$ for all $\mathbf{x} \in \{0,1\}^n$

             $\mathbf{0} \leq \mathbf{w} \leq \mathbf{1}$

Let λ^* be the optimum value of this linear program. First note that $\lambda_A \leq \lambda^*$ since $(\mathbf{w}^*, \lambda)$ satisfies the constraints. Next we show that $\lambda_A \geq \lambda^*$. Suppose, towards contradiction, that $\lambda_A < \lambda^*$. Then there exists $\mathbf{w}' \in [0,1]^n$ such that

$$\|\mathbf{A}(\mathbf{w}' - \mathbf{x})\|_\infty \geq \mathbf{b}_x(\mathbf{w}' - \mathbf{x}) \geq \lambda^* > \lambda_A$$

for every $\mathbf{x} \in \{0,1\}^n$. Since $\lambda_A = \text{lindisc}(\mathbf{A})$, we cannot have $\text{lindisc}(\mathbf{A}, \mathbf{w}') > \lambda_A$. Thus $\lambda^* = \text{lindisc}(\mathbf{A})$. Since this LP has n variables, the number of bits required to express the linear discrepancy and some deep-hole $\mathbf{w}^*$ of $\mathbf{A}$ are polynomial in n and the bit complexity of the largest entry of $\mathbf{A}$ [19].   ◀

A.2   Largest Empty Ball Problem

Let V be a set of n points in the plane and let $\text{ch}(V)$ denote the convex hull of V. The largest empty circle problem, denoted LEC, takes V and outputs both a radius r and point $\mathbf{x}^* \in \text{ch}(V)$ such that the circle centered at $\mathbf{x}^*$ with radius r is the largest empty circle not containing any point of V. We generalize this problem to other norms and to higher

dimensions as follows: V is a set of n points in $\mathbb{R}^d$, and the goal is to compute a point $\mathbf{x}^*$ in ch(V) such that $\mathbf{x}^* + rB$ does not contain any point of V, where B is the unit ball of either the ℓ_2^d or the ℓ_∞^d norm. In the following we present an algorithm which solves this largest empty ball (LEB) problem.

▶ **Lemma 11** (LEC in Higher Dimensions). *Let V be a set of n points in $\mathbb{R}^d$ for some fixed constant d. The LEB of V, in both ℓ_2- and ℓ_∞-norms, can be computed in time $O(n^d)$.*

Proof. We use the following terminology. Define a face F of the Voronoi diagram vd(V) of V to be a subset of $\mathbb{R}^d$ such that, for some $S \subseteq V$, and every $\mathbf{x} \in F$, S are the points in V closest to $\mathbf{x}$. In particular, this means that any $\mathbf{x} \in F$ is equidistant from all points in S.

The algorithm of Toussaint [20] computes the LEB of n points V in the plane with respect to the ℓ_2-norm as follows,

1. Compute vd(V). Note that vd(V) is the union of Voronoi faces of dimension k, the set of which we denote vd$_k(V)$, over all $k = 0, ..., d-1$.
2. Compute the convex hull of V, denoted ch(V). Let h be the number of facets of ch(V).
3. Preprocess the points of ch(V) so that queries of the form "Is a point x in ch(V)?" can be answered in time $O(\log h)$. For every $v \in \mathrm{vd}_0(V)$, determine if $v \in \mathrm{ch}(V)$. Let $C_1 = \{v \in \mathrm{vd}_0(V) : v \in \mathrm{ch}(V)\}$.
4. Determine the intersection points of faces in vd$_k(V)$ with faces of ch(V) of co-dimension k, for pairs of such faces that intersect at a unique point. Let C_2 be the set of all such intersection points.
5. For all points $v \in C_1 \cup C_2$, find the largest empty circle centered at v. Output a v which maximizes this radius.

We find the analogue of each step for points in $\mathbb{R}^d$ with respect to the ℓ_2-norm, and then adapt the algorithm to the ℓ_∞-norm.

In the following let $N = n^{\lceil d/2 \rceil}$. The complexity, i.e. total number of faces of every dimension, of the ℓ_2-Voronoi diagram in $\mathbb{R}^d$ for fixed d is $O(N)$ and can be computed in time $O(N + n \log n)$ by a classic result of Chazelle [7]. The complexity of ch(V) is $O(N)$ and can also be computed in time $O(N + n \log n)$.

To determine the set C_1 of Voronoi intersection points inside the convex hull, we let $\mathcal{H}$ be the set of bounding hyperplanes of ch(V). Assume, without loss of generality, that ch(V) contains the origin, and, for each $H \in \mathcal{H}$, let H^- be the half-space with H as its boundary containing the origin. Then ch$(V) = \bigcap_{H \in \mathcal{H}} H^-$. We simply test, for each Voronoi intersection point $\mathbf{v}$, whether $\mathbf{v} \in H^-$ for each $H \in \mathcal{H}$, in total time $O(N)$. Since there are at most $O(N)$ Voronoi intersection points, we can find C_1 in time $O(N^2)$.

To determine the set C_2 of all unique intersection points of k-faces of vd$_k(V)$ and faces of ch(V) of co-dimension k will require solving several linear systems. Note that the points in each face F in vd$_k(V)$ satisfy $d - k$ equality constraints $\langle \mathbf{a}_1, \mathbf{x} \rangle = b_1, \langle \mathbf{a}_k, \mathbf{x} \rangle = b_k$ for linearly independent vectors $\mathbf{a}_1, ... \mathbf{a}_k \in \mathbb{R}^d$. Similarly, the points in each face of co-dimension k of ch(V) satisfy k linearly independent equality constraints. Since there are at most $O(2^d N) = O(N)$ faces of ch(V), there are at most that many faces of ch(V) of co-dimension k. We can then go over all Voronoi faces F of dimension k, and all faces G of ch(V) of co-dimension k, and solve the corresponding system of $(d-k) + k = d$ linear equations. If the system has a unique solution, we check if that solution is in $F \cap G$, and, if so, we add it to C_2. Thus, for constant d, the size of C_2 and the time to compute it are bounded bounded above by $O(N \cdot 2^d N) = O(N^2)$.

In total there are at most $O(N + N^2)$ points in $C_1 \cup C_2$ which can be computed in time $O(N^2)$. Thus solving the largest empty ball problem in dimension d for constant d takes time $O(n^d)$.

Next we consider the largest empty ball problem in ℓ_∞-norm. The convex hull remains the same, so we just have to consider the Voronoi diagram with respect to the ℓ_∞-norm. Again, constructing the Voronoi diagram can be done in expected time $O(n^{\lceil d/2 \rceil} \log^{d-1} n)$ using the randomized algorithm of Boissonnat et al. [5]. Next we consider the number of intersections between the Voronoi diagram and the convex hull. First note that Voronoi diagrams with respect to the ℓ_∞-norm need not consist of only hyperplanes and their intersections. Indeed, in $\mathbb{R}^d$, for two points with the same y-coordinate, there exists regions with affine dimension two which are equidistant to both points. To remedy this, we assume that no two points in V have the same i-th coordinate, for any $i \in [d]$. This is without loss of generality, by perturbing the points in V slightly. It remains to consider the complexity of each bisector in ℓ_∞-norm. By Claim 12, in constant dimension d, each such bisector can have at most $O(d^2)$ facets. Therefore, the complexity of any face of the Voronoi diagram, being the intersection of at most d bisectors, is bounded by a function of d. Thus the bounds of the ℓ_2-norm algorithm still hold, up to constant factors that depend on d. ◀

▷ **Claim 12** (Bound on Number of Facets of ℓ_∞ Bisectors). Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ be such that assume that $u_i \neq v_i$ for all $i \in [d]$. Then the bisector $\{\mathbf{x} : \|\mathbf{x} - \mathbf{u}\|_\infty = \|\mathbf{x} - \mathbf{v}\|_\infty\}$ has at most $O(d^2)$ facets.

Proof. Let $\mathbf{x}$ be a point in the bisector at ℓ_∞ distance r from $\mathbf{u}$ and $\mathbf{v}$. Pick coordinates i and j and signs σ and τ in $\{-1, +1\}$ such that

$$\sigma_i(x_i - u_i) = \tau_j(x_j - v_j) = r. \tag{4}$$

Moreover, let us make this choice so that either $i \neq j$ or $\sigma_i \neq \tau_i$. This is always possible, since, otherwise, the assumption on $\mathbf{u}$ and $\mathbf{v}$ is violated. Then, (4) defines a hyperplane in $\mathbb{R}^d$, namely $H_{i,j,\sigma,\tau} = \{\mathbf{x} : \sigma_i x_i - \tau_j x_j = \sigma_i u_i - \tau_j v_j\}$. Note that there are at most $\binom{2d}{2} \in O(d^2)$ such hyperplanes, and each $\mathbf{x}$ in the bisector lies in at at least one of them. Moreover, a point $\mathbf{x}$ in $H_{i,j,\sigma,\tau}$ lies in the bisector if and only if it satisfies the inequalities

$$|x_k - u_k| \leq \sigma_i(x_i - u_i) \quad \forall k \in [d],$$
$$|x_k - v_k| \leq \tau_j(x_j - v_j) \quad \forall k \in [d].$$

Thus, the bisector is the union of $(d-1)$-dimensional convex polyhedra, one per each of the $O(d^2)$ hyperplanes $H_{i,j,\sigma,\tau}$. ◁