Complexity of Sublinear Algorithms for Convexity in Higher Dimensions

by

Abhinav Bommireddi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2022

© Abhinav Bommireddi 2022

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Sofya Raskhodnikova Professor, College of Arts and Sciences, Boston University
Supervisor:	Eric Blais Associate Professor, Dept. of Computer Science, University of Waterloo
Internal Member:	Lap Chi Lau Professor, Dept. of Computer Science, University of Waterloo
Internal Member:	Shalev Ben-David Assistant Professor, Dept. of Computer Science, University of Waterloo
Internal-External Member:	Kanstantsin Pashkovich Assistant Professor, Dept. of Combinatorics & Optimization, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The main results in the thesis are from the following papers I co-authored.

1. *On Testing and Robust Characterization of Convexity.* Joint work with: Eric Blais. In, RANDOM 2020 [11].

2. *Testing Convexity of Functions Over Finite Domains.* Joint work with: Aleksandrs Belovs, Eric Blais. In, SODA 2020 [6].

3. *Minimizing Discrete Convex Functions.* Joint work with: Aleksandrs Belovs, Eric Blais. Unpublished.

Abstract

Convexity plays a prominent role in both mathematics and computer science. It is defined for sets and functions, and many related problems can be solved efficiently if the set/function is convex. In this thesis, we focus on three problems related to convexity where we don't have that guarantee.

The first problem we consider is the decision problem: is a given unknown set convex? We study it under the framework of property testing. In property testing, instead of distinguishing objects that satisfy the property from the ones that do not, we distinguish objects that satisfy the property from the ones that are "far" from satisfying the property. We approach the problem of testing convex sets by studying a closely related problem of *robust characterizations of convex sets*, in which we study different characterizations of convex sets and determine their robustness. We say a characterization is *robust*, if sets that are "far" from convex are also "far" from satisfying the characterization. We examine the robustness of three characterizations of convex sets: line characterization, convex hull characterization and supporting hyperplane characterization. Furthermore, we discuss the implications of the robustness/non-robustness of these characterizations on the testing problem.

The second problem we consider is the decision problem: is a given unknown function over the hypergrid $[n]^d$ convex? The function is given via a valuation oracle, which we query to get the function value. We study this problem also under the framework of property testing. We refer to an algorithm that performs the testing task as *non-adaptive* if it submits all its queries before looking at the function value on any of the points, and adaptive otherwise. We show that any non-adaptive algorithm that tests convexity of functions $f\colon [n]^d \to \mathbb{R}$ for $d \geq 2$, has query complexity that is linear in n and exponential in d. To understand if adaptivity helps, we consider the problem of testing convexity of functions over a *stripe*, $[3] \times [n]$, and show that there exists an adaptive testing algorithm that does exponentially better than any non-adaptive one.

The third problem we consider is minimizing convex functions over the hypergrid when given access to the comparison oracle to the function on the points in the hypergrid. The *comparison oracle* to a function takes as input two points and determines which of them has the smaller function value. The discrete (non-convex) nature of the domain makes the problem challenging, since we do not have the property that local minimum implies global minimum, which is crucial for the minimization of convex functions over continuous domains. We, however, show that there is a minimization algorithm in two dimensions with polynomial time and query complexity, and for any constant dimension greater than two, there is a minimization algorithm with quasi-polynomial time and query complexity.

Acknowledgements

The last 7 years of my life, during my masters and PhD, have been transformative. I had the opportunity to meet lots of wonderful and smart people who have had an impact on me during this phase.

First and foremost, I would like to express my sincere gratitude to my advisor Eric Blais. He has been very patient and kind with me and gave me the freedom to explore, not just in research but also in life. He was very supportive when I told him I was going to take a break during my PhD to explore other things. He always had my best interests at heart. My PhD experience would not have been as enjoyable with a different advisor.

I also would like to thank Sofya Raskhodnikova, Kanstantsin Pashkovich, Lap Chi Lau, and Shalev Ben-David for agreeing to read this thesis and to serve on the examining committee.

The rest of the acknowledgement is dedicated to all the wonderful people I met during my stay at Waterloo without mentioning specific names, as I'm afraid I will miss someone. I always felt home at DC (Davis Center). Not just my fellow students, but all the professors who I had an opportunity to interact with were very kind and welcoming. I will cherish the long Avalon nights in MC (Mathematics and Computer building), pick-up soccer during summer, and the long conversations in DC corridors. The coffee shop at Williams needs a special mention, as I spent more time there than in my office. It is also the place where most of my masters and PhD thesis was written.

PhD is a demanding journey and it is hard to navigate it without a strong support system. Over the last 7 years, while I was away from my family, my friends have been my family. They have been there to celebrate the tiniest of my accomplishments, and have stood by me during my tough times. I will forever cherish the memories of late-night Friday hangouts, and the numerous trips we went on.

Finally, I cannot thank my family enough for their unconditional love and steadfast support during this journey; especially, my granddad who has been my biggest cheerleader.

Dedication

Dedicated to those who are not afraid to be different.

Table of Contents

List of Figures

Chapter 1

Introduction

Convexity is a mathematician's best friend. A set or a function being convex enables one to solve complex problems pertaining to them, as it expands the set of tools at one's disposal. Informally, we say a set is convex if the line joining any two points in the set lies entirely inside the set, and we say a function is convex if the line joining any two points on the plot of the function lies entirely above the function. Traditionally convex functions have been defined over convex domains, but the definition can be extended to discrete domains, and there are applications where a discrete domain is more natural.

The impact of convexity is not just limited to mathematics; it has applications in economics [59, 25] and many areas of computer science, like signal processing [49, 40], machine learning [42, 57], and game theory [56, 47].

The wide range of applications of convexity led to the extensive study of its various algorithmic aspects, and its unique structure enabled the design of efficient algorithms for them. The most important algorithmic problem related to convexity is the minimization of convex functions [13, 46]. The algorithms for this problem rely on the property of convex functions that a local minimum implies global minimum, i.e. if a point is minimum among all its neighbours, then it is the global minimum. Note that this property holds only for convex functions over convex domains and not for convex functions over discrete domains that we consider in this thesis.

For convex sets some of the important and well studied problems are rounding, sampling, and volume estimation [21, 30, 39, 58]. The algorithms for these problems make use of two properties of convex sets: the first is that for any point outside the set, there exists a hyperplane that separates the point from the set. The second is that any way we cut the set, the surface area of the cut is "significant" compared to the minimum of the volumes

of the resulting subsets. The second property ensures that random walks over the set mix fast.

As we can see, all the above algorithms rely heavily on the convexity property of the function/set. In many applications, the function/set is unknown, and convexity is an assumption we make on the unknown object. What if the unknown object is not convex? Is there a way to quickly check that? The first two problems we study in this thesis are motivated by this question.

The first of them is the problem of *testing convex sets*, which is an *approximate* version of the decision problem "Is a given unknown set convex?". We approach it by considering a different but closely related problem of *robust characterization of convex sets*, which is interesting in its own right. In this, we consider different characterizations of convexity and determine if they are *robust*. It ties in to the question of testing convex sets as follows: a robust characterization of convex sets could lead to a tester for convex sets, and a non-robust characterization rules out a natural class of algorithms. In Section 1.1, we state the results we obtain studying the robustness of three different characterizations of convex sets, and discuss the implications of these results on the testing convex sets problem.

The second problem we consider is that of *testing convex functions over discrete domains*, which is an *approximate* version of the decision problem "Is a given unknown discrete function convex?". We focus on functions over the hypergrid $[n]^d$, where $[n] = \{1, 2, \ldots, n\}$. This problem can be solved efficiently in one dimension as shown by Parnas, Ron, and Rubinfeld in [50]. Though it has been over 20 years since the problem was first studied, not much is known for higher dimensions, $d \geq 2$. We show that a certain class of algorithms called *non-adaptive* algorithms cannot solve the problem efficiently in higher dimensions. The results we obtain for this problem are stated in Section 1.2.

For convex functions over the hypergrid $[n]^d$, where $d \geq 2$, not just the problem of testing but the problem of minimization is also quite challenging, as the discreteness of the domain causes the function to lose the *local-optimality-implies-global-optimality* property, which is crucial for the efficient minimization of convex functions over convex domains. The discreteness of the domain also rules out many techniques established for convex domains. This is the third problem we consider in this thesis. Despite the challenges posed by the discrete setting, we show that there exist efficient algorithms for minimization of convex functions over the hypergrid. We describe our results for this problem in Section 1.3.

1.1 On Testing and Robust Characterizations of Convex Sets

A set $K \subset \mathbb{R}^d$ is *convex* if for every two points $x, y \in K$ and every parameter $0 \leq \lambda \leq 1$, the point $z = \lambda x + (1 - \lambda)y$ is also in K. The general set K can be degenerate. To get around this, we work with a special class of sets called bodies. A *body* is a subset of $\mathbb{R}^d$ that is compact—i.e., closed and bounded—and has a non-empty interior.

An *ϵ-tester for convex bodies* is an algorithm that when given an unknown body $K \subset \mathbb{R}^d$

- accepts with probability $\frac{2}{3}$ if K is convex; and
- rejects with probability $\frac{2}{3}$ is K is ϵ-far from convex

where a body K is *ϵ-far from convex* if every convex body C satisfies $\text{Vol}(K \triangle C) \geq \epsilon \, \text{Vol}(K)$, i.e. we need to change at least an ϵ fraction of the volume of the body to make it convex.

Access to the unknown body is given in the form of oracles which we query to gain information about the unknown body. We consider two standard oracles: random oracle and membership oracle. A *random oracle*, when queried, gives a point uniformly at random from the body, and a *membership oracle*, when queried at a point, determines if the point is in the body or not.

The problem of testing convexity of geometric objects was introduced by Raskhodnikova in [52], where they considered testing convexity of discretized images in two dimensions. The problem of testing convex bodies, as described above, was introduced by Rademacher and Vempala [51]. They showed that it is possible to ϵ-test convexity of bodies in $\mathbb{R}^d$ using $O((d/\epsilon)^d)$ calls to the random oracle and $\text{poly}(d)/\epsilon$ calls to the membership oracle. The main open problem in testing convex bodies is to determine whether there is an ϵ-tester for convex bodies in $\mathbb{R}^d$ that has total query complexity polynomial in d.

The problem of testing convex bodies has been considered in many different settings. Tsur and Ron in [53], like Raskhodnikova in [52], considered testing convexity of discretized images in two dimensions, but for a different distance metric. Berman, Murzabulatov, and Raskhodnikova [9, 8] studied testing convex bodies in two dimensions, but in a slightly different setting. They assume the unknown set is contained in a larger convex set, which they get uniform samples from, and distance to convexity is also measured relative to the larger convex set. Chen, Freilich, Servedio, and Sun [17] considered testing convex bodies when the unknown body is accessed via samples from a standard normal distribution over

$\mathbb{R}^d$, and distance between bodies is measured using the Gaussian measure. Though testing convex bodies has been studied in many different settings, no progress has been made on the problem stated above, since the work of Rademacher and Vempala.

The problem of testing convex bodies is closely related to that of identifying robust characterizations of convex bodies. We say a characterization of a property is *robust*, if any entity that is far from satisfying the property has lots of violations of the characterization. In this thesis, we study different characterizations of convexity and determine if they are robust or not, and discuss the implications of these results for the testing convex bodies problem.

The three main characterizations of convexity are:

Line characterization
For every pair of points $x, y \in K$, the line segment $\overline{xy}$ joining x and y is contained in K.

Convex hull characterization
For every set X of m points in K, the convex hull of X is contained in K.

Supporting hyperplane characterization
For every point x on the boundary ∂K of K, there exists a vector $v \in \mathbb{R}^d$ such that $\langle v, x\rangle = 0$ and every point $y \in K$ satisfies $\langle v, y\rangle \geq 0$.

Line characterization

Rademacher and Vempala [51] showed that the line characterization of convexity is not very robust: there is a body $K \subset \mathbb{R}^d$ which is $\Omega(\frac{1}{d^2})$-far from convex but for which the line segment $\overline{xy}$ joining two points x and y drawn uniformly at random from K satisfies $\Pr[\overline{xy} \not\subseteq K] = 2^{-\Omega(d)}$. We strengthen this result by showing that there are also bodies $K \subset \mathbb{R}^d$ which are $\Omega(1)$-far from convex for which the same bound holds.

Theorem 1.1. *There exists a body $K \subset \mathbb{R}^d$ that is $\frac{1}{8}$-far from convex for which*

$$\Pr_{x,y \in K}\left[\overline{xy} \not\subseteq K\right] = 2^{-\Omega(d)}. \tag{1.1}$$

The robustness of the line characterization of convexity is closely related to the natural *line tests* for convexity. A line test draws two points x, y uniformly at random from K, draws a point $z \in \overline{xy}$, and checks whether z is in K or not. The result of Rademacher

and Vempala [51] shows that an ϵ-tester for convexity obtained by running a line test multiple times cannot have query complexity that is polynomial in *both* d and $1/\epsilon$, but it does not rule out the possibility that a line tester could yield a convexity tester with query complexity polynomial in d when ϵ is a constant. Theorem 1.1 eliminates this possibility, by showing that an ϵ-tester for convexity obtained from a line tester must still have query complexity exponential in d even when $0 < \epsilon \leq \frac{1}{8}$ is constant.

Convex hull characterization

For any $m \geq 1$, the *convex hull* of a set $X = \{x^{(1)}, \ldots, x^{(m)}\}$ of m points in $\mathbb{R}^d$ is

$$\mathrm{conv}\Big(X\Big) := \left\{ y : \exists \lambda_1, \ldots, \lambda_m \geq 0 \text{ s.t. } \sum_{i=1}^{m} \lambda_i = 1 \text{ and } y = \sum_{i=1}^{m} \lambda_i x^{(i)} \right\},$$

the set of points that can be obtained by taking a convex combination of the points in X. A natural extension of the line test is the *convex hull test*: for some $m \geq 2$, draw a set X of m points from K uniformly and independently at random, draw a point $z \in \mathrm{conv}(X)$, and check if z is in K. When $m = 2$, the convex hull test is equivalent to the line test which, as we have seen above, cannot lead to an efficient tester for convexity. For $m \geq 3$, however, it is possible that it leads to much more efficient testing algorithms. Indeed, Berman, Murzabulatov, and Raskhodnikova [9, 8] showed that in a slightly different property testing model, the convex hull test can be used to test convexity with a number of queries that is polynomial in $1/\epsilon$ in the two-dimensional setting where $K \subset \mathbb{R}^2$.

Our next result, however, rules out the possibility of obtaining an efficient tester for convexity in the high-dimensional setting using the convex hull test by showing that the convex hull characterization is not robust, even when taking the convex hull of an *exponential* (in the dimension d) number of points.

Theorem 1.2. *There exist a body $K \subset \mathbb{R}^d$ that is $\frac{1}{8}$-far from convex and a constant $c > 0$ such that a set $X = \{x^{(1)}, \ldots, x^{(m)}\} \in K$ of $m = 2^{cd}$ points drawn uniformly and independently at random from K satisfies*

$$\Pr_X \Big[\mathrm{conv}(X) \not\subseteq K \Big] = 2^{-\Omega(d)}.$$

Theorem 1.2 strengthens a result of Chen, Freilich, Servedio, and Sun [17]. While they considered another different property testing setting (based on the Gaussian measure on $\mathbb{R}^d$), one of their lower bound arguments carries over to our current setting and shows that

any ϵ-tester for convexity that only accesses the random oracle of K and always accepts when K is convex must have query complexity $2^{\Omega(d)}$. Theorem 1.2 shows that this query complexity lower bound still holds even if the tester also has free access to a (very strong) "convex hull oracle" that determines whether the convex hull of the sampled points contains *any* point that is outside of K.

Supporting hyperplane characterization

A hyperplane $H = \{y \in \mathbb{R}^d : \langle v, y - x\rangle = 0\}$ defined by $v \in \mathbb{R}^d \setminus \{0\}$ and $x \in \mathbb{R}^d$ is a *supporting hyperplane* for the body $K \subset \mathbb{R}^d$ if $x \in K$ and every other point $y \in K$ satisfies $\langle v, y - x\rangle \geq 0$. The supporting hyperplane characterization of convexity guarantees that for every point x on the boundary ∂K of a convex body $K \subset \mathbb{R}^d$, there is a supporting hyperplane for K that contains x. Our main result in this section shows that, in contrast to the line and convex hull characterizations of convexity, this characterization is robust. In the theorem statement and beyond, the *ball* of radius $r > 0$ around the origin is the set $B(r) = \{x : \|x\| \leq r\}$.

Theorem 1.3. *Fix some $R > r > 0$. For every body $K \subset \mathbb{R}^d$ that satisfies $B(r) \subseteq K \subseteq B(R)$ and is ϵ-far from convex, when $x \in \mathbb{R}^d$ is drawn uniformly at random from the boundary ∂K of K then*

$$\Pr_{x \in \partial K}\left[\forall c \in \mathbb{R}^d \, \exists y \in K \ \text{s.t.} \ \langle c, y - x\rangle < 0\right] \geq \frac{\epsilon r}{3Rd}. \tag{1.2}$$

The condition $B(r) \subseteq K \subseteq B(R)$ is a standard assumption in convex geometry (see, e.g., [27]) to avoid degenerate examples. Furthermore, the condition is necessary in Theorem 1.3, as the bound in (1.2) is tight in d, r, and R. We can see this by considering a simple example of a cylinder of radius $r < \frac{R}{10}$ and length $\frac{9R}{10}$. Removing a constant fraction of the cylinder in the middle yields a body that is $\Omega(1)$-far from convex but for this body the fraction of points on its boundary that do not have a supporting hyperplane is $O(\frac{r}{dR})$. See Section 2.3.3 for the details.

Theorem 1.3 can be interpreted algorithmically as saying that convexity testing can be done efficiently by algorithms which have access to a "random boundary" oracle that returns a point uniformly at random from the boundary of the input set and a *separation oracle* defined below.

Separation oracle. Given as input a point $x \in \mathbb{R}^d$, the oracle SO returns a vector $v \in \mathbb{R}^d$ such that for every $y \in K$, $\langle v, y - x\rangle \geq 0$ if such a vector exists and returns $\perp$ otherwise.

For the setting where the algorithm does not have access to a random boundary oracle but does have access to a separation oracle, random walks can be used to obtain a simple and efficient algorithm for testing convexity.

Proposition 1.4. *There is an ϵ-tester for convexity of bodies $K \subset \mathbb{R}^d$ that satisfy the promise $B(r) \subseteq K \subseteq B(R)$ which makes $\mathrm{poly}(d, \frac{1}{\epsilon}, \log \frac{R}{r})$ queries to the membership and separation oracles for K.*

The algorithm in the proof of Proposition 1.4 has time complexity that is polynomial in d, $\frac{1}{\epsilon}$ and $\log \frac{R}{r}$. However, it does not settle the problem of testing convex bodies using only membership and random access oracles, since in general the separating hyperplane oracle cannot be simulated with these two oracles.

1.2 Testing Discrete Convex Functions

Let X be a subset of $\mathbb{Z}^d$. A function $f\colon X \to \mathbb{R}$ is called a *discrete convex function* if for every finite collection of points $x^{(1)}, x^{(2)}, \ldots, x^{(k)} \in X$ and non-negative reals $\lambda_1, \ldots, \lambda_k \geq 0$ satisfying $\sum_i \lambda_i = 1$ and $\sum_i \lambda_i x^{(i)} \in X$, we have

$$f\Big(\sum_i \lambda_i x^{(i)}\Big) \leq \sum_i \lambda_i f(x^{(i)}).$$

Equivalently, the function f is a discrete convex function if and only if there exists a convex function $\widetilde{f}\colon \mathbb{R}^d \to \mathbb{R}$ that satisfies $\widetilde{f}(x) = f(x)$ for every $x \in X$. We use the terms convex functions over discrete domains and discrete convex functions interchangeably.

An *ϵ-tester for discrete convex functions* is an algorithm that when given a discrete function $f : X \to \mathbb{R}$,

- accepts with probability $\frac{2}{3}$ if f is convex; and
- rejects with probability $\frac{2}{3}$ is f if ϵ-far from convex

where we say that a function $f : X \to \mathbb{R}$ is *ϵ-far from convex* if for every discrete convex function $h : X \to \mathbb{R}$, we have $|\{x \in X : h(x) \neq f(x)]\}| \geq \epsilon |X|$.

Access to the unknown function, like in the case of convex sets, is via an oracle which we query to gain information about the underlying function. We consider the standard *valuation oracle*, which, when queried at a point, returns the function value at that point. We refer to a tester as *non-adaptive* if it selects all the inputs to query before observing the value of f on any of those inputs; otherwise the tester is *adaptive*.

Testing convexity on the line

The problem of testing convexity of functions $f\colon [n] \to \mathbb{R}$ on the line was first considered by Parnas, Ron, and Rubinfeld [50]. They showed that $O(\frac{\log n}{\epsilon})$ queries suffice to ϵ-test convexity in this setting. A slightly better upper bound of $O(\frac{\log(\epsilon n)}{\epsilon})$ was shown by Ben-Eliezer [7]. It follows from his more general algorithm for testing local properties of arrays. We give a more direct algorithm.

Theorem 1.5. *There exists an ϵ-tester for convexity of functions $f : [n] \to \mathbb{R}$ over the line with complexity $O(\frac{\log(\epsilon n)}{\epsilon})$. The tester is non-adaptive and has 1-sided error.*

Blais, Raskhodnikova, and Yaroslavtsev [12] showed that the bound in Theorem 1.5 on the query complexity of *non-adaptive* convexity testers is optimal when $\epsilon > 0$ is a constant. In [6], Belovs, Blais and Bommireddi show that the bound in Theorem 1.5 is optimal for all values of $\epsilon \leq \frac{1}{9}$, even when the testers are allowed to be adaptive. This implies that adaptivity does *not* help to reduce query complexity when testing convexity of functions over the line, which is analogous to the situation for testing monotonicity of functions over the line [24].

Testing convexity over 2-dimensional domains

Parnas, Ron, and Rubinfeld [50] asked whether convexity can be tested efficiently for functions over the 2-dimensional domain. Prior to our work, there were no know non-trivial lower or upper bounds for this problem. We show that no non-adaptive algorithm can efficiently test convexity for functions over $[n]^2$.

Theorem 1.6 (Special case of Theorem 1.8 below)**.** *Any non-adaptive $\Omega(1)$-tester for convexity of $f\colon [n]^2 \to \mathbb{R}$ has query complexity $\Omega(n)$.*

Note that Theorem 1.6 does not eliminate the possibility that convexity of functions on $[n]^2$ can be tested with $\mathrm{poly}(\log(n))$ queries by *adaptive* algorithms. Based on the results for testing convexity in 1D, one may be tempted to guess that adaptivity does not help in this setting either and that the bound in the theorem could be strengthened to apply to adaptive algorithms as well. To test this intuition, we consider an intermediate domain between 1-dimensional and full 2-dimensional case: the *stripe* $[3] \times [n]$. The same intuition from the 1-dimensional case would suggest that adaptivity does not help in testing convexity of functions over the stripe. We show, however, that here adaptivity can be used to obtain an *exponential* improvement on the query complexity of convexity testers.

Theorem 1.7. *There exists a 1-sided-error algorithm that ϵ-tests a function $f\colon [3]\times[n]\to\mathbb{R}$ for convexity using $O(\frac{\log^2 n}{\epsilon})$ queries to f. By contrast, any non-adaptive $\Omega(1)$-tester for convexity of $f\colon [3]\times[n]\to\mathbb{R}$ has query complexity $\Omega(\sqrt{n})$.*

The exponential gap between the adaptive and non-adaptive query complexity of convexity testing in Theorem 1.7 stands in stark contrast to the situation for the related problem of testing monotonicity of real valued functions: there it is known that adaptivity does not yield *any* reduction in query complexity, as there is a non-adaptive monotonicity tester for functions $f\colon [n]^d\to\mathbb{R}$ with query complexity $O(d\log n)$ [15] and every monotonicity tester (adaptive or not) has query complexity $\Omega(d\log n)$ [16].

Testing convexity over high-dimensional domains

Our main contribution to the problem of testing discrete convex functions is to show that any non-adaptive algorithm for testing convexity of functions over $[n]^d$ has query complexity that is linear in n and exponential in d.

Theorem 1.8. *For every $d\geq 2$, $n>4d$ and any $\epsilon\leq\frac{1}{10}$, any non-adaptive ϵ-tester for convexity of functions over $[n]^d$ has query complexity $\Omega\big((\frac{n}{d})^{\frac{d}{2}}\big)$.*

Note that the trivial upper bound for testing convexity (or any other property) of functions over $[n]^d$ is n^d, so Theorem 1.8 shows that non-adaptive convexity testers cannot do significantly better (qualitatively) than the naïve brute-force testing algorithm.

Related work. Pallavoor, Raskhodnikova and Varma [48] considered the convexity testing problem over the line $[n]$, and gave a parameterized algorithm whose query complexity depends on the size of the range, rather than the size of the domain. Lahiri, Newman and Varma [31] again for the same problem gave a parametrized algorithm whose query complexity depends only on the number of distinct discrete derivatives. Dixit, Raskhodnikova, Thakurta and Varma [20] considered *erasure-resiliant* testing of convexity of functions over the line $[n]$. Berman, Raskhodnikova and Yaroslavtsev [10] studied testing convexity of functions over the hypergrid under ℓ_p distance. Other weaker notions of discrete convexity like linear convexity and separate convexity have also been studied under the property testing framework [7, 12]. We define these properties in Sections 3.2 and the testing results related to them in Section 4.5.

1.3 Minimizing Discrete Convex Functions

Consider the following scenario: A person has to choose from a large number of bundles of goods what he likes the most, subject to his budget constraint. Each bundle consists of d types of *indivisible* goods, and at most n goods of each type. It is hard for the person to process so many options simultaneously due to cognitive limitations. Can we devise an efficient algorithm that asks the person questions and helps them decide what they like the most? The problem is in the area of utility maximization in economics. The utility function measures preferences over a set of goods. The utility function being concave is a common assumption made. Under that assumption, the above problem of utility maximization is equivalent to the question of minimizing discrete convex functions stated below.

Discrete convex minimization (DCM). *Given discrete convex functions $f, g\colon \mathbb{Z}^d \to \mathbb{R}$, how efficiently can we solve the following optimization problem*[1]

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & g(x) \leq 0 \\ & x \in [n]^d \end{aligned}$$

We are interested in the case where access to the objective function f is given by a comparison oracle. A *comparison oracle* takes two points as input and determines which one has the smaller function value. The economics application we discussed at the beginning of the section is an example where a comparison oracle is more natural than a valuation oracle. It is hard for the person to quantify how much they like a bundle of goods, and they might find it easier to answer comparison questions.

For the constraint function g, access to just the comparison oracle is not enough, as using it, we can't even determine if a given point is feasible. Therefore, we assume access to valuation oracle for g. Our algorithms work even when given access to the comparison oracle to g and an oracle that can determine the feasibility of a point.

Discrete convex minimization for $d = 2$

The problem of minimizing discrete convex functions is easy when $d = 1$. A simple bisection algorithm finds the minimum of a discrete convex function $f\colon [n] \to \mathbb{R}$ with query complexity $O(\log n)$.

[1]Note that if we have multiple convex constraints $g_1, g_2, \ldots, g_m$, we can combine them into one convex constraint g by defining g as $g(x) = \max\{g_1(x), g_2(x), \ldots, g_m(x)\}$.

As soon as we move to two dimensions, $d = 2$, the problem becomes challenging. The trivial brute-force minimization algorithm in this setting has query complexity $O(n^2)$. It is possible to improve this query complexity to $O(n \log n)$ by using the efficient one-dimensional minimization algorithm to find the minimum value of f on each of the n rows. However, any attempt at improving the algorithm immediately runs into a fundamental problem with convex functions over discrete domains:

> **Local optimality does not imply global optimality.** That is, a point $x \in \mathbb{Z}^2$ might have the minimal value $f(x)$ among all of its neighbours, but still be far from the global minimum of f.

Even an easier problem of *minimum verification*, where given a point we need to determine if it is a minimum, is also challenging.

We overcome this challenge and give a non-trivial algorithm for minimizing discrete convex functions over $[n]^2$. Our algorithm has time and query complexity $O(\text{poly}(\log n))$.

Theorem 1.9. *Let $f, g \colon \mathbb{Z}^2 \to \mathbb{R}$ be discrete convex functions. Given access to the comparison oracle for f and the valuation oracle for g, we can solve the optimization problem*

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & g(x) \leq 0 \\ & x \in [n] \times [n] \end{aligned} \tag{1.3}$$

using $\text{poly}(\log n)$ *time and queries.*

Discrete convex minimization for $d > 2$

A special case of DCM when f, g are linear functions is the *integer linear programming* problem (ILP). It is known that ILP is NP-Complete [54]. As our problem is at least as hard as ILP, we can not solve it efficiently in terms of the dimension. We assume the dimension d is a constant and state the computational complexity as a function of n.

For any constant d, we show that discrete convex functions over the grid $[n]^d$ can be minimized using $2^{O(\text{poly}(\log \log n))}$ time and queries.

Theorem 1.10. *Let $f, g \colon \mathbb{Z}^d \to \mathbb{R}$ be discrete convex functions. Given access to the comparison oracle for f and the valuation oracle for g, we can solve the optimization*

problem

$$\begin{aligned} \min_{x} \quad & f(x) \\ & g(x) \leq 0 \\ & x \in [n]^d \end{aligned} \tag{1.4}$$

using $2^{O(\text{poly}(\log\log n))}$ *time and queries for any constant* d*.*

Our algorithms work for a more general class of functions called discrete strictly quasiconvex functions, of which discrete convex functions are a subset. A function $\widetilde{f} : \mathbb{R}^d \to \mathbb{R}$ is *strictly quasiconvex* if for all points $x, y \in \mathbb{R}^d$, and $\lambda \in (0,1)$, we have $\widetilde{f}(\lambda x + (1-\lambda)y) < \max\{\widetilde{f}(x), \widetilde{f}(y)\}$ when $\widetilde{f}(x) \neq \widetilde{f}(y)$. A function $f : \mathbb{Z}^d \to \mathbb{R}$ is *discrete strictly quasiconvex* if there exists a quasiconvex function $\widetilde{f} : \mathbb{R}^d \to \mathbb{R}$ such that for all $x \in \mathbb{Z}^d$, $f(x) = \widetilde{f}(x)$.

Remark. After finishing our work on minimizing discrete convex functions, we discovered the paper by Veselov, Gribanov, Zolotykh and Chirkov [60] that independently showed the DCM problem can be solved using $O(\text{poly}(\log n))$ queries to the comparison oracle of f and valuation oracle of g, when the dimension d is a constant. The paper was published in *Discrete Applied Mathematics* in September 2020 and was uploaded on arXiv in November 2020. We started this project in early 2019 and submitted preliminary results to SODA in July 2019. It was not accepted at the conference and we continued working on this in 2020/2021 and strengthened the results to the form in which we present them here. The approach of Veselov et al. is different from ours, and we describe it in Section 5.5.

1.3.1 Other Related Work

The DCM problem has been solved in many different settings when given "richer" access to the function. But none of those results apply to our current setting.

The DCM problem can be solved efficiently, approximately, when given access to degree-one oracles to the convex extensions of the functions f and g, and the guarantee that the function extensions are Lipschitz and/or strongly convex (see, e.g., [27, 2, 4, 33, 18, 1]). A *degree-one* oracle to a function, when queried at a point, returns the gradient at the point along with the function value.

Dadush [19] gave the first algorithm that solves the DCM problem exactly, when given access to the degree-one oracles to the convex extensions of functions f, g, without any additional assumptions on the convex extensions. They show that there exists an algorithm with query complexity $O(\text{poly}(\log n))$ for a constant dimension d. Note that given a general

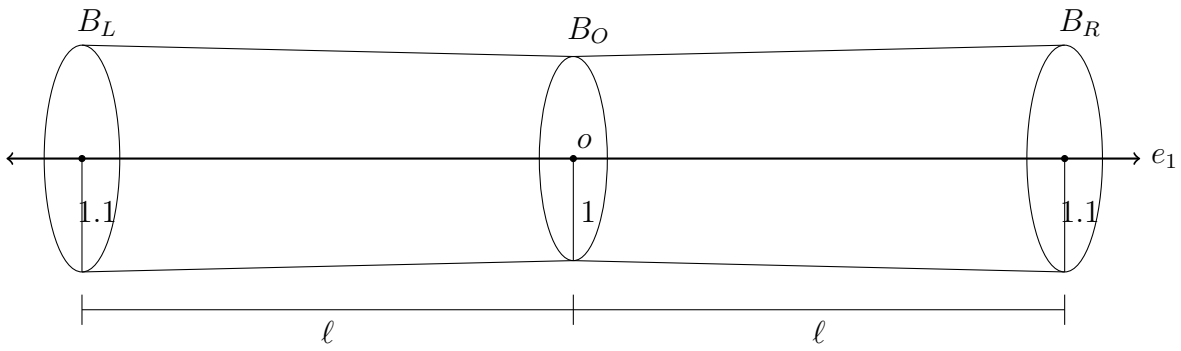

Figure 1.1: The body constructed by taking the union of the truncated cones $\text{conv}(B_L, B_O)$ and $\text{conv}(B_O, B_R)$, where B_L, B_R, and B_O are $(d-1)$-dimensional balls in the hyperplanes $\{x \in \mathbb{R}^d : x_1 = -\ell\}$, $\{x \in \mathbb{R}^d : x_1 = \ell\}$, and $\{x \in \mathbb{R}^d : x_1 = 0\}$, respectively.

discrete convex function, f, it is not possible to get the function values for a convex extension $\tilde{f}$ efficiently, unless the function is linear.

The only work we were aware of, while working on the problem, that attempts to solve the DCM problem without access to the convex extension of function f was by Larson et al. [32]. They assume access to the valuation oracles of the functions f, g and give an algorithm that in the worst case ends up querying all the points in the set $[n]^d$.

There is another line of work where they make additional structural assumptions (such as L-convexity, M-convexity) on the discrete convex function to get around the challenge that local optimality does not imply global optimality [43, 44, 26, 45]. The majority of the techniques in this line of work are inspired by the existing algorithms in continuous optimization like gradient descent.

1.4 Main Contributions

Though the problems we consider in this thesis are fundamental and widely applicable, not much is known about them. This is primarily because, though these problems are related to convexity, we don't have the convexity property. In the first two problems, the objects we deal with are not convex, and in the third, the domain of the function is not convex. Despite the absence of convexity property, we prove exciting and non-trivial results for these problems. We list our main contributions to each of the problems we study below.

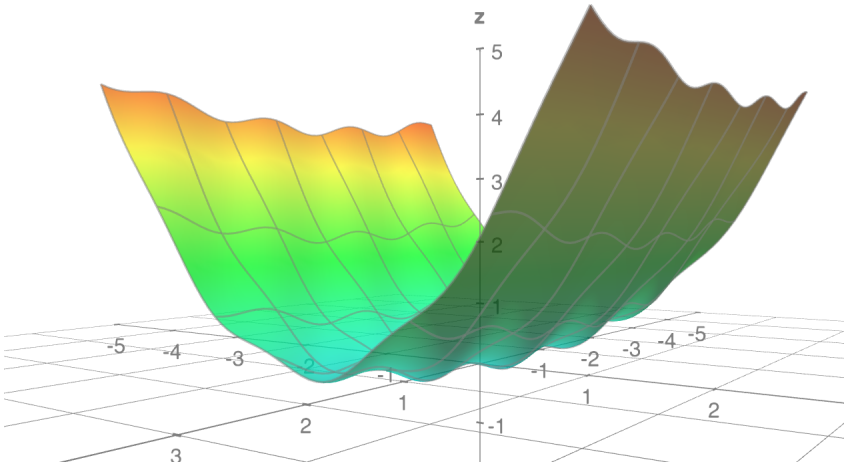

Figure 1.2: Illustration of a function "far" from convex, with all the violations of convexity along one direction.

Testing convex sets. The main contribution of the thesis towards this problem is the construction of a body that is "far" from convex for which the convex hull of points picked uniformly at random almost always lies completely inside. Intuitively, we expect such a body to be a convex body with some volume added to or deleted from the surface. In fact, these are the kinds of hard examples that have been considered to prove lower bounds for the problem of testing convex bodies and a related problem of learning convex bodies. Contrary to that intuition, the body we construct is close to being disconnected. Refer to Figure 1.1 for an illustration. Though the body looks close to a cylinder, if we plot the volume of the body with respect to the first coordinate, we observe that almost all the volume is towards the ends. This anomaly is because the volume of a ball of radius r in dimension d is proportional to r^d.

Testing discrete convex functions. The main contribution of the thesis towards the testing convexity of functions over the hypergrid problem is the construction that proves the lower bound for non-adaptive algorithms. We construct functions that are "far" from convex, with all the violations of convexity hidden in just one direction, i.e., unless we query two points along that direction, we can not discover that the function is not convex. There are several such possible directions, which make it hard for an algorithm to catch the violation. For a high-level explanation of the construction refer to Section 4.1, and for an illustration refer to Figure 1.2.

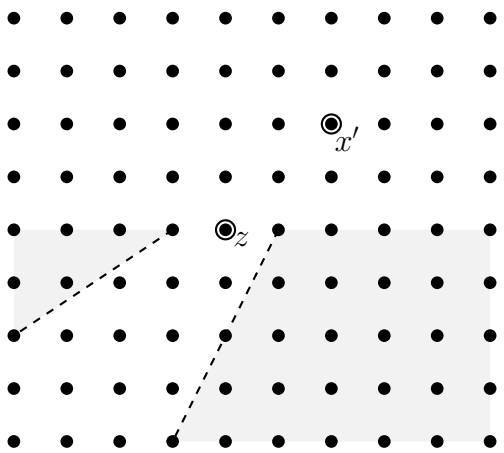

Figure 1.3: Illustration of the *fencing* idea in the minimization algorithm. The shaded region corresponds to the region that is "eliminated" when x' and z are the minimum in the corresponding rows with $f(x') \leq f(z)$.

Minimizing discrete convex functions. The main contribution to the minimizing discrete convex functions problem is a new algorithmic approach to minimization. This approach in two dimensions consists of four main ideas. The first is *fencing*, which is used to eliminate points in the domain as illustrated in Figure 1.3. The second is *finding fences*, which identifies rows to build fences so that after the fencing step, all the remaining points are "close" to a few lines. The third idea, *concentrating change of basis*, is that we can use a change of basis to transform all the points close to the line into a smaller grid and solve the sub-problem recursively. The fourth is an observation that all the points that are not eliminated are close to just two lines, which reduces the number of sub-problems to two. The high-dimensional minimization algorithm extends the ideas in two dimensions to higher dimensions. For a detailed overview of the main ideas in both the two-dimensional and high-dimensional minimization algorithms, refer to Section 5.1.

1.5 Organization

The thesis is divided into two parts. In the first part, we present our results related to convex sets and in the second part, we present the ones related to discrete convex functions. The fist part consists of one chapter, Chapter 2, in which we prove the results stated in Section 1.1 about the robustness of different characterizations of convex sets. The second part consists of three chapters, Chapters 3-5. In Chapter 3, we present some facts about

discrete convexity and lattices that we use in Chapters 4 and 5. In Chapters 4 and 5, we prove the results stated in Sections 1.2 and 1.3 about testing and minimization of discrete convex functions, respectively. Finally, in Chapter 6, we mention some interesting open problems in the areas we tackled in the thesis.

Part I

Convex Sets

Chapter 2

On Testing and Robust Characterizations of Convex Sets

In this chapter, we prove the results related to robust characterizations of convex bodies. In Section 2.1, we present the necessary background related to convex geometry to prove the theorems in this chapter. In Section 2.2, we prove that the line and convex hull characterizations of convex bodies are not robust, and in Section 2.3, we show that the supporting hyperplane characterization of convex bodies is robust.

2.1 Convex Geometry

We use standard notions and results regarding high-dimensional convex sets. For general introductions to the topic see [3, 28, 41, 27, 58].

2.1.1 Convex Bodies and Slices

For a body $K \subseteq \mathbb{R}^d$, a point $x \in K$ is on the *boundary* ∂K of K if there exists a direction $c \in \mathbb{R}^d$ such that for any constant $\delta > 0$, $x + \delta c \notin K$. We refer to the set $K \setminus \partial K$ as the *interior* of the body K. We use $\mathrm{Vol}(K)$ to denote the Lebesgue measure of the set $K \subset \mathbb{R}^d$. We use $\mathrm{Vol}_{d-1}(\partial K)$ to denote the $(d-1)$-dimensional Minkowski measure of the set $\partial K \subset \mathbb{R}^d$ which is defined as the limit of the volume of the $\frac{\delta}{2}$ neighbourhood of ∂K divided by δ, when $\delta \to 0$ (volume means Lebesgue measure). The Minkowski measure is used to measure the volume of the boundary.

The *distance* between two bodies $A, B \in \mathbb{R}^d$ is defined to be

$$\text{dist}(A, B) = \text{Vol}(A \triangle B) = \text{Vol}(A \setminus B) + \text{Vol}(B \setminus A),$$

the measure of the symmetric difference of the two bodies. We repeatedly use the following simple lower bound on the distance between two bodies.

Proposition 2.1. *The distance between two bodies $A, B \subset \mathbb{R}^d$ is bounded below by*

$$\text{dist}(A, B) \geq \max\big\{\text{Vol}(A) - \text{Vol}(B), \text{Vol}(B) - \text{Vol}(A)\big\}.$$

Furthermore, equality holds whenever $A \subseteq B$ or $B \subseteq A$.

Proof. The distance between A and B is bounded below by

$$\text{dist}(A, B) = \text{Vol}(A \setminus B) + \text{Vol}(B \setminus A) \geq \max\{\text{Vol}(A \setminus B), \text{Vol}(B \setminus A)\}.$$

The lower bound then follows from the observation that

$$\text{Vol}(A \setminus B) = \text{Vol}(A) - \text{Vol}(A \cap B) \geq \text{Vol}(A) - \text{Vol}(B)$$

and, similarly, that $\text{Vol}(B \setminus A) \geq \text{Vol}(B) - \text{Vol}(A)$.

Finally, when $A \subseteq B$, then $\text{Vol}(A \setminus B) = 0$ and $\text{Vol}(B \setminus A) = \text{Vol}(B) - \text{Vol}(A)$, as $\text{Vol}(A \cap B) = \text{Vol}(A)$ so equality holds. Similarly, equality also holds when $B \subseteq A$. □

Much of our analysis in Section 2.2 is concerned with various slices of a high-dimensional body. The t-th *slice* of a body $A \subset \mathbb{R}^d$ is

$$A_t = \{x \in A : x_1 = t\}.$$

For $t_1 \leq t_2 \in \mathbb{R}$, we also define $A_{[t_1,t_2]} = \{x \in A : t_1 \leq x_1 \leq t_2\}$ to be the set of points in A with first coordinate between t_1 and t_2.

A fundamental property of the slices of a convex body is that the $(d-1)$-th root of their volumes is a concave function.

Brunn's Theorem (Theorem 5.1 in [3])**.** *For any convex body $C \subset \mathbb{R}^d$, the function $t \mapsto \text{Vol}_{d-1}(C_t)^{\frac{1}{d-1}}$ is concave on its support.*

2.1.2 High-Dimensional Balls and Cones

We use $B_d(x,r) = \{y \in \mathbb{R}^d : \|y - x\| \leq r\}$ to denote the ball of radius r around a point $x \in \mathbb{R}^d$. We use $B(r)$ as a shorthand for $B_d(o,r)$, and $B_{d-1}(r)$ for $B_{d-1}(o,r)$, where o is the origin. Similarly, we use $S_d(x,r) = \{y \in \mathbb{R}^d : \|y - x\| = r\}$ to denote the sphere of radius r around a point $x \in \mathbb{R}^d$. We use the following standard approximation for the volume of the ball.

Proposition 2.2. *The volume of ball $B(r) \subset \mathbb{R}^d$ with radius r is*

$$\mathrm{Vol}(B(r)) = r^d \cdot \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)} = r^d \cdot \frac{1}{\sqrt{\pi d}}\Big(\frac{2\pi e}{d}\Big)^{\frac{d}{2}} \cdot (1 + O(d^{-1})),$$

where Γ is Euler's gamma function.

We also use the following standard concentration inequality for high-dimensional balls.

Proposition 2.3. *Let $x \in \mathbb{R}^d$ be drawn uniformly at random from $B(r)$. Then*

$$\Pr[|x_1| \geq \frac{r}{100}] \leq 2^{-\Omega(d)}.$$

The convex hull of a set is defined below.

Definition 2.4. For a set $X \subset \mathbb{R}^d$, a point $y \in \mathbb{R}^d$ is in the convex hull of X, $\mathrm{conv}(X)$, if for some $0 < k \in \mathbb{Z}$ there exist $x^{(1)}, \ldots, x^{(k)} \in X$ and $\lambda_1, \ldots, \lambda_k \geq 0$ with $\sum_{i=1}^k \lambda_i = 1$ such that $y = \sum_{i=1}^k \lambda_i x^{(i)}$.

Definition 2.5. Let $H \subset \mathbb{R}^d$ be a hyperplane, $S \subset H$ be a $(d-1)$-dimensional convex body, and $x \in \mathbb{R}^d \setminus H$ be a point outside the hyperplane. Then the *cone* with x as vertex and S as base is the convex hull of x with the body S

$$\mathrm{cone}(x, S) = \mathrm{conv}(x \cup S).$$

We use the following result on the volume of cones.

Proposition 2.6. *Let $H \subset \mathbb{R}^d$ be a hyperplane, $S \subset H$ be a $(d-1)$-dimensional convex body, and $x \in \mathbb{R}^d \setminus H$ be a point at a distance $h = \min_{y \in H} \|x - y\|$ from the hyperplane H. Then the volume of the cone is*

$$\mathrm{Vol}(\mathrm{cone}(x, S)) = \frac{h}{d}\mathrm{Vol}_{d-1}(S).$$

The above proposition can be used to obtain a relationship between the volume of a ball and volume of the boundary of a ball.

Proposition 2.7. *Let $B_d(x,r)$ be an d dimensional ball of radius r centered at x and $S_d(x,r)$ be the sphere of radius r around $x \in \mathbb{R}^n$. Then we have*

$$\frac{\mathrm{Vol}(B_d(x,r))}{\mathrm{Vol}_{d-1}(S_d(x,r))} = \frac{r}{d}.$$

Below is the definition of a truncated cone. Truncated cones play a crucial role in the proofs of Theorem 1.1 and Theorem 1.2.

Definition 2.8. A *truncated cone* is the convex hull of two balls $B_{d-1}((t_1, 0\ldots,0), r_1)$ and $B_{d-1}((t_2, 0\ldots,0), r_2)$, $\mathrm{conv}(B_{d-1}((t_1, 0\ldots,0), r_1) \cup B_{d-1}((t_2, 0\ldots,0), r_2))$.

2.2 Non-Robustness of Line and Convex Hull Characterizations

We complete the proofs of Theorems 1.1 and 1.2 in this section, showing that neither the line characterization nor convex hull characterization of convexity are robust. For the reader's convenience we restate the theorems below.

Theorem 1.1. *There exists a body $K \subset \mathbb{R}^d$ that is $\frac{1}{8}$-far from convex for which*

$$\Pr_{x,y \in K}\left[\overline{xy} \not\subseteq K\right] = 2^{-\Omega(d)}. \tag{1.1}$$

Theorem 1.2. *There exist a body $K \subset \mathbb{R}^d$ that is $\frac{1}{8}$-far from convex and a constant $c > 0$ such that a set $X = \{x^{(1)}, \ldots, x^{(m)}\} \in K$ of $m = 2^{cd}$ points drawn uniformly and independently at random from K satisfies*

$$\Pr_{X}\left[\mathrm{conv}(X) \not\subseteq K\right] = 2^{-\Omega(d)}.$$

Both non-robustness results are established by analyzing a construction obtained by taking the union of two truncated cones, as described in Section 2.2.1. The main technical component of the proofs is in Section 2.2.2, where we show that the union of truncated cones is far from convex. The proof of Theorem 1.1 is completed in Section 2.2.3, where we show that a line segment connecting two points drawn at random from the body is contained within the body with high probability. Finally, in Section 2.2.4, we generalize this result to show that the convex hull of a set of points picked uniformly at random lies inside the body with high probability.

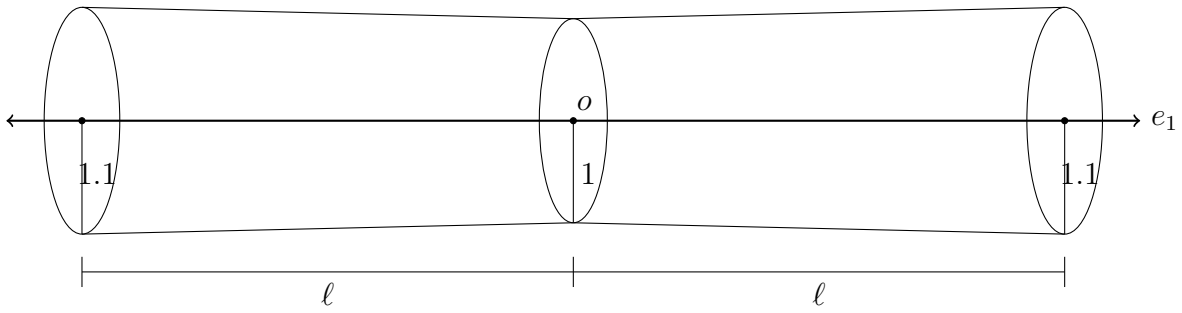

Figure 2.1: The body D obtained by taking the union of two truncated cones.

2.2.1 Union of Truncated Cones

The body $D \subset \mathbb{R}^d$ that will show the non-robustness of the line and convex hull definition is defined as follows. First, let $\ell > 0$ be some distance parameter. This distance parameter does not affect the results in the following sections; the reader may fix $\ell = 1$ for simplicity.

Let B_L, B_R, and B_O be three $(d-1)$-dimensional balls in the hyperplanes $\{x \in \mathbb{R}^d : x_1 = -\ell\}$, $\{x \in \mathbb{R}^d : x_1 = \ell\}$, and $\{x \in \mathbb{R}^d : x_1 = 0\}$, respectively. We define the balls B_L and B_R to have radius 1.1 each and be centered on the points $(-\ell, 0, 0, \ldots, 0)$ and $(\ell, 0, 0, \ldots, 0)$, respectively, while B_O has radius 1 and is centered at the origin. Define the body D to be the union of the truncated cones $\mathrm{conv}(B_L, B_O)$ and $\mathrm{conv}(B_O, B_R)$,

$$D = \mathrm{conv}(B_L, B_O) \cup \mathrm{conv}(B_O, B_R).$$

The definition of the body D is illustrated in Figure 2.1.

One of the basic properties of the body D that we will use in later sections is that a point drawn uniformly at random from D will have a large value in its first coordinate with high probability.

Proposition 2.9. *Let $x = (x_1, \ldots, x_d)$ be drawn uniformly at random from D. Then*

$$\Pr\left[|x_1| \leq \frac{\ell}{2}\right] = 2^{-\Omega(d)}.$$

Proof. Using the formula in Proposition 2.6 for the volume of a cone, the volume of the

body D is bounded below by

$$\begin{aligned}\operatorname{Vol}(D) = 2\operatorname{Vol}(D_{[0,\ell]}) &= 2\left(\frac{11\ell}{d}(1.1)^{d-1}\operatorname{Vol}_{d-1}\big(B_{d-1}(1)\big) - \frac{10\ell}{d}(1)^{d-1}\operatorname{Vol}_{d-1}(B_{d-1}(1))\right)\\ &\geq \frac{2\ell}{d}(1.1)^{d-1}\operatorname{Vol}_{d-1}(B_{d-1}(1)).\end{aligned}$$

Similarly, the volume of the body $D_{[-\frac{\ell}{2},\frac{\ell}{2}]}$ is bounded above by

$$\begin{aligned}\operatorname{Vol}(D_{[-\frac{\ell}{2},\frac{\ell}{2}]}) = 2\operatorname{Vol}(D_{[0,\frac{\ell}{2}]}) &= 2\left(\frac{10.5\ell}{d}(1.05)^{d-1}\operatorname{Vol}_{d-1}(B_{d-1}(1)) - \frac{10\ell}{d}(1)^{d-1}\operatorname{Vol}_{d-1}(B_{d-1}(1))\right)\\ &\leq \frac{21\ell}{d}(1.05)^{d-1}\operatorname{Vol}_{d-1}(B_{d-1}(1)).\end{aligned}$$

Therefore, the probability that the absolute value of the first coordinate of the point is less than or equal to $\frac{\ell}{2}$ is bounded by

$$\Pr\left[|x_1| \leq \frac{\ell}{2}\right] = \frac{\operatorname{Vol}(D_{[-\frac{\ell}{2},\frac{\ell}{2}]})}{\operatorname{Vol}(D)} \leq \frac{\frac{21\ell}{d}(1.05)^{d-1}\operatorname{Vol}_{d-1}(B_{d-1}(1))}{\frac{2\ell}{d}(1.1)^{d-1}\operatorname{Vol}_{d-1}(B_{d-1}(1))} \leq \frac{1}{2^{\Omega(d)}}.$$

□

2.2.2 D is Far from Convex

In this section, we prove that the body D is $\frac{1}{8}$-far from convex. We prove this in three steps. First, we show that the closest convex body to D must be symmetric about the axis $e_1 = (1, 0, 0, \ldots, 0)$. Next, we prove that if the closest convex body is symmetric about e_1, then it has to be a truncated cone. Finally, we prove that every truncated cone is $\frac{1}{8}$-far from our body D.

A partial converse to Brunn's Theorem

The first step in the proof—showing that the closest convex body to D must be symmetric—uses Brunn's Theorem as well as the following partial converse result.

Lemma 2.10. *Let $K \subset \mathbb{R}^d$ be a body such that for each $t \in \mathbb{R}$ the slice K_t is a $(d-1)$ dimensional ball and the function $t \mapsto \operatorname{Vol}_{d-1}(K_t)^{\frac{1}{d-1}}$ is concave on its support. Then K is convex.*

Proof. Assume for the sake of contradiction that K is a non-convex body that satisfies the conditions of the theorem. Then there exist points $x, y \in K$ and $0 \leq \lambda \leq 1$ such that the point $z = \lambda x + (1-\lambda)y \notin K$.

Let $r : \mathbb{R} \to \mathbb{R}$ be the function defined by setting $r(t)$ to be the radius of the $(d-1)$-dimensional ball with volume $\mathrm{Vol}_{d-1}(K_t)$. By applying the formula for volume of the ball from Proposition 2.2, we have that

$$r(t) = \frac{\Gamma(\frac{d-1}{2}+1)^{\frac{1}{d-1}}}{\sqrt{\pi}} \mathrm{Vol}_{d-1}(K_t)^{\frac{1}{d-1}}.$$

Since the function $t \mapsto \mathrm{Vol}_{d-1}(K_t)^{\frac{1}{d-1}}$ is concave, the function r is concave as well.

The concavity of r and the fact that $x, y \in K$ imply that

$$r(z_1) = r(\lambda x_1 + (1-\lambda)y_1) \geq \lambda r(x_1) + (1-\lambda) r(y_1) \geq \lambda \sqrt{\sum_{2 \leq i \leq d} x_i^2} + (1-\lambda)\sqrt{\sum_{2 \leq i \leq d} y_i^2}.$$

The fact that $z \notin K$ also implies that

$$r(z_1) < \sqrt{\sum_{2 \leq i \leq d} z_i^2} = \sqrt{\sum_{2 \leq i \leq d} (\lambda x_i + (1-\lambda)y_i)^2}.$$

But by the convexity of Euclidean norm and Jensen's inequality, $\sqrt{\sum_{2 \leq i \leq d}(\lambda x_i + (1-\lambda)y_i)^2} \leq \lambda\sqrt{\sum_{2 \leq i \leq d} x_i^2} + (1-\lambda)\sqrt{\sum_{2 \leq i \leq d} y_i^2}$ so the last two inequalities yield the desired contradiction. □

Symmetry of the closest convex body

A body $K \subset \mathbb{R}^d$ is *symmetric* about $e_1 = (1, 0, 0, \ldots, 0)$ if for every point $x \in K$, all points $y \in \mathbb{R}^d$ that satisfy $x_1 = y_1$ and $\sum_{i=2}^d x_i^2 = \sum_{i=2}^d y_i^2$ are also in K. In other words, a body K is symmetric about e_1 if it is invariant under rotations about the axis e_1.

We now prove that the closest convex body to D is symmetric about e_1.

Lemma 2.11. *There exists a closest convex body to D that is symmetric about e_1.*

Proof. Fix C to be any convex body which minimizes $\mathrm{dist}(D, C)$. Any point $x \in C$ should have $-\ell \leq x_1 \leq \ell$, otherwise we can truncate C and get a convex body closer to D. Let

C^s be the body where for every $t \in \mathbb{R}$, the slice C^s_t of the body is an $(d-1)$-dimensional ball centered at $(t, 0, 0, \ldots, 0)$ and has volume $\mathrm{Vol}_{d-1}(C^s_t)$ equal to the volume of the slice C_t of C. By this construction, C^s is symmetric about e_1. To complete the proof, we need to show that it satisfies $\mathrm{dist}(D, C^s) \leq \mathrm{dist}(D, C)$ and that it is convex.

We first establish the inequality $\mathrm{dist}(D, C^s) \leq \mathrm{dist}(D, C)$. The distance between D and C is

$$\mathrm{dist}(D, C) = \int_{t=-\ell}^{\ell} \mathrm{dist}(D_t, C_t)\, \mathrm{d}t = \int_{t=-\ell}^{\ell} \mathrm{Vol}_{d-1}(D_t \triangle C_t)\, \mathrm{d}t.$$

By Proposition 2.1, for every $t \in \mathbb{R}$, the volume of the symmetric difference between the slices D_t and C_t is bounded below by

$$\mathrm{Vol}_{d-1}(D_t \triangle C_t) \geq \max\Big\{\mathrm{Vol}_{d-1}(C_t) - \mathrm{Vol}_{d-1}(D_t), \mathrm{Vol}_{d-1}(D_t) - \mathrm{Vol}_{d-1}(C_t)\Big\}.$$

Since $\mathrm{Vol}_{d-1}(C_t) = \mathrm{Vol}_{d-1}(C^s_t)$, we then obtain

$$\mathrm{dist}(D, C) \geq \int_{t=-\ell}^{\ell} \max\Big\{\mathrm{Vol}_{d-1}(C^s_t) - \mathrm{Vol}_{d-1}(D_t), \mathrm{Vol}_{d-1}(D_t) - \mathrm{Vol}_{d-1}(C^s_t)\Big\}\, \mathrm{d}t.$$

Both D_t and C^s_t are balls with the same center, so one is a strict subset of the other and so we can apply the equality condition of Proposition 2.1 to obtain

$$\mathrm{dist}(D, C) \geq \int_{t=-\ell}^{\ell} \mathrm{Vol}_{d-1}(D_t \triangle C^s_t)\, \mathrm{d}t = \int_{t=-\ell}^{\ell} \mathrm{dist}(D_t, C^s_t)\, \mathrm{d}t = \mathrm{dist}(D, C^s),$$

as we wanted to show.

We now complete the proof of the lemma by showing that C^s is convex. From Brunn's Theorem, the function $t \mapsto \mathrm{Vol}_{d-1}(C_t)^{\frac{1}{d-1}}$ is concave on its support. And from the construction we have that $\mathrm{Vol}_{d-1}(C^s_t)^{\frac{1}{d-1}} = \mathrm{Vol}_{d-1}(C_t)^{\frac{1}{d-1}}$. Hence, the function $t \mapsto \mathrm{Vol}_{d-1}(C^s_t)^{\frac{1}{d-1}}$ is also concave on its support and by Lemma 2.10, the body C^s is convex. □

The closest convex body is a truncated cone

We now show that the closest symmetric convex body to D is a truncated cone. The proof of this claim uses the following standard result about the separation of convex and concave functions.

Lemma 2.12. *Fix any $\ell_1 \leq \ell_2 \in \mathbb{R}$. Let $f : [\ell_1, \ell_2] \to \mathbb{R}$ be a convex function and $g : [\ell_1, \ell_2] \to \mathbb{R}$ be a concave function such that $\forall t \in [\ell_1, \ell_2]$, $f(t) \geq g(t)$. Then there exists an affine function $h : [\ell_1, \ell_2] \to \mathbb{R}$ such that $g(t) \leq h(t) \leq f(t)$ for all $t \in [\ell_1, \ell_2]$.*

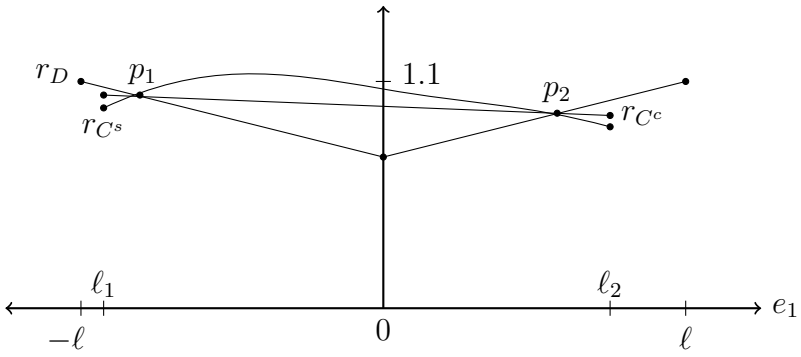

Figure 2.2: Illustration of the radius functions r_D, r_{C^s}, and r_{C^c} and of the points p_1 and p_2 in the proof of Lemma 2.13.

Proof. The proof follows from the fact that any two convex sets have a separating hyperplane. Let $S_1 = \{(t, x) : t \in [\ell_1, \ell_2], x \geq f(t)\}$ and $S_2 = \{(t, x) : t \in [\ell_1, \ell_2], x \leq g(t)\}$. The sets S_1 and S_2 are convex and their separating hyperplane corresponds to the function h. □

Lemma 2.13. *The closest convex body to D that is symmetric about e_1 is a truncated cone.*

Proof. Let C^s be a convex body that is symmetric about e_1 and minimizes $\text{dist}(D, C^s)$. We will construct a truncated cone C^c that is also symmetric about e_1 and satisfies $\text{dist}(D, C^c) \leq \text{dist}(C^s, D)$.

Define the functions $r_D, r_{C^s} : \mathbb{R} \to \mathbb{R}$ where $r_D(t)$ and $r_{C^s}(t)$ are the radii of the $(d-1)$-dimensional balls D_t and C^s_t, respectively. Let ℓ_1 be the infimum of t for which $r_{C^s}(t) > 0$ and ℓ_2 be the supremum of t for which $r_{C^s}(t) > 0$. Note that $-\ell \leq \ell_1 < \ell_2 \leq \ell$ since any point $x \in C^s$ satisfies $-\ell \leq x_1 \leq \ell$.

We define an affine function $r_{C^c} : [\ell_1, \ell_2] \to \mathbb{R}$. We further define C^c to be the body whose slices C^c_t are $(d-1)$-dimensional balls of radius $r_{C^c}(t)$, for $t \in [\ell_1, \ell_2]$. Clearly C^c is a truncated cone. We define r_{C^c} differently for different cases mentioned below. In Case 1 we define it directly and in Cases 2, 3 we define two values a_1, a_2. The affine function $r_{C^c} : [\ell_1, \ell_2] \to \mathbb{R}$ corresponding to the values a_1, a_2 is defined by the line joining the points $p_1 = (a_1, r_D(a_1)), p_2 = (a_2, r_D(a_2))$. See Figure 2.2 for an illustration of this construction.

- **Case 1:** $\forall t \in (\ell_1, \ell_2)$, $r_{C^s}(t) \leq r_D(t)$
  Since r_D is convex and r_{C^s} is concave, from Lemma 2.12, there exists an affine function $r_{C^c} : [\ell_1, \ell_2] \to \mathbb{R}$ such that $r_D(t) \geq r_{C^c}(t) \geq r_{C^s}(t)$ for all $t \in [\ell_1, \ell_2]$.

- **Case 2:** $\forall t \in (\ell_1, \ell_2)$, $r_{C^s}(t) > r_D(t)$

  Let $a_1 = \ell_1$, $a_2 = \ell_2$.

- **Case 3:** $\exists t_1, t_2 \in [\ell_1, \ell_2]$ such that $r_{C^s}(t_1) \leq r_D(t_1)$ and $r_{C^s}(t_2) > r_D(t_2)$

  This case be further divided into three sub-cases.

  - **Case 3a:** $r_{C^s}(\ell_1) \leq r_D(\ell_1)$ and $r_{C^s}(\ell_2) \leq r_D(\ell_2)$

    In this case since r_{C^s} is concave and r_D is convex the curves have exactly two points of intersection. Let a_1, a_2 be the values of t where the curves intersect.

  - **Case 3b:** $r_{C^s}(\ell_1) \leq r_D(\ell_1)$ and $r_{C^s}(\ell_2) > r_D(\ell_2)$

    In this case since r_{C^s} is concave and r_D is convex the curves have exactly one point of intersection. Let a_1 be the value of t where the curves intersect and let $a_2 = \ell_2$.

  - **Case 3c:** $r_{C^s}(\ell_1) > r_D(\ell_1)$ and $r_{C^s}(\ell_2) \leq r_D(\ell_2)$

    In this case since r_{C^s} is concave and r_D is convex the curves have exactly one point of intersection. Let a_2 be the value of t where the curves intersect and let $a_1 = \ell_1$.

Since the function r_{C^c} is affine, it is also concave and so by Lemma 2.10 the body C^c is convex. To complete the proof, we need to show that $\mathrm{dist}(D, C^c) \leq \mathrm{dist}(D, C^s)$ in all three cases.

By definition, the distance between D and C^s is

$$\mathrm{dist}(D, C^s) = \mathrm{Vol}(D \setminus C^s) + \mathrm{Vol}(C^s \setminus D) = \int_{-\ell}^{\ell} \mathrm{Vol}_{d-1}(D_t \setminus C^s_t) + \mathrm{Vol}_{d-1}(C^s_t \setminus D_t)\,\mathrm{d}t.$$

For Case 1, since D, C^s, C^c are symmetric about e_1 and $r_D(t) \geq r_{C^c}(t) \geq r_{C^s}(t)$ for every $t \in [\ell_1, \ell_2]$,

$$\begin{aligned}
\mathrm{dist}(D, C^s) &= \int_{-\ell}^{\ell_1} \mathrm{Vol}_{d-1}(D_t)\,\mathrm{d}t + \int_{\ell_1}^{\ell_2} \mathrm{Vol}_{d-1}(D_t) - \mathrm{Vol}_{d-1}(C^s_t)\,\mathrm{d}t + \int_{\ell_2}^{\ell} \mathrm{Vol}_{d-1}(D_t)\,\mathrm{d}t \\
&\geq \int_{-\ell}^{\ell_1} \mathrm{Vol}_{d-1}(D_t)\,\mathrm{d}t + \int_{\ell_1}^{\ell_2} \mathrm{Vol}_{d-1}(D_t) - \mathrm{Vol}_{d-1}(C^c_t)\,\mathrm{d}t + \int_{\ell_2}^{\ell} \mathrm{Vol}_{d-1}(D_t)\,\mathrm{d}t \\
&= \mathrm{dist}(D, C^c).
\end{aligned}$$

For Cases 2 and 3, for $t \in (\ell_1, a_1) \cup (a_2, \ell_2)$, the ball D_t contains the ball C^s_t. Conversely, for every $t \in [a_1, a_2]$, C^s_t contains D_t. And for $t \in (-\ell, \ell_1) \cup (\ell_2, \ell)$ the ball C^s_t has zero radius Hence, the distance between D and C^s is

$$\begin{aligned}\operatorname{dist}(D, C^s) = \int_{-\ell}^{\ell_1} \operatorname{Vol}_{d-1}(D_t)\,\mathrm{d}t + \int_{\ell_2}^{\ell} \operatorname{Vol}_{d-1}(D_t)\,\mathrm{d}t + \int_{\ell_1}^{a_1} \operatorname{Vol}_{d-1}(D_t) - \operatorname{Vol}_{d-1}(C^s_t)\,\mathrm{d}t \\ + \int_{a_1}^{a_2} \operatorname{Vol}_{d-1}(C^s_t) - \operatorname{Vol}_{d-1}(D_t)\,\mathrm{d}t + \int_{a_2}^{\ell_2} \operatorname{Vol}_{d-1}(D_t) - \operatorname{Vol}_{d-1}(C^s_t)\,\mathrm{d}t.\end{aligned}$$

For every $t \in (\ell_1, a_1) \cup (a_2, \ell_2)$, we have that $r_D(t) \geq r_{C^c}(t) \geq r_{C^s}(t)$. And for every $t \in (a_1, a_2)$, we have the reverse inequalities $r_D(t) \leq r_{C^c}(t) \leq r_{C^s}(t)$. Therefore,

$$\begin{aligned}\operatorname{dist}(D, C^s) &\geq \int_{-\ell}^{\ell_1} \operatorname{Vol}_{d-1}(D_t)\,\mathrm{d}t + \int_{\ell_2}^{\ell} \operatorname{Vol}_{d-1}(D_t)\,\mathrm{d}t + \int_{\ell_1}^{a_1} \operatorname{Vol}_{d-1}(D_t) - \operatorname{Vol}_{d-1}(C^c_t)\,\mathrm{d}t \\ &\quad + \int_{a_1}^{a_2} \operatorname{Vol}_{d-1}(C^c_t) - \operatorname{Vol}_{d-1}(D_t)\,\mathrm{d}t + \int_{a_2}^{\ell_2} \operatorname{Vol}_{d-1}(D_t) - \operatorname{Vol}_{d-1}(C^c_t)\,\mathrm{d}t \\ &= \operatorname{Vol}(D \setminus C^c) + \operatorname{Vol}(C^c \setminus D) = \operatorname{dist}(D, C^c).\end{aligned}$$

□

Every truncated cone is far from D

As the last step in the proof that D is far from convex, we show that it is far from every truncated cone.

Lemma 2.14. *Every truncated cone is $\frac{1}{8}$-far from D.*

Proof. Let C^c be a truncated cone. Without loss of generality let the truncated cone have larger radius towards the left side. We consider the two cases where $\operatorname{Vol}(C^c_{[0,\ell]}) \leq \frac{1}{2}\operatorname{Vol}(D_{[0,\ell]})$ and where $\operatorname{Vol}(C^c_{[0,\ell]}) > \frac{1}{2}\operatorname{Vol}(D_{[0,\ell]})$ separately.

Case 1: $\operatorname{Vol}(C^c_{[0,\ell]}) \leq \frac{1}{2}\operatorname{Vol}(D_{[0,\ell]})$.

In this case, Proposition 2.1 and the case condition yield

$$\operatorname{Vol}(D \triangle C^c) \geq \operatorname{Vol}(D_{[0,\ell]} \triangle C^c_{[0,\ell]}) \geq \operatorname{Vol}(D_{[0,\ell]}) - \operatorname{Vol}(C^c_{[0,\ell]}) \geq \frac{1}{2}\operatorname{Vol}(D_{[0,\ell]}) = \frac{1}{4}\operatorname{Vol}(D).$$

Case 2: $\operatorname{Vol}(C^c_{[0,\ell]}) \geq \frac{1}{2}\operatorname{Vol}(D_{[0,\ell]})$

In this case, if $C^c_{[-\ell,-\frac{\ell}{2}]} = \emptyset$, then from Proposition 2.9

$$\mathrm{Vol}(D \triangle C^c) \geq \mathrm{Vol}(D_{[-\ell,-\frac{\ell}{2}]} \triangle C^c_{[-\ell,-\frac{\ell}{2}]}) = \mathrm{Vol}(D_{[-\ell,-\frac{\ell}{2}]}) \geq \frac{1}{4}\mathrm{Vol}(D).$$

If $C^c_{[-\ell,-\frac{\ell}{2}]} \neq \emptyset$, then using the fact that C^c is a truncated cone with larger radius on the left side we get

$$\mathrm{Vol}(C^c_{[-\frac{\ell}{2},\frac{\ell}{2}]}) \geq \mathrm{Vol}(C^c_{[0,\ell]}) \geq \frac{1}{2}\mathrm{Vol}(D_{[0,\ell]}) \geq \frac{1}{4}\mathrm{Vol}(D).$$

Then Proposition 2.1 implies that

$$\mathrm{Vol}(D \triangle C^c) \geq \mathrm{Vol}(D_{[-\frac{\ell}{2},\frac{\ell}{2}]} \triangle C^c_{[-\frac{\ell}{2},\frac{\ell}{2}]}) \geq \mathrm{Vol}(C^c_{[-\frac{\ell}{2},\frac{\ell}{2}]}) - \mathrm{Vol}(D_{[-\frac{\ell}{2},\frac{\ell}{2}]}) \geq \frac{1}{4}\mathrm{Vol}(D) - \mathrm{Vol}(D_{[-\frac{\ell}{2},\frac{\ell}{2}]}).$$

From Proposition 2.9, we also have that $\mathrm{Vol}(D_{[-\frac{\ell}{2},\frac{\ell}{2}]})$ is exponentially smaller than $\mathrm{Vol}(D)$. Hence,

$$\mathrm{Vol}(D \triangle C_C) \geq \frac{1}{4}\mathrm{Vol}(D) - o\big(\mathrm{Vol}(D)\big) \geq \frac{1}{8}\mathrm{Vol}(D).$$

□

Putting our last three lemmas together completes the proof of the main result from this section.

Theorem 2.15. *The body D is $\frac{1}{8}$-far from convex.*

Remark. In fact, the above argument shows that D is $(\frac{1}{4} - o(1))$-far from convex. With more careful calculations, it is possible to show that D is $(\frac{1}{2} - o(1))$-far from convex. This result is tight, since the body D is $(\frac{1}{2} - o(1))$-close to the convex body obtained by deleting the right half of D and extending the truncated cone in the left half to ℓ.

2.2.3 Non-Robustness of the Line Characterization

We are now ready to prove Theorem 1.1 by showing that when two points x and y are drawn uniformly at random from D, then with high probability the line segment $\overline{xy}$ that connects x to y is completely contained within D.

Theorem 2.16. *When $x, y \in D$ are drawn uniformly at random from D, then the line segment $\overline{xy}$ that joins x and y satisfies*

$$\Pr\left[\overline{xy} \not\subseteq D\right] = 2^{-\Omega(d)}.$$

Proof. Let $x = (\alpha, x_2, \ldots, x_d)$ and y be drawn independently and uniformly at random from D. By the symmetry of D with respect to reflection on the axis e_1, we can assume without loss of generality that $\alpha \leq 0$. Furthermore, since D is symmetric with respect to rotations around e_1, we can also assume that $x_2 \geq 0$ and rest of the $x_i = 0$. Hence, without loss of generality let $x = (\alpha, x_2, 0, 0, \ldots, 0)$ and let $y = (\beta, y_2, \ldots, y_d)$.

If $\beta \leq 0$, then both x and y lie in the same half of D, and that half is a convex set so the line segment $\overline{xy}$ is contained in D.

Consider the case now where $\beta > 0$. By Proposition 2.9, with probability $1 - 2^{-\Omega(d)}$ we have $\alpha \leq -\ell/2$ and $\beta \geq \ell/2$. Furthermore, for any given β since $(y_2, \ldots, y_d)$ is uniformly distributed over an $(d-1)$-dimensional ball of radius at most 1.1, from Proposition 2.3 we have that $\Pr[|y_2| \leq \frac{1}{10}] = 1 - 2^{-\Omega(d)}$. In the rest of the proof, assume that all three inequalities $\alpha \leq -\frac{\ell}{2}$, $\beta \geq \frac{\ell}{2}$, and $|y_2| \leq \frac{1}{10}$ hold. We will show that in this case, the line passes through the center slice D_0 and, therefore, the line segment $\overline{xy}$ is contained in D, thus completing the proof of the theorem.

Consider the point $z = \frac{|\alpha|}{|\alpha|+|\beta|}(\beta, y_2, \ldots, y_d) + \frac{|\beta|}{|\alpha|+|\beta|}(\alpha, x_2, 0, 0, \ldots, 0)$. The point z has $z_1 = 0$. We want to show that it is contained in the slice D_0 or, equivalently, that $\|z\|^2 \leq 1$. By definition,

$$\begin{aligned}
\|z\|^2 &= \left(\frac{1}{|\alpha| + |\beta|}\right)^2 \left(|\alpha| y_2 + |\beta| x_2\right)^2 + \left(\frac{|\alpha|}{|\alpha| + |\beta|}\right)^2 \sum_{i=3}^{d-1} y_i^2 \\
&= \left(\frac{|\beta| x_2}{|\alpha| + |\beta|}\right)^2 + \left(\frac{1}{|\alpha| + |\beta|}\right)^2 2|\alpha||\beta| x_2 y_2 + \left(\frac{|\alpha|}{|\alpha| + |\beta|}\right)^2 \sum_{i=2}^{d} y_i^2.
\end{aligned}$$

Since x and y are in D, then $\sum_{i=2}^{d} y_i^2 \leq (1.1)^2$ and $x_2 \leq 1.1$. And we have that $y_2 \leq 0.1$. Substituting these bounds into the above expression, we obtain

$$\|z\|^2 \leq \left(\frac{1.1|\beta|}{|\alpha| + |\beta|}\right)^2 + \left(\frac{1}{|\alpha| + |\beta|}\right)^2 .22|\alpha||\beta| + \left(\frac{1.1|\alpha|}{|\alpha| + |\beta|}\right)^2 = (1.1)^2 - 2.2|\alpha||\beta| \left(\frac{1}{|\alpha| + |\beta|}\right)^2.$$

Defining $\delta = |\alpha|/|\beta|$, the above equation simplifies to $\|z\|^2 \leq 1.21 - 2.2\frac{\delta}{(1+\delta)^2}$. Since $|\alpha|$ and $|\beta|$ are both in the range $[\frac{\ell}{2}, \ell]$, then $\delta \in [\frac{1}{2}, 2]$. The minimum value of the function $\frac{\delta}{(1+\delta)^2}$ in the interval $[\frac{1}{2}, 2]$ is $\frac{2}{9}$, so $\|z\|^2 \leq 1.21 - 2.2 \cdot \frac{2}{9} \leq 1$. □

Theorem 1.1 follows immediately from Theorems 2.15 and 2.16.

2.2.4 Non-Robustness of the Convex Hull Characterization

In this section, we complete the proof of Theorem 1.2 by combining Theorem 2.16 with the following structural result about the body D.

Lemma 2.17. *For any finite set $X \subset D$, if the line connecting any two points $x, y \in X$ satisfies $\overline{xy} \subseteq D$, then*

$$\text{conv}(X) \subseteq D.$$

Proof. We prove the claim by induction on the number of points in X. The base case where $|X| = 2$ is trivially true. For the base case where $|X| = 3$, let $X = \{x, y, z\}$ be any set that satisfies $\overline{xy}, \overline{xz}, \overline{yz} \subseteq D$. We can assume without loss of generality that $x, y \in D_{\geq 0}$. If $z \in D_{\geq 0}$ as well, then $\text{conv}(X) \subseteq D$ since $D_{\geq 0}$ is a convex set. Let us now consider the case where $z \in D_{<0}$. Note that a line joining two points $a \in D_{\leq 0}, b \in D_{\geq 0}$ is contained in the body if and only if $\overline{ab} \cap D_0 \neq \emptyset$. From this observation, we get that $\overline{xz} \cap D_0 \neq \emptyset$ and $\overline{yz} \cap D_0 \neq \emptyset$. Define $x' = \overline{xz} \cap D_0$ and $y' = \overline{yz} \cap D_0$. Let w be any point on the line $\overline{xy}$ and define $w' \in \overline{zw}$ be such that $w'_1 = 0$. Since $w \in \overline{xy}$, we have that $w' \in \overline{x'y'}$. Since $x', y' \in D_0$ and D_0 is convex, we must also have that $w' \in D_0$, and so $\overline{zw} \subseteq D$. Since every point in the convex hull of X is on the line $\overline{zw}$ for some $w \in \overline{xy}$, this means that $\text{conv}(X) \subseteq D$.

For the induction step, we assume that the claim is true for all sets with at most k points for some fixed $k \geq 2$. Fix any set $X \subseteq D$ with $k+1$ elements such that every line $\overline{xy}$ connecting $x, y \in X$ is contained in D. We want to show that $\text{conv}(X) \subseteq D$.

Let $x \in X$ be an element for which there exists $y \in X$ that satisfies $x_1 y_1 \geq 0$, i.e x, y are in the same half of D. Such an element x is guaranteed to exist since $|X| \geq 3$. Without loss of generality, assume $x \in D_{\leq 0}$. Define $X_k = X \setminus \{x\}$. By the induction hypothesis, we must have that $\text{conv}(X_k) \subseteq D$. Furthermore, if every $x' \in \text{conv}(X_k)$ satisfies $\overline{x'x} \subseteq D$, then $\text{conv}(X) = \text{conv}(x \cup X_k) \subseteq D$. To complete the proof, let us now assume that there exists $x' \in \text{conv}(X_k)$ for which $\overline{x'x} \not\subseteq D$ and show that this leads to a contradiction.

Define $X_1 = \{y : y \in X_k \cap D_{\leq 0}\}$ and $X_2 = X_k \setminus X_1$. By our choice of x, $|X_1| \geq 1$ and so $|X_2| \leq k - 1$. And since $x' \in \text{conv}(X_k) = \text{conv}(X_1 \cup X_2)$, there exist two points $x'' \in \text{conv}(X_1)$ and $x''' \in \text{conv}(X_2)$ such that $x' \in \overline{x''x'''}$. We have $\overline{xx''} \subseteq D$ as $x, x'' \in D_{\leq 0}$ and $D_{\leq 0}$ is convex. And $\overline{xx'''} \subseteq D$ because $\text{conv}(\{x\} \cup X_2) \subseteq D$ from the induction hypothesis. Finally, since $x'', x''' \in \text{conv}(X_k)$ we also have that $\overline{x''x'''} \subseteq D$. Hence, the three points x, x'', x''' satisfy $\overline{xx''} \subseteq D$, $\overline{xx'''} \subseteq D$, and $\overline{x''x'''} \subseteq D$. Therefore, from the induction hypothesis on the set $\{x, x'', x'''\}$, $\text{conv}(x, x'', x''') \subseteq D$. This implies $\overline{xx'} \subseteq D$, which is a contradiction. Therefore, $\text{conv}(X) = \text{conv}(\{x\} \cup X_K) \subseteq D$. □

There exists a small constant $c > 0$ such that if we pick $m = 2^{cd}$ points, X, uniformly at random then the probability that $\forall x, y \in X$, $\overline{xy} \subset D$ is greater than $1 - \frac{1}{2^{\Omega(d)}}$. We get this by applying a union bound on Theorem 2.16. This combined with Lemma 2.17 completes the proof of Theorem 1.2.

2.3 Robustness of the Supporting Hyperplane Characterization

In this section we prove Theorem 1.3 and Proposition 1.4. For the reader's convenience we restate them below.

Theorem 1.3. *Fix some $R > r > 0$. For every body $K \subset \mathbb{R}^d$ that satisfies $B(r) \subseteq K \subseteq B(R)$ and is ϵ-far from convex, when $x \in \mathbb{R}^d$ is drawn uniformly at random from the boundary ∂K of K then*

$$\Pr_{x \in \partial K}\left[\forall c \in \mathbb{R}^d \, \exists y \in K \text{ s.t. } \langle c, y - x\rangle < 0\right] \geq \frac{\epsilon r}{3Rd}. \tag{1.2}$$

Proposition 1.4. *There is an ϵ-tester for convexity of bodies $K \subset \mathbb{R}^d$ that satisfy the promise $B(r) \subseteq K \subseteq B(R)$ which makes $\mathrm{poly}(d, \frac{1}{\epsilon}, \log \frac{R}{r})$ queries to the membership and separation oracles for K.*

The proof of Theorem 1.3 is obtained by considering a body K that is contained within a convex body $C \subset \mathbb{R}^d$. We establish an isoperimetric-type inequality that gives a lower bound on the fraction of the boundary of K that must be strictly contained within C.

Lemma 2.18. *Fix some $\gamma > 0$ and $R > r > 0$. Let $C \subset \mathbb{R}^d$ be a convex body and let $K \subseteq C$ be a body that satisfies $B(r) \subseteq K \subseteq B(R)$ and $\mathrm{Vol}(K) \leq \frac{1}{1+\gamma}\mathrm{Vol}(C)$. Then*

$$\mathrm{Vol}_{d-1}(\partial K \setminus \partial C) \geq \frac{\min\{1, \gamma\} \cdot r}{3Rd} \mathrm{Vol}_{d-1}(\partial K).$$

When $\mathrm{Vol}_{d-1}(\partial K) \geq \frac{d}{r}\mathrm{Vol}(K)$, Lemma 2.18 gives a stronger lower bound on $\mathrm{Vol}_{d-1}(\partial K \setminus \partial C)$ than the the classic isoperimetric inequality for convex bodies of Lovász and Simonovits [36].

Lovász–Simonovits isoperimetric inequality. *Let $C \subset \mathbb{R}^d$ be a convex body with diameter R and let $K \subseteq C$. Then*

$$\mathrm{Vol}_{d-1}(\partial K \setminus \partial C) \geq \frac{2}{R} \min\left\{\mathrm{Vol}(K), \mathrm{Vol}(C \setminus K)\right\}.$$

For instance, when $\iota = \frac{1}{2^{2^d}}$ is some small value and $K = B(1) \cup \{x \in \mathbb{R}^d : d - \iota \leq \|x\| \leq d\}$ is the union of a unit ball and a thin shell of radius $R = d$, then the volume of K and of its boundary satisfies $\mathrm{Vol}_{d-1}(\partial K) \geq 2^{\Omega(d)}\mathrm{Vol}(K)$ and the bound in Lemma 2.18 is much stronger.

We prove the isoperimetric-type inequality of Lemma 2.18 in Section 2.3.1 and show how it implies the robustness of the supporting hyperplane definition of convexity of Theorem 1.3 in Section 2.3.2. We establish the tightness of Theorem 1.3 in Section 2.3.3 and finally establish Proposition 1.4 regarding convexity testing with a separating hyperplane oracle in Section 2.3.4.

2.3.1 Isoperimetric-type Inequality

We complete the proof of Lemma 2.18 in this section. Let us fix two bodies K and C that satisfy the conditions of the theorem. We define $E = \partial K \cap \partial C$ to be the portion of the boundary of K that is also on the boundary of the containing convex body C. We refer to E as the *external boundary* of K(with respect to C). Let $F = \partial K \setminus \partial C$ be the portion of the boundary of K that is strictly contained in C. We refer to this as the *internal boundary* of K (with respect to C). Lemma 2.18 gives a lower bound on the measure of the internal boundary of K with respect to C.

A central concept in the proof of Lemma 2.18 is the notion of *generalized cones*, which we define as follows. Throughout this section, we use o to refer to the origin in $\mathbb{R}^d$.

Definition 2.19. A set $S \subset \mathbb{R}^d \setminus \{o\}$ is *origin-visible* if for every two points $x, x' \in S$ satisfy $x' \notin \overline{ox}$. The *generalized cone* of an origin-visible S is the set

$$S^\vee = \{y : \exists x \in S \text{ such that } y \in \overline{ox}\}.$$

We use the following bounds relating the volume of the generalized cone of S to the volume of S when S is a subset of the boundary of a convex set.

Lemma 2.20. *Fix any $R > r > 0$. Let $B(r) \subset C \subset B(R)$ be a convex body and $S \subseteq \partial C$. Then*

$$\frac{r}{d}\mathrm{Vol}_{d-1}(S) \leq \mathrm{Vol}(S^\vee) \leq \frac{R}{d}\mathrm{Vol}_{d-1}(S).$$

Proof. Let dx be an infinitesimally small element in S. Then $(dx)^\vee$ is a (standard) cone from the origin with base dx. By Proposition 2.6, we have $\mathrm{Vol}((dx)^\vee) = \frac{h(x)}{d}\mathrm{Vol}_{d-1}(dx)$

where $h(x)$ is the distance of the origin from the supporting hyperplane at x. This gives us

$$\mathrm{Vol}(S^\vee) = \int_{x\in S} \mathrm{Vol}((dx)^\vee) = \int_{x\in S} \frac{h(x)}{d}\mathrm{Vol}_{d-1}(dx).$$

Since x also belongs to ∂C and $B(r) \subset C \subset B(R)$ we have the distance of the origin from the supporting hyperplane at x is between r and R i.e. $r \le h(x) \le R$. Therefore

$$\int_{x\in S} \frac{h(x)}{d}\mathrm{Vol}_{d-1}(dx) \ge \int_{x\in S} \frac{r}{d}\mathrm{Vol}_{d-1}(dx) = \frac{r}{d}\mathrm{Vol}_{d-1}(S)$$

and

$$\int_{x\in S} \frac{h(x)}{d}\mathrm{Vol}_{d-1}(dx) \le \int_{x\in S} \frac{R}{d}\mathrm{Vol}_{d-1}(dx) = \frac{R}{d}\mathrm{Vol}_{d-1}(S).$$

□

We complete the proof of Lemma 2.18 by considering the cases where $\mathrm{Vol}(K) \ge \frac{1}{2}\mathrm{Vol}(E^\vee)$ and $\mathrm{Vol}(K) < \frac{1}{2}\mathrm{Vol}(E^\vee)$ separately.

Case 1: $\mathrm{Vol}(K) \ge \frac{1}{2}\mathrm{Vol}(E^\vee)$.

By the Lovász–Simonovits isoperimetric inequality, the volume of the internal boundary F of K is bounded below by

$$\mathrm{Vol}_{d-1}(F) \ge \frac{2}{R}\min\big\{\mathrm{Vol}(K), \mathrm{Vol}(C\setminus K)\big\}.$$

The theorem assumption that $\mathrm{Vol}(K) \le \frac{1}{1+\gamma}\mathrm{Vol}(C)$ guarantees that $\mathrm{Vol}(C\setminus K) \ge \gamma\mathrm{Vol}(K)$ and so

$$\mathrm{Vol}_{d-1}(F) \ge \frac{2\cdot\min\{\gamma,1\}}{R}\mathrm{Vol}(K).$$

Note that the volume of K does not immediately give any useful bound on the volume of the boundary ∂K. However, we can now use our case assumption that $\mathrm{Vol}(K) \ge \frac{1}{2}\mathrm{Vol}(E^\vee)$ and the lower bound $\mathrm{Vol}(E^\vee) \ge \frac{r}{d}\mathrm{Vol}_{d-1}(E)$ from Lemma 2.20 to obtain

$$\mathrm{Vol}_{d-1}(F) \ge \frac{\min\{1,\gamma\}}{R}\mathrm{Vol}(E^\vee) \ge \frac{\min\{1,\gamma\}\,r}{Rd}\mathrm{Vol}_{d-1}(E).$$

The sets E and F partition the boundary ∂K, so $\mathrm{Vol}_{d-1}(E) = \mathrm{Vol}_{d-1}(\partial K) - \mathrm{Vol}_{d-1}(F)$ and substituting this identity in the above inequality yields

$$\left(1 + \frac{\min\{1,\gamma\}\,r}{Rd}\right)\mathrm{Vol}_{d-1}(F) \ge \frac{\min\{1,\gamma\}\,r}{Rd}\mathrm{Vol}_{d-1}(\partial K).$$

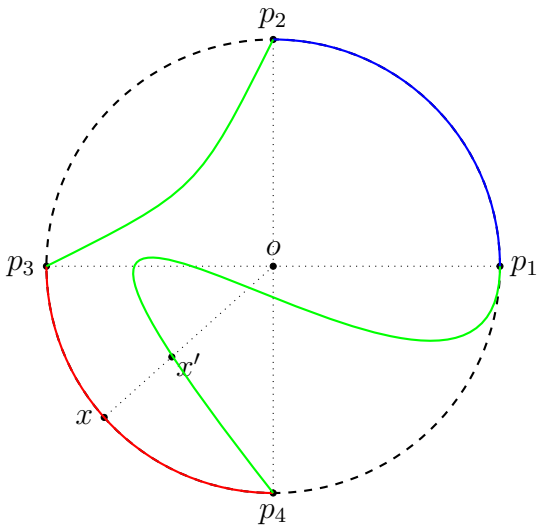

Figure 2.3: In this figure the convex body C is the ball centered at o. The boundary of the body K is defined by the solid curves p_1, p_2, p_3, p_4, p_1. The external boundary of K with respect to C, E, are the arcs p_1, p_2 and p_3, p_4. The internal boundary of K with respect to C, F, are the curves p_2, p_3 and p_4, p_1. The visible external boundary, V, is the blue arc p_1, p_2. The invisible external boundary, I, is the red arc p_3, p_4. I' is the set of x' corresponding to the set of $x \in I$. The set $V^\vee$ is the sector o, p_1, p_2. The set $I^\vee$ is the sector o, p_3, p_4. The set $E^\vee$ is the union of sets $V^\vee, I^\vee$.

The trivial bound $\frac{r}{Rd} < 1$ implies that $\left(1 + \frac{\min\{1,\gamma\}\, r}{Rd}\right) < 2$ and so

$$\mathrm{Vol}_{d-1}(F) \geq \frac{\min\{1,\gamma\}\, r}{2Rd} \mathrm{Vol}_{d-1}(\partial K),$$

completing the proof of Case 1.

Case 2: $\mathrm{Vol}(K) < \frac{1}{2}\mathrm{Vol}(E^\vee)$.

We say a point $x \in \mathbb{R}^d$ is *visible* from the origin if the line $\overline{ox}$ joining the origin o to the point x lies completely inside K.[1] Otherwise we say x is *invisible* from the origin. Partition the external boundary E of K into the two sets V, I, where V is the set of points in E that are visible from the origin and I is the set of points in E invisible from the origin.

[1] I.e. if $\overline{ox} \subseteq K$. Note that by the theorem's assumption that $B(r) \subset K$, the origin is inside K.

For every point $x \in I$, the line joining the origin o with x can intersect the boundary of K at multiple points. For any fixed $x \in I$, define $\phi_K(x)$ to be the point in $\overline{ox} \cap \partial K \setminus \{x\}$ that is closest to x. From this definition, the line joining x and $\phi_K(x)$ lies completely inside the body K. Define

$$I' = \{\phi_K(x) : x \in I\}.$$

Note that for each point $x \in I$, the point $\phi_K(x)$ lies on the boundary of K and does not lie on the boundary of C, so $\phi_K(x) \in F$ and, therefore, $I' \subseteq F$ is a subset of the internal boundary of K. From this observation, the definition of generalized cones, and the fact that $I' \subset B(R)$ we get

$$\mathrm{Vol}_{d-1}(F) \geq \mathrm{Vol}_{d-1}(I') \geq \frac{1}{R}\mathrm{Vol}(I'^{\vee}). \tag{2.1}$$

We now show that the volume of $I'^{\vee}$ cannot be much smaller than that of $E^{\vee}$.

Claim 2.21. $\mathrm{Vol}(I'^{\vee}) > \frac{1}{2}\mathrm{Vol}(E^{\vee})$.

Proof. Since V and I are disjoint subsets of the boundary of the convex body C, the generalized cones $V^{\vee}$ and $I^{\vee}$ are disjoint and so

$$\mathrm{Vol}(K) \geq \mathrm{Vol}(K \cap V^{\vee}) + \mathrm{Vol}(K \cap I^{\vee}).$$

By the definition of visible points, $K \cap V^{\vee} = V^{\vee}$. And by the definition of I', $I^{\vee} \setminus I'^{\vee} \subseteq K$. Clearly, $I^{\vee} \setminus I'^{\vee} \subseteq I^{\vee}$ as well so $I^{\vee} \setminus I'^{\vee} \subseteq K \cap I^{\vee}$. Therefore,

$$\mathrm{Vol}(K) \geq \mathrm{Vol}(V^{\vee}) + \mathrm{Vol}(I^{\vee} \setminus I'^{\vee}) = \mathrm{Vol}(V^{\vee}) + \mathrm{Vol}(I^{\vee}) - \mathrm{Vol}(I'^{\vee}) = \mathrm{Vol}(E^{\vee}) - \mathrm{Vol}(I'^{\vee}).$$

By the Case 2 assumption, $\mathrm{Vol}(K) < \frac{1}{2}\mathrm{Vol}(E^{\vee})$ so $\mathrm{Vol}(I'^{\vee}) > \frac{1}{2}\mathrm{Vol}(E^{\vee})$. □

Applying the inequality in Claim 2.21 and the lower bound in Lemma 2.20 to inequality (2.1), we obtain

$$\mathrm{Vol}_{d-1}(F) \geq \frac{1}{2R}\mathrm{Vol}(E^{\vee}) \geq \frac{r}{2Rd}\mathrm{Vol}_{d-1}(E).$$

This inequality combined with the fact that $\mathrm{Vol}_{d-1}(E) = \mathrm{Vol}_{d-1}(\partial K) - \mathrm{Vol}_{d-1}(F)$ yields

$$\left(1 + \frac{r}{2Rd}\right)\mathrm{Vol}_{d-1}(F) \geq \frac{r}{2Rd}\mathrm{Vol}_{d-1}(\partial K).$$

The trivial bound $\frac{r}{2Rd} < \frac{1}{2}$ implies that $\left(1 + \frac{r}{2Rd}\right) < \frac{3}{2}$ and so

$$\mathrm{Vol}_{d-1}(F) \geq \frac{r}{3Rd}\mathrm{Vol}_{d-1}(\partial K).$$

This completes the proof of Case 2 and of Lemma 2.18. □

2.3.2 The Supporting Hyperplane Characterization is Robust

The robustness of the supporting hyperplane characterization established in Theorem 1.3 is a direct consequence of Lemma 2.18 and the following basic fact.

Claim 2.22. *For any body $K \subset \mathbb{R}^d$, if $x \in \text{conv}(K) \setminus \partial\text{conv}(K)$, then K does not have a separating hyperplane at x.*

Proof. Fix any point $x \in \text{conv}(K) \setminus \partial\text{conv}(K)$. Since x is an internal point of $\text{conv}(K)$, for any direction $c \in \mathbb{R}^d$ there exists a $\delta > 0$ such that $x + \delta c \in \text{conv}(K)$. Assume the claim is false and x has a separating hyperplane $c' \in \mathbb{R}^d$ such that $\forall y \in K$, $\langle c', x - y\rangle \geq 0$. Since all the points in K lie on one side of the hyperplane c', all the points in $\text{conv}(K)$ do as well. Hence, for every $y \in \text{conv}(K)$, $\langle c', x - y\rangle \geq 0$. Consider the direction c'. For any $\delta > 0$, the point $y = x + \delta c'$ lies on the other side of the hyperplane i.e. $\langle c', x - (x + \delta c')\rangle < 0$. Therefore $y = x + \delta c' \notin \text{conv}(K)$ for any $\delta > 0$. This is a contradiction as x is an internal point of $\text{conv}(K)$. □

We are now ready to complete the proof of Theorem 1.3.

Proof of Theorem 1.3. Let $K \subset \mathbb{R}^d$ be a body that is ϵ-far from convex. Since it is ϵ-far from convex it is ϵ-far from its convex hull $\text{conv}(K)$ as well. Hence $\text{Vol}(K) \leq \frac{1}{1+\epsilon}\text{Vol}(\text{conv}(K))$.

Applying Lemma 2.18 with $C = \text{conv}(K)$ and $\gamma = \epsilon$, we get that

$$\text{Vol}_{d-1}(\partial K \setminus \partial\text{conv}(K)) \geq \frac{\epsilon\, r}{3dR}\text{Vol}_{d-1}(\partial K).$$

Therefore, by Claim 2.22, the probability that K has a supporting hyperplane at a point x drawn uniformly at random from ∂K is at most

$$1 - \frac{\text{Vol}_{d-1}(\partial K \setminus \partial\text{conv}(K))}{\text{Vol}_{d-1}(\partial K)} \leq 1 - \frac{\epsilon r}{3dR}.$$

□

2.3.3 Tightness of the Robust Supporting Hyperplane Characterization

In this section, we present a simple argument to show that Theorem 1.3 is essentially tight.

Theorem 2.23. *For any $R > r > 0$ that satisfy $r < \frac{R}{10}$, there exists a body $K \subset \mathbb{R}^d$ that satisfies $B(r) \subseteq K \subseteq B(R)$ and is $\frac{1}{12}$-far from convex such that a point $x \in \partial K$ drawn uniformly at random drawn from the boundary of K satisfies*

$$\Pr_{x \in \partial K}\left[\forall c \in \mathbb{R}^d \, \exists y \in K \text{ s.t. } \langle c, y - x\rangle < 0\right] \leq \frac{4r}{3Rd}.$$

Proof. Consider the four $(d-1)$-dimensional balls, $B_{LL}, B_{LR}, B_{RL}, B_{RR}$, perpendicular to the axis e_1, $B_{LL} = B_{d-1}((-\frac{9R}{10}, 0, \ldots, 0), r)$, $B_{LR} = B_{d-1}((-\frac{R}{6}, 0, \ldots, 0), r)$, $B_{RL} = B_{d-1}((\frac{R}{3}, 0, \ldots, 0), r)$ and $B_{RR} = B_{d-1}((\frac{9R}{10}, 0, \ldots, 0), r)$. Let K be the union of the two cylinders $\mathrm{conv}(B_{LL}, B_{LR})$ and $\mathrm{conv}(B_{RL}, B_{LR})$. The closest convex body to K is the cylinder $\mathrm{conv}(B_{LL}, B_{RR})$ and so K is $O(1)$-far from convex. The body K also contains a ball of radius $r < \frac{R}{10}$ and is contained in a ball of radius R. A point on the boundary of this body does not have a supporting hyperplane only when it is in $B_{LR} \cup B_{RL}$. Therefore, since the volume of the curved boundary of a cylinder of radius r and length ℓ is $\mathrm{Vol}_{d-1}(S_{d-1}(r)) \cdot \ell$,

$$\Pr_{x \in \partial K}\left[\forall c \in \mathbb{R}^d \, \exists y \in K \text{ s.t. } \langle c, y - x\rangle < 0\right] \leq \frac{2\mathrm{Vol}_{d-1}(B_{d-1}(r))}{\mathrm{Vol}_{d-1}(S_{d-1}(r)) \cdot \frac{3}{2}R} \leq \frac{4r}{3Rd}.$$

The last inequality is due to Proposition 2.7. □

2.3.4 Testing Convexity using the Supporting Hyperplane Oracle

We establish Proposition 1.4 and show that convexity can be tested using membership and separation oracles in this section. The central building block of our convexity testing algorithm is a random walk algorithm that generates nearly-uniformly random points from a convex body. Multiple random walk algorithms have been developed (e.g., [39, 38, 37]). The random walk algorithm of [37] suffices for our purposes.

Theorem 2.24 (Lovász, Simonovits [37])**.** *Fix any $R > r > 0$ and $\gamma > 0$. Given a convex body $B(r) \subseteq K \subseteq B(R)$ and access to the membership oracle of the body MO. There exists an algorithm $\mathcal{W}$ that using $O(\mathrm{poly}(d, \log \frac{R}{r}, \log \frac{1}{\gamma}))$ queries to MO outputs a $x \sim \mu$ such that for any subset $S \subseteq K$, $|\mu(S) - \mathcal{U}(S)| \leq \gamma$, where $\mathcal{U}$ is a uniform distribution.*

Proof of Proposition 1.4. Our first observation is that the MO and SO oracles to K enable us to essentially simulate a MO to its convex hull $\mathrm{conv}(K)$: for any query x to the MO to $\mathrm{conv}(K)$, we query both the MO and SO oracles of K on x. If the MO oracle of K

returns "yes", we do as well; otherwise, we return "yes" if and only if the SO oracle finds no separating hyperplane for K at x.

Consider the test that simulates the random walk algorithm $\mathcal{W}$ on $\text{conv}(K)$ with parameter $\gamma = \frac{\epsilon}{10}$ and then checks whether the final point x output by $\mathcal{W}$ is in K. Our convexity testing algorithm runs this test $O(1/\epsilon)$ times and accept if and only if each of the checked points are in K. The query complexity of this algorithm is $O(\frac{1}{\epsilon}) \cdot \text{poly}(d, \log \frac{R}{r}, \log \frac{1}{\epsilon}) = \text{poly}(d, \log \frac{R}{r}, \frac{1}{\epsilon})$.

The testing algorithm always accepts when K is convex because it is equal to its convex hull so $\mathcal{W}$ run on $\text{conv}(K)$ always returns a point in K.

Fix any body K that is ϵ-far from convex. Then $\text{Vol}(\text{conv}(K) \setminus K) \geq \epsilon \text{Vol}(K)$. By Theorem 2.24, the probability that $\mathcal{W}$ returns a point in $\text{conv}(K) \setminus K$ satisfies

$$|\mu(\text{conv}(K) \setminus K) - \mathcal{U}(\text{conv}(K) \setminus K)| \leq \frac{\epsilon}{10}.$$

Therefore,

$$\mu(\text{conv}(K) \setminus K) \geq \mathcal{U}(\text{conv}(K) \setminus K) - \frac{\epsilon}{10} \geq \frac{9\epsilon}{10}$$

and so with probability at least $\frac{2}{3}$ at least one of the points checked by the testing algorithm is not in K and the algorithm correctly rejects. □

Part II

Discrete Convex Functions

Chapter 3

Discrete Convexity and Lattices

In this chapter, we present some preliminaries about discrete convexity and lattices that are useful in the next two chapters. In Section 3.1, we establish some basic facts about convex functions over finite subsets of $\mathbb{R}^d$. In Section 3.2, we describe other notions of discrete convexity that have been studied in the context of property testing. Finally, in Section 3.3 we define lattices and present some useful background knowledge related to them.

3.1 Basic Facts about Discrete Convexity

We write $[n] = \{1, 2, \ldots, n\}$ for $n > 0 \in \mathbb{Z}$. For $x \in \mathbb{R}^d$, we write $x_1, x_2, \ldots, x_d \in \mathbb{R}$ to refer to its d coordinates. When $a < b \in \mathbb{R}$, we use $[a, b]_{\mathbb{Z}}$ to denote the set of integers in the range between a and b inclusively, and when $a < b \in \mathbb{Z}$, we use $x_{[a,b]_{\mathbb{Z}}}$ to denote the coordinates $x_a, x_{a+1}, \ldots, x_b$ of an input x. All the results in this section are standard; we provide the proofs for completeness.

For a function $f : \mathbb{Z}^d \to \mathbb{R}$ we define $f_{x_d=k} : \mathbb{Z}^{d-1} \to \mathbb{R}$ as $f_{x_d=k}(x) = f(x, k)$. The *restriction* of a function $f\colon X \to \mathbb{R}$ to a domain $Y \subseteq X$ is the function $f|_Y\colon Y \to \mathbb{R}$ defined by $f|_Y(y) = f(y)$ for each $y \in Y$. Our first basic observation is that restriction preserves convexity. When we say a function over a discrete domain is convex, we mean it is a discrete convex function. Refer to Section 1.2 for the definition of discrete convex functions.

Proposition 3.1. *Let $f\colon X \to \mathbb{R}$ be a convex function and $Y \subseteq X$. Then the function $f|_Y\colon Y \to \mathbb{R}$ restricted to Y is also convex.*

Now we will move on to extending convex functions and establishing minimal requirements for a function f to be convex. To define the extension of convex functions, we first need the notion of a centered simplex.

Definition 3.2. A *simplex* in $\mathbb{R}^d$ is a set of affinely independent points. A *centered simplex* in $\mathbb{R}^d$ is a collection of points $x^{(1)}, \dots, x^{(k)}, z$ such that $x^{(1)}, \dots, x^{(k)}$ form a simplex, and z can be (uniquely) expressed as

$$z = \sum_i \lambda_i x^{(i)}, \tag{3.1}$$

where all $\lambda_i > 0$ and $\sum_i \lambda_i = 1$. The point z is called the *center* of the simplex, and we say that the simplex is *centered at* z when this condition is satisfied.

In other words, $x^{(1)}, \dots, x^{(k)}, z$ is a centered simplex if z is inside the convex hull of $x^{(1)}, \dots, x^{(k)}$ and no $x^{(i)}$ can be removed from the simplex without breaking this property. When X is a finite subset of $\mathbb{R}^d$ and $x^{(1)}, \dots, x^{(k)}, z \in X$, we say that the centered simplex is *of* X.

Definition 3.3. The centered simplex $x^{(1)}, \dots, x^{(k)}, z$ of X is *minimal* iff z is the only point of X inside the convex hull of the simplex $x^{(1)}, \dots, x^{(k)}$ except for its vertices.

Lemma 3.4. *Let $f\colon X \to \mathbb{R}$ be a convex function with X a finite subset of $\mathbb{R}^d$. Then the function can be extended to a convex function on the whole space $\mathbb{R}^d$. That is, there exists a convex function $\tilde{f}\colon \mathbb{R}^d \to \mathbb{R}$ such that $\tilde{f}(x) = f(x)$ for all $x \in X$. Moreover, for a point z in the convex hull of X the function $\tilde{f}$ can be defined as*

$$\tilde{f}(z) = \min_{x^{(1)}, \dots, x^{(k)}} \sum_i \lambda_i f(x^{(i)}), \tag{3.2}$$

where $x^{(1)}, \dots, x^{(k)}$ range over all simplices of X centered at z, and λ_i are as in Equation (3.1).

Proof. By convexity of f, we have that $\tilde{f}(x) = f(x)$ for all $x \in X$. Hence, $\tilde{f}$ indeed extends f. It remains to prove that $\tilde{f}$ is convex.

Claim 3.5. *The definition of $\tilde{f}$ in Equation (3.2) does not change if we minimise over all possible convex combinations $z = \lambda_1 x^{(1)} + \dots + \lambda_k x^{(k)}$, where $x^{(1)}, \dots, x^{(k)}$ need not form a simplex.*

Proof. Let $\tilde{f}(z)$ be defined as in the statement of this claim. Take a linear combination $z = \lambda_1 x^{(1)} + \cdots + \lambda_k x^{(k)}$ which minimises $\sum_i \lambda_i f(x^{(i)})$ and such that k is as small as possible. We claim that then $x^{(1)}, \ldots, x^{(k)}$ form a simplex.

Indeed, assume $x^{(1)}, \ldots, x^{(k)}$ are not affinely independent. Then, there exists a non-trivial linear combination $\beta_1 x^{(1)} + \cdots + \beta_k x^{(k)} = 0$ such that $\beta_1 + \cdots + \beta_k = 0$. Changing the sign of each β_i if necessary, we may assume that $\beta_1 f(x^{(1)}) + \cdots + \beta_k f(x^{(k)}) \geq 0$. Let $t \geq 0$ be the maximal real number such that $\lambda_i - t\beta_i \geq 0$ for all i.

Let $\lambda'_i = \lambda_i - t\beta_i$. We have that $\lambda'_i \geq 0$, $\sum_i \lambda'_i = 1$, $\sum_i \lambda'_i x^{(i)} = z$, and $\sum_i \lambda'_i f(x^{(i)}) \leq \sum_i \lambda_i f(x^{(i)})$. Moreover, at least one of λ'_i is equal to 0, which contradicts minimality of k. □

Claim 3.6. *The function $\tilde{f}$ is convex on the convex hull of X.*

Proof. Consider a convex combination $z = \mu_1 z^{(1)} + \cdots + \mu_k z^{(k)}$, where all $z^{(i)}$ lie in the convex hull of X. For each $z^{(i)}$ choose a convex combination $z^{(i)} = \sum_{x \in X} \lambda_{i,x} x$ such that $\tilde{f}(z^{(i)}) = \sum_{x \in X} \lambda_{i,x} f(x)$. Then, $\lambda_x = \sum_i \mu_i \lambda_{i,x}$ give a convex combination over the elements of X such that $z = \sum_{x \in X} \lambda_x x$. By Claim 3.5,

$$\tilde{f}(z) \leq \sum_{x \in X} \lambda_x f(x) = \mu_1 \tilde{f}(z^{(1)}) + \cdots + \mu_k \tilde{f}(z^{(k)}),$$

proving that the function $\tilde{f}$ is convex. □

By [61], the function $\tilde{f}$ can be extended from the convex hull of X to the whole $\mathbb{R}^d$. This completes the proof of Lemma 3.4. □

From the above lemma, we see that if $f \colon X \to \mathbb{R}$ is a convex function with $X \subseteq \mathbb{R}^d$ finite, and $X \subseteq Y \subseteq \mathbb{R}^d$, then the function f can be extended to a convex function on Y. This is how we will usually use the above lemma.

We say that a function $f \colon X \to \mathbb{R}$ is convex on a centered simplex $x^{(1)}, \ldots, x^{(k)}, z$ if its restriction to this set of points is convex. This is equivalent to

$$f(z) \leq \sum_i \lambda_i f(x^{(i)}),$$

where λ_i are as in Equation (3.1). This notion provides a characterization of convexity that we will use to test convex functions.

Theorem 3.7. *A function $f\colon X \to \mathbb{R}$ is convex if and only if it is convex on every minimal centered simplex of X.*

Proof. Assume that f is not convex. Then there exists a convex combination $z = \lambda_1 x^{(1)} + \cdots + \lambda_k x^{(k)}$ such that $f(z) > \lambda_1 f(x^{(1)}) + \cdots + \lambda_k f(x^{(k)})$. Choose such a convex combination that k is as small as possible and the convex hull of $x^{(1)}, \ldots, x^{(k)}$ is inclusion-wise minimal. We claim that then $x^{(1)}, \ldots, x^{(k)}, z$ form a minimal centered simplex.

Using the same argument as in Claim 3.5, we get that $x^{(1)}, \ldots, x^{(k)}$ is a simplex. Assume it contains more than two points in its convex hull minus the vertices. Let z be such that the violation $f(z) - \lambda_1 f(x^{(1)}) - \cdots - \lambda_k f(x^{(k)}) > 0$ is as large as possible. Let y be any other point in the convex hull of $x^{(1)}, \ldots, x^{(k)}$ except for its vertices. Then,

$$f(z) - \lambda_1 f(x^{(1)}) - \cdots - \lambda_k f(x^{(k)}) \geq f(y) - \mu_1 f(x^{(1)}) - \cdots - \mu_k f(x^{(k)}), \tag{3.3}$$

where $y = \mu_1 x^{(1)} + \cdots \mu_k x^{(k)}$. Let $t \geq 0$ be the largest real number such that $\lambda_i - t\mu_i \geq 0$ for all i. Let $\lambda'_i = \lambda_i - t\mu_i$. We have the following convex combination:

$$z = ty + \lambda'_1 x^{(1)} + \cdots + \lambda'_k x^{(k)}.$$

Moreover, one of λ'_i is equal to 0. As $z \neq y$, we have that $t < 1$. This together with Equation (3.3) yields

$$f(z) > tf(y) + \lambda'_1 f(x^{(1)}) + \cdots + \lambda'_k f(x^{(k)}).$$

This contradicts inclusion-wise minimality of $x^{(1)}, \ldots, x^{(k)}$. □

Let us apply the general Theorem 3.7 to the setting where f is a function over the line. Let $X = \{x^{(1)}, x^{(2)}, x^{(3)}, \ldots\} \subseteq \mathbb{R}$ where $x^{(1)} < x^{(2)} < x^{(3)} < \cdots$. A centered simplex in this case is a *triple* $x < y < z$ and y is the center of the triple. A function f is convex on the triple if and only if

$$\frac{f(y) - f(x)}{y - x} \leq \frac{f(z) - f(y)}{z - y}. \tag{3.4}$$

A minimal centered simplex is a *minimal triple* of the form $x^{(i)} < x^{(i+1)} < x^{(i+2)}$. Thus, we get the following corollary.

Corollary 3.8. *Let $X = \{x^{(1)}, x^{(2)}, x^{(3)}, \ldots\} \subseteq \mathbb{R}$ with $x^{(1)} < x^{(2)} < x^{(3)} < \cdots$. Then the function $f\colon X \to \mathbb{R}$ is convex if and only if it is convex on every triple $x^{(i)} < x^{(i+1)} < x^{(i+2)}$ of consecutive points.*

3.2 Other Notions of Discrete Convexity

Other notions of discrete convexity like linear convexity and separate convexity have also been considered in the context of property testing. In this section, we define them and give an example that distinguishes them from discrete convex functions. Later, in Section 4.5 we discuss the testing results related to them.

Linear convexity

The function $f\colon [n]^d \to \mathbb{R}$ is *linearly convex* if for every $x, y \in [n]^d$ and every $0 \leq \lambda \leq 1$ for which $\lambda x + (1-\lambda)y \in [n]^d$, we have $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$.

Separate convexity

A function, $f : [n]^d \to \mathbb{R}$ is *separately convex*, if for every $i \in [d]$ and $x \in [n]^d$ the function $g : [n] \to \mathbb{R}$ defined by $g(y) = f(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_d)$ is convex.

Below we show that there exists functions that are not convex, but are linearly and separately convex.

Proposition 3.9. *Every convex function is linearly convex and every linearly convex function is separately convex. However, there exists functions that are linearly convex but are not convex.*

Proof. That every convex function is linearly convex and every linearly convex function is separately convex follows directly from the definitions. For the second statement, consider the function $f\colon [3] \times [3] \to \mathbb{R}$ defined by

$$\begin{array}{lll} f(0,2) = 3 & f(1,2) = 1 & f(2,2) = 5 \\ f(0,1) = 1 & f(1,1) = 2 & f(2,1) = 3 \\ f(0,0) = 5 & f(1,0) = 3 & f(2,0) = 1 \end{array}$$

The function f is linearly convex, but it has a violation of convexity on the point $(1,1)$ with respect to the points $(2,0)$, $(0,1)$, and $(1,2)$. □

Other notions of discrete convexity have also been considered in the context of minimization to get around the problem of local optimality does not imply global optimality. See [44] and the references therein for more details on those notions.

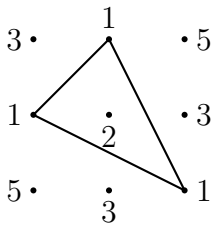

Figure 3.1: Illustration of the function f constructed in the proof of Proposition 3.9

3.3 Lattices

Lattices play an important role in Chapters 4 and 5.

Definition 3.10. A set $\mathcal{L} \subset \mathbb{R}^d$ is a *lattice* if the following statements hold:

- The origin is in $\mathcal{L}$.
- For any $\alpha \in \mathbb{Z}$ and $x \in \mathcal{L}$, the point $\alpha x \in \mathcal{L}$.
- For any $x, y \in \mathcal{L}$, the point $x + y \in \mathcal{L}$.

For a matrix $B \in \mathbb{R}^{m \times n}$ we use $b^{(1)}, b^{(2)}, \ldots, b^{(m)} \in \mathbb{R}^n$ to represent the columns of the matrix $B = (b^{(1)}\, b^{(2)}\, \ldots\, b^{(m)})$. We use $B_{S \times T}$ to represent the sub-matrix defined by the rows $S \subset [m]$ and columns $T \subset [n]$.

Definition 3.11. The matrix $B = (b^{(1)}, \ldots, b^{(k)}) \in \mathbb{R}^{d \times k}$ is a *basis of the lattice* $\mathcal{L} \in \mathbb{R}^d$ if:

- The column vectors $b^{(1)}, \ldots, b^{(k)}$ are linearly independent.
- $\forall i\ b^{(i)} \in \mathcal{L}$.
- For every $y \in \mathcal{L}$, we have $x \in \mathbb{Z}^k$ such that $y = \sum_{i=1}^{k} x_i b^{(i)}$.

If the size of the basis of a lattice is k, we call it a *k-dimensional lattice*. The Euclidean space $\mathbb{Z}^d$ is a d-dimensional lattice with $(e_1\, e_2\, \ldots\, e_d)$ as a basis.

The *representation of $x \in \mathbb{Z}^d$ according to the basis B* is

$$x^B = B^{-1}x.$$

For a function $f : \mathbb{Z}^d \to \mathbb{R}$, a basis $B \in \mathbb{Z}^{d\times d}$, and a point $x' \in \mathbb{Z}^d$, we write $f_{B,x'}(x) = f(Bx + x')$. It represents the function we get when we translate the origin to x' and change the basis to B.

The basis of a lattice is not unique and can be transformed using a unimodular matrix defined below.

Definition 3.12. A matrix $U \subset \mathbb{Z}^{d\times d}$ is referred to as *unimodular* if the determinant is ± 1.

Proposition 3.13 (Corollary 4.3a [54])**.** *Let $B \in \mathbb{Z}^{d\times d}$ be the basis of a lattice $\mathcal{L} \subset \mathbb{Z}^d$. For any unimodular matrix U, the matrix $B' = UB$ is also a basis of $\mathcal{L}$.*

The intersection of a hyperplane, passing through the origin, with a lattice $\mathcal{L} \in \mathbb{Z}^d$ is a lattice and the basis of it can be computed in time polynomial in the dimension d and the binary representation of the hyperplane.

Proposition 3.14 (Lemma 14.13 [14])**.** *Let $L \subset \mathbb{Z}^d$ be a lattice and $H = \{x \in \mathbb{Z}^d : \langle a, x\rangle = 0\}$ be a subspace. The set $L \cap H$ is a lattice and its basis $B \subset \mathbb{Z}^{d-1\times d}$ can be found in time $O(\text{poly}(d, \log \|a\|_\infty))$.*

Definition 3.15. We say a matrix $B \subset \mathbb{Z}^{d\times d}$ is in *Hermite normal form*(HNF) if:

1. B is an lower triangular matrix i.e. $b_j^{(i)} = 0$ for $i < j$.
2. The leading coefficient(first non-zero entry from the top, also called *pivot*) in a column is below the leading coefficient of the column to the left of it, and is positive.
3. The elements to the right of the pivots are zero and elements to the left are non-negative and strictly smaller than the pivot.

Given an integral matrix, we can efficiently find a unimodular matrix that transforms the integral matrix into HNF, as indicated in the proposition below.

Proposition 3.16 (Theorem 5.2 [54])**.** *Given an integral matrix $B \subset \mathbb{Z}^{d\times d}$, we can find a unimodular matrix $U \in \mathbb{Z}^{d\times d}$, in time $O(\text{poly}(d, \log \|B\|_\infty))$, such that $A = BU$ is in Hermite normal form and $\|U\|_\infty = O(\text{poly}(\log \|B\|_\infty))$.*

Chapter 4

Testing Convex Functions Over the Hypergrid

In this chapter we present the results related to testing convex functions over the hypergrid. In Section 4.1, we describe the main ideas in our proofs. We establish our algorithmic results in Sections 4.2 and 4.3, and give the proofs for our hardness results in Sections 4.4. Specifically, the proof of Theorem 1.5 for testing convexity over one-dimensional domains is presented in Section 4.2. The upper bound in Theorem 1.7 for testing convexity of functions on the stripe is established in Section 4.3. The lower bound in Theorem 1.8 for testing convexity over high-dimensional domains is presented in Section 4.4; the lower bound for the stripe in Theorem 1.7 is found in Section 4.4.4. Finally, in Section 4.5 we discuss related work on testing linear convexity and separate convexity. For the reader's convenience we restate the theorems we prove in this chapter below.

Theorem 1.5. *There exists an ϵ-tester for convexity of functions $f : [n] \to \mathbb{R}$ over the line with complexity $O(\frac{\log(\epsilon n)}{\epsilon})$. The tester is non-adaptive and has 1-sided error.*

Theorem 1.7. *There exists a 1-sided-error algorithm that ϵ-tests a function $f\colon [3] \times [n] \to \mathbb{R}$ for convexity using $O(\frac{\log^2 n}{\epsilon})$ queries to f. By contrast, any non-adaptive $\Omega(1)$-tester for convexity of $f\colon [3] \times [n] \to \mathbb{R}$ has query complexity $\Omega(\sqrt{n})$.*

Theorem 1.8. *For every $d \geq 2$, $n > 4d$ and any $\epsilon \leq \frac{1}{10}$, any non-adaptive ϵ-tester for convexity of functions over $[n]^d$ has query complexity $\Omega\big((\frac{n}{d})^{\frac{d}{2}}\big)$.*

4.1 Main Ideas

Convexity tester for functions over $[n]$ (Theorem 1.5)

Ours is the first triple tester for convexity. We select triples independently (not uniformly) at random and check for violations of convexity. The previous testers of Parnas, Ron and Rubinfeld [50], and Ben-Eliezer [7] use a binary search approach. The parameterized tester by Pallavoor, Raskhodnikova, and Varma [48] also uses the binary search approach as they use the tester in [50] as a subroutine. The parameterized tester of Lahiri, Newman and Varma [31] uses our tester as a subroutine.

The testing algorithm is inspired by the monotonicity testers on the line [5, 23]. The idea there is to query pairs of points that are different distances apart and check those pairs for violation of monotonicity. Testing convexity is different because we need three points even to catch a violation of convexity. So we pick two points that are different distances apart as they do for testing monotonicity, and the third point is chosen to be adjacent to one of the first two points picked. We want the first two points in the triplet to cover different scales of distance so that it is hard to hide the violations of convexity. For instance, if the first two points in the triplet are close together for all our queries, then it is hard to catch the violation of convexity in functions like, $f(x) = x$ for $x \in [\frac{n}{2}]$ and $f(x) = x - n$ for $x \in [n] \setminus [\frac{n}{2}]$. The above function is far from convex, but to catch the violation of convexity we need at least one point in the first half and one in the second half, which we may not get if the points in our triplet are close together. If the first two points in our triplets are far for all our triplets, then we fail to catch functions like $f(x) = x - \lfloor \frac{x}{3} \rfloor$, which are far from convex but the violations of convexity can only be caught by triplets with points close together.

Convexity tester for functions over $[3] \times [n]$ (Theorem 1.7)

The algorithm relies on Theorem 3.7, which shows that a function is convex if it is convex on every minimal centered simplex in the domain. Refer to Section 3.1 for the definitions of a simplex, centered simplex, and a minimally centered simplex. As we are in two dimensions the two kinds of centered simplices possible are a line centered simplex, in which the simplex is two points, and a triangle centered simplex, in which the simplex is three points which form a triangle. We refer to the violation of convexity of a line centered simplex as *line violation of convexity*, and a triangle centered simplex as *triangle violation of convexity*. The first observation we make is that any violation of convexity with the center in the first or third column is a line violation of convexity, with the points entirely contained in the first or third column respectively. The second observation we make is that any violation of convexity with the center in the second column is of three types: 1) a line

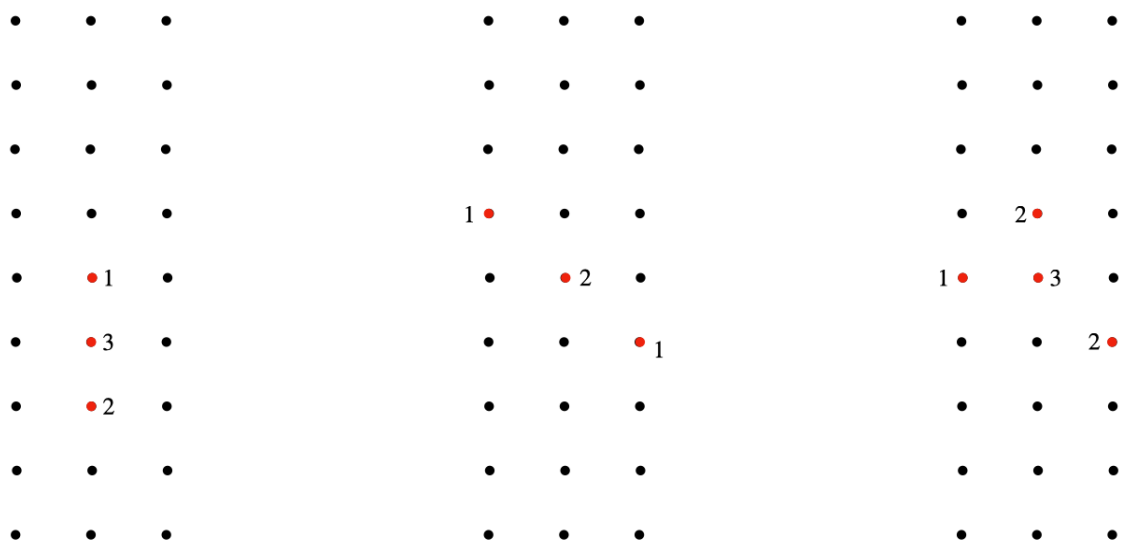

Figure 4.1: The three figures above illustrate the different kinds of violations of convexity possible with the center in the second column, for the domain $[3] \times [n]$.

violation of convexity with both the end points in the second column 2) a line violation of convexity with one end point each in the first and third column, and 3) a triangle violation of convexity with one point in each column. Refer to Figure 4.1 for an illustration.

The line violations of convexity with all the points in the same column are handled by our line tester. To handle the line violations of convexity with the end points in the first and third column, and the triangle violations of convexity, we introduce a new function, h, on the second column over the points $\{0, \frac{1}{2}, 1, \frac{3}{2}, \ldots, n\}$. For a more detailed overview of the algorithm refer to Section 4.3.1.

Non-adaptive lower bound for testing convexity over $[n]^d$ (Theorem 1.8)

The lower bounds in Theorems 1.7 and 1.8 are both obtained using the same general construction. We describe it in the setting of functions over $[n]^d$ for simplicity.

The key idea is that we can construct convex functions whose increase in slope (i.e., second derivative) is small in a particular direction and large in the rest of the directions. We can perturb the values of such functions by ± 1 in a way that yields functions which are far from convex but for which the only violation of convexity on the hypergrid will contain at least two points that form a line along the direction where the slope was increasing

slowly. So any algorithm that does not query two points which give a line in that direction cannot catch any violations of convexity. To get a strong lower bound from this key idea, we show that it is possible to "hide" the slowly-increasing direction among $\Omega\big((\frac{n}{d})^d\big)$ possible directions. Since a set of q queries contains pairs of points that form at most q^2 different directions, this construction shows that any non-adaptive convexity testing algorithm with one-sided error—i.e., that always accepts convex functions—must have query complexity at least $\Omega\big((\frac{n}{d})^{d/2}\big)$.

To generalize this argument in a way that gives a lower bound for non-adaptive testing algorithms with two-sided error as well, we consider a different perturbation of the convex functions of ± 1 that preserves convexity. We can do this by performing the same perturbation (i.e., either all $+1$ or all -1) for every point along a line in the slowly-increasing direction. The perturbations for each line are chosen independently at random; by ensuring that the slope of the original function is large enough in all other directions, these independent perturbations do not violate convexity.

4.2 Algorithms for Testing Convexity Over the Line

In this section, we prove Theorem 1.5.

Definition 4.1. Let $a \in [n]$. A *triple test* rooted at a is a (non-necessarily sorted) triple (a, b, c) such that

- $b \in 2^k \lfloor \frac{a-1}{2^k} \rfloor, 2^k \lceil \frac{a+1}{2^k} \rceil$ for some integer k satisfying $1 \leq 2^k < n$, and
- c is either $a+1$ or $b+1$.

The element a is called the *root*, and b is called a *hub* of a. The integer 2^k is called the *height* of the triple. We say that a passes the triple test if the function f is convex on $\{a, b, c\}$. We say that a passes all its triple tests if it passes all the triple tests rooted at it.

Claim 4.2. *If $x < y - 1$, then x and y have a common hub with height not exceeding $2(y - x)$.*

Proof. Let k be such that $\frac{y-x}{2} \leq 2^k \leq y - x$. There can be either one or two multiples of 2^k between x and y. If there is just one then we are done and that is the common hub. If there are two then there will be exactly one multiple of 2^{k+1} between x and y and that is their common hub. □

Lemma 4.3. *Assume $x < y < z$ is a non-convex triple. Then at least one of x, y or z fails some of its triple tests with height not exceeding $2 \cdot \max\{y - x, z - y\}$.*

Proof. Assume for now that $y - x \geq 2$ and $z - y \geq 2$. Let h be the common hub between x, y with height not exceeding $2(y - x)$ and h' be the common hub between y, z with height not exceeding $2(z - y)$. Consider the function f restricted to the domain $x < h < y < h' < z$. By Proposition 3.1, we know the function is not convex, hence, by Corollary 3.8, it is non-convex on at least one of the triples (x, h, y), (h, y, h'), or (y, h', z). Let us consider the three cases separately:

- f is non-convex on the triple x, h, y. Consider the function f on the domain $x < h < h + 1 \leq y$. Using Corollary 3.8 if needed, we get that the function f is non-convex on one of the triples $(x, h, h + 1)$ or $(h, h + 1, y)$. Each of them constitutes a triple test: $a = x$, $b = h$, $c = h + 1$, or $a = y$, $b = h$, $c = h + 1$, respectively.
- f is non-convex on the triple h, y, h'. Consider the function f on the domain $h < y < y + 1 \leq h'$. The function f is non-convex on one of the triples $(h, y, y + 1)$ or $(y, y + 1, h')$. Again, each of them constitutes a triple test: $a = y$, $b = h$, $c = y + 1$, or $a = y$, $b = h'$, $c = y + 1$, respectively.
- f is non-convex on the triple y, h', z. This case is analogous to the first one.

If $y = x + 1$, then the above analysis works with $h = x$ (the first case never holds, and $x = y - 1$ is a hub of y). If $z = y + 1$, the above analysis works with $h' = z$ (the third case never holds, and $z = y + 1$ is a hub of y). Finally, if both $y = x + 1$ and $z = y + 1$, we can use the triple test with $a = x$, $b = y$, and $c = z$. □

A simple consequence of this lemma is that the function f is convex on the set of points passing all their triple tests. This allows us to formulate the following notion.

Definition 4.4. A *convex replacement* of a function $f\colon [n] \to \mathbb{R}$ is a convex function $\hat{f}\colon [n] \to \mathbb{R}$ such that $f(x) = \hat{f}(x)$ for all x that pass all their triple tests.

We are now ready to complete the proof of Theorem 1.5.

Proof of Theorem 1.5. The algorithm is a triple tester. It selects a root a of the triple uniformly at random from $[n]$, select a triple rooted at a with height at most $2\epsilon n$ uniformly at random, and tests it for convexity. For completeness, let us restate the algorithm:

Algorithm 1: ConvexityTest1D

1 Draw a uniformly at random from $[n]$;
2 Draw k uniformly at random from $\{0, \ldots, \lceil \log_2(2\epsilon n) \rceil\}$;
3 Let b be either the largest multiple of 2^k strictly smaller than x, or the smallest multiple of 2^k strictly larger than x, each case with probability $1/2$;
4 Let c be either $a+1$ or $b+1$, each with probability $1/2$;
5 Query $f(a)$, $f(b)$ and $f(c)$ and test whether a, b, c form a convex triple;

We claim that if the function f is ϵ-far from convex, then this test fails with probability $\Omega(\epsilon/\log(\epsilon n))$. Thus, this test has to be repeated $O(\log(\epsilon n)/\epsilon)$ times.

We will construct a subset $A \subseteq [n]$ of size ϵn such that every $a \in A$ fails one of its triple tests with height at most $2\epsilon n$. Start with $S \leftarrow [n]$. We treat S as a sorted list. While $|[n] \setminus S| < \epsilon n$, the function $f|_S$ is non-convex. Choose three neighbouring elements $x < y < z$ in S that violate convexity. Let (a, b, c) be the non-convex triple constructed in Lemma 4.3. The height of this triple is at most $2\epsilon n$. Remove a from S. When $|[n] \setminus S| \geq \epsilon n$, let $A \leftarrow [n] \setminus S$. □

4.3 Algorithm for Testing Convexity on the $[3] \times [n]$ Stripe

In this section, we prove the upper bound in Theorem 1.7.

4.3.1 High-level description

Our approach to testing convexity on the stripe $[3] \times [n]$ is as follows. This set is close to the 1-dimensional line, so we can draw a lot from the tester of Section 4.2. In this vein, for $i \in [3]$, let $f_i \colon [n] \to \mathbb{R}$ be the restrictions of f to the column $\{i\} \times [n]$. We will construct a convex replacement $\hat{f}$ of f so that every point where f and $\hat{f}$ disagree fails some test. Sampling $a \in [3] \times [n]$ uniformly at random and executing the test on a will give us a tester of convexity.

Any simplex centered at a point in the line $\{1\} \times [n]$ or $\{3\} \times [n]$ is completely contained inside this line. Hence, for f_1 and f_3 we can simply take convex replacements $\hat{f}_1$ and $\hat{f}_3$

from Definition 4.4, and assume that $\hat{f}$ restricted to $\{1\} \times [n]$ or $\{3\} \times [n]$ is $\hat{f}_1$ or $\hat{f}_3$, respectively.

Let us define a function $h\colon \{1, \frac{3}{2}, 2, \frac{5}{2}, \ldots, n\} \to \mathbb{R}$ as

$$h(x) = \min_{\delta} \frac{\hat{f}_1(x-\delta) + \hat{f}_3(x+\delta)}{2}. \tag{4.1}$$

Note that $h(x) = g(2, x)$ where g is the convex extension, as in Equation (3.2), of the function $\hat{f}$ restricted to $\{1, 3\} \times [n]$. (We have not defined $\hat{f}$ on the line $\{2\} \times [n]$ yet.) By Lemma 3.4 and Proposition 3.1, the function h is convex. Its value can be computed by minimising the convex function $\delta \mapsto (\hat{f}_1(x-\delta) + \hat{f}_3(x+\delta))/2$. This is exactly the place where our tester uses adaptivity.

The main part of our algorithm deals with interplay between the functions h and f_2. Let us give some relations between h and f_2 for the case when f is convex. First, the function f is convex on any simplex of the form $(1, x-\delta), (3, x+\delta)$ centered at $(2, x)$, which implies that $f_2(x) \leq h(x)$ for every $x \in [n]$. Next, for every $\{x, x+1\} \subseteq [n]$, let $\psi\colon \mathbb{R} \to \mathbb{R}$ be the affine function agreeing with f_2 at x and $x+1$. We have that the function f is convex on any simplex of the form $(1, z-\delta), (2, x), (3, z+\delta)$ centered at $(2, x+1)$, which implies that $\psi(z) \leq h(z)$ for all $z > x+1$. [1] Similarly, considering simplices $(1, z-\delta), (2, x+1), (3, z+\delta)$ centered at $(2, x)$, we get that $\psi(z) \leq h(z)$ for all $z < x$. Our tester will check these conditions.

4.3.2 Subroutines

We are now ready to describe the subroutines used by our tester. The first subroutine is the convexity test for the line from Section 4.2.

Algorithm 2: 1DTest(i, x)

1 Execute all the triple tests rooted on x for the function f_i as in Definition 4.1;
2 If at least one of the tests fails, output that f is not convex and terminate the algorithm;

The complexity of this subroutine is $O(\log n)$. The following claim is a direct consequence of Definition 4.4.

[1] Note that this observation does not immediately follow from the first observation and convexity of f_1, because it also incorporates half-integer values of z, where f_1 is not defined.

Claim 4.5. *If* 1DTest(i, x) *does not fail for* $i = 1$ *or* $i = 3$*, then* $\hat{f}_i(x) = f_i(x)$.

The next subroutine evaluates the function h.

Algorithm 3: Evaluate(x)

1 Find a point δ^* where the function $g(\delta) = \frac{1}{2}(f_1(x-\delta) + f_3(x+\delta))$ attains its minimum, assuming this function is convex;
2 For $y \in \{x - \delta^* - 1, x - \delta^*, x - \delta^* + 1\}$ perform 1DTest$(1, y)$;
3 For $y \in \{x + \delta^* - 1, x + \delta^*, x + \delta^* + 1\}$ perform 1DTest$(3, y)$;
4 Check that $g(\delta^*) \le g(\delta^* - 1)$ and $g(\delta^*) \le g(\delta^* + 1)$;
5 Return $g(\delta^*)$;

Claim 4.6. *The subroutine either finds a violation of convexity or returns* $h(x)$*. The complexity of the subroutine is* $O(\log n)$.

Proof. Define $\hat{g}(\delta) = \frac{1}{2}(\hat{f}_1(x-\delta) + \hat{f}_3(x+\delta))$ so that $h(x) = \min_\delta \hat{g}(\delta)$. Steps 2 and 3 of the subroutine ensure that g and $\hat{g}$ agree on $\delta^* - 1, \delta^*$ and $\delta^* + 1$. If Step 4 fails, we get that $g \neq \hat{g}$, meaning that the function f is not convex. Otherwise, we get that $\hat{g}(\delta^*) \le \hat{g}(\delta^* - 1)$ and $\hat{g}(\delta^*) \le \hat{g}(\delta^* + 1)$. As $\hat{g}$ is convex, this implies that the minimum of $\hat{g}$ is attained at δ^*. The complexity estimate is obvious. □

4.3.3 The Algorithm

Now let us state the test for convexity over the stripe.

Algorithm 4: ConvexityTestStripe

1 Sample (i, x) uniformly at random from $[3] \times [n]$;
2 Perform 1DTest(i, x);
3 **if** $i = 2$ **then**
4 Evaluate $h(x)$ and verify that $f_2(x) \leq h(x)$;
5 Perform 1DTest$(2, x-1)$ and 1DTest$(2, x+1)$;
6 Let $\psi^-\colon [n] \to \mathbb{R}$ be the affine function satisfying $\psi^-(x-1) = f_2(x-1)$ and $\psi^-(x) = f_2(x)$;
7 Minimise the convex function $h - \psi^-$ on the interval $\{x+1, x+\frac{3}{2}, x+2, \ldots, n\}$ and verify that the minimum is non-negative;
8 Let $\psi^+\colon [n] \to \mathbb{R}$ be the affine function satisfying $\psi^+(x) = f_2(x)$ and $\psi^+(x+1) = f_2(x+1)$;
9 Minimise the convex function $h - \psi^+$ on the interval $\{1, \frac{1}{2}, 1, \ldots, x-1\}$ and verify that the minimum is non-negative;

Each run of the tester $O(\log^2 n)$ queries to f, since steps 7 and 9 each require $O(\log n)$ calls to the Evaluate subroutine, which in turn makes $O(\log n)$ queries to f.

By the discussion at the beginning of the section, any convex function f passes the test with probability 1. Let $f\colon [3] \times [n] \to \mathbb{R}$ be any function that is ϵ-far from convex. Let S be the set of points that pass the test. We claim that f restricted to S is convex. Hence, the error probability of the test is at least ϵ, and it suffices to repeat the test $O(1/\epsilon)$ times.

In order to prove that f is convex on S, we extend it to a slightly larger domain. This is done to better handle possible minimal centered simplices. Let as above $\hat{f}_i$ be convex replacement of f_i. We claim that the function $\hat{f}\colon (\{1,3\} \times [n]) \cup S \to \mathbb{R}$ defined by

$$\hat{f}(i,x) = \begin{cases} \hat{f}_i(x), & \text{if } i = 1 \text{ or } i = 3; \\ f(i,x), & \text{if } (i,x) \in S; \end{cases}$$

is convex (the two values are equal when both conditions apply). As f and $\hat{f}$ agree on S, this implies that f restricted to S is convex.

By Theorem 3.7 and the above discussion, it suffices to consider minimal simplices centered at points of the form $(2, x) \in S$. From Lemma 4.3, we get that the function $\hat{f}$ is convex on a centered simplex of the form $\{(2,x), (2,y), (2,z)\} \subseteq S$. The function $\hat{f}$ is also convex on a simplex $(1, x-\delta), (3, x+\delta)$ centered at $(2, x)$ because $h(x) \geq f_2(x)$ by Step 4.

Any other minimal simplex centered at $(2, x)$ is of the form $(1, a), (2, b), (3, c)$. Let $y = (a + c)/2$, and assume $b < x < y$ (the case $y < x < b$ is similar). From Step 7 of the algorithm, we know that the function $g\colon \{x - 1, x, y\} \to \mathbb{R}$ defined by

$$\begin{aligned} g(x-1) &= f_2(x-1), \\ g(x) &= f_2(x), \\ g(y) &= h(y) \end{aligned}$$

is convex. Both $(2, b)$ and $(2, x)$ are in S and so they pass the test. Thus, from Steps 2 and 5, we have that f_2 and $\hat{f}_2$ agree on b, $x-1$ and x. As $\hat{f}_2$ is convex, and using Corollary 3.8, we have that the function $g'\colon \{b, x - 1, x, y\} \to \mathbb{R}$ defined by

$$\begin{aligned} g'(b) &= f_2(b), \\ g'(x-1) &= f_2(x-1), \\ g'(x) &= f_2(x), \\ g'(y) &= h(y) \end{aligned}$$

is also convex. Finally, since $h(y) \leq (\hat{f}_1(a) + \hat{f}_3(c))/2$, we get that $\hat{f}$ is convex on the simplex $(1, a), (2, b), (3, c)$ centered at $(2, x)$.

4.4 Lower Bounds for Testing Convexity in High Dimensions

In this section we prove the $\Omega((\frac{n}{d})^{\frac{d}{2}})$ lower bound for non-adaptive algorithms that test convexity on the $[n]^d$ grid in Theorem 1.8 and the $\Omega(\sqrt{n})$ lower bound for non-adaptive algorithms that test convexity over the stripe $[3] \times [n]$ in Theorem 1.7.

4.4.1 Preliminaries

We use the following standard results in our proof.

Lemma 4.7 (Hoeffding's inequality)**.** *Let $x_1, \ldots, x_n \in \mathbb{R}$ be negatively correlated random variables bounded by $x_i \in [b_i, a_i]$ and define $\overline{x} = \frac{1}{n}(x_1 + \cdots + x_n)$. Then*

$$\Pr\left[|\overline{x} - \mathbb{E}[\overline{x}]| \geq t\right] \leq 2e^{\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}}.$$

Lemma 4.8 (Yao's minimax)**.** *Fix any disjoint sets $\mathcal{P}$ and $\mathcal{N}$ of functions mapping $\mathcal{X}$ to $\mathcal{Y}$. Let $\mathcal{D}_{\mathcal{P}}$ and $\mathcal{D}_{\mathcal{N}}$ be probability distributions on functions mapping $\mathcal{X}$ to $\mathcal{Y}$ that satisfy*

$$\Pr_{f\sim\mathcal{D}_{\mathcal{P}}}[f\in\mathcal{P}]=1 \qquad \text{and} \qquad \Pr_{g\sim\mathcal{D}_{\mathcal{N}}}[f\in\mathcal{N}]=1-o(1).$$

Let $\mathcal{D}$ be the distribution where with probability $\frac{1}{2}$ we sample from $\mathcal{D}_Y$ and with probability $\frac{1}{2}$ we sample from $\mathcal{D}_N$. If any non-adaptive deterministic algorithm Π with query complexity q can not answer correctly with probability $\frac{2}{3}$, then any non-adaptive randomized algorithm that decides whether $f\in\mathcal{P}$ or $f\in\mathcal{N}$ with error at most $\frac{1}{4}$ makes $\Omega(q)$ queries.

Proposition 4.9 (Theorem 332 [29])**.** *Let $a,b\in[n]$ be two numbers picked uniformly at random. The probability that the pair (a,b) is co-prime is >0.5.*

Definition 4.10. Given any vector $a\in\mathbb{Z}^d$ whose first two coordinates a_1 and a_2 are coprime, the *canonical basis completion* of a is the basis $B(a)=(b^{(1)}(a),\ldots,b^{(d)}(a))\in\mathbb{Z}^{d\times d}$ whose ith column is

$$b^{(i)}(a)=\begin{cases}a & \text{if } i=1\\ c_1e_1+c_2e_2 & \text{if } i=2\\ e_i & \text{if } 3\le i\le d\end{cases}$$

where c_1 and c_2 are the integers that satisfy $a_1c_1-a_2c_2=1$ and $e_i\in\mathbb{Z}^d$ is the vector with value 1 in the ith coordinate and 0 in all other coordinates.

The next proposition shows that the canonical basis completion of any vector $a\in\mathbb{Z}^d$ that satisfies the condition of the above definition is a basis for the lattice $\mathbb{Z}^d$.

Proposition 4.11. *Given any vector $a\in\mathbb{Z}^d$ whose first two coordinates a_1 and a_2 are coprime, the canonical basis completion $B(a)$ of a is a basis for the lattice $\mathbb{Z}^d$.*

Proof. The matrix $B(a)$ is a unimodular matrix. From Proposition 3.13 it follows that $B(a)$ is a basis of the lattice $\mathbb{Z}^d$. □

4.4.2 Constructions

In this subsection we show how to construct the distributions $\mathcal{D}_Y,\mathcal{D}_N$. We also prove that every function in $\mathcal{D}_Y$ is convex and every function in $\mathcal{D}_N$ is $\frac{1}{20}$-far from convex.

Let $\mathcal{B}$ be the distribution over bases obtained by drawing a vector $a\in\mathbb{Z}^d$ uniformly at random among all vectors whose coordinates are in the range $0\le a_1,a_2,\ldots,a_d\le\frac{n}{4d}$ and

whose first two coordinates a_1 and a_2 are coprime and returning the canonical basis $B(a)$ for a. Refer to Definition 4.10 for the definition of $B(a)$.

The distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$ are both obtained by drawing a basis from $\mathcal{B}$ and starting with a convex function g_B associated with that basis that we will call the *canonical* convex function for B.

Definition 4.12. The *canonical convex function* for a basis B of $\mathbb{Z}^d$ is the function $g_B \colon \mathbb{Z}^d \to \mathbb{Z}$ defined by

$$g_B(x) = (x_1^B)^2 + 2\sum_{i=2}^{d}(x_i^B)^2.$$

Our distribution on convex functions is obtained by shifting the values of the canonical convex function g_B in a way that preserves convexity.

Definition 4.13 ($\mathcal{D}_Y$)**.** Let $\mathcal{S}^B$ to be the distribution on functions $h \colon [n]^d \to \mathbb{Z}$ obtained by drawing values $\sigma(z) \in \{\pm 1\}$ independently and uniformly at random for each $z \in \mathbb{Z}^{d-1}$ and defining

$$h(x) = g_B(x) + \sigma(x_2^B, \ldots, x_d^B)$$

for each $x \in [n]^d$. Let $\mathcal{D}_Y$ be the distribution obtained by drawing $B \sim \mathcal{B}$ and then drawing a function $h \sim \mathcal{S}^B$.

Our distribution on functions that are far from convex is similar, except that the shifts of the canonical convex function g_B are now constructed in a way that will create many disjoint violations of convexity.

Definition 4.14 ($\mathcal{D}_N$)**.** Let $\mathcal{A}^B$ be the distribution on functions $h \colon [n]^d \to \mathbb{Z}$ obtained by drawing values $\sigma(z) \in \{\pm 1\}$ independently and uniformly at random for each $z \in \mathbb{Z}^{d-1}$ and defining

$$h(x) = g_B(x) + \sigma(x_2^B, \ldots, x_d^B) \cdot (-1)^{x_1^B}$$

for each $x \in [n]^d$. Let $\mathcal{D}_N$ be the distribution obtained by drawing $B \sim \mathcal{B}$ and then drawing a function $h \sim \mathcal{A}^B$.

We complete this section by showing that the functions in the support of $\mathcal{D}_Y$ are indeed convex and that the functions in the support of $\mathcal{D}_N$ are far from convex.

Claim 4.15. *Every function in the support of $\mathcal{D}_Y$ is convex.*

Proof. Fix any B in the support of $\mathcal{B}$, any h in the support of $\mathcal{S}^B$, and any points $z, x^{(1)}, \ldots, x^{(k)} \in \mathbb{Z}^d$ such that $z = \sum_{i=1}^k \lambda_i x^{(i)}$ is a convex combination of the points $x^{(1)}, \ldots, x^{(k)}$, $\lambda_1, \ldots, \lambda_k \geq 0$ and $\sum_{i=1}^k \lambda_i = 1$. We will show that $\sum_{i=1}^k \lambda_i h(x^{(i)}) \geq h(z)$.

Let us define $\delta^{(1)}, \ldots, \delta^{(k)} \in \mathbb{Z}^d$ to be the vectors for which $(x^{(i)})^B = z^B + \delta^{(i)}$ for each $i \in [k]$. Then the identity $\sum_{i=1}^k \lambda_i((x^{(i)})^B - z^B) = 0$ implies that $\sum_{i=1}^k \lambda_i \delta_j^{(i)} = 0$ for every $j \in [d]$ and that

$$\begin{aligned}
\sum_{i=1}^k \lambda_i g_B(x^{(i)}) &= \sum_{i=1}^k \lambda_i \Big(\big((x^{(1)})_1^B\big)^2 + 2\sum_{j=2}^d \big((x^{(i)})_j^B\big)^2 \Big) \\
&= \sum_{i=1}^k \lambda_i \Big((z_1^B + \delta_1^{(i)})^2 + 2\sum_{j=2}^d (z_j^B + \delta_j^{(i)})^2 \Big) \\
&= g_B(z) + \sum_{i=1}^k \lambda_i \Big((\delta_1^{(i)})^2 + 2\sum_{j=2}^d (\delta_j^{(i)})^2 \Big).
\end{aligned}$$

Define $I = \{i \in [k] \mid z^B_{[2,d]_\mathbb{Z}} \neq (x^{(i)})^B_{[2,d]_\mathbb{Z}}\}$. For each $i \in I$, the vector $\delta^{(i)}$ satisfies $\sum_{j=2}^d (\delta_j^{(i)})^2 \geq 1$ so we have that

$$\sum_{i=1}^k \lambda_i g_B(x^{(i)}) - g_B(z) \geq 2\sum_{i=1}^k \lambda_i \sum_{j=2}^d (\delta_j^{(i)})^2 \geq 2\sum_{i \in I} \lambda_i.$$

Furthermore, since $\sigma((x^{(i)})^B_{[2,d]_\mathbb{Z}}) - \sigma(z^B_{[2,d]_\mathbb{Z}})$ is always bounded below by -2 and the difference is zero whenever $i \notin I$, we obtain

$$\sum_{i=1}^k \lambda_i h(x^{(i)}) - h(z) \geq \sum_{i=1}^k \lambda_i g_B(x^{(i)}) - g_B(z) - 2\sum_{i \in I} \lambda_i \geq 0.$$

□

Claim 4.16. *Every function in the support of $\mathcal{D}_N$ is $\frac{1}{20}$-far from convex.*

Proof. Fix any B in the support of $\mathcal{B}$ and any h in the support of $\mathcal{A}^B$. For any points $x, y, z \in [n]^d$ that satisfy $y^B = x^B + e_1$ and $z^B = y^B + e_1$, if we have

$$h(x) = g_B(x) - 1, \qquad h(y) = g_B(y) + 1, \text{ and} \qquad h(z) = g_B(z) - 1$$

Then the triple (x, y, z) is a witness of non-convexity of h since

$$\tfrac{1}{2}h(x) + \tfrac{1}{2}h(z) = \tfrac{1}{2}g_B(x) + \tfrac{1}{2}g_B(z) - 1 = g_B(y) < h(y) = h(\tfrac{1}{2}x + \tfrac{1}{2}z).$$

Hence from how we defined h, any four points $w, x, y, z \in [n]^d$ that satisfy $x^B = w^B + e_1$, $y^B = x^B + e_1$ and $z^B = y^B + e_1$ one of (w, x, y), (x, y, z) is a witness on non-convexity. Let $L = [\frac{n}{2d}, n - \frac{n}{2d}]^d$. For $s \in \mathbb{Z}^{d-1}$, let $L_s = \{x \mid x \in [n]^d, \exists y \in L \text{ s.t } y^B_{[2,d]_\mathbb{Z}} = x^B_{[2,d]_\mathbb{Z}} = s\}$. Since $a_1, a_2, ..., a_d < \frac{n}{4d}$ we have that $|L_s| \geq 4$. And since any 4 consecutive points with the same $[2, d]_\mathbb{Z}$ coordinates, in basis B, have a witness of non-convexity, the number of witnesses in L_s is $\geq \frac{|L_s|}{7}$. Also $L \subseteq \cup_{s \in \mathbb{Z}^{d-1}} L_s$, hence the number of disjoint witnesses of non-convexity is greater than $\frac{|L|}{7} = \frac{1}{7}(1 - \frac{1}{d})^d n^d \geq \frac{1}{20}n^d$. In every disjoint non-convexity witness we have to change the value of at least one point to make the function convex. Therefore h is $\frac{1}{20}$-far from convex. □

4.4.3 Proof of Theorem 1.8

Let $\mathcal{D}$ be the distribution where with probability $\frac{1}{2}$ we pick something from $\mathcal{D}_Y$ and with probability $\frac{1}{2}$ we pick something from $\mathcal{D}_N$. In this section we prove that there does not exist a non-adaptive deterministic algorithm with query complexity $q < 0.01(\frac{n}{4d})^{\frac{d}{2}}$ that answers correctly with probability $\frac{2}{3}$ on the distribution $\mathcal{D}$. From Lemma 4.8 this would prove Theorem 1.8 as from Claim 4.15 and Claim 4.16 we know that every function in the support of $\mathcal{D}_Y$ is convex and every function in the support of $\mathcal{D}_N$ is $\frac{1}{10}$-far from convex.

Let us assume there exists such a deterministic algorithm Π that answers correctly on a distribution $\mathcal{D} = \frac{1}{2}\mathcal{D}_Y + \frac{1}{2}\mathcal{D}_N$ with probability greater than $\frac{2}{3}$. We can think of the distribution $\mathcal{D}$ as pick a $B \sim \mathcal{B}$ and pick a $\sigma \colon \mathbb{Z}^{d-1} \to \pm 1$ uniformly at random. And at the end with probability $\frac{1}{2}$ we choose whether we want a function in the support of $\mathcal{D}_Y$ or $\mathcal{D}_N$. Let the points the algorithm Π queries be $Q = x^{(1)}, x^{(2)}, ..., x^{(q)} \in \mathbb{Z}^d$.

We refer to a B in the support of $\mathcal{B}$ to be *exposed* if there exists $i, j < q$ such that $(x^{(i)})^B_{[2,d]_\mathbb{Z}} = (x^{(j)})^B_{[2,d]_\mathbb{Z}}$, otherwise we refer to it as *hidden*.

Claim 4.17. *On the distribution $\mathcal{D}$ the probability that Π answers correctly is less than 0.6.*

Proof. When B is hidden then there is no way the algorithm Π can answer correctly with probability greater than $\frac{1}{2}$. This is because $\Pr_{f \sim \mathcal{D}_Y | B \text{ is hidden}}[f|_Q = \alpha] = \Pr_{g \sim \mathcal{D}_N | B \text{ is hidden}}[g|_Q = \alpha]$. In fact it is even stronger, along with function values at the queried points even if we give

what the hidden basis B is, the algorithm can not answer correctly with probability greater than $\frac{1}{2}$. This is because, as for any $i, j < q$, $(x^{(i)})^B_{[2,d]_\mathbb{Z}} \neq (x^{(j)})^B_{[2,d]_\mathbb{Z}}$, we have $f|_Q - g_B|_Q = s$, for each $s \in \{-1, +1\}^q$, with probability $\frac{1}{2^q}$ irrespective of f being in $\mathcal{S}^B$ or $\mathcal{A}^B$. We can assume that the algorithm always answers correctly when B is exposed. The probability that the algorithm Π answers correctly is $\leq \Pr[B \text{ is exposed}] \cdot 1 + \Pr[B \text{ is hidden}] \cdot \frac{1}{2}$.

Since there are only $\binom{q}{2} < q^2$, $i, j < q$ pairs, there are at most q^2 exposed B. From Proposition 4.9 and the construction of $\mathcal{B}$ we know that $|\mathcal{B}| \geq 0.5(\frac{n}{4d})^d$ and if $q < 0.01(\frac{n}{4d})^{\frac{d}{2}}$ the probability that a $B \sim \mathcal{B}$ is exposed is $\leq \frac{1}{100}$.

Hence the success probability of the algorithm is $\leq \frac{1}{100} \cdot 1 + \frac{99}{100} \cdot \frac{1}{2} \leq \frac{101}{200}$. □

This a contradiction on the assumption that the algorithm answers correctly with probability $\frac{2}{3}$. Hence there can not exist such a non-adaptive deterministic algorithm Π.

4.4.4 Non-Adaptive Lower Bound for $[3] \times [n]$

In this section we prove a $\Omega(\sqrt{n})$ lower bound for non-adaptively testing convexity on the $[3] \times [n]$ grid. The proof is almost the same as the the higher dimensional setting with slight changes.

Let $\mathcal{B}$ be the distribution over bases obtained by drawing a vector $a \in \mathbb{Z}^2$ uniformly at random among all vectors whose first coordinate is 1 and the second coordinate is in the range $0 \leq a_2 \leq \frac{n}{100}$ and returning the canonical basis $B(a)$ for a.

Define the distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$ as above with the one modification that the domain of h is set to be $[3] \times [n]$ instead of $[n]^d$. In this setting, we again have that every function in the support of $\mathcal{D}_Y$ is convex, using the same argument as in Claim 4.15. But now it is no longer true that every function in the support of $\mathcal{D}_N$ is $\frac{1}{10}$-far from convex. Instead, we have that a function $f \sim \mathcal{D}_N$ is $\frac{1}{10}$-far from convex with probability $1 - o(1)$.

Claim 4.18. *A function $f \sim \mathcal{D}_N$ is $\frac{1}{10}$-far from convex with probability $1 - o(1)$.*

Proof. For any B in the support of $\mathcal{B}$, a function $h \sim \mathcal{A}^B$ is $\frac{1}{10}$-far from convex with probability $1 - o(1)$. Let $X = \{x \mid x_1 = 0, 0 \leq x_2 \leq \frac{9n}{10}\}$. For any points $x \in X$ and $y, z \in \mathbb{Z}^2$ that satisfy $y^B = x^B + e_1$ and $z^B = y^B + e_1$ we have that $y, z \in [3] \times [n]$ and

$$h(x) = g_B(x) - 1, \qquad h(y) = g_B(y) + 1, \text{ and } \qquad h(z) = g_B(z) - 1$$

with probability $\frac{1}{2}$. Therefore, x, y, z form a witness for non-convexity with probability $\frac{1}{2}$. This is true for all $x \in X$. Using Hoeffding's inequality the probability that the number of

witnesses for non-convexity is less than $\frac{n}{3}$ is $\leq e^{-cn}$. Hence with probability $1 - e^{-cn}$ the distance to convexity is at least $\frac{\frac{n}{3}}{3n} \geq \frac{1}{10}$. □

Any non-adaptive deterministic algorithm which performs $q < \frac{\sqrt{n}}{100}$ can not answer correctly with probability grater than 0.6. The proof is similar to that of Claim 4.17. From Lemma 4.8 this completes the proof of the lower bound in Theorem 1.7.

4.5 Related Work: Testing Results for Other Notions of Convexity

In this section, we discuss the testing results related to separate convexity and linear convexity defined in Section 3.2.

Separate convexity

Ben-Eliezer [7] showed that there exists a non-adaptive algorithm for testing separate convexity of functions over $[n]^2$, with query complexity $O(n)$, and a general non-adaptive algorithm for testing separate convexity of functions over the hypergrid $[n]^d$ with query complexity $O(n^{d-1})$. This result follows from their general result for testing local properties and the fact that separate convexity is a local property. They refer to a property as *k-local* if it can be defined by a family of $k \times k \times \ldots \times k$ forbidden consecutive patterns. Convexity over the line $[n]$ is 3-local, as the function is convex if there is no violation of convexity on every three consecutive points. The same is the case for separate convexity. However, discrete convexity over higher dimensions is not a local property. There are functions that are far from being discrete convex but every local region satisfies convexity. Our lower bound construction for Theorem 1.8 is one such example. Therefore their algorithms do not work in our setting.

Linear convexity

Blais, Raskhodnikova, and Yaroslavtsev [12] showed that non-adaptive algorithms that test linear convexity of functions over the hypergrid $[n]^d$ have query complexity $\Omega(d \log n)$.

Chapter 5

Minimizing Discrete Convex Functions

In this chapter, we prove the results we obtained for minimizing discrete convex functions over the hypergrid. In Section 5.1, we discuss the main ideas used in our two-dimensional and high-dimensional minimization algorithms. Next, in Section 5.2, we present the necessary background for proving the theorems in this chapter. In Section 5.3, we present the two-dimensional minimization algorithm and prove its correctness and complexity bounds. Similarly, in Section 5.4 we present the high-dimensional minimization algorithm and prove its correctness and complexity bounds. Finally, in Section 5.5 we give an overview of the minimization algorithm by Veselov et al. For the reader's convenience, we restate the theorems we prove in this chapter below.

Theorem 1.9. *Let $f, g\colon \mathbb{Z}^2 \to \mathbb{R}$ be discrete convex functions. Given access to the comparison oracle for f and the valuation oracle for g, we can solve the optimization problem*

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & g(x) \leq 0 \\ & x \in [n] \times [n] \end{aligned} \tag{1.3}$$

using $\mathrm{poly}(\log n)$ *time and queries.*

Theorem 1.10. *Let $f, g\colon \mathbb{Z}^d \to \mathbb{R}$ be discrete convex functions. Given access to the comparison oracle for f and the valuation oracle for g, we can solve the optimization*

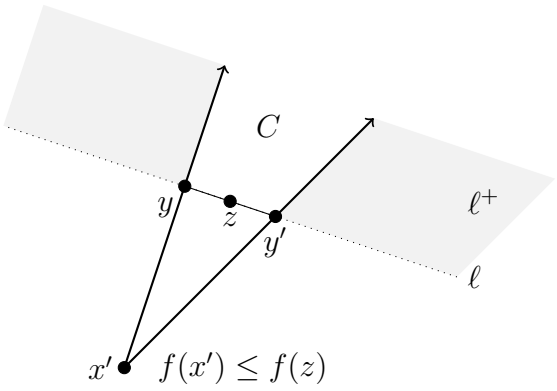

Figure 5.1: Illustration of the *fencing* idea in the minimization algorithm. The shaded region corresponds to the set $\ell^+ \setminus C$ that is "eliminated" when $f(x') \leq f(z) \leq \min\{f(y), f(y')\}$.

problem

$$\begin{aligned} \min_{x} \quad & f(x) \\ & g(x) \leq 0 \\ & x \in [n]^d \end{aligned} \tag{1.4}$$

using $2^{O(\text{poly}(\log \log n))}$ *time and queries for any constant* d*.*

5.1 Overviews

5.1.1 Two-Dimensional Minimizer Overview

The algorithm uses four main ideas. Throughout this overview, fix $f\colon \mathbb{Z}^2 \to \mathbb{R}$ to be any discrete convex function. Let $\widetilde{f} : \mathbb{R}^2 \to \mathbb{R}$ be a convex extension of f.

Fencing. The idea of fencing is to eliminate points from consideration like we do in cutting plane methods, but we need new ideas as we are in the discrete setting. If we have a line l and a point $x' \in \mathbb{Z}^2$ such that for any $x \in l$, we have $\widetilde{f}(x) \geq f(x')$, then we can eliminate all the points on the side of the line not containing x', as the function is convex. But unfortunately since we can access the function on only integer points, there is no way to know what the least value on the line is. For instance, consider the set of points $[n]$. Let $f(\lfloor \frac{n}{2} \rfloor)$ be the minimum of the set. Since the function is convex, we know all the

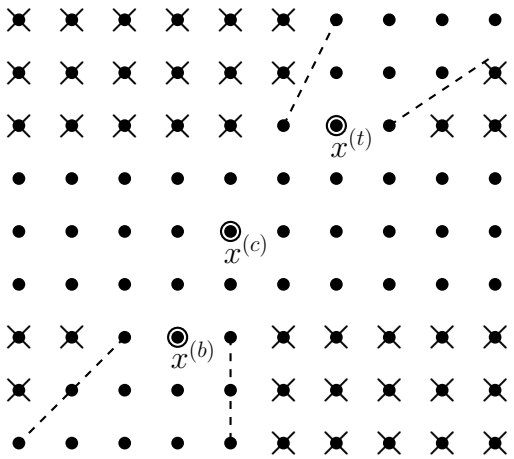

Figure 5.2: Finding fences when $x^{(c)}$ is the minimum of the three points. We process the region above t and region below b. The points between the lines in the processed region are not eliminated.

points between 1 and $\lfloor \frac{n}{2} \rfloor - 1$ and between $\lfloor \frac{n}{2} \rfloor + 1$ and n will have function value greater than $f(\lfloor \frac{n}{2} \rfloor)$ but there could exist a non-integer point between $\lfloor \frac{n}{2} \rfloor - 1, \lfloor \frac{n}{2} \rfloor + 1$ whose function value is much smaller than $f(\lfloor \frac{n}{2} \rfloor)$. We refer to this region where the fractional minimum could be less than the integer minimum as the *region of uncertainty*. Hence, in the discrete setting for a point x' we can not have a line where all the points have function value greater than $f(x')$. Instead what we can have is a line where almost all the points have function value greater than $f(x')$ with a region of uncertainty where the function value could potentially be less than $f(x')$. With the help of such a line we can eliminate many points on the side of the line not containing x' but not all of them. A more formal description of fences in two dimensions is described below.

Let $x' \in [n]^2$ be a point, ℓ be a line passing through $[n]^2$, $z \in [n]^2$ be the minimum of the set $\ell \cap [n]^2$ and $y, y' \in [n]^2$ be the adjacent points of z on the line ℓ. If $f(x') \leq f(z)$ then we can use the pair x', ℓ to restrict the subset of points in the domain of f where its minimum can occur. Let ℓ^+ be all the points on the side of the line ℓ not containing x' and let C be the region between the rays $\text{ray}(x', y)$, $\text{ray}(x', y')$. Then every point $w \in (\ell^+ \setminus C) \cap \mathbb{Z}^2$ satisfies $f(w) \geq f(z) \geq f(x')$. Hence we can eliminate all the points in the region $\ell^+ \setminus C$ when looking for the minimum value of f, as illustrated in Figure 5.1. We call this process of restricting our search space *fencing*.

FINDING FENCES. For the fencing idea to work, we need to address two challenges.

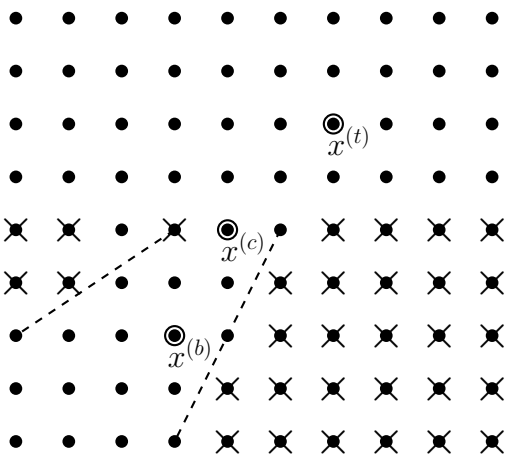

Figure 5.3: Finding fences when $x^{(t)}$ is the minimum of the three points. We process the region below row c. The points between the lines in the processed region are the not eliminated.

The first is to come up with an (x', ℓ) pair. One way to address this is via the following strategy. Pick any two rows $j \neq j' \in [h]$, where h is the number of rows in the grid. Assume the minimum value of f on the jth row is less than or equal to its minimum value on the j'th row. Choose $x' = (i, j)$ to be the point at which f attains its minimum on row j, and ℓ be the line passing through row j'.

The second challenge is to come up with a good (x', ℓ) pair such that we eliminate a significant fraction of the remaining points. Let our rectangle have h rows and without loss of generality assume $j < j'$. During the fencing step we eliminate some points between the rows j' and h(or between 0 and j if minimum of j' were smaller). When this happens, we say the region between the rows j' and h is *processed*. If the fence in the region is going upward, we say it is an *upward processed* region and if the fence is going downward, we say it is a *downward processed* region. We want j, j' to be far from the boundaries so that we can process a larger part of the rectangle with fewer queries. We also prefer j, j' to be reasonably far enough so that the number points not eliminated is small in the processed region. For example consider j, j' to be $\frac{h}{2} - 1, \frac{h}{2}$ versus $\frac{h}{4}, \frac{h}{2}$. In the first case the number of points not eliminated in the rows between j' and h is $O(h^2)$ while in the second case it is $O(h)$. To ensure this, we pick three rows $\frac{h}{4}, \frac{h}{2}, \frac{3h}{4}$. We check which of the three rows has the least function value. If row $\frac{h}{2}$ has the least function value, then regions between rows $0, \frac{h}{4}$ and $\frac{3h}{4}, h$ are processed. If row $\frac{h}{4}$ has the least then the region between rows $\frac{h}{2}, h$ is processed and if row $\frac{3h}{4}$ has the lest then the region between the rows $0, \frac{h}{2}$ is processed.

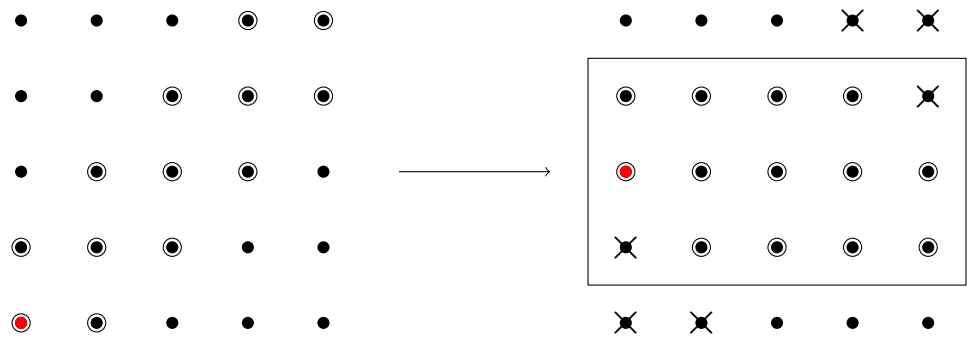

Figure 5.4: Illustration of a *concentrating change of basis* operation. The circled points in the left grid are the points that have not yet been eliminated. The red point has the coordinate $(0, 0)$. The second grid illustrates the position of the circled points after we perform a change the basis to $[(1, 1)\,(1, 0)]$. The red point again has the coordinates $(0, 0)$ in the second grid, and all the circled points are now in a rectangle with only three rows. The crossed-out points in the second grid represent the points that were not present in the first grid.

We keep repeating the process with the remaining region. At the end, we are still left with a few points not eliminated in the processed regions which need to be taken care of.

Concentrating change of basis. Notice that all the points not eliminated in a region are horizontally close to a line. For example in Figure 5.3 all the points not eliminated in the rows between $0, \frac{h}{2}$ are at a horizontal distance at most 6 from the line joining $x^{(t)}$ with any of the not eliminated points in the region. Similarly in Figure 5.2 we can observe that all the points not eliminated between the rows $\frac{3h}{4}, h$ are horizontally at a distance at most 4 from a line joining $x^{(c)}$ with any of the points not eliminated in the region. The same can be observed for the region between the rows $0, \frac{h}{4}$.

After the finding fences step we are left with a few lines and points horizontally close to them. However, repeating the same fencing procedure by finding the minimum value of f on other rows will not eliminate a significant fraction of these points since the fenced-in region will still contain a large fraction of the points that have not yet been eliminated. Consider a simple example where the minimum value of f on each row is attained at a point on the diagonal: in this case, each point (j, j) is not eliminated by any step of the fencing procedure that does not explicitly find the minimum value of f on the jth row, so a naïve use of this procedure has time and query complexity $\Omega(h)$.

What the simple example suggests is that we want to repeat the fencing procedure in

a "different direction", where we find the minimum value of f along other lines instead of along the rows. We do this via a *concentrating change of basis* operation. Identifying a good basis change is not completely straightforward, but nevertheless we show that there is always a concentrating change of basis which guarantees that all the points in the processed region which have not been eliminated by the current fencing step end up in a much smaller rectangle after the change of basis, as illustrated in Figure 5.4.

Close to two lines. The final observation/idea which is crucial to achieving the desired $O(\text{poly}(\log h))$ query complexity is that, all the points not eliminated in the rectangle after the finding fences step are horizontally close to just two lines. After finding fences step, we end up with $\log h$ sets of points where each set is horizontally close to a line. We can perform a change of basis and cast each of these sets of points into a smaller rectangle but that would result in $\log h$ sub-problems and a time complexity of $(\log h)^{\log \log h}$. Showing that the points are close to just two lines reduces the number of sub-problems from $\log h$ to 2. The proof for this relies on the property that all the processed regions above the row containing the minimum of the minimums, encountered so far, are upward processed and all the regions processed below it are downward processed. This is the reason behind picking the $\frac{h}{4}, \frac{h}{2}, \frac{3h}{4}$ strategy of choosing lines in the finding fences step over the natural idea to pick j, j' to be $\frac{h}{3}, \frac{2h}{3}$. Even though in the second approach also, a constant fraction of the rectangle is processed in each step and in each processed region there are at most $O(h)$ points that are not eliminated.

5.1.2 High-Dimensional Minimizer Overview

The high-dimensional minimizer is similar to the two-dimensional minimization algorithm in spirit. But due to the complexity added by higher dimensions, some of the ideas in two dimensions do not extend, and we need to come up with new ones. Wherever we use new ideas we discuss the need for them and compare them with the corresponding ideas in two dimensions.

Slabs. A natural extension of the fencing idea to higher dimensions is to use a point and a hyperplane instead of a point and a line. But unfortunately it doesn't work, as the region of uncertainty of a hyperplane is more complicated than the line. Therefore in higher dimensions, instead of fences we construct polytopes called *slabs*. They perform a similar function as a fence, as in they let us eliminate points outside them. Each slab is defined by three parameters: A hyperplane H, direction e_i, and width w, and contains all points that are at a distance w along the direction e_i from the hyperplane H.

Constructing slabs. We use an algorithm, MinRegion, that constructs these slabs by recursing on the dimension. The algorithm takes in as input the d-dimensional hypergrid $[n]^d$, along with the functions, and outputs a point from $[n]^d$ and a set of slabs such that all the points in $[n]^d$ that are outside the slabs have a larger function value than the point.

For the MinRegion algorithm we borrow some ideas from the finding fences step in two dimensions. We pick hyperplanes at one-quarter, half, and three-quarter mark along the dimension d, and run the MinRegion algorithm on the intersection of these hyperplanes with $[n]^d$. Using the points and slabs output by the MinRegion algorithm on these $(d-1)$-dimensional sub-problems, we construct slabs for half of the region. Which region we construct slabs for, like in the two-dimensional case, depends on which of the points output by the sub-problems has the least function value. Once this is done, we repeat the same for the remaining region like in two dimensions. One main difference with the two-dimensional counterpart is that, there we find the minimum on the rows and compare the minimums, and here we compare the points returned by the MinRegion algorithm on these sub-problems, which need not necessarily be the minimums on the hyperplanes.

Concentrating change of basis. The idea of concentrating change of basis extends to higher dimensions, albeit with some additional challenges due to the high dimensional setting. We get around those challenges and show that there exists a concentrating change of basis that transforms all the points in a slab into a "small" rectangle. In two dimensions we had that all the points that are not in the fenced in regions are close to two lines. We do not have this property in higher dimensions and need to handle points in each slab separately. Therefore the number of sub-problems we need to solve is the number of slabs, which is $O(\text{poly}(\log n))$. This gives us a quasi-polynomial time and query complexity of $2^{O(\text{poly}(\log\log n))}$ in higher dimensions, instead of the polynomial complexity in two dimensions.

5.2 Preliminaries

In Section 5.2.1, we introduce a new class of relations over the points in $\mathbb{R}^d$ and prove some properties about them. These relations makes the presentation of our proofs much simpler. In Section 5.2.2, we introduce different geometric notations we use and define a notion of distance which measures distance along a direction and prove some interesting properties related to it. Since we are dealing with integer points, diophantine equations invariably show up and we mention the results we need related to them in Section 5.2.3.

5.2.1 Preference Relation

Given two functions, $f, g : \mathbb{R}^d \to \mathbb{R}$, we use $\leq_{f,g}$ to represent the *preference relation* over the points in $\mathbb{R}^d$ with respect to f, g.

Definition 5.1. Let $f, g : \mathbb{R}^d \to \mathbb{R}$. For any $x, y \in \mathbb{R}^d$, we say $x \leq_{(f,g)} y$ if $\max\{g(x), g(y)\} \leq 0$ and $f(x) \leq f(y)$, or if $\max\{g(x), g(y)\} > 0$ and $g(x) \leq g(y)$.

We use $\min^{\leq_{(f,g)}}$, $\max^{\leq_{(f,g)}}$ to denote that the comparisons are made using the relation $\leq_{(f,g)}$.

For two points $x, y \in \mathbb{R}^d$, we say x *is preferred to* y if they satisfy the preference relation $x \leq_{(f,g)} y$. Intuitively, think of f as the objective function and g as the feasibility function. If both the points satisfy the feasibility constraint defined by the feasibility function, then we look at the objective function value to decide which of the points is preferred and if at least one of the points violates the feasibility constraint defined by the feasibility function, then we look at just the feasibility function values to decide which of the points is preferred.

The preference relation defined above is both transitive and complete, as we show in Propositions 5.2 and 5.3 respectively.

Proposition 5.2. *For any $f, g : \mathbb{R}^d \to \mathbb{R}$, the relation $\leq_{(f,g)}$ is transitive i.e. for any $x, y, z \in \mathbb{R}^d$ such that $x \leq_{(f,g)} y$ and $y \leq_{(f,g)} z$ we have $x \leq_{(f,g)} z$.*

Proof. We prove it separately for $g(y) > 0$ and $g(y) \leq 0$.

Case 1: $g(y) > 0$

Since $g(y) > 0$, $x \leq_{(f,g)} y$ and $y \leq_{(f,g)} z$ we get $g(x) \leq g(y)$ and $g(y) \leq g(z)$. Therefore $x \leq_{(f,g)} z$, as $g(z) > 0$ and $g(x) \leq g(z)$.

Case 2: $g(y) \leq 0$

Since $x \leq_{(f,g)} y$ and $g(y) \leq 0$ we get $f(x) \leq f(y)$. Given that $y \leq_{(f,g)} z$, if $g(z) \leq 0$ then $f(y) \leq f(z)$ and therefore $x \leq_{(f,g)} z$ as $g(x), g(z) \leq 0$ and $f(x) \leq f(z)$. If $g(z) > 0$ then $x \leq_{(f,g)} z$ as $g(z) > 0$ and $g(x) \leq g(z)$. □

Proposition 5.3. *For any $f, g : \mathbb{R}^d \to \mathbb{R}$ the relation $<_{(f,g)}$ is complete i.e. for any $x, y \in \mathbb{R}^d$ either $x \leq_{(f,g)} y$ or $y \leq_{(f,g)} x$.*

Proof. We split the proof in two cases: $\max\{g(x), g(y)\} > 0$ and $\max\{g(x), g(y)\} \leq 0$.

Case 1: $\max\{g(x), g(y)\} > 0$

If $g(x) \leq g(y)$, then $x \leq_{(f,g)} y$. If $g(y) \leq g(x)$, then $y \leq_{(f,g)} x$.

Case 2: $\max\{g(x), g(y)\} \leq 0$

If $f(x) \leq f(y)$, then $x \leq_{(f,g)} y$. If $f(y) \leq f(x)$, then $y \leq_{(f,g)} x$. □

When the functions f, g are convex the preference relation, $\leq_{(f,g)}$, satisfies the additional property stated below.

Proposition 5.4. *Let $f, g : \mathbb{R}^d \to \mathbb{R}$ be convex functions. For any $x, z \in \mathbb{R}^d$, if there exists $y = \lambda x + (1-\lambda)z$, $0 \leq \lambda \leq 1$ such that $x \leq_{(f,g)} y$ then we have $x \leq_{(f,g)} z$.*

Proof. We prove it for the cases when $\max\{g(x), g(y)\} \leq 0$ and $\max\{g(x), g(y)\} > 0$ separately.

Case 1: $\max\{g(x), g(y)\} \leq 0$

We have $f(x) \leq f(y)$ as $x \leq_{(f,g)} y$. This combined with the fact that x, y, z lie on a line, and f is a convex function, gives us $f(x) \leq f(z)$. If $g(z) > 0$, then $x \leq_{(f,g)} z$ as $g(x) \leq g(z)$. If $g(z) \leq 0$, then $x \leq_{(f,g)} z$ as $f(x) \leq f(z)$.

Case 2: $\max\{g(x), g(y)\} > 0$

We have $g(x) \leq g(y)$ and $g(y) > 0$ as $x \leq_{(f,g)} y$ and $\max\{g(x), g(y)\} > 0$. This combined with the fact that x, y, z lie on a line, and g is a convex function, gives us $g(x) \leq g(y) \leq g(z)$, $g(z) > 0$. Therefore $x \leq_{(f,g)} z$. □

5.2.2 High Dimensional Linear Spaces

For any two distinct points $x, y \in \mathbb{R}^d$, the *line* going through x and y is $\text{line}(x, y) = \{x + t(y - x) : t \in \mathbb{R}\}$. We write $\text{line}(x) = \text{line}(o, x)$ to denote the line going through the origin $o = (0, 0, \ldots, 0)$ and the point x. The *ray* with initial point x going through y is the half-line $\text{ray}(x, y) = \{x + t(y - x) : t \in \mathbb{R}, t \geq 0\}$. We write $\text{ray}(x) = \text{ray}(o, x)$ to denote the ray with the origin as initial point going through x.

Definition 5.5. Let $x, y, z, z' \in \mathbb{R}^d$. The point $z' \in \text{cone}_z(x, y)$ if $\exists x' \in \text{ray}(z, x), \exists y' \in \text{ray}(z, y)$ such that $z' = \lambda x' + (1-\lambda)y'$ for some $0 \leq \lambda \leq 1$. We write $\text{cone}(x, y) = \text{cone}_o(x, y)$.

Definition 5.6. Let $z \in \mathbb{R}^d$, $c \in \mathbb{R}$. Then $H = \{x \in \mathbb{R}^d : \langle z, x\rangle = c\}$ is a *hyperplane* with normal z.

Definition 5.7. Let $H \subset \mathbb{R}^d$ be a hyperplane, $w \in \mathbb{R}$ and $z \in \mathbb{R}^d$. We define the polytope $\text{slab}(H, z, w)$ as follows:

$$\text{slab}(H, z, w) = \left\{x \in \mathbb{R}^d : \exists\, |\alpha| \leq w \text{ s.t } x + \alpha \tfrac{z}{\|z\|} \in H\right\}.$$

We refer to H as the *hyperplane defining the slab*. We refer to z as the *thickness direction of the slab* and w as the *thickness of the slab* in the direction z.

For two points $a, b \in \mathbb{Z}^d$ that satisfy $a_i < b_i$ for each $i = 1, \ldots, d$, the *rectangle* $\text{rect}(a, b) = [a_1, b_1]_{\mathbb{Z}} \times [a_2, b_2]_{\mathbb{Z}} \times \cdots \times [a_d, b_d]_{\mathbb{Z}}$ is the set of all points $x \in \mathbb{Z}^d$ whose d coordinates all satisfy $a_i \leq x_i \leq b_i$. Given a rectangle $R = \text{rect}(a, b)$, we define $\min_i(R) = a_i$ and $\max_i(R) = b_i$. The *length* of a rectangle R in dimension i is $\text{length}_i(R) = \max_i(R) - \min_i(R)$. For $k \in [d]$, $\min_k(R) \leq i \leq \max_k(R)$ we use $R_{x_k > i}$ to refer to points in R with $x_k > i$ and similarly $R_{x_k < i}, R_{x_k = i}$ to refer to points in R with $x_k < i$ and $x_k = i$ respectively. The *convex extension of the rectangle* $R = \text{rect}(a, b)$ is the set $\widetilde{R} = \{x \in \mathbb{R}^d : \forall i \in [d], a_i \leq x_i \leq b_i\}$.

Definition 5.8. Let $S \subset \mathbb{R}^d$, $x' \in \mathbb{R}^d$. We define the affine space $\text{affine}(x', S)$ as follows

$$\text{affine}(x', S) = \{x \in \mathbb{R}^d : x = y + \alpha(y' - x') \text{ for } y, y' \in S, \alpha \in \mathbb{R}\}.$$

For a set $S \subset \mathbb{R}^d$ and a point $x' \in \mathbb{R}^d$ we define $S - x' = \{x : x + x' \in S\}$.

The *distance along the jth dimension* of a point $x \in \mathbb{R}^d$ from the hyperplane $H \subseteq \mathbb{R}^d$ is represented by $\text{dist}_j(x, H)$. We say $\text{dist}_j(x, H) = |c|$, if $\exists c \in \mathbb{R}$ such that $x + ce_j \in H$. Otherwise, $\text{dist}_j(x, H) = \infty$. We use $e_1, e_2, \ldots, e_d$ to represent the standard basis vectors of $\mathbb{R}^d$.

The proposition below gives a relation between the distance of a point from a hyperplane and distance of the point from the hyperplane along a direction e_j.

Proposition 5.9. *For any $y, a \in \mathbb{R}^d$, $\phi \in \mathbb{R}^+$, and $j \in [d]$ we have $|\langle a, y\rangle| \leq \phi$ if and only if $\text{dist}_j(y, \langle a, x\rangle = 0) \leq \frac{\phi}{|a_j|}$.*

Proof. Assume $\text{dist}_j(y, \langle a, x\rangle = 0) \leq \frac{\phi}{|a_j|}$. Then there exists $\delta \in [-\frac{\phi}{|a_j|}, \frac{\phi}{|a_j|}]$ such that $\langle a, (y + \delta e_j)\rangle = 0 \implies |\langle a, y\rangle| \leq \phi$.

Assume $|\langle a, y\rangle| \leq \phi$. There exists $\delta \in [0, \frac{\phi}{|a_j|}]$ such that $|\langle a, y\rangle| = \delta\, |a_j|$. Therefore we get either $\langle a, (y + \delta e_j)\rangle = 0$ or $\langle a, (y - \delta e_j)\rangle = 0$ which implies $\text{dist}_j(y, \langle a, x\rangle = 0) \leq \frac{\phi}{|a_j|}$. □

The proposition below bounds the distance between two hyperplanes along a direction e_j, in a rectangle.

Proposition 5.10. *Let $R \subset \mathbb{Z}^d$ be a rectangle with $|\min_i(R)|, |\max_i(R)| \leq h_i \ \forall i$. For any $y' \in R$, $z, a \in \mathbb{R}^d$ such that $\langle z, y' \rangle = 0$ we have*

$$\text{dist}_j(y', \langle a, x \rangle = 0) \leq \sum_{i=1}^{d} h_i \left| \frac{a_i}{a_j} - \frac{z_i}{z_j} \right|$$

Proof. Let $\langle a, y' + \delta e_j \rangle = 0$, where $|\delta| = \text{dist}_j(y', \langle a, x \rangle = 0)$. Solving for y'_j from $\langle z, y' \rangle = 0$ we get $y'_j = \sum_{i \in [d] \setminus j} y'_i \frac{z_i}{z_j}$. Substituting the value of y'_j in $\langle a, y' + \delta e_j \rangle = 0$ we get $|\delta| \leq \sum_{i=1}^{d} |y'_i| \left| \frac{a_i}{a_j} - \frac{z_i}{z_j} \right|$. As $y' \in R$ substituting $|y'_i| \leq h_i$ into the previous equation we get $|\delta| \leq \sum_{i=1}^{d} h_i \left| \frac{a_i}{a_j} - \frac{z_i}{z_j} \right|$. □

5.2.3 Diophantine Equations and Approximation

Diophantine equations are equations whose solutions of interest are integers. They can be solved efficiently, in terms of their binary representation, when the dimension is fixed.

Proposition 5.11 (Corollary 5.2a, 5.3b [54])**.** *Let $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$. If the equation $Ax = b$ has an integral solution, then there exists a solution $x' \in \mathbb{Z}^d$ with $\|x'\|_\infty = O(\text{poly}(\|A\|_\infty, \|b\|_\infty))$ and can be found in time $O(\text{poly}(\log(\|A\|_\infty), \log(\|b\|_\infty)))$ for fixed m, n.*

Proposition 5.12 ([35])**.** *Let $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$. If the equation $Ax \leq b$ has an integral solution then it can be found in time $O(\text{poly}(\log(\|A\|_\infty), \log(\|b\|_\infty)))$ for fixed m, n.*

We also use the following classic theorem about approximating real numbers with fractions with small denominator.

Theorem 5.13 (Dirichlet's approximation theorem)**.** *For any $\alpha \in \mathbb{R}^d$ and $N \in \mathbb{R}^+$, there exist $p \in \mathbb{Z}^d$ and $q \in \mathbb{Z}^+$ that satisfy $q \leq N$ and*

$$\left| \alpha_i - \frac{p_i}{q} \right| \leq \frac{1}{q N^{\frac{1}{d}}}$$

for each $i = 1, \ldots, d$.

5.3 Minimization in Two Dimensions

In Section 5.3.1, we present the 2D-Minimizer algorithm. In Section 5.3.2, we present the subroutines that were used in 2D-Minimizer. In Section 5.3.3, we show the correctness and analyse the computational complexity of the 2D-Minimizer. Note that the 2D-Minimizer algorithm can not be used directly to minimize discrete convex functions over the hypergrid $[n]^2$. We need to manipulate the feasibility function given as input for it to perform the desired task. We discuss more on this and prove Theorem 1.9 in Section 5.3.4.

5.3.1 2D Minimization Algorithm

The algorithm $2D$-Minimizer takes in as input discrete convex functions $f, g : \mathbb{Z}^2 \to \mathbb{R}$ and a rectangle R and outputs a point $y \in \mathbb{Z}^2$ such that for any $x \in R$, $y \leq_{(f,g)} x$. Note that the point y output by the algorithm need not be in the rectangle. The formal lemma is stated below.

Lemma 5.14. *Let $f, g : \mathbb{Z}^2 \to \mathbb{R}$ be discrete convex functions and $R \subset \mathbb{Z}^2$ be a rectangle. The algorithm* 2D-Minimizer(f, g, R) *outputs a point $y \in \mathbb{Z}^2$ such that for any $x \in R$, $y \leq_{(f,g)} x$. The algorithm has time and query complexity $O(\text{poly}(\log h))$, where $h_1 = \text{length}_1(R)$, $h_2 = \text{length}_2(R)$ and $h = \max\{h_1, h_2\}$.*

We prove the lemma in Section 5.3.3.

Description

In lines 1-3 we check if the rectangle has few rows and if that is the case we solve the problem by brute-force. In lines 4-9 we find the rows on which fences are constructed. All the rows we visit as a part of it are added to the set V, which is the set of rows for which we have already computed the minimum. How we find these rows is explained in the overview. In line 10 we translate the rectangle so that $x^{(c^*)}$ is the new origin and f, g, V are also modified accordingly.

In lines 11-13 we assign two sets of points $S_{>j}, S_{<j}$ for each $j \in V'$. We use $S_{>j}$ to represent the set of points which are above row j and are part of all fences $\text{Fence}(f'', g'', R'', i, j)$ where $j > i \in V$'. Similarly $S_{<j}$ to represent the set of points which are below row j and part of all fences $\text{Fence}(f'', g'', R'', i, j)$ where $j < i \in V'$.

In lines 14-15 we find $z^{\text{low}}, z^{\text{high}}$. The point z^{low} is the lowest point which is part of all the sets $S_{<j}$ for all rows j above z_{low} and in V'. Similarly, z^{high} is the highest point which is part of all the sets $S_{>j}$ for all rows j below z^{high} and in V'. On the high level think of the fences as constraints and z^{low} is the lowest point in the rectangle which satisfies all the constraints applicable to it and z^{high} is the highest point which satisfies all the constraints applicable to it. The expression to find these looks complex because the set of constraints applicable to a point depends on which row it is in. For example, the constraint $\text{Fence}(f', g', R', i, j)$ does not apply to any point below row j if $i < j$.

All the points not eliminated above row 0, previously c^*, are horizontally close to the line $\text{line}(z^{\text{high}})$ and all the points not eliminated below row 0 are horizontally close to the line $\text{line}(z^{\text{low}})$.

In lines 16-18 we find the minimum of all the points below row 0 in the rectangle. In line 16 we find a change of basis B_{low} which has the property that for any $y' \in R$, if $\text{dist}_1(y', \text{line}(z^{\text{low}})) \leq 6$, then $y'^{B_{\text{low}}} \in [-10h_2, 10h_2] \times [-9\sqrt{h_2}, 9\sqrt{h_2}]$. In lines 19-21 we do the same for points above row 0.

In line 22 we compare the minimum candidates so far and output the minimum. We add x' to undo the translations before we return the minimum.

Algorithm 5: 2D-MINIMIZER(f, g, R)

// Find the minimum using bruteforce if there are only few rows

Let $x^{(k)} = \text{1D-MINIMIZER}(f_{x_2=k}, g_{x_2=k}, R_{x_2=k})$, for any $k \in [\min_2(R), \max_2(R)]_{\mathbb{Z}}$;

if $\text{length}_2(R) \le 100$ **then**

    Return $\min^{\le_{(f,g)}}_{i \in [\min_2(R), \max_2(R)]_{\mathbb{Z}}} x^{(i)}$;

// Picking rows to build fences

$h' \leftarrow \text{length}_2(R)$; $c = \frac{\min_2(R)+\max_2(R)}{2}$;

while $h' > 1$ **do**

    Let $t \leftarrow c + \frac{h'}{4}$; $b \leftarrow c - \frac{h'}{4}$; $V \leftarrow V \cup \{t, c, b\}$; // V is the set of visited rows

    $c \leftarrow \text{argmin}^{\le_{(f,g)}}_{i \in \{t,c,b\}} x^{(i)}$; // The comparison used is $\le_{(f,g)}$

    $h' \leftarrow \frac{h'}{2}$;

$c^* \leftarrow c$; $x' \leftarrow x^{(c^*)}$;

$R' \leftarrow R - x'$; $f' \leftarrow f_{I,x'}$; $g' \leftarrow g_{I,x'}$; $V' \leftarrow \{i - x'_2 : i \in V\}$;

// Construct the fences

for $j \in V'$ **do**

    $S_{>j} \leftarrow \cap_{i \in V', i<j, \text{Fence}(f',g',R',i,j) \text{ is valid}} \text{Fence}(f', g', R', i, j)$;

    $S_{<j} \leftarrow \cap_{i \in V', i>j, \text{Fence}(f',g',R',i,j) \text{ is valid}} \text{Fence}(f', g', R', i, j)$;

$z^{\text{low}} \leftarrow \text{argmin}_{y: \forall q \in V', q \ge y_2 \ y \in S_{<q}} y_2$;

$z^{\text{high}} \leftarrow \text{argmax}_{y: \forall q \in V', q \le y_2 \ y \in S_{>q}} y_2$;

// Find the minimum of f below row 0

$B_{\text{low}} \leftarrow \text{2D-FINDCONCBASIS}\big(z^{\text{low}}, h_2\big)$;

$x \leftarrow \text{2D-MINIMIZER}(f'_{(B^{-1}_{\text{low}},(0,0))}, g'_{(B^{-1}_{\text{low}},(0,0))}, [-10h_2, 10h_2] \times [-9\sqrt{h_2}, 9\sqrt{h_2}])$;

$x^{\text{low}} \leftarrow B^{-1}_{\text{low}} x$;

// Find the minimum of f above row 0

$B_{\text{high}} \leftarrow \text{2D-FINDCONCBASIS}\big(z^{\text{high}}, h_2\big)$;

$x \leftarrow \text{2D-MINIMIZER}(f'_{(B^{-1}_{\text{high}},(0,0))}, g'_{(B^{-1}_{\text{high}},(0,0))}, [-10h_2, 10h_2] \times [-9\sqrt{h_2}, 9\sqrt{h_2}])$;

$x^{\text{high}} \leftarrow B^{-1}_{\text{high}} x$;

Return $\min^{\le_{(f',g')}}\{x^{\text{low}}, (0,0), x^{\text{high}}\} + x'$;

5.3.2 Subroutines

The algorithm 2D-Minimizer uses three subroutines 1D-Minimizer, 2D-FindConcBasis, and Fence which we describe in this section.

Minimization in One Dimension(1D-Minimizer)

The 1D-Minimizer algorithm takes in as input discrete convex functions $f, g : \mathbb{Z} \to \mathbb{R}$ and a rectangle $R \subset \mathbb{Z}$ and outputs a point $y \in R$ such that for any $x \in R$, we have $y \leq_{(f,g)} x$.

Algorithm 6: 1D-Minimizer(f, g, R).

```
1 Let a ← min_1(R); b ← max_1(R) ;
2 h ← b − a;   c = (b+a)/2;
3 while h > 1 do
4     Let t ← c + h/4; b ← c − h/4;
5     c ← min^{≤_(f,g)}_{x∈{t,c,b}} x;
6     h ← h/2;
7 Return c;
```

Though there are a lot of possible one-dimensional minimizers that work on the same principle we present it the following way to match the finding fences step in the 2D-Minimizer algorithm. Refer to Figure 5.5 for an illustration of how the elimination step works.

Lemma 5.15. *Let $f, g : \mathbb{Z} \to \mathbb{R}$ be convex functions and $R \subset \mathbb{Z}$ be a rectangle. The algorithm* 1D-Minimizer(f, g, R) *returns a point $x' \in R$ such that for any $y \in R$, $x' \leq_{(f,g)} y$. The time and query complexity of the algorithm is $O(\log n)$, where $n = \text{length}_1(R)$.*

Proof. We show that for any point y eliminated by the algorithm there is a point y' in the remaining points such that $y' \leq_{(f,g)} y$. In the algorithm there are three possibilities depending on which of $c - \frac{h}{4}, c, c + \frac{h}{4}$ is the minimum according to $\leq_{(f,g)}$. We show the analysis for one of them and it similar in the rest. Assume $c \leq_{(f,g)} c - \frac{h}{4}, c + \frac{h}{4}$. In this case we eliminate points to the left of $c - \frac{h}{4}$ and points to the right of $c + \frac{h}{4}$. From Proposition 5.4, since the function is convex, any y that is eliminated has $c \leq_{(f,g)} y$. Therefore the point x' that is returned by the algorithm has the property that, for any $y \in R$, $x' \leq_{(f,g)} y$.

Time and query complexity analysis. In each iteration of the algorithm the number of points still under consideration is halved. Hence the number of times we enter the while

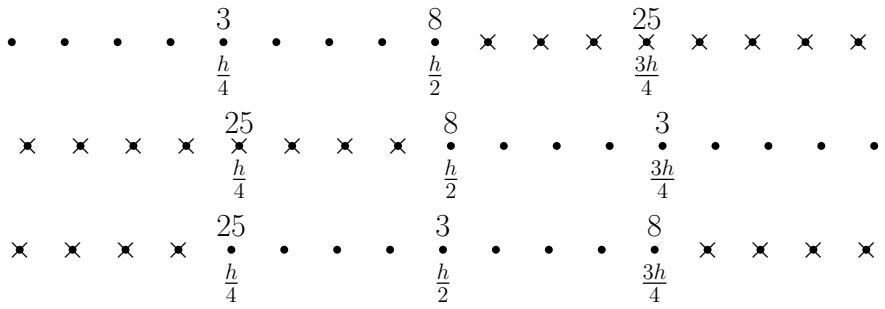

Figure 5.5: Illustration of the points eliminated from considering after the first round of the 1D algorithm when the minimum of f on the three points $\frac{h}{4}$, $\frac{h}{2}$, and $\frac{3h}{4}$ occurs at (i) $\frac{h}{4}$ (top); (ii) $\frac{h}{2}$ (middle); and (iii) $\frac{3h}{4}$ (bottom).

loop is $O(\log n)$. Each time in the loop we query f, g at most $O(1)$ times and rest of the operations take $O(1)$ time. Therefore the total time and query complexity is $O(\log n)$. □

Building a Fence(Fence)

This subroutine builds a fence like discussed in the overview, when given two row indices i, j of the rectangle along with the discrete convex functions f, g. In the scenario when it is not possible to construct a fence it outputs Invalid. This happens when minimum on row j is lesser than the minimum on row i. Find the algorithm in Algorithm 7. The Lemma below formally states the guarantees on the output of the algorithm.

Algorithm 7: FENCE(f, g, R, i, j)

For any $k \in [\min_2(R), \max_2(R)]_{\mathbb{Z}}$, let $x^{(k)} = $ 1D-MINIMIZER$(f_{x_2=k}, g_{x_2=k}, R_{x_2=k})$;
if $x^{(i)} \leq_{(f,g)} x^{(j)}$ **then**
    if $i < j$ **then**
        Return $\text{cone}_{x^{(i)}}(x^{(j)} - e_1, x^{(j)} + e_1) \cap R_{>j}$;
    else
        Return $\text{cone}_{x^{(i)}}(x^{(j)} - e_1, x^{(j)} + e_1) \cap R_{<j}$;
else
    Return Invalid;

Lemma 5.16. *Let $f, g : \mathbb{Z}^2 \to \mathbb{R}$ be discrete convex functions, $R \subset \mathbb{Z}^2$ be a rectangle and $i, j \in [\min_2(R), \max_2(R)]_{\mathbb{Z}}$. If $x^{(i)} \leq_{(f,g)} x^{(j)}$ and $i < j$ for any point $y \in R_{x_2>j} \setminus$*

$\textsc{Fence}(f, g, R, i, j)$, $x^{(i)} \leq_{(f,g)} y$. *Similarly, if* $x^{(i)} \leq_{(f,g)} x^{(j)}$ *and* $i > j$ *for any point* $y \in R_{x_2<j} \setminus \textsc{Fence}(f, g, R, i, j)$, $x^{(i)} \leq_{(f,g)} y$. *The algorithm has time and query complexity* $O(\log h_1)$, *where* $h_1 = \text{length}_1(R)$.

Proof. We prove the Lemma for the case when $x^{(i)} \leq_{(f,g)} x^{(j)}$ and $i < j$, and it is similar for the other case.

Consider the line line$(x^{(i)}, y)$ and let it intersect the row j at y'. Note that y' need not have integer coordinates. As $y \notin \text{Fence}(f, g, R, i, j)$, we get $\left|y'_1 - x_1^{(j)}\right| > 1$. Assume $y'_1 - x_1^{(j)} > 1$. From Proposition 5.4, we have $x^{(j)} \leq_{(\widetilde{f},\widetilde{g})} y'$, as $x^{(j)} \leq_{(\widetilde{f},\widetilde{g})} x^{(j)} + e_1$ and $x^{(j)}, x^{(j)} + e_1, y'$ lie on a line. Using transitivity we get $x^{(i)} \leq_{\widetilde{f},\widetilde{g}} y'$, as $x^{(i)} \leq_{(\widetilde{f},\widetilde{g})} x^{(j)}$ and $x^{(j)} \leq_{(\widetilde{f},\widetilde{g})} y'$. Combining this again with Proposition 5.4 we get $x^{(i)} \leq_{\widetilde{f},\widetilde{g}} y$, as $x^{(j)}, y', y$ lie on a line.

The major computation in the algorithm is finding $x^{(i)}$ and $x^{(j)}$ which takes time and queries $O(\log h_1)$. Rest of the algorithm only takes $O(1)$ queries and time. The output set can be represented using $O(1)$ linear inequalities. Therefore that the total time and query complexity of the algorithm is $O(\log h_1)$. □

Finding Concentrating Basis in Two Dimensions (2D-FindConcBasis)

This subroutine takes in as input, $y \in \mathbb{Z}^2, h \in \mathbb{Z}$, and outputs a concentrating basis in which all the points "close" to line(y) lie in a "small" grid whose dimensions are a function of h.

Algorithm 8: 2D-$\textsc{FindConcBasis}(y, h)$

1 Let z be the normal vector of the line line(y);
2 $(a_1, a_2) \leftarrow$ co-prime integer values that satisfy the linear inequalities

$$\left|\frac{z_2}{z_1} - \frac{a_2}{a_1}\right| \leq \frac{2}{a_1\sqrt{h}} \text{ and}$$
$$|a_1| \leq \sqrt{h}$$

3 Let $b, b' \in \mathbb{Z}^2$ be such that $< a, b >= 0$, $< a, b' >= 1$, and $|b_2| \geq |b'_2|$;
4 Return (b, b');

We next show that there exists $a_1, a_2 \in \mathbb{Z}$ and $b, b' \in \mathbb{Z}^2$ which satisfy the properties claimed in the 2D-FindConcBasis algorithm, and that they can be computed efficiently. We defer the proof that the basis, $(b\, b')$, output by the algorithm is concentrating to Lemma 5.19 in Section 5.3.3.

Lemma 5.17. *Let $y \in \mathbb{R}^2$ and $h \in \mathbb{Z}$. Let $z \in \mathbb{R}^2$ be the normal of the line* line(y)*. Then in time $O(\text{poly}(\log h))$ we can find $a, b, b' \in \mathbb{Z}^2$ that satisfy the following conditions in the algorithm.*

1. $\left|\frac{z_2}{z_1} - \frac{a_2}{a_1}\right| \leq \frac{2}{a_1\sqrt{h}}$, $|a_1| \leq \sqrt{h}$ *and* a_1, a_2 *are co-prime.*
2. $< a, b >= 0$, $< a, b' >= 1$, *and* $|b_2| \geq |b_2'|$.

Proof. We need to prove the following: first that there exist $a_1, a_2 \in \mathbb{Z}$ and $b, b' \in \mathbb{Z}^2$ that satisfy the conditions in the algorithm. Second that they can be found in time $O(\text{poly}(\log h))$.

We first show that there exist $a_1, a_2 \in \mathbb{Z}$ that satisfy the first condition in the Lemma. Apply Dirichlet's theorem with $N = \sqrt{h}$, $\alpha_1 = \frac{z_2}{z_1}$, and let $a_1 = q$, $a_2 = p_1$. From the theorem we get that $\left|\frac{a_2}{a_1} - \frac{z_2}{z_1}\right| \leq \frac{1}{a_1\sqrt{h}}$ and $|a_1| \leq \sqrt{h}$. If a_1, a_2 are not co-prime then divide by their gcd to make them co-prime.

We next move on to show the existence of b, b' that satisfy the second condition. Choose $b = (-a_2, a_1)$. This satisfies $\langle a, b\rangle = 0$. Pick b' to be any vector so that $\langle a, b'\rangle = 1$, there exists such a b' as a_1, a_2 are co-prime. If $|b_2| < |b_2'|$, then there always exists $k \in \mathbb{Z}$ such that $b' \leftarrow b' + kb$ has the desired property. Note that even for the new b' the property that $\langle a, b'\rangle = 1$ still holds.

The last step remaining is to show that a, b, b' can be found in time $O(\text{poly}(\log h))$. The bulk of the time is taken to find (a_1, a_2). Once we have (a_1, a_2) finding (b, b') can be done in $O(1)$ time as mentioned above. Finding (a_1, a_2) is a feasibility problem in integer programming with 4 constraints. From [22] an integer programming problem with constant number of constraints can be solved in $O(s)$ time where s is the length of the encoding. Without loss of generality let $z_1 > z_2$ and $z_1 = 1$. Let's approximate z_2 with z_2' which has $2\log h$ length encoding. Then the integer programming problem $\left|\frac{z_2'}{z_1} - \frac{a_2}{a_1}\right| \leq \frac{1}{a_1\sqrt{h}}$, $|a_1| \leq \sqrt{h}$ can be solved in $O(\log h)$ time as discussed above. Because of restricting the number of bits to represent z_2' to $2\log h$ we have $|z_2 - z_2'| \leq \frac{1}{h^2}$. Therefore the solution of the integer programming problem satisfies the inequality $\left|\frac{z_2}{z_1} - \frac{a_2}{a_1}\right| \leq \frac{1}{a_1\sqrt{h}} + \frac{1}{h^3}$ and as $|a_1| \leq \sqrt{h}$ we get $\left|\frac{z_2}{z_1} - \frac{a_2}{a_1}\right| \leq \frac{2}{a_1\sqrt{h}}$. □

5.3.3 Analysis of 2D-Minimizer

In this section we prove Lemma 5.14. We first present two technical lemmas which are useful in proving the correctness of Lemma 5.14.

Main Technical Lemmas

In the rectangle R', in the algorithm 2D-Minimizer, the point $(0,0)$ is preferred over any point that is horizontally far from both the lines $\text{line}(z^{\text{low}})$ and $\text{line}(z^{\text{high}})$. This implies it is okay to eliminate all the points in R' that are far from both the lines $\text{line}(z^{\text{low}})$ and $\text{line}(z^{\text{high}})$. This property is formally stated and proved in the lemma below.

Lemma 5.18. *Let $z^{\text{high}}, z^{\text{low}} \in \mathbb{Z}^2$, $R' \subset \mathbb{Z}^2$, $f', g' : \mathbb{Z}^2 \to \mathbb{R}$ be as defined in the* 2D-Minimizer *algorithm. Any $y' \in R'$ with $y'_2 > 0$ and $\text{dist}_1(y', \text{line}(z^{\text{high}})) > 8$ satisfies $(0,0) \leq_{(f',g')} y'$. Similarly, any $y' \in R'$ with $y'_2 < 0$ and $\text{dist}_1(y', \text{line}(z^{\text{low}})) > 8$ satisfies $(0,0) \leq_{(f',g')} y'$.*

Proof. We prove the claim for $y'_2 > 0$ and for $y'_2 < 0$ it is similar. For this proof let $x^{(k)} = \text{1D-Minimizer}(f'_{x_2=k}, g'_{x_2=k}, R'_{x_2=k})$ to factor in the translations made in line 10 of the algorithm.

Any point $y' \in R'$ with $y'_2 > z_2^{\text{high}}$ satisfies $(0,0) \leq_{(f',g')} y'$. For any y' with $y'_2 > z_2^{\text{high}}$ there exists $i, j \in V'$ with $i < j < y'_2$ and $\text{Fence}(f', g', R', i, j)$ valid such that $y' \notin \text{Fence}(f', g', R', i, j)$. Since $y' \notin \text{Fence}(f', g', R', i, j)$, from Lemma 5.16, we get $x^{(i)} \leq_{(f',g')} y'$. From how we picked the lines in V' in the algorithm $(0,0) \leq_{(f',g')} x^{(i)}$. Using transitivity $(0,0) \leq_{(f',g')} y'$.

We next show that for any point $y' \in \text{Fence}(f', g', R', i, j)$ with $0 < y'_2 \leq z_2^{\text{high}}$, $i, j \in V'$ and $y'_2 - j \leq 2(j - i)$ we have $\text{dist}_1(y', \text{line}(z^{\text{high}})) \leq 8$. Since the points $y', z^{\text{high}} \in \text{Fence}(f', g', R', i, j)$ and $y'_2 - j \leq j - i$ we have $\text{dist}_1(y', \text{line}(x^{(i)}, z^{\text{high}})) \leq 6$. Consider the lines $\text{line}(z^{\text{high}})$ and $\text{line}(x^{(i)}, z^{\text{high}})$. They satisfy $\text{dist}_1(x^{(j)}, \text{line}(z^{\text{high}})) \leq 1$ and $\text{dist}_1(x^{(j)}, \text{line}(x^{(i)}, z^{\text{high}})) \leq 1$. Therefore by triangle inequality the horizontal distance between the lines on the row j is at most 2. As both the lines intersect at z^{high} the horizontal distance between the lines is at most 2 on any row above j. Since $\text{dist}_1(y', \text{line}(x^{(i)}, z^{\text{high}})) \leq 6$ and $y'_2 > j$ we have $\text{dist}_1(y', \text{line}(z^{\text{high}})) \leq 8$.

The last step remaining is to show that for any $y' \in R'$ with $0 \leq y'_2 \leq z_2^{\text{high}}$ even when $\text{dist}_1(y', \text{line}(z^{\text{high}})) > 8$ we have $(0,0) \leq_{(f',g')} y'$. Let the snapshot of the variables when the region containing y' is processed be $t', c', b', h' \in V'$. Since $y'_2 > 0$, the region is upward

processed and either $x^{(c')} \leq_{(f',g')} \{x^{(t')}, x^{(b')}\}$ or $x^{(b')} \leq_{(f',g')} \{x^{(c')}, x^{(t')}\}$. We prove it for the case when $x^{(c')} \leq_{(f',g')} \{x^{(t')}, x^{(b')}\}$ and for the other case it is similar. Since $c', t' \in V'$, $y'_2 \leq z_2^{\text{high}}$ and $y'_2 - t' \leq t' - c'$ from what we showed above if $y' \in \text{Fence}(f', g', R', c', t')$, then $\text{dist}_1(y', \text{line}(z^{\text{high}})) \leq 8$. But $\text{dist}_1(y', \text{line}(z^{\text{high}})) > 8$, therefore $y' \notin \text{Fence}(f', g', R', c', t')$. From Lemma 5.16 it implies $x^{(c')} \leq_{(f',g')} y'$ and since $(0,0) \leq_{(f',g')} x^{(c')}$ by transitivity we get $(0,0) \leq_{(f',g')} y'$. □

The basis output by 2D-FindConcBasis(y, h) has the property that all the integer points that are horizontally "close" to the line line(y) fit in a "small" rectangle by performing a change of basis with it. The lemma below formalizes this notion and proves it.

Lemma 5.19. *Let $y \in \mathbb{R}^2$, $h \in \mathbb{Z}$, and z be the normal to* line(y). *Let $B = (b\, b')$ be the output of the algorithm* 2D-FindConcBasis(y, h). *Then $B = (b\, b')$ is a basis of $\mathbb{Z}^2$ and for any $x' \in \mathbb{Z} \times [h]$ such that* $\text{dist}_1(x', \langle z, x\rangle = 0) \leq 8$ *we have $x'^B \in [-10h, 10h] \times [-9\sqrt{h}, 9\sqrt{h}]$.*

Proof. We first show b, b' form a basis for the integer lattice $\mathbb{Z}^2$. From Proposition 3.13, to show this it is enough to show the determinant of the matrix $B = (b\, b')$ is ± 1. The determinant of the matrix $B = (b, b')$ is $b_1 b'_2 - b'_2 b_1 = -a_2 b'_2 - a_1 b'_1 = -\langle a, b'\rangle = -1$.

To show that $x'^B \in [-10h, 10h] \times [-9\sqrt{h}, 9\sqrt{h}]$ we first show the second coordinate of x'^B lies in the range $[-9\sqrt{h}, 9\sqrt{h}]$. Let (α_1, α_2) be the coordinates x'^B. From Proposition 5.10 and 2D-FindConcBasis algorithm, for any $x' \in \mathbb{Z} \times [h]$ such that $\langle z, x'\rangle = 0$ we have $\text{dist}_1(x', \langle a, x\rangle = 0) \leq \frac{\sqrt{h}}{|a_1|}$. Combining this with triangle inequality we get, for any point $x' \in \mathbb{Z} \times [h]$ with $\text{dist}_1(x', \langle z, x\rangle = 0) \leq 8$ has $\text{dist}_1(x', \langle a, x\rangle = 0) \leq \frac{\sqrt{h}}{|a_1|} + 8$. From Proposition 5.9 we get $|\langle a, x'\rangle| \leq \sqrt{h} + 8\,|a_1| \leq 9\sqrt{h}$. The last inequality is because $|a_1| \leq \sqrt{h}$. Since $x' = \alpha_1 b + \alpha_2 b'$, $\langle a, b\rangle = 0$ and $\langle a, b'\rangle = 1$ we also have $\langle a, x'\rangle = \alpha_2$. Therefore $|\alpha_2| \leq 9\sqrt{h}$.

We complete the proof by bounding $|\alpha_1|$. We have $x'_2 = \alpha_1 b_2 + \alpha_2 b'_2$. Taking absolute value on both sides $|x'_2| = |\alpha_1 b_2 + \alpha_2 b'_2| \geq |\alpha_1 b_2| + |\alpha_2 b'_2|$. Rearranging the terms we get $|\alpha_1|\,|b_2| \leq |x'_2|\,|b_2| + |\alpha_2|\,|b'_2| \leq h\,|b_2| + |\alpha_2|\,|b_2|$. The last inequality is because $|x'_2| \leq h$ and $|b_2| \geq |b'_2|$. Therefore $|\alpha_1| \leq h + |\alpha_2| \leq h + 9\sqrt{h} \leq 10h$. □

Correctness

To prove the correctness of the lemma we need to show two things. One, for every $y' \in R$ we eliminate there is a $y \in R$ that is not eliminated and $y \leq_{(f,g)} y'$. Two, the algorithm

terminates. The fact that the algorithm terminates is implied from upper bound on the time complexity which we prove in the next section.

For every $y' \in R$ we eliminate there is a $y \in R$ that is not eliminated and $y \leq_{(f,g)} y'$ is implied by the following two statements. The first is, any point $y' \in R$ that is at a horizontal distance more than 8 from both the lines $\text{line}(z^{\text{high}})$, $\text{line}(z^{\text{low}})$ satisfies $(0,0) \leq_{(f,g)} y'$. The second is, all the points that are horizontally at a distance atmost 8 form the lines $\text{line}(z^{\text{high}})$ or $\text{line}(z^{\text{low}})$ are included in our new sub-problems. The second statement follows Lemma 5.19 and the fact that we call 2D-FindConcBasis with (z^{low}, h_2) and (z^{high}, h_2). The first statement was proved in Lemma 5.18.

Time and Query Complexity Analysis

The time and query complexity of the algorithm can be split into two parts: The first is the time and queries taken by the two sub-problems and the second is the time and queries taken by rest of the operations in the algorithm.

We first bound the time and queries taken by rest of the operations in the algorithm:

If $h_2 \leq 100$ we find the minimum in the rectangle by brute-force which takes $O(h_2 \log h_1)$ time and queries. When $h_2 \geq 100$, the time and query complexity is contributed by the following:

1. Finding the minimum for $\log h_2$ rows, which takes $O(\log h_1 \log h_2)$ time and queries.
2. Finding the points $z_{\text{low}}, z_{\text{high}}$, which takes $O(\text{poly}(\log h))$ time. We can find $z_{\text{low}}, z_{\text{high}}$ by solving an integer programming problem with at most $O((\log h_2)^2)$ constraints. Each constraint is a linear inequality with binary representation at most $O(\log h)$ as they are lines passing through two points in the grid. From Proposition 5.12 this system of inequalities can be solved in time $O(\text{poly}(\log h))$.
3. Finding the basis for $z_{\text{low}}, z_{\text{high}}$ takes $O(\text{poly}(\log h_2))$ time as proved in Lemma 5.17.

Hence the total time and queries taken other than the sub-problems is $O(\text{poly}(\log h))$.

Factoring in the two sub-problems, the time and query complexity T of the algorithm satisfies the relation

$$T(h_1, h_2) = 2T(20h_2, 18\sqrt{h_2}) + \text{poly}(\log h).$$

Solving the above recursive relation we get the total time and query complexity to be $O(\text{poly}(\log h))$.

5.3.4 Proof of Theorem 1.9

To prove Theorem 1.9 we use 2D-Minimizer, but not directly. We manipulate the feasibility constraint function we give as input to the 2D-Minimizer so that the rectangle constraint is also included in the feasibility constraint. We need this manipulation as running the 2D-Minimizer directly does not guarantee that the output is from the rectangle even when there are feasible points in the rectangle. This manipulation ensures that the algorithm outputs a point outside the rectangle only when the original rectangle does not have any feasible points.

The proof uses a function $g^{[n]^2} : \mathbb{Z}^2$ which we define first. Let $R = [n]^2$. The function value of $g^{[n]^2}$ at a point, $x \in \mathbb{Z}^2$, is the distance of the point x from the set $\widetilde{R}$. Since the set $\widetilde{R}$ is convex, the function that represents the distance from the set is convex as well. The function value $g^{[n]^2}(x)$ can be computed efficiently.

We next describe how to solve the optimization problem in the Theorem 1.9 using the 2D- Minimizer algorithm. First change the feasibility constraint function from g to $g' = \max\{g, g^{[n]^2}\}$. Since $g, g^{[n]^2}$ are convex, g' is convex as well. Next, run the 2D-Minimizer algorithm with $(f, g', [n]^2)$ as the input. Let $x' = \text{2D-Minimizer}(f, g', [n]^2)$. If $g'(x') \leq 0$, then from Lemma 5.14 we get that for every $y \in [n]^2$ either $g(y) > 0$ or $f(x') \leq f(y)$. Therefore x' is the solution of the optimization problem. If $g'(x') > 0$, then from Lemma 5.14 we get that for any $y \in [n]^2$, $g'(x') \leq g'(y)$. This combined with the fact that $g^{[n]^2}(y) \leq 0$, gives $g(y) > 0$. This shows that the initial rectangle $[n]^2$ does not have any feasible points.

The time and query complexity is $O(\text{poly}(\log n))$ from the time and query complexity of the 2D-Minimizer algorithm.

5.4 High-Dimensional Minimization

In Section 5.4.1, we present the high dimensional minimization algorithm HD-Minimizer. The algorithm by itself is quite short and simple. Most of the heavy lifting is done by the subroutines in the algorithm which we present in Section 5.4.2. In Section 5.4.3, we show the correctness of the algorithm and prove bounds on its time and query complexity. Like the 2D-Minimizer, the HD-Minimizer algorithm does not directly solve the problem of minimizing discrete convex functions over the hypergrid $[n]^d$. We need to manipulate the constraint feasibility function given as input. We discuss more on this and prove Theorem 1.10 in Section 5.4.4.

5.4.1 Algorithm(HD-Minimizer)

The algorithm HD-Minimizer takes in as input a parameter η, discrete convex functions $f, g : \mathbb{Z}^d \to \mathbb{R}$ and a rectangle R, such that $\text{length}_i(R) \geq \text{length}_{i'}(R)$ for $i \geq i'$, and outputs a point $y \in \mathbb{Z}^d$ such that for any $x \in R$ we have $y \leq_{(f,g)} x$. The formal lemma is stated below and the algorithm can be found in Algorithm 9.

Lemma 5.20. *Let $f, g : \mathbb{Z}^d \to \mathbb{R}$ be discrete convex functions, $R \subset \mathbb{Z}^d$ be a rectangle with $\text{length}_i(R) \geq \text{length}_{i'}(R)$ for $i \geq i'$ and $\eta = 2^{d^2 \log_{\frac{d}{d-1}} \log h}$, where $d << h = \max_{i \in [d]} \text{length}_i(R)$. Then the algorithm HD-Minimizer(f, g, R, η) outputs $y \in \mathbb{Z}^d$ such that for any $x \in R$ we have $y \leq_{(f,g)} x$, and the time and query complexity of the algorithm is $2^{O(\text{poly}(\log \log h))}$ for a any constant d.*

Description

In lines 1-2, we check if the length of the rectangle in dimension d is less than η. If that is the case we find the minimum in the rectangle by solving $\text{length}_d(R)$ number of sub-problems in dimension $d - 1$.

In line 3, using the MinRegion algorithm we find the regions, slabs, where the minimum could possibly lie. In lines 4-12, we cast all the slabs we found in line 3 into smaller sub-problems by performing a change of basis and then solve them, compare the minimums and output the minimum of the minimums.

Algorithm 9: HD-MINIMIZER(f, g, R, η).

1 **if** $\text{length}_d(R) \leq \eta$ **then**
2 Return $\min^{\leq_{(f,g)}}_{i \in [\min_d(R), \max_d(R)]_{\mathbb{Z}}}$ HD-MINIMIZER$(f_{x_d=i}, g_{x_d=i}, R_{x_d=i})$;
3 Let $(y', \mathcal{M}) \leftarrow$ MINREGION(f, g, R);
4 $x_{\min} \leftarrow o$;
5 **for** $M \in \mathcal{M}$ **do**
6 Let $x' \in M_1 \cap R$;
  // x' is on the hyperplane corresponding to the slab M
7 $R' \leftarrow R - x'$; $M' \leftarrow M - x'$;
8 $(B, R^*) \leftarrow$ FINDCONCBASIS(M', R');
9 $x \leftarrow$ HD-MINIMIZER$(f_{B^{-1},x'}, g'_{B^{-1},x'}, R^*, \eta)$;
10 **if** $B^{-1}x + x' \leq_{(f,g)} x_{\min}$ **then**
11 $x_{\min} \leftarrow B^{-1}x + x'$;
12 Return $x^{\min}$;

5.4.2 Subroutines

The algorithm HD-MINIMIZER uses two subroutines, MINREGION and HD-FINDCONCBASIS, which we present in this section.

Region of Fractional Minimum(MinRegion)

The MINREGION algorithm takes in as input discrete convex functions $f, g : \mathbb{Z}^d \to R$ and a rectangle $R \subset \mathbb{Z}^d$ and outputs a point $y \in R$ and a set of slabs $\mathcal{M}$ such that any point in $x \in \widetilde{R}$ with $x \leq_{(\tilde{f},\tilde{g})} y$ should lie in one of the slabs in $\mathcal{M}$. Find a formal statement in the lemma below. The algorithm is presented in Algorithm 10.

Lemma 5.21. *Given discrete convex functions $f, g : \mathbb{Z}^d \to \mathbb{R}$ and a rectangle $R \subset \mathbb{Z}^d$. The algorithm* MINREGION(f, g, R) *outputs a point, $y \in R$, and a set of slabs, $\mathcal{M} \subset \mathbb{R}^d$, such that for any $y' \in \widetilde{R} \setminus \mathcal{M}$, $y \leq_{(\tilde{f},\tilde{g})} y'$. The time and query complexity of the algorithm is $O(\log^d h)$, where $h = \max_{i \in [d]} \text{length}_i(R)$.*

Description of the algorithm

In lines 1-3, we handle the case when dimension is one.

In lines 4-20, we construct slabs using the slabs in $d-1$ dimensions. We choose the hyperplanes we use to construct slabs similar to how we chose the rows to build fences in two dimensions. We pick the hyperplanes $R_{x_d=t}, R_{x_d=c}, R_{x_d=b}$, where $t = \frac{3h'}{4}$, $c = \frac{h'}{2}, b = \frac{h'}{4}$ and h' is the length of the rectangle in dimension d, and run the MinRegion algorithm on these $d-1$ dimensional problems. Let $(x^{(b)}, \mathcal{M}^{(b)})$, $(x^{(c)}, \mathcal{M}^{(c)})$, $(x^{(t)}, \mathcal{M}^{(t)})$ be the respective outputs. We compare the points $x^{(t)}, x^{(c)}$ and $x^{(b)}$ using the relation $\leq_{(f,g)}$, and pick the most preferred point and use that with the slabs in $\mathcal{M}^{(t)}, \mathcal{M}^{(c)}, \mathcal{M}^{(b)}$ to construct the slabs in dimension d.

In lines 8-10, we handle the case when $x^{(t)}$ is the most preferred. In that case we use $x^{(t)}$ and the slabs in $\mathcal{M}^{(c)}$ to construct slabs for the region $\{x \in \mathbb{R}^d : c - \frac{h'}{2} \leq x_d \leq c\}$. Similarly, in lines 11-13 we handle the case when $x^{(b)}$ is the most preferred and in that case we use $x^{(b)}$ and the slabs in $\mathcal{M}^{(c)}$ to construct slabs for the region $\{x \in \mathbb{R}^d : c \leq x_d \leq c + \frac{h'}{2}\}$. In lines 14-17 we handle the case when $x^{(c)}$ is the most preferred. This case is slightly different from the above two cases, as here we construct slabs for two regions. We use $x^{(c)}$ and the slabs in $\mathcal{M}^{(t)}, \mathcal{M}^{(b)}$ to construct slabs for the regions $\{x \in \mathbb{R}^d : c \leq x_d \leq c + \frac{h'}{2}\}$ and $\{x \in \mathbb{R}^d : c - \frac{h'}{2} \leq x_d \leq c\}$ respectively.

In Lines 10, 13, 17, we update c and h' so that c is the center of the remaining region and h' is the length in dimension d. Note that since in each iteration we are constructing slabs for half the region, the length gets halved each time, and the new center depends on which of the regions is handled.

In lines 9, 12, 15, 16, we add a slab in d dimensions corresponding to each of the slabs in $d-1$ dimensions. For the new slab we need to determine the hyperplane, thickenss direction, and thickness. For the hyperplane, we extend the hyperplane of the slab in $d-1$ dimensions with the help of the point, to a hyperplane in d dimensions. This extension is done by the affine operator. The direction of thickness of the slab remains the same as the slab in $d-1$ dimensions, and the thickness at most triples because of how we chose the hyperplanes at one-quarter, half, and three-quarter mark.

In line 19, we add a slab to cover the last remaining region.

Algorithm 10: MinRegion(f, g, R)

if $d = 1$ **then**
  Let $x' \leftarrow$ 1D-Minimizer(f, g, R);
  Return $(x', \text{slab}(x', e_1, 1))$;
Let $h' \leftarrow \text{length}_d(R)$; $c \leftarrow \frac{\min_d(R)+\max_d(R)}{2}$; $\mathcal{M} \leftarrow \emptyset$;
while $h' > 1$ **do**
  Let $t \leftarrow c + \frac{h'}{4}$; $b \leftarrow c - \frac{h'}{4}$;
  Let $(x^{(i)}, \mathcal{M}^{(i)}) \leftarrow$ MinRegion$(f_{x_d=i}, g_{x_d=i}, R_{x_d=i})$ for any $i \in [\min_d(R), \max_d(R)]_{\mathbb{Z}}$;
  // $x^{(i)}$ need not be the minimum in $R_{x_d=i}$
  if $x^{(t)} \leq_{(f,g)} \{x^{(c)}, x^{(b)}\}$ **then**
    $\mathcal{M} = \mathcal{M} \cup \{\text{slab}(\text{affine}(x^{(t)}, \{x \in \mathbb{R}^d : x_d = c, (x_1, \ldots, x_{d-1}) \in H\}), e_j, 3\text{width}) : \text{slab}(H, e_j, \text{width}) \in \mathcal{M}^{(c)}\}$;
    $c \leftarrow t$; $h' \leftarrow \frac{h'}{2}$;
  else if $x^{(b)} \leq_{(f,g)} x^{(c)}, x^{(t)}$ **then**
    $\mathcal{M} = \mathcal{M} \cup \{\text{slab}(\text{affine}(x^{(b)}, \{x \in \mathbb{R}^d : x_d = c, (x_1, \ldots, x_{d-1}) \in H\}), e_j, 3\text{width}) : \text{slab}(H, e_j, \text{width}) \in \mathcal{M}^{(c)}\}$;
    $c \leftarrow b$; $h' \leftarrow \frac{h'}{2}$;
  else
    $\mathcal{M} = \mathcal{M} \cup \{\text{slab}(\text{affine}(x^{(c)}, \{x \in \mathbb{R}^d : x_d = t, (x_1, \ldots, x_{d-1}) \in H\}), e_j, 2\text{width}) : \text{slab}(H, e_j, \text{width}) \in \mathcal{M}^{(t)}\}$;
    $\mathcal{M} = \mathcal{M} \cup \{\text{slab}(\text{affine}(x^{(c)}, \{x \in \mathbb{R}^d : x_d = b, (x_1, \ldots, x_{d-1}) \in H\}), e_j, 2\text{width}) : \text{slab}(H, e_j, \text{width}) \in \mathcal{M}^{(b)}\}$;
    $h' \leftarrow \frac{h'}{2}$;
$c^* \leftarrow c$;
$\mathcal{M} = \mathcal{M} \cup \text{slab}(x_d = c^*, e_d, 2)$;
Return $(x^{(c^*)}, \mathcal{M})$;

We prove the Lemma 5.21 towards the end of the section. We first prove a few properties about the set $\mathcal{M}$, which are useful in the proof of the lemma and the proofs in the next section.

Properties of $\mathcal{M}$

The proposition below gives an upper bound on the number of slabs, and the thickness of a slab in $\mathcal{M}$.

Proposition 5.22. *Let $f, g : \mathbb{Z}^d \to \mathbb{R}$ be discrete convex functions, $R \subset \mathbb{Z}^d$ be a rectangle, and $(x', \mathcal{M}) = \text{MinRegion}(f, g, R)$. Then $|\mathcal{M}| \leq (3 \log h)^d$, where $h = \max_{i \in [d]} \text{length}_i(R)$, and for any slab $\text{slab}(H, e_i, w) \in \mathcal{M}$ we have the width $w \leq 3^d$.*

Proof. We prove the proposition by induction on the dimension.

Base case: For dimension 1 the proposition is trivially true as $|\mathcal{M}| = 1, w = 1$.

Induction step: Let the proposition be true till dimension $k - 1$.

If w' is the width in dimension $k - 1$ then $w \leq 3w'$ is the width in dimension k. From the induction hypothesis $w' \leq 3^{k-1}$. Therefore $w \leq 3^k$.

We next bound the size of $\mathcal{M}$ in dimension k. Slabs are added to the set $\mathcal{M}$ in lines 9, 12, 15, 16, 19. In line 19 there is only one slab added. In rest of the lines the number of slabs added is at most $2 \log(\text{length}_k(R)) \left|\mathcal{M}^{(l)}\right|$ where $l \in [\min_k(R), \max_k(R)]_{\mathbb{Z}}$. From induction hypothesis $\left|\mathcal{M}^{(l)}\right| \leq (3 \log h')^{k-1}$ where $h' = \max_{i \in [k-1]} \text{length}_i(R)$. Since $h' \leq h$, we get $|\mathcal{M}| \leq (3 \log h)^k$. □

The next proposition is useful in proving the correctness of the HD-FindConcBasis subroutine in the next section.

Proposition 5.23. *Let $i \in [d]$, $c \in \mathbb{R}$, $z \in \mathbb{R}^d$ and $H = \{x \in \mathbb{R}^d : \langle z, x \rangle = c\}$. If $\text{slab}(H, e_i, \textit{width}) \in \mathcal{M}$, then $z_i \neq 0$.*

Proof. We prove the proposition by induction on the dimension.

Base case: For $d = 1$, there is exactly one slab in $\mathcal{M}$. The thickness direction is e_1, and the hyperplane defining the slab is $x_1 = x'$. Therefore the coefficient corresponding to dimension 1 is non-zero as desired.

Induction step: Assume the proposition is true till dimension $k-1$. Let $\text{slab}(H, e_i, w) \in \mathcal{M}$ be a slab in dimension k. We split the proof in two cases: $i = k$ and $i \in [k-1]$.

Case 1 $i = k$**:** There is exactly one slab with $i = k$. For that slab the hyperplane defining the slab is $x_k = c^*$. The coefficient corresponding to dimension k is non-zero as desired.

Case 2 $i = [k-1]$**:** The slabs with $i \in [k-1]$ are generated using slabs in one dimension less. Let the slab used to generate $\text{slab}(H, e_i, w)$ be $\text{slab}(H', e_i, w')$. Let $z' \in \mathbb{R}^{k-1}$, $c' \in \mathbb{R}$, $H' = \{x \in \mathbb{R}^{k-1} : \langle z', x \rangle = c'\}$ and $z \in \mathbb{R}^k$, $c \in \mathbb{R}$, $H = \{x \in \mathbb{R}^k : \langle z, x \rangle = c\}$. From the algorithm $H = \text{affine}(x', \{x \in \mathbb{Z}^d : x_d = b, (x_1, \ldots, x_{d-1}) \in H'\})$, and from the induction hypothesis $z'_i \neq 0$. Combining these two we get $z_i \neq 0$. Otherwise, if $z_i = 0$, then for a

point $y \in \{x \in \mathbb{R}^d : x_d = b, (x_1, \ldots, x_{d-1}) \in H'\} \subset H$ when we change y_i it should still belong to $\{x \in \mathbb{R}^d : x_d = b, (x_1, \ldots, x_{d-1}) \in H'\} \subset H$. This is a contradiction as $z'_i \neq 0$. If we take a point in H' and change just the ith coordinate it should not be in the set H'.

□

Proof of Lemma 5.21

We prove the Lemma using induction on the dimension.

Base case d = 1:. Let $y' \in \widetilde{R} \setminus \mathcal{M}$. This means $|y' - x'| > 1$, where $x' =$ 1D-MINIMIZER(f, g, R). We show the proposition is true for the case when $y' - x' > 1$, the proof similar for the other case. From Lemma 5.15, $x' \leq_{(\widetilde{f},\widetilde{g})} x' + 1$ and as $x', x' + 1, y'$ lie on a line, from Proposition 5.4 we get $x' \leq_{(\widetilde{f},\widetilde{g})} y'$.

Induction step. Assuming the lemma is true till dimension $k-1$. Let $y' \in \widetilde{R} \setminus \mathcal{M}$ and t, c, b, h' be the snapshot of variables when the region containing y' is processed. We prove it for the case when $t \leq y'_k \leq t + \frac{h'}{4}$ and $x^{(c)} \leq_{(f,g)} x^{(t)}, x^{(b)}$ and it is similar in rest of the scenarios. Let the line line$(x^{(c)}, y')$ intersect the hyperplane $x_k = t$ at y''. Since $y' \notin \mathcal{M}$, the point $y'' \notin \mathcal{M}^{(t)}$. If $y'' \in \mathcal{M}^{(t)}$, then $\exists$slab$(H', e_i, w) \in \mathcal{M}^{(t)}$ such that $y'' \in$ slab(H', e_i, w). From how we defined $\mathcal{M}$, slab(affine$(x^{(c)}, \{x \in \mathbb{R}^d : x_d = t, (x_1, \ldots, x_{d-1}) \in H'\}), e_i, 2w) \in \mathcal{M}$, and as $x^{(c)}, y'', y'$ lie on a line and $y'' \in \{x \in \mathbb{Z}^d : x_d = t, (x_1, \ldots, x_{d-1}) \in H'\}$ we get $y' \in \mathcal{M}$, which is a contradiction. From the induction hypothesis $x^{(t)} \leq_{(\widetilde{f},\widetilde{g})} y''$, and as $x^{(c)} \leq_{(\widetilde{f},\widetilde{g})} x^{(t)}$, from transitivity we get $x^{(c)} \leq_{(\widetilde{f},\widetilde{g})} y''$. As $x^{(c)}, y'', y'$ lie on a line and $x^{(c)} \leq_{(\widetilde{f},\widetilde{g})} y''$, from Proposition 5.4, we get $x^{(c)} \leq_{(\widetilde{f},\widetilde{g})} y'$. Since c^* is the last center and from how we picked the centers we have $x^{(c^*)} \leq_{(\widetilde{f},\widetilde{g})} x^{(c)}$, and due to transitivity $x^{(c^*)} \leq_{(\widetilde{f},\widetilde{g})} y'$.

TIME AND QUERY COMPLEXITY ANALYSIS. The algorithm calls the MINREGION algorithm in one dimension less at most $\log h$ times. So the time and query complexity $M_{(d,n)} = \log n M_{(d-1,n)} + \alpha$, where α is a constant. Therefore the total time and query complexity is $O(\log^d h)$.

Finding Concentrating Basis in Higher Dimensions(HD-FindConcBasis)

The algorithm HD-FINDCONCBASIS takes in as input a slab, slab(H, e_j, w), and a rectangle R, and outputs a basis B and a "small" rectangle R^* such that, if we perform a change of basis with the basis B, then all the points $y \in$ slab(H, e_j, w) fit in the rectangle R^*.

Algorithm 11: HD-FindConcBasis(slab$(H, e_j, w), R$)

Let $h_i \leftarrow \text{length}_i(R)$; $h = \max_{i\in[d]} h_i$;
// We are promised that $h_i \leq h_{i'}$ for $i \leq i'$.
Let $H = \{x : \langle z, x\rangle = 0\}$; $\beta_i \leftarrow \frac{z_i}{z_j}$;
$(a_1, a_2, \ldots a_d) \leftarrow$ co-prime integer values that satisfy the linear inequalities

$$\forall i \left|\beta_i - \frac{a_i}{a_j}\right| \leq \frac{1}{a_j h^{\frac{1}{2d}}} \text{ and}$$
$$|a_j| \leq \sqrt{|h|}$$

Let $B = (b^{(1)}\, b^{(2)}\, \ldots\, b^{(d)})$ be such that $B_{[d-1]\times[d-1]}$ is in HNF and $\left|b_i^{(d)}\right| \leq \left|b_i^{(i)}\right|$ for $i \in [d]$. Also $\left\langle a, b^{(d)}\right\rangle = 1$, and $b^{(1)}, \ldots, b^{(d-1)}$ form a basis for $\{x \in \mathbb{Z}^d : \langle a, x\rangle = 0\}$;
Let R' be a rectangle with $\min_d(R') = -3wdh^{(1-\frac{1}{2d})}$, $\max_d(R') = 3wdh^{(1-\frac{1}{2d})}$ and $\min_i(R') = -w2^{2d}(h_i + h^{(1-\frac{1}{2d})})$, $\max_i(R') = w2^{2d}(h_i + h^{(1-\frac{1}{2d})})$ for $i \in [d-1]$;
Let $B' = (b^{(d)}\, b^{(1)}\, \ldots\, b^{(d-1)})$ and R^* be the corresponding rectangle;
// R^* has the lengths of its dimensions in increasing order
Return (B', R^*);

In the next lemma we show there exists a vector $a \in \mathbb{Z}^d$, and a basis $B \in \mathbb{Z}^{d\times d}$, that satisfy the constraints in the algorithm.

Lemma 5.24. *Let $R \subset \mathbb{Z}^d$ be a rectangle with $h_i = \text{length}_i(R)$, $j \in [d]$, $H = \{x \in \mathbb{R}^d : \langle z, x\rangle = 0)\}$ be a hyperplane. Then in time $O(\text{poly}(d \log h))$, $h = \max_{i\in[d]} h_i$, we can find $a, b^{(1)}, b^{(2)}, \ldots, b^{(d)} \in \mathbb{Z}^d$ that satisfy the following conditions in the algorithm:*

1. $\left|\frac{z_i}{z_j} - \frac{a_i}{a_j}\right| \leq \frac{1}{a_j h^{\frac{1}{2d}}}$, $0 < |a_j| \leq \sqrt{h}$, *and $a_1, a_2, \ldots, a_d$ are co-prime.*
2. *Let $B = (b^{(1)}\, b^{(2)}\, \ldots\, b^{(d)}) \in \mathbb{Z}^{d\times d}$ be such that $B_{[d-1]\times[d-1]}$ is in HNF and $\left|b_i^{(d)}\right| \leq \left|b_i^{(i)}\right|$ for $i \in [d]$. Also $\left\langle a, b^{(d)}\right\rangle = 1$ and $b^{(1)}, \ldots, b^{(d-1)}$ form a basis for $\{x \in \mathbb{Z}^d : \langle a, x\rangle = 0\}$.*
3. *B is a basis of the integer lattice $\mathbb{Z}^d$.*

Proof. We need to prove the following: First that there exist $a \in \mathbb{Z}^d$ and $b_1, b_2, \ldots, b_d \in \mathbb{Z}^d$ that satisfy the conditions in the lemma. Second that they can be found in time $O(\text{poly}(d \log h))$.

We start by showing that there exists $a \in \mathbb{Z}^d$ that satisfies the first condition in the Lemma. Apply Dirichlet's theorem with $N = \sqrt{h}$, $\alpha_i = \frac{z_i}{z_j}$, $a_j = q$, $a_i = p_i$ for $i \neq j$. From the theorem we get that $\forall i \left|\frac{a_i}{a_j} - \frac{z_i}{z_j}\right| \leq \frac{1}{a_j h^{\frac{1}{2d}}}$ and $0 < |a_j| \leq \sqrt{h}$. To apply Dirichlet's theorem we need to have $z_j \neq 0$, which follows from Proposition 5.23. If $a_1, a_2, \ldots, a_d$ are not co-prime divide them by their gcd to make them co-prime.

We next show there exists $b^{(1)}, b^{(2)}, \ldots, b^{(d)} \in \mathbb{Z}^d$ that satisfy the second condition in the lemma. Let $C = (c^{(1)}\, c^{(2)}\, \ldots\, c^{(d-1)}) \in \mathbb{Z}^{d \times d-1}$ be the basis of the lattice $\{x \in \mathbb{Z}^d : \langle a, x\rangle = 0\} \cap \mathbb{Z}^d$. The existence of C follows from Proposition 3.14. Let $U \in \mathbb{Z}^{d-1 \times d-1}$ be a unimodular matrix such that in $C' = (c'^{(1)}\, c'^{(2)}\, \ldots\, c'^{(d-1)}) = CU$ the submatrix $C'_{[d-1]\times[d-1]}$ is in HNF. The existance of such a U follows from Proposition 3.16. The vectors $c'^{(1)}, c'^{(2)}, \ldots, c'^{(d-1)}$ form a basis of the lattice $\{x \in \mathbb{Z}^d : \langle a, x\rangle = 0\} \cap \mathbb{Z}^d$ as U is unimodular. Let $b^{(i)} = c'^{(i)}$ for $i \in [d-1]$. Let $c'^{(d)}$ be such that $\left\langle a, c'^{(d)}\right\rangle = 1$. There exists such a $c'^{(d)}$ as the coordinates of a are co-prime. Let $\alpha \in \mathbb{Z}^{d-1}$ be such that $b^{(d)} = c'^{(d)} + \sum_{i=1}^{d-1} \alpha_i b^{(i)}$ satisfies $\left|b_i^{(d)}\right| \leq \left|b_i^{(i)}\right|$ for $i \in [d-1]$. The vectors $b^{(1)}, \ldots, b^{(d-1)}$ form a basis for $\{x \in \mathbb{Z}^d : \langle a, x\rangle = 0\}$ as $b^{(i)} = c'^{(i)}$ for $i \in [d-1]$. We have $\left\langle a, b^{(d)}\right\rangle = 1$ as $\forall i \in [d-1]$, $\left\langle a, b^{(i)}\right\rangle = 0$ and $\left\langle a, c'^{(d)}\right\rangle = 1$. The matrix $B_{[d-1]\times[d-1]}$ is in HNF as $C'_{[d-1]\times[d-1]}$ is in HNF.

The vectors $b^{(1)}, b^{(2)}, \ldots, b^{(d)}$ form a basis for the integer lattice $\mathbb{Z}^d$. We showed above that the vectors $b^{(1)}, b^{(2)}, \ldots, b^{(d-1)}$ form a basis for the lattice $\{x \in \mathbb{Z}^d : \langle a, x\rangle = 0\}$. Let $y \in \mathbb{Z}^d$ be any point and $\langle a, y\rangle = l$. Then we can express y as $y = lb^{(d)} + y'$, where $\langle a, y'\rangle = 0$. Since $y' \in \mathbb{Z}^d$ lies on the lattice $\{x \in \mathbb{Z}^d : \langle a, x\rangle = 0\}$ it can be represented as $y' = \sum_{i=1}^{d-1} \gamma_i b^{(i)}$ where $\gamma_i \in \mathbb{Z}$.

Time complexity analysis The time complexity in the algorithm is primarily contributed by Lines 4, 5.

Line 4: To bound the time taken to find a we need to first bound the sizes of binary representations of β_i. Assume $\forall i \in [d]$, $\frac{1}{h^d} \leq z_i \leq 1$. If this is not the case divide each z_i with $\max_{i\in[d]} z_i$ and round it up so that new $z_i \geq \frac{1}{h^d}$. Dividing with $\max_{i\in[d]} z_i$ does not change the hyperplane as the right hand side is zero and rounding up to $\frac{1}{h^d}$ is an insignificant change, as for $x \in R$, $x_i \leq h$. So we ignore the change in the right hand side due to it. Now as $\beta_i = \frac{z_i}{z_j}$, we get $\beta_i \leq h^d$. We round up β_i so that the decimal part is grater than $\frac{1}{h^d}$, and like before we ignore the impacts of the rounding as they are insignificant.

The vector $a \in \mathbb{Z}^d$ can be computed in time $O(\text{poly}(d \log h))$ from Proposition 5.12 as

it is obtained by solving d inequalities and the binary representation of the coefficients is $O(d \log h)$. We also get $a_i \leq \beta_i a_j + 1 \leq h^{d+1}$, for $i \in [d] \setminus j$.

Line 5: The time complexity of this line can be derived from the existence proof we presented above for $b^{(1)}, \ldots, b^{(d)}$. The matrix C' can be found in time $O(\text{poly}(\log \|a\|_\infty)) = O(\text{poly}(d \log h))$ from Propositions 3.14 and 3.16. The vector $c'^{(d)}$ can be computed in time $O(\text{poly}(\log \|a\|_\infty)) = O(\text{poly}(d \log h))$ from Proposition 5.11. Rest of the operations require only $O(d)$ time. Therefore the time complexity of this line is $O(\text{poly}(d \log h))$.

The time taken for each of the lines in $O(\text{poly}(d \log h))$. Therefore the total time complexity is $O(\text{poly}(d \log h))$. □

The next Lemma proves that the basis returned by the HD-FindConcBasis algorithm is concentrating.

Lemma 5.25. *Let $R \subset \mathbb{Z}^d$ be a rectangle with $h_i \geq h_{i'}$ for $i \geq i'$, where $h_i = \text{length}_i(R)$ and $h = \max_{i \in [d]} h_i$. Let $j \in [d]$, $z \in \mathbb{R}^d, c, w \in \mathbb{R}$ and $H = \{x : \langle z, x \rangle = 0\}$ be a hyperplane. Let $B = (b^{(1)}, b^{(2)}, \ldots, b^{(d)})$ be the basis that satisfies the constraints in the* HD-FindConcBasis *algorithm. Then for every $x' \in R \cap \text{slab}(H, e_j, w)$ we have $(x')^B \in R'$, where R' is a rectangle with $\min_d(R') = -3wdh^{(1-\frac{1}{2d})}$, $\max_d(R') = 3wdh^{(1-\frac{1}{2d})}$ and $\min_i(R') = -2^{2d}(h_i + wh^{(1-\frac{1}{2d})})$, $\max_i(R') = 2^{2d}(h_i + wh^{(1-\frac{1}{2d})})$ for $i \in [d-1]$.*

Proof. We first show that the dth coordinate of $(x')^B$ is in the desired range. Let $(x')^B = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ i.e. $x' = \sum_{i=1}^d \alpha_i b^{(i)}$. Take dot product with a on both sides. We get $\alpha_d = \langle a, x' \rangle$. From Proposition 5.10 and HD-FindConcBasis algorithm we get that the distance between the hyperplanes $\langle z, x \rangle = 0, \langle a, x \rangle = 0$ along the jth direction, within the rectangle R, is upper bound by $\frac{dh^{(1-\frac{1}{2d})}}{a_j}$. This combined with triangle inequality gives $\text{dist}_j(x', \langle a, x \rangle = 0) \leq w + \frac{dh^{(1-\frac{1}{2d})}}{a_j}$. By Proposition 5.9 this implies $\langle a, x' \rangle \leq dh^{(1-\frac{1}{2d})} + w\,|a_j|$. Substituting this in $\alpha_d = \langle a, x' \rangle$ we get $|\alpha_d| \leq dh^{(1-\frac{1}{2d})} + w\,|a_j| \leq 2wdh^{(1-\frac{1}{2d})}$ as $|a_j| \leq \sqrt{h}$.

We next bound the length of the rectangle in rest of the dimensions. The first coordinate in all the basis vectors except $b^{(1)}$, $b^{(d)}$ is zero, so we get $x'_1 = \alpha_d b_1^{(d)} + \alpha_1 b_1^{(1)}$. Taking absolute value on both sides $|x'_1| = \left|\alpha_d b_1^{(d)} + \alpha_1 b_1^{(1)}\right|$. This implies $\left|\alpha_1 b_1^{(1)}\right| \leq |x'_1| + \left|\alpha_d b_1^{(d)}\right| \leq h_1 \left|b_1^{(1)}\right| + |\alpha_d| \left|b_1^{(1)}\right|$. The last inequality follows from $x' \in R$ and $\left|b_1^{(d)}\right| \leq \left|b_1^{(1)}\right|$. Therefore $|\alpha_1| \leq h_1 + |\alpha_d| \leq h_1 + 3wdm^{(1-\frac{1}{2d})}$.

Similarly, we get $|\alpha_2| \leq h_2 + |\alpha_d| + |\alpha_1| \leq h_1 + h_2 + 2.3wdm^{(1-\frac{1}{2d})}$ and $|\alpha_3| \leq h_3 + |\alpha_d| + |\alpha_1| + |\alpha_2| \leq 2h_1 + h_2 + h_3 + 4.3wdm^{(1-\frac{1}{2d})}$. In general $|\alpha_i| \leq h_i + \alpha_d + \sum_{k=1}^{i-1} |\alpha_k|$

$\leq h_i + 2^i h_1 + 2^{i-1} h_2 + \ldots + 2h_{i-1} + 2^i 3wdm^{1-\frac{1}{2d}} \leq 2^{2d}(h_i + wm^{(1-\frac{1}{2d})})$. The last inequality is because $h_i \geq h_{i'}$ for $i \geq i'$. Therefore all the points we are interested in fit in a R' with $\min_d(R') = -3wdh^{(1-\frac{1}{2d})}$, $\max_d(R') = 3wdh^{(1-\frac{1}{2d})}$ and $\min_i(R') = -2^{2d}(h_i + wh^{(1-\frac{1}{2d})})$, $\max_i(R') = 2^{2d}(h_i + wh^{(1-\frac{1}{2d})})$ for $i \in [d-1]$. □

5.4.3 Analysis of HD-Minimizer

In this section we prove Lemma 5.20.

Correctness

To show that the algorithm is correct we need to show two things: One, for every $z' \in R$ we eliminate there is a $z \in R$ that is not eliminated with $z \leq_{(f,g)} z'$. Two, the algorithm terminates. The termination part is implied from the time complexity analysis. The first part follows from Lemma 5.21 and Lemma 5.25. In Lemma 5.21 we showed that for every point in $\widetilde{R} \setminus \mathcal{M}$ there is a better point in the set $\mathcal{M} \cap \mathbb{Z}^d$ and in Lemma 5.25 we showed that all the points in $\mathcal{M} \cap \mathbb{Z}^d$ are passed on correctly to the sub-problems.

Time and Query Complexity Analysis

The HD-Minimizer algorithm recursively calls itself on a smaller grid of the same dimension till the length of the rectangle in dimension d becomes small enough to brute-force.

The lemma below upper bounds the depth of the recursion tree.

Lemma 5.26. *Let $R \subset \mathbb{Z}^d$ be a rectangle with $h = \max_{i \in [d]} \text{length}_i(R)$. Then the depth of the recursion tree is at most $O(\log_{\frac{d}{d-1}} \log h)$, and for all rectangles in the tree, the largest length is at most $2^{O(d^2)}h$.*

Proof. We first show that the depth of the recursion tree is at most $O(\log_{\frac{d}{d-1}} \log h)$. We prove this by showing that as we go depth d in the recursion tree, the length in dimension d, which is the largest length, goes from h to $2^{O(d^2)}h^{1-\frac{1}{d}}$. Therefore in $O(\log_{\frac{d}{d-1}} \log h)$ depth the length of dimension d goes to $2^{d^2 \log_{\frac{d}{d-1}} \log h}$, at which point we brute-force.

At depth one, the length of the rectangle in the first dimension is $3^d h^{(1-\frac{1}{d})}$. This follows from Lemma 5.25, and Proposition 5.22. The length of new rectangle in dimension d is

$2^d(h_{d-1} + 3^d h_d^{1-\frac{1}{d}}) \leq 2^{3d}h$. At depth two in the recursion tree, the length of the rectangle in dimension two is at most $2^{5d}h^{1-\frac{1}{d}}$. At depth d, the length of the rectangle in the dth dimension is at most $2^{d^2}h^{1-\frac{1}{d}}$. If we keep repeating the process, at depth $O(\log_{\frac{d}{d-1}} \log h)$ the largest length of the rectangle is $O(2^{d^2 \log_{\frac{d}{d-1}} \log h})$.

We next show that the largest length of all the rectangles in the tree is at most $2^{O(d^2)}h$. We showed that at depth d the largest length of the rectangle is at most $2^{d^2}h^{1-\frac{1}{d}}$, which is less than h, as $d << h$. Therefore, the rectangle with the largest length has to be within depth d and as shown above the largest length increases by a factor of at most 2^{3d} as we go one depth lower. This gives that the upto depth d the largest length of the rectangle is at most $2^{O(d^2)}h$. □

Time and query complexity analysis

We prove it using induction on the dimension. Let the time and query complexity of HD-Minimizer$(f, g, R, 2^{d^2 \log_{\frac{d}{d-1}} \log h})$ be $2^{(\log_{\frac{d}{d-1}} \log h)^{O(d^2)}}$, where $h = \max_{i \in [d]} \text{length}_i(R)$. The hypothesis trivially holds for the one-dimensional minimizer.

We next show the time and query complexity for dimension d, assuming the hypothesis is true for dimension $d-1$. The depth of the recursion tree, as shown above, is $O(\log_{\frac{d}{d-1}} \log h)$. We also showed that the largest length of the rectangle for the nodes in the tree is at most $2^{O(d^2)}h$. Therefore, for the analysis, we use that to be the largest length. Each node has at most $|\mathcal{M}|$ children, which from Proposition 5.22 is less than $(3 \log 2^{d^2}h)^d$. Hence the total number of nodes is at most $(3 \log 2^{d^2}h)^{O(d \log_{\frac{d}{d-1}} \log h)}$. The time and query complexity at each node comes from the following:

1. Brute-forcing to find the minimum when the length in the largest dimension is $2^{d^2 \log_{\frac{d}{d-1}} \log h}$, which takes $2^{d^2 \log_{\frac{d}{d-1}} \log h} 2^{(\log_{\frac{d-1}{d-2}} \log 2^{d^2}h)^{O((d-1)^2)}}$ time and queries from the induction hypothesis.
2. Finding the MinRegion in line 3 which, as we showed in Lemma 5.21, takes $(\log 2^{d^2}h)^{O(d)}$ time and queries.
3. Finding the basis transformation for the elements in $\mathcal{M}$. It takes $|\mathcal{M}| \text{ poly}(d \log 2^{d^2}h) = (\log 2^{d^2}h)^{O(d)}$ time from Lemma 5.24.
4. Comparing the function values of the minimums, which takes $O(|\mathcal{M}|) = (\log 2^{d^2}h)^{O(d)}$ time and queries.

The first item dominates the rest, so the total time and complexity at each node is $2^{d^2 \log_{\frac{d}{d-1}} \log h} 2^{(\log_{\frac{d-1}{d-2}} \log 2^{d^2} h)^{O((d-1)^2)}}$. The number of nodes is $(3 \log 2^{d^2} h)^{O(d \log_{\frac{d}{d-1}} \log h)}$. Therefore the total time and query complexity is $(3 \log 2^{d^2} h)^{O(d \log_{\frac{d}{d-1}} \log h)} 2^{d^2 \log_{\frac{d}{d-1}} \log h} 2^{(\log_{\frac{d-1}{d-2}} \log 2^{d^2} h)^{O((d-1)^2)}}$, which is clearly less than $2^{(\log_{\frac{d}{d-1}} \log h)^{O(d^2)}}$ as $d << h$. When d is a constant, the time and query complexity can be written as $2^{O(\text{poly}(\log \log h))}$.

5.4.4 Proof of Theorem 1.10

The idea behind the proof of Theorem 1.10 is similar to that of Theorem 1.9. To prove Theorem 1.10 we use HD-Minimizer, but not directly. We manipulate the feasibility constraint function given as input to the HD-Minimizer so that the rectangle constraint is also included in the feasibility constraint. We need this manipulation as running the HD-Minimizer directly, does not guarantee that the output is from the rectangle, even when there are feasible points in the rectangle. This manipulation ensures that the algorithm outputs a point outside the rectangle only when the original rectangle does not have any feasible points.

The proof uses a function $g^{[n]^d} : \mathbb{Z}^d$ which we define first. The function value of $g^{[n]^d}$ at a point, $x \in \mathbb{Z}^d$, is the distance of the point x from the set $\widetilde{R}$, where $R = [n]^d$. Since the set $\widetilde{R}$ is convex the function that represents the distance from the set is also convex. The function value $g^{[n]^d}(x)$ can be computed efficiently.

We next describe how to solve the optimization problem in the theorem using the HD-Minimizer algorithm. We first change the feasibility constraint function from g to $g' = \max\{g, g^{[n]^d}\}$. Since, $g, g^{[n]^d}$ are convex, g' is convex as well. We next run the HD-Minimizer algorithm with $(f, g', [n]^d, 2^{d^2 \log_{\frac{d}{d-1}} \log n})$ as input and let x' be the output. If $g'(x') \leq 0$, then from Lemma 5.20 we get that for every $y \in [n]^d$ either $g(y) > 0$, or $f(x') \leq f(y)$. Therefore x' is the solution of the optimization problem. If $g'(x') > 0$, then from Lemma 5.20 we get that for any $y \in [n]^d$, $g'(x') \leq g'(y)$. This combined with the fact that $g^{[n]^d}(y) \leq 0$, gives $g(y) > 0$. This shows that the initial rectangle $[n]^d$ does not have any feasible points.

The time and query complexity is $2^{O(\text{poly}(\log \log n))}$ from the time and query complexity of the HD-Minimizer algorithm.

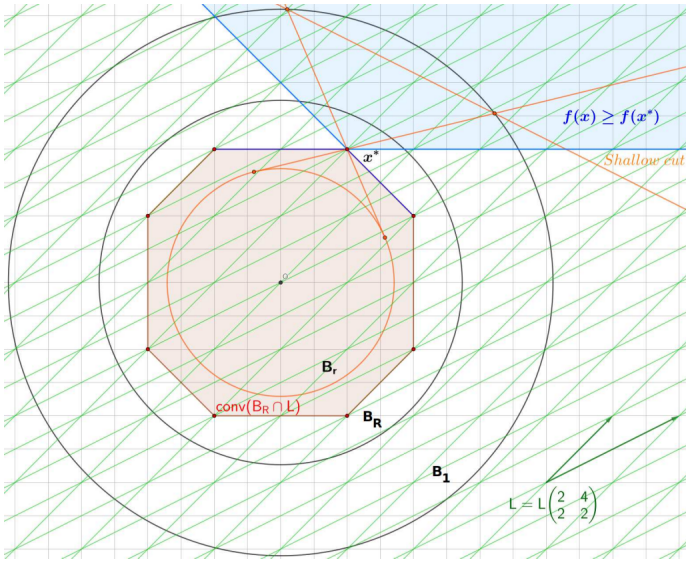

Figure 5.6: The above picture, borrowed from [60], shows how to construct a shallow cut to the ellipsoid. For the ease of presentation they transform the ellipsoid to a ball of radius one.

5.5 Technique of Veselov et al.

In [60] Veselov et al. give an algorithm that solves the problem of minimizing discrete convex functions over $B(n) \cap \mathbb{Z}^d$, where $B(n)$ is a ball of radius n. The algorithm maintains an ellipsoid that contains all the candidate points that could be the minimum. At every step it finds a cut to the ellipsoid that eliminates a good fraction of the points in the ellpisoid without eliminating the minimum. It then finds a smaller ellipsoid that contains all of the remaining points and keeps repeating the same process till the ellipsoid is small enough that all the integer points in the ellipsoid lie only on a "few" hyperplanes. Then it recursively solves the sub-problems, on these hyperplanes, that are in one dimension less.

The challenging part in the above algorithm is finding the cut that satisfies the conditions. We refer to this cut as the *shallow cut*. The key contribution of [60] is to show that a shallow cut to the ellipsoid can be found efficiently even when given access to just the comparison oracle on the integer points. For their algorithm, and our explanation, we assume the set of points is always contained in a ball instead of an ellipsoid as we can always transform an ellipsoid to a ball.

They use two main ideas to find the shallow cut to the ellipsoid. The first is the

observation that if we consider a ball of radius R, $B(R)$, and a ball of radius $r < R$, $B(r)$, that is contained in the convex hull of the integer points in $B(R)$, then we can eliminate all the points in the conic region we get by the tangents to $B(r)$ passing through the maximum integer point in $B(R)$. Refer to Figure 5.6 for an illustration. If we choose the right values for R and r, then from the conic region we get a shallow cut to the ellipsoid such that the eliminated points do not contain the minimum. Note that the maximum point x^* can not lie in $B(r)$, because $B(r)$ is completely contained in the convex hull of integer points in $B(R)$, and for a convex function the maximum of a set of points can not lie in the interior of the convex hull of those points.

The above idea by itself does not work, because to get a shallow cut that eliminates a desired fraction of the ellipsoid, we need the value of R to be cn, for some constant c, and in that case finding the maximum of all the integer points in the ball takes $2^{O(n)}$ queries. To get around this, instead of finding the maximum of all the integer points in $B(R)$, they find the maximum of all the lattice points of a sparsified lattice in $B(R)$. This is the second main idea. They sparsify it in such a way that the number of lattice points in $B(R)$ is not too many. They also show that the ball $B(r)$ is still contained in the convex hull of the sparsified lattice points in $B(R)$.

Note that the domain of the discrete convex function they consider is the set of integer points in the ball of radius n. This does not fundamentally change the problem as any algorithm that minimizes over the ball can minimize over the hypergrid $[n]^d$, by considering a ball of radius $\sqrt{d}n$ and adding the rectangle as an additional constraint. Similarly, any algorithm that minimizes discrete convex functions over $[n]^d$ can also minimize over the ball of radius n by just adding the ball as an additional constraint.

Chapter 6

Open Problems

6.1 Testing Discrete Convex Functions

Our results suggest two main open problems.

Open Problem 1. *Is it possible to* $\Omega(1)$*-test convexity of functions* $f\colon [n] \times [n] \to \mathbb{R}$ *with* $\mathrm{poly}(\log(n))$ *queries?*

Parnas, Ron, and Rubinfeld [50] also raised the problem of determining the query complexity for testing convexity in $d \geq 2$, and the upper bound in Theorem 1.7 provides the first suggestion that the query complexity of the problem might be exponentially smaller than—and not just sublinear in—the domain size. As the lower bound in the same theorem shows, however, any algorithm that would provide a positive answer to this question would have to be adaptive.

We can also generalize Open Problem 1 to ask whether convexity testing of $f\colon [n]^d \to \mathbb{R}$ can be done with query complexity $\mathrm{poly}(\log(n))$ for every constant value of d. For high-dimensional settings, it is also natural to ask about the dependence on d.

Open Problem 2. *Must every* $\Omega(1)$*-tester for convexity of functions* $f\colon [n]^d \to \mathbb{R}$ *have query complexity* $2^{\Omega(d)}$*?*

Theorem 1.8 gives a positive answer to this question for non-adaptive algorithms, but it still allows for the possibility that there is a convexity tester with query complexity that is polynomial in d. It is also possible that the best query complexity of convexity testers is

subexponential in d, even if it is not polynomial in d. (C.f., for instance, the submodularity testing problem, where it is known that $2^{\tilde{O}(\sqrt{d})}$ queries suffice to test submodularity of functions $f : \{0,1\}^d \to \mathbb{R}$ [55]. It is possible that a similar bound holds for testing convexity as well.)

6.2 Testing Convex Sets

There are two main open problems suggested by our results that are worth discussing in more detail.

Efficient convexity testing without supporting hyperplane oracles. Is it possible to test convexity of bodies $K \subset \mathbb{R}^n$ with a poly(n) queries to the membership and random oracles of K even when we don't have access to a separation oracle for K? Note that when K is convex, then it is known that an *approximate* separation oracle can be simulated using poly(n) queries to membership oracle [27, 34], but this simulation argument does not extend to non-convex bodies.

An efficient membership-and-random-oracle tester for convexity might be obtained by examining the robustness of other characterizations of convexity. It might also be useful to explore the role of *expansion* of bodies in testing convexity. A body $K \subseteq \mathbb{R}^n$ is α-*expanding* if for any set $K' \subseteq K$, we have $\mathrm{Vol}_{n-1}(\partial K' \setminus \partial K) \geq \alpha \min(\mathrm{Vol}(K'), \mathrm{Vol}(K \setminus K'))$. Convex bodies are expanding, and the body we construct in the proofs of Theorems 1.1 and 1.2 is far from expanding, a fact that makes this particular body easy to distinguish from convex bodies using random walks. Meanwhile, we have not been able to rule out the possibility that even the convex hull test by itself could suffice to efficiently test convexity under the assumption that the input body is expanding.

Testing convexity of functions over continuous domains. There is a close connection between convexity of sets and convexity of functions over continuous domains. Namely, a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if its epigraph is a convex set in $\mathbb{R}^{n+1}$. The definitions of distance to convexity, however, make the problems of testing convex functions and testing convex sets quite different in general. Nonetheless, as Berman, Raskhodnikova, and Yaroslavtsev [10] pointed out, the two problems are closely connected when we consider the testing of convex functions under the ℓ_1 norm. It would be interesting to see if the techniques or results introduced here could yield any progress on the problem of testing convex functions with a polynomial number of queries. (See Problem 70 on `sublinear.info` and [10] for more details on this problem.)

References

[1] Michel Baes, Timm Oertel, Christian Wagner, and Robert Weismantel. Mirror-descent methods in mixed-integer convex optimization. In *Facets of Combinatorial Optimization: Festschrift for Martin Grötschel*, pages 101–131. Springer, 2013.

[2] Michel Baes, Alberto Del Pia, Yurii Nesterov, Shmuel Onn, and Robert Weismantel. Minimizing lipschitz-continuous strongly convex functions over integer points in polytopes. *Math. Program.*, 134(1):305–322, 2012.

[3] Keith Ball. An elementary introduction to modern convex geometry. In Silvio Levy, editor, *Flavors of geometry*, chapter 1, pages 1–58. MSRI Publications, 1997.

[4] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.

[5] Aleksandrs Belovs. Adaptive lower bound for testing monotonicity on the line. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018*, pages 31:1–31:10, 2018.

[6] Aleksandrs Belovs, Eric Blais, and Abhinav Bommireddi. Testing convexity of functions over finite domains. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2030–2045. SIAM, 2020.

[7] Omri Ben-Eliezer. Testing local properties of arrays. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, pages 11:1–11:20, 2019.

[8] Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. The power and limitations of uniform samples in testing properties of figures. *Algorithmica*, 81(3):1247–1266, 2019.

[9] Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Testing convexity of figures under the uniform distribution. *Random Struct. Algorithms*, 54(3):413–443, 2019.

[10] Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p testing. In *STOC*, pages 164–173, 2014.

[11] Eric Blais and Abhinav Bommireddi. On testing and robust characterizations of convexity. In *APPROX/RANDOM*, pages 18:1–18:15, 2020.

[12] Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. In *Proceedings of the 29th Conference on Computational Complexity (CCC)*, pages 309–320, 2014.

[13] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[14] Murray Bremner. *Lattice basis reduction*. CRC Press New York, 2011.

[15] Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and lipschitz testing over hypercubes and hypergrids. In *Symposium on Theory of Computing Conference (STOC '13)*, pages 419–428, 2013.

[16] Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014.

[17] Xi Chen, Adam Freilich, Rocco A. Servedio, and Timothy Sun. Sample-based high-dimensional convexity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 37:1–37:20, 2017.

[18] Daniel Dadush, Chris Peikert, and Santosh Vempala. Enumerative lattice algorithms in any norm via M-ellipsoid coverings. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 580–589, 2011.

[19] Daniel Nicolas Dadush. *Integer programming, lattice algorithms, and deterministic volume estimation*. Georgia Institute of Technology, 2012.

[20] Kashyap Dixit, Sofya Raskhodnikova, Abhradeep Thakurta, and Nithin Varma. Erasure-resilient property testing. *SIAM J. Comput.*, 47(2):295–329, 2018.

[21] Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM (JACM)*, 38(1):1–17, 1991.

[22] F. Eisenbrand and Sören Laue. A linear algorithm for integer programming in the plane. *Mathematical Programming*, 2005.

[23] Funda Ergün, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000.

[24] Eldar Fischer. On the strength of comparisons in property testing. *Inf. Comput.*, 189(1):107–116, 2004.

[25] Peter C Fishburn. Utility theory for decision making. Technical report, Research analysis corp McLean VA, 1970.

[26] Satoru Fujishige and Kazuo Murota. Notes on L/M-convex functions and the separation theorems. *Mathematical Programming*, 88(1):129–146, 2000.

[27] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[28] Peter M. Gruber and Jörg M. Wills, editors. *Handbook of Convex Geometry*. North-Holland, Amsterdam, 1993.

[29] Godfrey Harold Hardy and Edward Maitland Wright. *An introduction to the theory of numbers*. Oxford University Press, 1979.

[30] Ravi Kannan, László Lovász, and Miklós Simonovits. Random walks and an $O(n^5)$ volume algorithm for convex bodies. *Random Structures & Algorithms*, 11(1):1–50, 1997.

[31] Abhiruk Lahiri, Ilan Newman, and Nithin Varma. Parameterized convexity testing. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 174–181. SIAM, 2022.

[32] Jeffrey Larson, Sven Leyffer, Prashant Palkar, and Stefan M. Wild. A method for convex black-box integer global optimization. *J. Glob. Optim.*, 80(2):439–477, 2021.

[33] Jon Lee and Sven Leyffer. *Mixed Integer Nonlinear Programming*. Springer, 2011.

[34] Yin Tat Lee, Aaron Sidford, and Santosh S. Vempala. Efficient convex optimization with membership oracles. In *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018.*, pages 1292–1294, 2018.

[35] Hendrik W Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of operations research*, 8(4):538–548, 1983.

[36] László Lovász and Miklós Simonovits. The mixing rate of markov chains, an isoperimetric inequality, and computing the volume. In *Proceedings [1990] 31st annual symposium on foundations of computer science*, pages 346–354. IEEE, 1990.

[37] László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Struct. Algorithms*, 4(4):359–412, 1993.

[38] László Lovász and Santosh Vempala. Hit-and-run from a corner. *SIAM J. Comput.*, 35(4):985–1005, 2006.

[39] László Lovász and Santosh Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *J. Comput. Syst. Sci.*, 72(2):392–417, 2006.

[40] Zhi-Quan Luo and Wei Yu. An introduction to convex optimization for communications and signal processing. *IEEE Journal on selected areas in communications*, 24(8):1426–1438, 2006.

[41] Jiří Matoušek. *Lectures on discrete geometry*, volume 108. Springer, 2002.

[42] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[43] Kazuo Murota. *Discrete convex analysis*. SIAM, 2003.

[44] Kazuo Murota. On steepest descent algorithms for discrete convex functions. *SIAM Journal on Optimization*, 14(3):699–707, 2004.

[45] Kazuo Murota and Akihisa Tamura. New characterizations of M-convex functions and their applications to economic equilibrium models with indivisibilities. *Discrete Applied Mathematics*, 131(2):495–512, 2003.

[46] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[47] Hukukane Nikaidô and Kazuo Isoda. Note on non-cooperative convex games. *Pacific Journal of Mathematics*, 5(S1):807–815, 1955.

[48] Ramesh Krishnan S Pallavoor, Sofya Raskhodnikova, and Nithin Varma. Parameterized property testing of functions. *ACM Transactions on Computation Theory (TOCT)*, 9(4):17, 2018.

[49] Daniel P Palomar and Yonina C Eldar. *Convex optimization in signal processing and communications*. Cambridge university press, 2010.

[50] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. On testing convexity and submodularity. *SIAM J. Comput.*, 32(5):1158–1184, 2003.

[51] Luis Rademacher and Santosh Vempala. Testing geometric convexity. In *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science, 24th International Conference, Chennai, India, December 16-18, 2004, Proceedings*, pages 469–480, 2004.

[52] Sofya Raskhodnikova. Approximate testing of visual properties. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, pages 370–381. Springer, 2003.

[53] Dana Ron and Gilad Tsur. Testing properties of sparse images. *ACM Trans. Algorithms*, 10(4):17:1–17:52, 2014.

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

[55] C. Seshadhri and Jan Vondrák. Is submodularity testable? *Algorithmica*, 69(1):1–25, 2014.

[56] Lloyd S Shapley. Cores of convex games. *International journal of game theory*, 1(1):11–26, 1971.

[57] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning*. MIT Press, 2012.

[58] Santosh S Vempala. Recent progress and open problems in algorithmic convex geometry. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010.

[59] Maarten CM Vendrik and Geert B Woltjer. Happiness and loss aversion: Is utility concave or convex in relative income? *Journal of Public Economics*, 91(7-8):1423–1448, 2007.

[60] Sergey I Veselov, Dmitry V Gribanov, N Yu Zolotykh, and A Yu Chirkov. A polynomial algorithm for minimizing discrete convic functions in fixed dimension. *Discrete Applied Mathematics*, 283:11–19, 2020.

[61] Min Yan. Extension of convex function. *arXiv preprint arXiv:1207.0944*, 2012.