Computational Aspects of some Problems from Discrete Geometry in Higher Dimensions

Dissertation zur Erlangung des Doktorgrades

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

August 2012

von

Daniel Werner

Institut für Informatik

Freie Universität Berlin

Takustraße 9

14195 Berlin

daniel.werner@fu-berlin.de

gefördert durch die DFG im Rahmen des
Graduiertenkollegs “Methods for Discrete Structures”

Betreuer:

Prof. Dr. Günter Rote
Arbeitsgruppe Theoretische Informatik
Institut für Informatik
Fachbereich Mathematik und Informatik
Freie Universität Berlin
Takustraße 9
D-14195 Berlin
Tel.: +49 - 30 - 838 751 50

Gutachter:

Prof. Dr. Günter Rote
Arbeitsgruppe Theoretische Informatik
Institut für Informatik
Fachbereich Mathematik und Informatik
Freie Universität Berlin
Takustraße 9
D-14195 Berlin
Tel.: +49 - 30 - 838 751 50

Prof. Dr. Emo Welzl
Theory of Combinatorial Algorithms Group
Institute of Theoretical Computer Science
Department of Computer Science
ETH Zürich, CAB G39.2
CH-8092 Zürich
Switzerland
Tel.: +41 - 44 - 632 73 70

Vorlage zur Begutachtung: 03. August 2012
Datum der Disputation: 18. Januar 2013

Für Nina und Charlotte

Acknowledgments

First of all, I would like to thank my supervisor Günter Rote for his support and for giving me all the opportunities that a Ph.D. student can possibly hope for, and in particular for his careful proofreading of this thesis.

I am also very thankful to Christian Knauer for countless fruitful discussions and jokes, and to his successor Wolfgang Mulzer for our common work and all the things he taught me. I would like to thank the second referee Emo Welzl for agreeing to review my thesis, and him and János Pach for their invitation to the awesome Special Semester on Discrete and Computational Geometry in Lausanne, which brought about many new experiences and many new friends. Special thanks go to Nabil Mustafa for being probably the best (and surely the most enthusiastic) teacher I had during this time.

I am very grateful to the entire AG Theoretische Informatik for providing me with such a nice atmosphere to work in, most of all to my roommates Panos Giannopoulos and Rafel Jaume, and to Helmut Alt, Frank Hoffmann and Klaus Kriegel for our daily inspiring chats. I would also like to thank Sylvain Lazard and Sue Whitesides for inviting me to Barbados for an unforgettable seminar.

I am especially thankful to the RTG "Methods for Discrete Structures" for their generous scholarship—without it, this entire thesis would have been impossible to finish—and in particular to my BMS Mentor Günter M. Ziegler for our interesting chats and his caring advice. Moreover, I thank Elke Pose and Dorothea Kiefer for always being of great help to me in all questions around the scholarship.

Finally, and most of all, I would like to thank all my family and friends who supported me while writing this thesis, and who had to stand back more than once.

Daniel Werner, August 2012

Contents

x

Abstract

We will investigate computational aspects of several problems from discrete geometry in higher dimensions. In the plane, many of them are well understood and can be solved efficiently, but as the dimension increases, most of them seem to be considerably harder to solve. In this thesis, we make progress towards explaining this phenomenon by showing computational hardness for some of these problems. To this end, we also make use of parameterized complexity theory in order to show stronger relative lower bounds than those possible with classical complexity theory only. For one of the problems, we moreover develop several approximation algorithms. In the process, we pay particular attention to the exact dependence of the running time on the dimension.

We will use and develop different techniques for showing hardness of the problems in unbounded dimension. These include the technique of deconstructing the space into orthogonal planes, into which gadgets are placed. Using this technique, we are able to show a relative lower bound of $n^{\Omega(d)}$ for several problems related to testing the discrepancy of a point set and verifying ε-nets.

We then present a more natural reduction technique that reduces from the d-Sum problem to show relative lower bounds for many problems arising from theorems in combinatorial geometry. These include computing minimal Helly sets, certain decision versions of the ham-sandwich problem, and computing the Tverberg depth of a point set.

We then turn to computing a maximum size subset of points in convex position. While all the previous problems admit straightforward $n^{O(\text{poly}(d))}$ algorithms in d dimensions, here we are able to show that the problem already becomes hard in 3 dimensions. This shows a strong dichotomy between a low and a higher dimensional case, because in the plane the problem was known to be solvable in polynomial time.

As a positive result, we then consider the problem of computing a point of high Tverberg depth in d dimensions. We present a novel lifting approach that allows us to compute deep points for a point set in high dimension from deep points of its projection to some lower dimensional space. The approach is very generic, and we show how to combine it with other known methods in order to get even better algorithms.

Finally, we give a short outlook and suggest further open problems on the subject.

Zusammenfassung

In dieser Arbeit betrachten wir algorithmische Aspekte verschiedener Probleme der diskreten Geometrie in höheren Dimensionen. Während viele von ihnen in der Ebene intensiv untersucht und effizient lösbar sind, werden sie deutlich schwieriger, wenn sich die Dimension erhöht. In dieser Arbeit machen wir einen Versuch, dieses Phänomen zu erklären, indem wir zeigen, dass sie im komplexitätstheoretischen Sinne schwer sind. Dazu benutzen wir unter anderem parametrisierte Komplexitätstheorie, die es uns erlaubt, schärfere relative untere Schranken zu zeigen, als sie alleine mit klassischer Komplexitätstheorie möglich sind. Für eines solcher Probleme entwerfen wir außerdem verschiedene Approximationsalgorithmen. Besonderes Augenmerk liegt dabei auf der Untersuchung der exakten Abhängigkeit der Laufzeit von der Dimension.

Wir entwerfen und benutzen verschiedene Techniken, um die Schwerheit der Probleme in unbeschränkter Dimension zu zeigen. Ein Beispiel dafür ist die Zerlegung des d-dimensionalen Raumes in orthogonale Ebenen, in die wir bestimmte Teile einer Konstruktion platzieren können. Damit ist es uns möglich, relative untere Schranken von $n^{\Omega(d)}$ für die Laufzeit von Algorithmen für verschiedene Probleme zu zeigen. Diese beinhalten das Berechnen der Diskrepanz von Punktmengen oder die Verifikation von ε-Netzen.

Wir stellen anschließend eine etwas natürlichere Reduktionstechnik vor, die vom d-Sum Problem reduziert. Damit zeigen wir relative untere Schranken für einige Entscheidungsprobleme, die zu Sätzen der Diskreten Geometrie gehören. Dieses sind zum Beispiel das Bestimmen einer minimalen Hellymenge, bestimmte Entscheidungsvarianten des Ham-Sandwich Problems, und das Berechnen der Tverbergtiefe eines Punktes.

Anschließend wenden wir uns dem Problem zu, eine maximal große Untermenge in konvexer Lage zu finden. Während die bisherigen Probleme durch einfache Algorithmen in $n^{O(\mathrm{poly}(d))}$ Zeit lösbar sind, tritt hier das Phänomen auf, dass das Problem schon in 3 Dimensionen NP-schwer ist. Das zeigt einen starken Unterschied zwischen einem niedrig- und einem höherdimensionalen Problem, da es in der Ebene in polynomieller Zeit gelöst werden kann.

Für ein positives Ergebnis wenden wir uns dann dem Problem zu, einen Punkt mit hoher Tverbergtiefe in d Dimensionen zu berechnen. Wir entwerfen einen neuen Ansatz, der höherdimensionale aus niedrigdimensionalen Tverbergpunkten berechnet. Unser Algorithmus ist sehr generisch und lässt sich einfach mit bisherigen Algorithmen kombinieren, um noch bessere Resultate zu erhalten.

Am Ende geben wir noch einen Ausblick auf mögliche weitere Forschungsvorhaben.

Introduction

0.1 Motivation and Overview

In computational geometry, many problems are well understood in the plane. Most often, they can be solved by optimal algorithms that match the respective lower bounds.

However, as the dimension goes up, many of them seem to become considerably harder, and their complexity status becomes less clear. Often, the problems become NP-hard as the dimension increases—a phenomenon commonly referred to as "the curse of dimensionality". Still, for many of these problems it is not clear *how* hard they actually become once the dimension goes up. While there are many problems that can be solved in time $n^{O(d)}$ in d dimensions by simple algorithms, others become NP-hard already in 3 dimensions.

In any case, one has to cope with the problems once they turn out to be NP-hard. The main approach to this have been approximation algorithms, but recently, a different approach has been developed: parameterized complexity. Using this theory, one tries to *bound the exponential dependence on the dimension*, and sometimes it is possible to indeed find algorithms that are asymptotically optimal in every dimension (even though they might have exponential factors in d in the running time). On the other hand, negative results enable us to prove stronger hardness results than those that are possible with the theory of NP-completeness only.

In this thesis, we are particularly interested in problems that arise from existence theorems of *discrete (or combinatorial) geometry*. Theorems of this type usually deal with geometric objects such as points, hyperplanes, or circles, and guarantee the existence of "interesting" objects under certain conditions. At the same time, they often easily generalize to arbitrary dimensions, and as such they provide an excellent starting point for studying the dichotomy in the computational complexity of lower and higher dimensional problems.

We omit a detailed introduction into the huge field of combinatorial geometry here, but instead state the theorems we are going to consider in the respective chapters. For a detailed introduction to the subject we refer the reader to the textbook by Matoušek [68], which contains all theorems related to the problems considered in this thesis.

In what follows, we first give a short introduction into the field of parameterized complexity theory. Then, we present earlier work that has dealt with bounding the dimension in terms of fixed-parameter tractability. Finally, we give a high level overview of our results.

0.2 Parameterized Complexity

Throughout this thesis, many results can be interpreted in terms of parameterized complexity. Here, we will give a short introduction into the subject that suffices to understand the results in the subsequent sections.

Parameterized complexity has been developed in the 1990's, most notably by Rod Downey and Mike Fellows. The idea is very natural: instead of taking one parameter as the input size to a problem and describing the running time depending on this single parameter, we now take as input two or more parameters. Several algorithms of that kind are known in computational geometry, often formulated using parameterized complexity only implicitly. One of a myriad of examples is Chan's convex hull algorithm [21] that computes the convex hull of a given point set in time $O(n \log h)$, where n is the number of points in the input, and h is the number of points on the boundary of the resulting convex hull.

However, the theory only develops its full strength when dealing with otherwise NP-hard problems. Problems that are tractable when one parameter is small are called *fixed-parameter tractable*, and this is the central concept of parameterized complexity. Moreover, also in this framework hardness results are known. The problems which are considered not to be fixed-parameter tractable are usually captured by the notion of *hardness for the complexity class W[1]*. As for usual NP-hardness proofs, hardness results are often obtained via *parameterized reductions* from existing hard problems.

In what follows, we will explain these concepts in detail. For more concise introductions to the subject of parameterized complexity, we refer the reader to the textbooks by Downey and Fellows [38], Flum and Grohe [49], and Niedermeier [82].

0.2.1 Fixed-Parameter Tractability

We begin with the central definition of a parameterized problem.

Definition 1. *A* parameterized problem L *is a subset* $L \subseteq \Sigma^* \times \mathbb{N}$*, where* Σ *is a finite alphabet. For an element* $(x, k) \in \Sigma^* \times \mathbb{N}$*,* k *is called the parameter.*

A standard example is the parameterized version of the VERTEX-COVER problem. For a graph $G = (V, E)$, a subset $C \subseteq V$ is called a *vertex cover* for G, if every edge of E is incident to a vertex of C.

Definition 2. *The parameterized problem* k-VERTEX-COVER *is defined as*

$$\{(G, k) \mid \textit{there exists a vertex cover } C \subseteq V \textit{ of size } k\}.$$

The notation suggests that k be the parameter in the problem. In order to make this more readable, in what follows we will usually write parameterized problems in the following way.

Definition 3. k-Vertex-Cover
Input: *A graph $G = (V, E)$, and $k \in \mathbb{N}$*
Parameter: k
Question: *Is there a vertex cover of size k for G?*

What have we gained from such a definition? We are now able to express the running time for solving this problem in terms of *two* parameters, namely $n := |G|$ and k. Clearly, the k-Vertex-Cover problem can be solved in time $O(n^{k+1})$, by simply checking each subset of vertices for being a vertex cover. But can we do better? For k-Vertex-Cover, indeed we can, and this leads to the central definition of parameterized tractability.

Definition 4. *Let L be a parameterized problem. L is said to be* fixed-parameter tractable*, if there is an algorithm that decides whether $(x, k) \in L$ in time*

$$O\left(f(k)|x|^c\right),$$

where f is a computable function which depends only on k, and c is a constant independent of $|x|$ and k. Here, $|x|$ denotes the encoding length of x.

An algorithm with that running time is called an *fpt algorithm*, and we say that the problem can be solved in *fpt time* (with respect to a certain parameter). The class of all parameterized problems that admit fpt algorithms is denoted as FPT.

How can we use this to find a better algorithm for the k-Vertex-Cover problem? We will present an algorithm that solves the problem in fpt time and is presumably due on Mehlhorn [76]. Like many fpt algorithms, it uses the paradigm of *bounded search trees.*

It is based on a simple observation, also used for the standard 2-approximation for Vertex-Cover: consider a graph $G = (V, E)$, and any edge $e = uv \in E$. Now any vertex cover for G must contain either u or v, for otherwise the edge e between them would not be covered.

We thus obtain an algorithm as follows: take any edge $e = uv$, and create two new instances $(G', k-1)$ and $(G'', k-1)$. G' is the graph obtained from deleting u and all incident edges, and G'' is the graph obtained by deleting v and all incident edges. Recurse on both instances separately. If $k = 0$, accept if and only if there are no edges left.

The correctness of the algorithm follows from the above observation. For the running time, it holds that $f(|G|, k) \leq 2f(|G|, k-1) + O(|G|)$, and so $f(|G|, k) \in O\left(2^k|G|\right)$. Thus, this algorithm runs in fpt time.

This means that, *for any fixed k*, the Vertex-Cover problem is solvable *linear time*—a huge improvement compared to the trivial $O(n^{k+1})$ algorithm. Of course, this raises the question of parameterized hardness. Is it always possible to find such algorithms?

0.2.2 Parameterized Hardness and Reductions

Consider the parameterized version of the Clique problem:

Definition 5. k-Clique
Input: *A graph $G = (V, E)$, and $k \in \mathbb{N}$*
Parameter: k
Question: *Does G contain a complete subgraph with k vertices?*

Is it possible to also find an algorithm solving this problem in fpt time? So far, no one has managed to find such an algorithm, and this motivates the notion of parameterized hardness.

As usual, hardness results will mostly be obtained by reductions between parameterized problems. In order for this to work, we will need the reductions to capture the notion of fixed-parameter tractability—and "normal" NP-hardness reductions are often not strong enough for this. Thus, for a problem L' to be at least as hard as another problem L, we must be able to reduce L to L' by a *parameter preserving* reduction.

Definition 6. *Let L, L' be two parameterized problems. We say that L reduces to L' by a* parameterized reduction*, $L \leq_{fpt} L'$, if there are computable functions $\phi, \psi \colon \mathbb{N} \to \mathbb{N}$ and $f \colon \Sigma^* \times \mathbb{N} \to \Sigma^*$ such that*

- *$f(x, k)$ can be computed in time $O(\phi(k)\,\mathrm{poly}(|x|, k))$ and*
- *$(x, k) \in L$ if and only if $(f(x, k), \psi(k)) \in L'$.*

While the second point is what we would expect from any reduction, the definition further ensures that the size of the new parameter only depends on the old parameter, but not on the entire input. From this, we derive the main property of such reductions:

Proposition 1. *If $L \leq_{fpt} L'$, and L' can be solved in fpt time, then L can also be solved in fpt time.*

We now define the complexity class W[1], which can be seen as the analog of NP in classical computational complexity. As in the case of NP, which is captured by non-deterministic Turing machines, the lowest level of parameterized hardness can be defined by reductions from the following machine-type problem:

Definition 7. k-Short-Turing-Machine-Acceptance
Input: *A non-deterministic Turing machine M, given by its transition table, and $k \in \mathbb{N}$*
Parameter: k
Question: *Does M halt in k computation steps on the empty input?*

Intuitively, this problem should take time $|M|^{\Omega(k)}$, and indeed no better algorithm is known. As this problem and k-Clique can be reduced to each other by a parameterized reduction, we can thus simply define the class W[1] as follows:

Definition 8. *The* class W[1] *is the class of all parameterized problems which can be reduced to k-*Clique *by a parameterized reduction. Consequently, a problem is said to be* W[1]-hard*, if the k-*Clique *problem (and thus every other problem in W[1]) can be reduced to it by a parameterized reduction.*

The prevailing conjecture is that $FPT \neq W[1]$, i.e., that there are problems in W[1] which do not permit fpt algorithms.

We should mention that the classical W[1]-hard problem is defined via a circuit problem with unbounded *weft*, which is also were the "W" stems from. We do not find it very instructive to define the problem here; intuitively, it defines a class of circuit problems where a bounded search tree approach will not work.

A most simple example of another problem that is W[1]-hard with respect to the size of the solution is k-Independent-Set: the reduction from k-Clique simply maps G to the complement graph, and lets the parameter remain unchanged. On the other hand, the standard NP-hardness reduction from Independent-Set to Vertex-Cover is *not* a parameterized reduction, and thus does not show any hardness in terms of parameterized complexity.

In what follows, we will give an example of a different parameter very suitable for geometric problems.

0.3 The Dimension as a Parameter

In addition to the size of a solution, for geometric problems the dimension of the underlying space is a very natural parameter—based on the motivation of lowering the running time for d-dimensional problems from $n^{\Omega(d)}$ to $f(d)n^c$. Without using parameterized complexity explicitly, Megiddo [73] showed that linear programming with d variables and n constraints can be solved in time $O\left(2^{2^d} n\right)$ (see Matoušek, Sharir, and Welzl [72] for the fastest, and Seidel [89] for the simplest algorithm with such a running time). This shows that linear programming is fixed-parameter tractable with respect to the dimension as a parameter, and can be solved in polynomial, even linear time, for any fixed number of variables.

This motivation has led several other researchers to work on the subject of lowering the dependence on the dimension for other geometric problems. More or less recent progress includes the following examples.

Subset-Congruence (Cabello, Giannopoulos, and Knauer [16]): The problem of deciding whether a point set Q is congruent to a subset of another set P is easily seen to be NP-hard if the dimension is part of the input (see Akutsu [3]). However, this does not exclude an algorithm with a running time of $O(f(d)n^c)$ in $\mathbb{R}^d$. The authors show that, unless FPT=W[1], such an algorithm is not possible. In fact, an even stronger result is derived: the problem is W[1]-hard with respect to *both* parameters, k and the size of the smaller set $|Q|$, excluding an $O(f(d, |Q|)n^c)$ algorithm for the problem.

Klee's measure problem (Chan [23]): For Klee's measure problem, we are given a set of rectangles in the d-dimensional cube, and want to determine the volume of their union. In this paper, a very small improvement over the previous best algorithm is given: it can be solved in $n^{d/2}2^{\log^* n}$ time. At the same time, it is shown that the decision problem is

W[1]-hard by a simple parameterized reduction from k-Clique, making an fpt algorithm very unlikely.

Geometric Clustering (Cabello et al. [17]): Here, the problem of finding a so called k-center in $\mathbb{R}^d$ is investigated with respect to different metrics. The problem is already NP-complete for $k = 2$ (for L_2) and $k = 3$ (for L_∞) if d is part of the input (see Megiddo [74]). However, in this paper it is shown that in d dimensions, for the case L_∞, the decision problem can be solved in time $O(6^d dn + dn \log n)$ for $k = 3$, making it fixed-parameter tractable. On the other hand, for the $k \geq 2$ and L_2, and $k \geq 4$ and L_∞, it is shown that no such algorithm exists: in these cases, the problem becomes W[1]-hard with respect to the dimension.

Covering Points with Hyperplanes (Langerman and Morin [67]): Here, the problem of covering n points with k hyperplanes in $\mathbb{R}^d$ is considered. The decision problem is known to be NP-hard even in $\mathbb{R}^2$ (see Megiddo [75]), if k is part of the input. The authors show that the problem is fixed-parameter tractable with respect to *both*, d and k, by giving an algorithm with a running time of $O(k^{dk} n)$.

0.4 Our Contribution—a High Level Overview

In this thesis, we are going to analyze computational problems in high dimensions in a similar manner. All problems considered come from "standard" theorems in combinatorial geometry. Yet, for some of them, no complexity results in higher dimension were known so far, while for others, the actual dependence on the dimension was unclear until now.

Discrepancy and ε-nets (Part I): Here, we investigate the question of computing the discrepancy of a given point set in $\mathbb{R}^d$ for implicit range spaces. While for the most common ranges this can be done in time $n^{O(d)}$, we show that it is not possible to reduce this dependence on d significantly. Our results hold for both the combinatorial as well as the continuous discrepancy. Subsequently, similar results are obtained for problems related to ε-net computation and verification. This part is based on the paper by Giannopoulos, Knauer, Wahlström, and Werner [52].

Affine degeneracy and ham-sandwich cuts (Part II): In this part, we revisit a technique by Erickson of embedding instances of the d-Sum problem into $(d+1)$-dimensional space. We first consider several decision problems related to basic problems from combinatorial geometry, such as computing minimum Helly sets or Carathéodory sets. Then, we apply our technique to the problem of deciding the existence of a linear ham-sandwich cut for a point set in $\mathbb{R}^d$, and to that of computing the Tverberg depth of a point set. This part is based on the paper by Knauer, Tiwary, and Werner [64].

Erdős-Szekeres theorem (Part III): In this part, we consider the problem of computing maximum subsets of points in convex position. This is motivated by the Erdős-Szekeres theorem, the starting point of geometric Ramsey theory. In the plane, it was known that maximum sets in convex position can be computed in polynomial time. Here, we are able to show that the dependence on the dimension is even worse than in the previous parts: the problem already becomes NP-hard in 3 dimensions. This also leads to several interesting problems for further research. This part is based on the preliminary paper by Knauer and Werner [65].

Tverberg points (Part IV): In this part, we present the fastest known algorithm for computing points of high Tverberg depth in $\mathbb{R}^d$. As the basis for this, we first give a novel lifting argument that allows us to compute high depth points in higher dimension from high depth points in lower dimensions. The technique developed here is highly generic and can be combined with any previous method for computing such points. We subsequently present several variants of this to improve the approximation factor as well as the running time of the algorithm. This part is based on the paper by Mulzer and Werner [80].

Part I

Discrepancy, Empty Boxes, and Verification of ε-Nets in $\mathbb{R}^d$

Chapter 1

Introduction and Motivation

Assume we are given a large set of points on which we have to perform some costly operation. In order to do this more efficiently, we would like to *approximate* it by a smaller set with similar properties. If we can guarantee that the smaller set behaves *not too differently* from the original set, we will be able to replace costly operations on the large set by much cheaper operations on the smaller one, while getting a *similar* result.

One property that is of particular interest in many cases is the behaviour of the point set with respect to certain measures. This leads to the notion of *geometric discrepancy*: intuitively, if the discrepancy of two sets is small, it means that one approximates the other quite well with respect to a certain measure.

Geometric discrepancy has led to significant applications in Computational Geometry, many of which can be found in the textbooks by Chazelle [25], Drmota and Tichy [39], and Matoušek [70]. Also, in several other areas, including optimization, statistics, combinatorics, and computer graphics, it is of high interest. In particular, the *star discrepancy* of a point set is important in multi-variate numerical integration, where the error of a quasi-Monte Carlo integration is bounded as a function of the star discrepancy of the point set used in the integration (by the Koksma–Hlawka inequality, see Niederreiter [83]).

Closely related is the notion of ε-nets and ε-approximations. Based on the seminal works of Vapnik and Chervonenkis [95] and Haussler and Welzl [57], they are of particular interest in the area of approximation algorithms for geometric problems (see also Har-Peled [56]). It is a vivid area of research to find small size ε-nets for particular range spaces (see Alon [4], Pach and Tardos [84], and Chan et al. [24] for recent celebrated results.), as often both the running time as well as the approximation ratio strongly depend on the size of the nets. This relation is stated in the seminal paper by Brönnimann and Goodrich [14], and in an alternative, somewhat simpler formulation also by Even et al. [47].

However, many constructions of small ε-nets are based on randomized algorithms and random sampling of points. As such, these approaches usually lead to Monte-Carlo algorithms, and consequently it is of particular interest to check in reasonable time whether such a construction has indeed produced an ε-net.

This motivates the main computational questions we want to investigate in this chapter:

- How quickly can we determine the discrepancy of a set with respect to certain range spaces in $\mathbb{R}^d$?

- How quickly can we decide whether a given set is an ε-net for a set P with respect to certain range spaces in $\mathbb{R}^d$?

First, we define the different notions of discrepancy in detail and build a connection to the theory of ε-nets and ε-approximations. Afterwards, we give a high level overview of the results in this part.

1.1 Discrepancy and ε-Nets

A set system, or hypergraph, $(X, \mathcal{R})$ consists of a (possibly infinite) *ground set* X, and a subset $\mathcal{R}$ of the power set of X. In many areas, these set systems arise from intersections of geometric objects, such as points and half-spaces, in which case we usually call them (geometric) *range spaces*. If the underlying set is clear, we will often denote a range space simply by $\mathcal{R}$.

A simple example is $\mathcal{H} := \{h \mid h \text{ is a half-plane}\}$, the range space of all half-planes with respect to all points in $\mathbb{R}^2$. For a particular finite set of points $P \subseteq \mathbb{R}^2$, we then denote as

$$\mathcal{H}_P := \{h \cap P \mid h \text{ is a half-plane}\}$$

the set of all combinatorially different ranges induced by intersections of P with the half-planes. Observe that even though there are uncountably many half-planes, $\mathcal{H}_P$ is a finite set. Again, we will denote a range space simply by $\mathcal{R}$ when the underlying set P is clear.

For many geometric ranges, the number of combinatorially different sets determined in $\mathbb{R}^d$ is of order $n^{\Theta(d)}$, where $n = |P|$ is the number of points, and d is the dimension of the underlying space. Thus, in order to define a reasonable decision problem, in the following considerations all range spaces are given implicitly—otherwise, the input would already be huge. That means that the input to our problem is the set of points, and the ranges are determined implicitly by all combinatorially different intersection with the possible ranges. In what follows, it is helpful to think of the ranges as the set of all *axis-parallel rectangles*, or *boxes* for short. Without loss of generality, we will assume that all points lie inside the unit cube $I^d := [0, 1]^d$ in $\mathbb{R}^d$.

1.1.1 Combinatorial Discrepancy

First, we want to define the notion of *combinatorial discrepancy*, or *red-blue discrepancy*. Here, we are given a set of points P and a range space $\mathcal{R}$, and we want to color the points red and blue so that every range contains *roughly* as many red points as blue points, i.e., we want to minimize the maximal difference between all red and blue points among all the ranges. While this seems like a somewhat artificial problem, it has significant implications, e.g., on the theory of ε-nets and many other fields in computational geometry (see the textbook by Chazelle [25]).

Formally, we define the combinatorial discrepancy, or red-blue discrepancy, as follows.

Definition 9. *Let $P = P_r \uplus P_b$ be a set of red points P_r and blue points P_b in $\mathbb{R}^d$, and let $\mathcal{R}$ be a set of ranges. The* combinatorial discrepancy *of P with respect to some $R \in \mathcal{R}$ is defined as*

$$\dot{D}_R(P_r, P_b) := ||R \cap P_r| - |R \cap P_b|| \,.$$

The combinatorial discrepancy of P with respect to $\mathcal{R}$ *is then defined as as the maximum discrepancy over all ranges:*

$$\dot{D}_{\mathcal{R}}(P_r, P_b) := \max_{R \in \mathcal{R}} \dot{D}_R(P_r, P_b) = \max_{R \in \mathcal{R}} ||R \cap P_r| - |R \cap P_b|| \,.$$

From this, we derive the canonical decision problem: for a set P of red and blue points and an implicit range space $\mathcal{R}$, determine whether the discrepancy is larger than some given value k. We will state the exact definitions of decision problems and its variants just before considering it in the respective sections.

We should mention that we would get a different definition by defining the discrepancy as a relative value: if we divide both terms by the respective cardinalities, we get a discrete probability measure, and for these, the discrepancy can be defined as well.

Definition 10. *Let $P = P_r \uplus P_b$ be a set of red points P_r and blue points P_b in $\mathbb{R}^d$, and let $\mathcal{R}$ be a set of ranges. The* relative combinatorial discrepancy *of P with respect to some $R \in \mathcal{R}$ is defined as*

$$\hat{D}_R(P_r, P_b) := \left| \frac{|R \cap P_r|}{|P_r|} - \frac{|R \cap P_b|}{|P_b|} \right| .$$

The relative combinatorial discrepancy of P with respect to $\mathcal{R}$ *is then defined as as the maximum discrepancy over all ranges:*

$$\hat{D}_{\mathcal{R}}(P_r, P_b) := \max_{R \in \mathcal{R}} \hat{D}_R(P_r, P_b) = \max_{R \in \mathcal{R}} \left| \frac{|R \cap P_r|}{|P_r|} - \frac{|R \cap P_b|}{|P_b|} \right| .$$

This definition has the advantage that it allows two sets of very different sizes to have small discrepancy, and thus can be used to determine how well a small set approximates a large set. This will be particularly helpful once we state the connection to ε-approximations. The way we have chosen to define it though, via absolute values, will be used later in the hardness proofs.

1.1.2 Continuous Discrepancy

With the *continuous discrepancy*, or *Lebesgue discrepancy*, we want to define how well the counting measure of a discrete point set approximates the continuous volume with respect to certain ranges. To this end, we replace one of the discrete measures by a continuous measure. What would we expect from such an approximation? Certainly, in order to approximate the continuous measure well, for each box the fraction of points in that box should roughly equal its volume.

Definition 11. *Let P be a set of points in $\mathbb{R}^d$, and let $\mathcal{R}$ be a set of ranges. The continuous discrepancy of P with respect to some $R \in \mathcal{R}$ is defined as*

$$\bar{D}_R(P) := ||R \cap P| - n \operatorname{vol}(R)| .$$

The continuous discrepancy of P with respect to $\mathcal{R}$ *is then defined as as the maximum discrepancy over all ranges:*

$$\bar{D}_\mathcal{R}(P) := \sup_{R \in \mathcal{R}} \bar{D}_R(P) = \sup_{R \in \mathcal{R}} ||R \cap P| - n \operatorname{vol}(R)| .$$

In most cases that we are considering, there are actually only a finite number of *significant* ranges, so we can replace the sup by a max. Again we could define *relative* continuous discrepancy, but in this case, converting one value to the other would simply mean multiplying (or dividing) by n. The way we have chosen to define it will simplify notation later.

A remark. To see that combinatorial discrepancy and continuous discrepancy are actually closely related, we want to point out that both definitions fall under a common, more general definition of discrepancy: for any two measures μ, μ', we can define how well one measure approximates the other:

Definition 12. *Let $\mu, \mu' \colon [0,1]^d \to [0,1]$ be measures with $\mu([0,1]^d) = \mu'([0,1]^d)$. We define the discrepancy of μ and μ' with respect to $R \in \mathcal{R}$ as*

$$D_R(\mu, \mu') := |\mu(R) - \mu'(R)| .$$

The discrepancy of μ and μ' with respect to $\mathcal{R}$ *is then defined as as the maximum discrepancy over all ranges:*

$$D_\mathcal{R}(\mu, \mu') := \sup_{R \in \mathcal{R}} D_R(\mu, \mu') = \sup_{R \in \mathcal{R}} |\mu(R) - \mu'(R)| .$$

This generalizes both definitions: If we set μ, μ' to be the (relative) counting measure on the red and blue set, respectively, we obtain the definition of (relative) combinatorial discrepancy. If we set μ_P to be the counting measure on P, and $\mu'(R) = |P| \operatorname{vol}(R)$, we get the definition of continuous discrepancy.

In any case, the first main question of this part stems from the canonical decision problems arising from these definitions: how quickly can we decide whether the discrepancy of a point set is at least some given value k?

1.1.3 ε-Nets and ε-Approximations

We now give a very short introduction into the theory of ε-nets and ε-approximations, a subject whose influence on computational geometry can hardly be overestimated.

The idea behind an ε-net is to have a *small* set of elements of X that *hit* all *heavy sets* in a range space $\mathcal{R}$.

Definition 13. *Let $(X, \mathcal{R})$ be a (finite) range space, and $\varepsilon > 0$. A set $N \subseteq X$ is an ε-net for X, if for all $Y \subseteq X$ with $|Y| \geq \varepsilon|X|$ we have $N \cap Y \neq \emptyset$.*

In terms of computational complexity, an ε-net is thus a hitting set for the set system defined by all sets of size at least $\varepsilon|P|$.

While this only requires each heavy set to contain at least one point of N, for the case of ε-approximations we require a property that is a lot stronger: it should *approximate* the given set well in the sense that in each range the fraction of points in the set should roughly equal the fraction of all points.

Definition 14. *Let $(X, \mathcal{R})$ be a (finite) range space, and $\varepsilon > 0$. A set $S \subseteq X$ is an ε-approximation for X, if*

$$\forall R \in \mathcal{R} : \left| \frac{|R \cap S|}{|S|} - \frac{|R \cap X|}{|X|} \right| \leq \varepsilon.$$

This justifies the name ε-approximations, and it is the point where the connection to discrepancy becomes more obvious: an ε-approximation is a set with small *relative discrepancy* with respect to $\mathcal{R}$; in the notation of the previous section, this means that $\hat{D}_{\mathcal{R}}(S, X) \leq \varepsilon$.

So far, these definitions are not so interesting—for example, every set is an ε-approximation for itself. The definitions will only become useful once we can guarantee that we can construct ε-nets (and -approximations) that are much smaller than the original set. For this to hold, we need some more information about the underlying range spaces.

To this end, we define the notion of Vapnik-Chervonenkis dimension of a range space, which was introduced in [95].

Definition 15. *Let $(X, \mathcal{R})$ be a range space. A subset $Y \subseteq X$ is* shattered *by $\mathcal{R}$, if*

$$\{R \cap Y \mid R \in \mathcal{R}\} = 2^Y.$$

Now the VC-dimension is the size of a largest set shattered by $\mathcal{R}$:

Definition 16. *Let $(X, \mathcal{R})$ be a range space. The* Vapnik-Chervonenkis dimension *of $(X, \mathcal{R})$ is defined as*

$$\max_{Y \subseteq X} \{|Y| \mid Y \textit{ is shattered by } \mathcal{R}\}.$$

It turns out that the VC-dimension of geometric range spaces is often strongly related to the dimension of the underlying space. For example, the range space $(\mathbb{R}^d, \mathcal{H})$ has VC-dimension $d + 1$, as one can derive from Radon's theorem (cf. Section 12.1). In general, it is known that range spaces of bounded description complexity (like boxes in $\mathbb{R}^d$) have bounded VC-dimension, see Sharir and Agarwal [90].

Set systems of finite VC-dimension are of particular interest in computational geometry, most notably because of the following two theorems due to Haussler and Welzl [57] and Vapnik and Chervonenkis [95], respectively.

Theorem (ε-Net theorem; Haussler and Welzl [57]). *Let $(X, \mathcal{R})$ be a range space of VC-dimension d and $P \subseteq X$ be a finite set. Let $\varepsilon > 0$. Then there exists an ε-net N of size at most*

$$c_d \frac{1}{\varepsilon} \log \frac{1}{\varepsilon},$$

where $c_d \in O(d)$.

And the analogous result for ε-approximations:

Theorem (Vanik and Chervonenkis [95]). *Let $(X, \mathcal{R})$ be a range space of VC-dimension d and $P \subseteq X$ be a finite set. Let $\varepsilon > 0$. Then there exists an ε-approximation S of size at most*

$$c_d \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon},$$

where c_d is a constant only depending on d.

This means that the sizes of ε-nets and ε-approximations only depend on the value of ε and the VC-dimension d, but are independent of the number of points in P.

Even though the two theorems about ε-nets and ε-approximations guarantee the existence of small sets, so far this does not give any method to efficiently *find* such them. While there are deterministic methods to generate small ε-nets and ε-approximations in time $f(d, \frac{1}{\varepsilon})n^c$ (Chazelle [25]), they are usually very complicated. But, fortunately, as one reads off from the proofs of the two theorems, something a lot stronger is actually true: a *random sample* S of points of P of the respective size forms an ε-net or ε-approximation with high probability! Thus, by picking each point with probability $c'_d \frac{1}{\varepsilon} \log \frac{1}{\varepsilon}/n$, we are likely to end up with an ε-net. While this is surely one of the simplest algorithm one can think of, it has a serious flaw: when given such a sample, we would like to know whether it really *is* a net. Thus, an efficient and simple algorithm for checking such a net would allow us to replace the complicated deterministic methods by a simpler sample-and-check algorithm that guarantees us to deliver such a net.

This leads us to the second main question of the chapter: can we, in reasonable time, check whether a given set is an ε-net for a given point set and an implicitly defined set of ranges?

1.2 Our Contribution

Unfortunately, every known method for computing the discrepancy of a point set is computationally intensive for most of the ranges; given a set P of n points in d dimensions, every known algorithm for getting even a constant-factor approximation of its star discrepancy has a running time of $n^{\Theta(d)}$.

The main question we ask here is whether this dependency on d is necessary. Specifically, we ask whether the decision version for star discrepancy (and other related problems) can be solved in $O(f(d)n^c)$ time, i.e., whether it is fixed parameter tractable with respect to the dimension (cf. Section 0.2). We will exclude the existence of such an algorithm under standard complexity theoretic assumptions, using tools from parameterized complexity, by showing that the problems considered are W[1]-hard with respect to the dimension. The reductions are based on a general framework by Cabello et al. [17] of deconstructing the space into orthogonal planes, into which point sets are placed.

We will concentrate on computing the discrepancy and the related problem of finding large monochromatic or empty boxes (which, by definition, have high discrepancy). Each of the six main variants of the problem considered in Chapters 2 and 3 is interesting in its own right, and often they have been considered in separate papers, as the related work shows. Fortunately, we are able to tackle all of them in a similar fashion by our approach.

First, in Chapter 2, we will consider the discrete versions of the problem, i.e., computing the combinatorial discrepancy and finding large monochromatic boxes. For this, the least technical detail is required. However, the essence of the reduction is already stated clearly there and forms the basis of the hardness proofs in later chapters.

In Chapter 3, we use this reduction to prove the hardness of the related problem of computing the continuous discrepancy. We consider several variants of the problems, such as computing a maximum volume box that does not contain any points, and each of these requires some new idea for the reduction to go through.

Using this technique, in Chapter 4 we state similar results with respect to half-spaces and simplices. We also elaborate the precise connection of these problems to the verification of ε-nets, partially answering the main question given in the introduction to this part. In particular, we show that it cannot be verified in time $O(f(d, 1/\varepsilon)n^c)$, whether or not a set is indeed an ε-net with respect to another.

Chapter 2

Combinatorial Discrepancy

In this section, we will consider the two discrete variants of the problem, namely computing an axis-parallel box with a maximum number of blue points that contains no red points, and computing the combinatorial discrepancy of a colored point set.

2.1 The Bichromatic Rectangle Problem

We will first consider the BICHROMATIC-RECTANGLE problem. We define the parameterized version of the decision problem as follows:

Definition 17. k-d-BICHROMATIC-RECTANGLE
Input: *A set P_r of red points and a set P_b of blue points in $\mathbb{R}^d$, and $k \in \mathbb{N}$*
Parameter: k, d
Question: *Is there an axis-parallel box H with $H \cap P_r = \emptyset$ and $|H \cap P_b| \geq k$?*

We remark that the name MONOCHROMATIC-RECTANGLE would have been much more suitable, but the former has been used throughout the literature.

The problem was shown to be NP-hard by Eckstein et al. [41], and in the same paper, a straightforward $O(n^{2d+1})$ algorithm was given, where n denotes the number of points. However, the hardness proof creates instances whose dimensions grows linearly in the size of the input, and thus is not a parameterized reduction.

Recently, an output-sensitive $O(m \log^{d-2} n)$ time algorithm was developed by Backer and Keil [9], where m is the number of significant boxes. In the worst case, this can be as much as $\Omega(n^d)$. Further, Aronov and Har-Peled [7] gave a $(1-\varepsilon)$-approximation algorithm that runs in time $O\left(n^{\lfloor d/2 \rfloor}(\varepsilon^{-2} \log n)^{\lceil d/2 \rceil}\right)$ for half-spaces as ranges. They ask whether their algorithm can be improved, and here we will show that in terms of asymptotic behaviour of the exponent of n, this is not the case.

All further results in this and the upcoming chapters heavily rely on the construction that follows now, so it should be read with care.

In what follows, we will show that the k-d-BICHROMATIC-RECTANGLE problem is W[1]-hard with respect to both these parameters, excluding the possibility of an algorithm with running time $O(f(d,k)n^c)$ for the problem. Observe that in terms of hardness, this is a stronger parameterization than the parameterization *only* by d: hardness for the latter would still leave the possibility of an algorithm with the aforementioned running time.

A box that does not contain any red point will be called *feasible*. Let $\mathcal{B}$ denote the set of (closed or open) axis-parallel boxes inside the unit cube I^d. For a given set of points $P = P_r \uplus P_b$, let

$$E(P_r, P_b) = E_{\mathcal{B}}(P_r, P_b) := \max_{B \in \mathcal{B}, B \cap P_r = \emptyset} |B \cap P_b|$$

denote the size of an optimal solution. We are thus interested in the problem of computing a feasible box $B \in \mathcal{B}$ that maximizes $|B \cap P_b|$.

2.1.1 The Idea

In order to show that the k-d-BICHROMATIC-RECTANGLE problem is $W[1]$-hard, we will give a reduction from the k-CLIQUE problem. For a given simple graph $G = ([n], E)$, where $[n] := \{1, \ldots, n\}$, we will construct point sets $P_r = P_r(G,k)$ and $P_b = P_b(G,k)$ in $\mathbb{R}^{2k}$ such that G has a clique of size k if and only if $E(P_r, P_b) \geq k+1$. Observe that both the size of the solution $k+1$ as well as the dimension $2k$ depend only on k and not on the size of G. Thus, it is a parameterized reduction.

We first describe the reduction on a higher level before presenting the details in the subsequent Section 2.1.2.

We will put blue and red points into k pairwise orthogonal coordinate planes. As we will see (cf. Observation 1), this will allow us to consider point sets in different planes independently of each other—whether or not a certain point is contained in a box B will only depend on the (two) non-zero coordinates. These points will be used to encode the vertices of G. Additional red points will then be used to encode the edge-set of G, each of which will lie in the product of two planes, and thus have four non-zero coordinates. Finally, we put a single blue point at the origin to make sure that a maximal monochromatic box contains the origin, simplifying our arguments a little.

Each of the k planes will contain n blue points, corresponding to the vertices of the graph, and $n-1$ red points. The red points are placed such that no feasible box can contain more than one blue point from a single plane. Thus, at most k of these blue points can be contained in any feasible box. At the same time, each choice of k points corresponds to a choice of k vertices from our input graph G. In our construction, a box can contain points x and y from two different planes if and only if the corresponding vertices are connected in G. This is achieved by putting red points into the product of the respective planes (which is a four-dimensional subspace).

With the origin as an additional blue points, this will ensure that any feasible box containing $k+1$ blue points corresponds to a k-clique in G, and vice versa.

2.1.2 The Construction

Preliminaries

First, let us introduce some basic notation. Let $\{e_i \mid 1 \leq i \leq 2k\}$ denote the standard basis of $\mathbb{R}^{2k}$. For a set of vectors $\{x_1, \ldots, x_l\}$, let $\langle x_1, \ldots, x_l \rangle$ denote their linear span, i.e., the set $\left\{\sum_{i=1}^{l} \alpha_i x_i \mid \alpha_i \in \mathbb{R}\right\}$. For $1 \leq i \leq k$, we then define the two-dimensional subspace

$$\mathcal{S}_i := \langle e_{2i-1}, e_{2i} \rangle \subseteq \mathbb{R}^{2k}.$$

Further, for $1 \leq i < j \leq k$, we then set $\mathcal{S}_{ij}$ to be the sum of $\mathcal{S}_i$ and $\mathcal{S}_j$, i.e., $\mathcal{S}_{ij} := \langle \mathcal{S}_i, \mathcal{S}_j \rangle$. For $p \in \mathcal{S}_i$ and $q \in \mathcal{S}_j$, observe that the unique point in $\mathcal{S}_{ij}$ that (orthogonally) projects to p (into $\mathcal{S}_i$) and to q (into $\mathcal{S}_j$) is $p + q$.

For an axis-parallel box in $\mathbb{R}^d$ defined as $B = \Pi_{i=1}^{d}[x_i, x_i']$, let us say that $(x_1, \ldots, x_d)$ is the *lower left corner*, and $(x_1', \ldots, x_d')$ is the *upper right corner*.

All reductions are based on the following observation:

Observation 1 (Decomposition)**.** *Let B be an axis-parallel box and let $x = (x_1, \ldots, x_d)$ be a point in $\mathbb{R}^d$. Then $x \in B$ if and only if $\mathrm{pr}_i(x) = x_i \in \mathrm{pr}_i(B)$ for all $1 \leq i \leq d$, where pr_i denotes the projection to the i-th coordinate.*

While this is quite obvious, it will prove to be very helpful in further considerations. Observe that for general boxes that are not axis-parallel, this does not hold.

As we will at first mainly consider boxes whose lower left corner is the origin, it will be helpful to also state this in a slightly different form.

Corollary 1. *Let $B = \Pi_{i=1}^{d}[0, y_i]$ be an axis-parallel box containing the origin and let $x = (x_1, \ldots, x_d)$ be a point in I^d. Then $x \in B$ if and only if $x_i \leq y_i$ for all $1 \leq i \leq d$.*

In particular, for every x, it suffices to consider the coordinates $x_i \neq 0$.

From now on, we often omit the word axis-parallel whenever we speak about axis-parallel boxes or (hyper-)rectangles.

The scaffold construction.

Let $1 \leq i \leq k$. We now define the set of n blue points $b_i(v)$ in the i-th plane $\mathcal{S}_i$. For a vertex $1 \leq v \leq n$, let

$$b_i(v) := (v, n + 1 - v) \in \mathcal{S}_i \subseteq \mathbb{R}^{2k}.$$

Then, let

$$(P_b)_i^{\text{scaffold}} := \{b_i(1), \ldots, b_i(n)\} \subseteq \mathcal{S}_i$$

denote the set of all blue points in the i-th plane. As we will see later, choosing a rectangle containing point $b_i(v)$ will correspond to choosing vertex v from G. Let

$$P_b^{\text{scaffold}} := \biguplus_{1 \leq i \leq k} (P_b)_i^{\text{scaffold}}$$

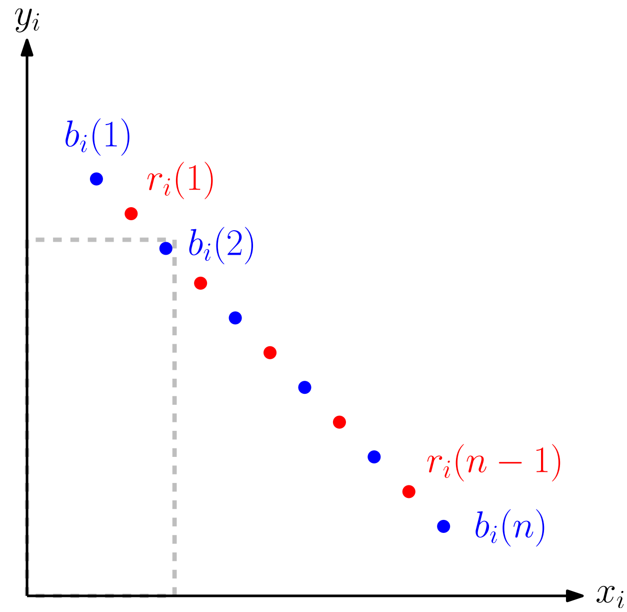


Figure 2.1: The scaffold construction in $\mathcal{S}_i$ with vertex 2 selected. Observe that any feasible box B can contain at most one blue point from each $(P_b)_i^{\text{scaffold}}$.

be the set of all these blue points.

As we do not want any feasible box to contain more than one point from a single $\mathcal{S}_i$, we have to add red points between them that forbid this. To this end, for $1 \leq v \leq n-1$, we define a red point $r_i(v) := (v + 1/2, n + 1 - (v + 1/2))$ and set

$$(P_r)_i^{\text{scaffold}} := \{r_i(1), \dots, r_i(n-1)\} \subseteq \mathcal{S}_i.$$

Finally, we define

$$P_r^{\text{scaffold}} := \biguplus_{1 \leq i \leq k} (P_r)_i^{\text{scaffold}}$$

to be the set of all red scaffolding points. See Figure 2.1 for an example of the scaffold construction in $\mathcal{S}_i$.

Encoding edges.

If we did not add any more points, we would now be able to pick n^k different feasible boxes containing k points, corresponding to the n^k ordered tuples of k vertices from G. In order to forbid certain tuples—namely the ones that contain a point pair corresponding to vertices which are *not* connected—we will now add additional red points to encode the edges.

To this end, we place points inside the product of different $\mathcal{S}_i$: for $1 \leq i < j \leq k$ and vertices $1 \leq u, v \leq n$, we define the point

$$r_{ij}^{\text{kill}}(uv) := b_i(u) + b_j(v) \in \mathcal{S}_{ij}.$$

We then define the set of all killing points in $\mathcal{S}_{ij}$:

$$(P_r)_{ij}^{\mathrm{E}} := \{r_{ij}^{\mathrm{kill}}(uv) \mid uv \notin E\}.$$

Observe that as G simple (i.e., contains no loops), all points of the form $r_{ij}(uu)$ are also added. Finally, we set

$$P_r^{\mathrm{E}} := \biguplus_{1 \le i \ne j \le k} (P_r)_{ij}^{\mathrm{E}}$$

to be the set of all killing points. See Figure 2.2 for an example where $uv \notin E$.

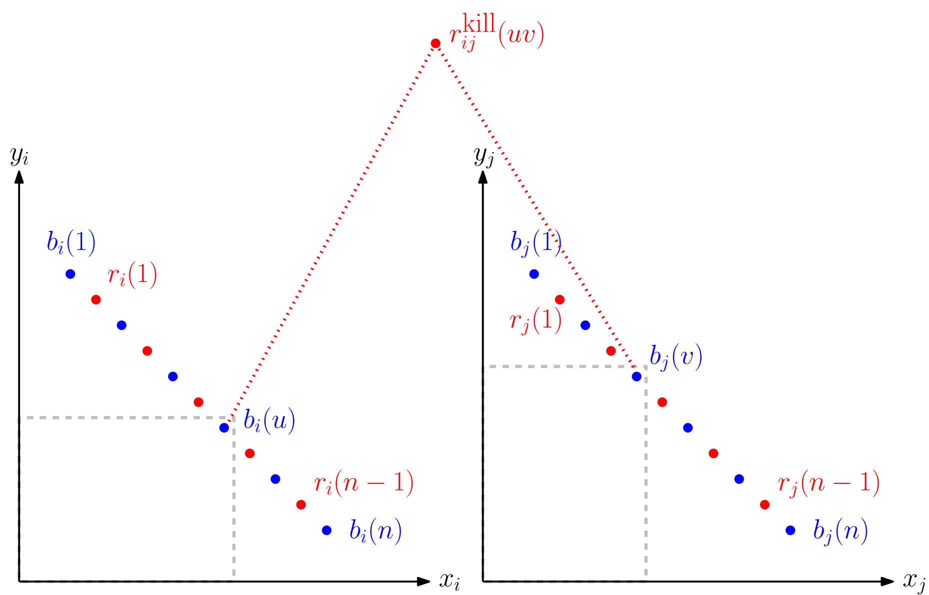


Figure 2.2: $b_i(u)$ is the projection of $r_{ij}^{\mathrm{kill}}(uv)$ to $\mathcal{S}_i$ and $b_j(v)$ is the projection of $r_{ij}^{\mathrm{kill}}(uv)$ to $\mathcal{S}_j$. As both points are contained in the dashed rectangles, the resulting box also contains $r_{ij}^{\mathrm{kill}}(uv)$.

Observation 1 states that whether or not a point is contained in a box depends only on the respective non-zero coordinates. Thus, for the killing points we get the following main property:

Observation 2. *A point $r_{ij}^{\mathrm{kill}}(uv)$ is contained in a box B (containing the origin) if and only if both $b_i(u)$ and $b_j(v)$ are contained in B.*

Putting things together

For $G = ([n], E)$ and $k > 0$ we construct point sets $P_r = P_r(G,k), P_b = P_b(G,k)$ in $\mathbb{R}^{2k}$ as follows:

- $P_r := P_r^{\mathrm{scaffold}} \cup P_r^{\mathrm{E}}$
- $P_b := \{\mathbf{0}\} \cup P_b^{\mathrm{scaffold}}$

The size of the point set is $O(k^2n^2)$ and the coordinates of the points can be encoded by $O(\log kn)$ bits. Clearly, the construction can be performed in time polynomial in both k and n.

We now come to prove the main lemma of this section.

Lemma 1. *G has a k-clique if and only if $E(P_r, P_b) = k + 1$.*

Proof. First, observe that any feasible box B can contain at most $k + 1$ points, as $|B \cap (P_b)_i^{\text{scaffold}}| \leq 1$, for $1 \leq i \leq k$. Thus, in order to contain $k+1$ blue points, any feasible box has to contain the (blue point at the) origin.

"$\Rightarrow$" Let $v_1, \ldots, v_k$ be a clique of size k. We choose a (closed) box B with upper right corner $b_i(v_i)$ in $\mathcal{S}_i$ and lower left corner $\mathbf{0}$.

B contains exactly one point from each of the $\mathcal{S}_i$, and additionally the origin, making it a total of $k + 1$ blue points. We show that B is feasible.

First, by definition, B contains no point of P_r^{scaffold}. Further, assume that B contains a point of P_r^E, say $r_{ij}^{\text{kill}}(uv) = b_i(u) + b_j(v) \in (P_r)_{ij}^E$. Then, by Observation 2, B contains both $b_i(u)$ and $b_j(v)$. But as u and v are connected, $r_{ij}^{\text{kill}}(uv)$ has not been added to P_r. Thus, B is feasible.

"$\Leftarrow$" Let B be any feasible box containing $k+1$ blue points. It can contain at most one point from each $\mathcal{S}_i$, and additionally the origin. Thus, such a selection of points corresponds to a selection of k vertices from G. Further, as the box does not contain any killing point, it means that these vertices are pairwise connected in G. Thus, they form a k-clique in G. $\square$

Summing up, we have created an instance of k-d-Bichromatic-Rectangle in dimension $2k$ in time polynomial in both n and k with the property that there is a feasible box with $k + 1$ points if and only if G has a clique of size k. In our reduction the number of points in a solution as well as the dimension only depend on the parameter k. As the k-Clique problem is W[1]-hard, we thus derive the following result:

Theorem 1. *The k-d-*Bichromatic-Rectangle *problem is $W[1]$-hard .*

That still leaves the question for approximation algorithms. As for approximation schemes, we are able to state a hardness result based on a known relation to parameterized complexity theory. An Efficient Polynomial-Time Approximation Scheme (EPTAS) is an algorithm that, given any $\varepsilon > 0$, produces a solution whose value is at least $(1 - \varepsilon)$ times the value of an optimal solution in $O(f(1/\varepsilon) \cdot n^c)$ time, for some constant $c > 0$. As noted in [20, Lemma 11], an integer-valued optimization problem that is $W[1]$-hard when parameterized by the size of the solution is unlikely to have an EPTAS. To see this, consider setting $\varepsilon = 1/(k + 1)$ for solution value k; the only $(1 - \epsilon)$-approximate solution is the optimum. Since we have shown that the problem is hard with respect to *both* the dimension and the size of the solution, the above implies the following:

Corollary 2. *The (optimization version of the) k-d-*Bichromatic-Rectangle *problem does not admit a $(1-\epsilon)$-approximation scheme that runs in $O(f(1/\varepsilon, d) \cdot n^c)$ time, for $c > 0$ constant, unless $W[1] = FPT$.*

In this sense, the algorithm by Aronov and Har-Peled [7] cannot be improved to have an fpt running time. In Section 4.3, we combine this result with a deeper result from parameterized complexity theory to strengthen it even further.

2.2 The Red-Blue Discrepancy Problem

We now turn to the problem of computing the red-blue discrepancy, or combinatorial discrepancy, of a point set in $\mathbb{R}^d$. Here, we want to maximize the difference between red and blue points inside a box.

With slight modifications, the NP-hardness of the canonical decision problem follows from the aforementioned hardness result in [41], but we are again aiming for a stronger statement. To this end, we choose the parameterization by the dimension d.

Definition 18. d-Red-Blue-Discrepancy
Input: *A set P_r of red points and a set P_b of blue points in $\mathbb{R}^d$, and $\delta \in \mathbb{N}$*
Parameter: d
Question: *Is the discrepancy of the set with respect to axis-parallel boxes at least δ?*

Observe that here we chose the weaker parameterization by d only. It is an interesting question whether our results also hold for the problem parameterized by both, d and δ.

In order to show W[1]-hardness for d-Red-Blue-Discrepancy, we have to overcome two obstacles that let the previous construction fail:

- The maximum discrepancy might be attained for boxes that have a lot more *red* than *blue* points.
- Even if we can ensure that a high discrepancy is attained for boxes with a lot more blue than red points, such boxes might still contain *some* red points.

In order for our construction to work, we thus have to make sure that a certain threshold can only be achieved by boxes containing only blue points. Fortunately, this can be easily fixed. To make sure that a high discrepancy is attained for a box with many blue points, starting from the above construction, we simply place $|P_r|$ additional blue points at the origin (or, if we do not want to allow multiple points, we place $|P_r|$ points close by).

We can then show the following lemma, similar to Lemma 1.

Lemma 2. *G has a k-clique if and only if $\dot{D}_{\mathcal{R}}(P_r, P_b) = k + |P_r| + 1 =: \delta$.*

Proof. "$\Rightarrow$" If there is a k-clique in G, we choose the box as in Lemma 1. It contains no red points, but the $|P_r|+1$ blue points at the origin and one blue point from each $\mathcal{S}_i$. Thus, it has discrepancy $k + |P_r| + 1$, as desired.
"$\Leftarrow$" Consider any box B with discrepancy δ. As there are only $|P_r|$ red points in total, this box cannot consist of more red points than blue points. Further, by construction, the discrepancy of the box with respect to each $\mathcal{S}_i$ is at most one. In order for B to have discrepancy at least $k + |P_r| + 1$, it must thus contain all blue points at the origin. Now

any box of such discrepancy cannot contain any red killing point: the discrepancy would at most be $k + |P_r| + 1 - 1$.

This means that among all points chosen in the respective $\mathcal{S}_i$, all corresponding vertices are connected. If we pick any set of points that contains one blue point from each $\mathcal{S}_i$, the corresponding vertices thus form a k-clique. □

From this, we derive the main theorem of this section.

Theorem 2. *The problem d-RED-BLUE-DISCREPANCY is W[1]-hard.*

2.3 General Position

In both constructions, the point sets created lie in degenerate position, in the sense that many points lie in low dimensional subspaces. In addition to this, many points coincide on some of the coordinates—an undesirable property when dealing with axis-parallel boxes as ranges. In order to avoid this, we now indicate how to put the point set in general position to show that the hardness results also hold for this case. To this end, we show how to maintain the main properties of the reduction for the case of the BICHROMATIC-RECTANGLE problem; the discrepancy case is then straightforward.

The plan is to modify our construction a little in order to make it more "robust" with respect to perturbations. Afterwards, we can simply perturb it a little in order to put the points in general position, while maintaining its combinatorial properties.

Definition 19. *A set P of points in $\mathbb{R}^d$ is called α-robust, if*

$$\min_i \{|\operatorname{pr}_i(x), \operatorname{pr}_i(y)| \mid x, y \in P\} \geq 2\alpha.$$

In particular, in a set that is α-robust for some $\alpha > 0$, no coordinates of any to points coincide.

We say that Q is a β-perturbation of P, if Q is created from P by perturbing each point by at most β in each direction. The following lemma justifies the name α-robust.

Lemma 3. *If P is α-robust and Q is a β-perturbation of P, for $\beta < \alpha$, then for any closed box B for P there is a box B' for Q that contains exactly the same (perturbed) points.*

Proof. Without loss of generality, assume that the endpoint of the box in each coordinate lies as some point from P. Enlarge B by a value of β in each direction to yield a box B'. Then, as every point is perturbed by at most β in each direction, for any point p, the corresponding perturbation is in the new box B'. Further, as the distance between the endpoint of B and any point not contained in B was at least $2\alpha > 2\beta$ in each coordinate, B' does not contain any additional points. □

In order to use this lemma, we now show how to make our construction α-robust for a suitable value of α. To this end, let n denote the total number of points, and set $\alpha := 1/(10n^2)$. Now consider a single $\mathcal{S}_i$. Let $X_i = \{x_i, \ldots, x_{2n-1}\}$ denote the set of all

$2n-1$ (red and blue) points in $\mathcal{S}_i$, ordered with respect to the $2i$-th coordinate. For $j \neq i$, we now set the j-th coordinate of each x_l in $\mathcal{S}_i$ to be $(2i(2n-1)+2l)\,\alpha$, and denote the corresponding points by $b_i(u)$ and $r_i(u)$, respectively.

Further, for the blocking points we now set $r_{ij}^{\text{kill}}(uv) := b_i(u) + b_j(v) - 1/10 \cdot \left(\sum_{l=1}^{2k} e_l\right)$.

We call this modified set P'. Analog to Lemma 1, P' admits a box containing $k+1$ blue and no red points if and only if G has a clique of size k. Further, the set created this way is α-robust, for $\alpha = 1/(10n^2)$, as each pair of points differs by at least 2α in each coordinate.

Finally, by perturbing each point of P' by a value of at most $\beta := \alpha/4$ in each direction, we obtain a point set in general position. Combining Lemma 3 and the fact that P' is α-robust, we thus get the following strengthening of the theorems above:

Theorem 3. *Theorems 1 and 2 still hold if the point sets are in general position.*

Chapter 3

Continuous Discrepancy

We now turn to the continuous versions of the problem: computing maximum volume boxes not containing any points, and computing the Lebesgue discrepancy of a point set.

So far, our instances have been very combinatorial. The measures were only defined by intersections of the ranges with a (finite) discrete set. To state it very vaguely: as one of our measures now becomes the volume of the range, we have to make the instances a bit more geometric.

As for the ranges considered in this section, we will restrict ourselves to the case of axis-parallel boxes $\mathcal{B}$, and axis-parallel boxes containing the origin $\mathcal{A}$, inside the unit cube $I^d = [0,1]^d$.

A crucial observation is that we can restrict ourselves to the *significant* boxes, i.e., all boxes that are products of open, half-open, or closed intervals and which touch one of the points on each side. This allows us to only consider finitely many boxes, and it shows that the problems considered can all be solved in time $n^{O(d)}$.

The proofs follow the ones of the discrete cases in Chapter 2—except that now we need to formulate everything in the continuous setting. In the upcoming constructions, the analog of a box containing many blue points is now a box that is "large" in terms of volume.

As the numbers appearing in this construction have a bit complexity polynomial in the input size, we can use the standard RAM as our computational model.

3.1 The Largest Empty Anchored Box Problem

First, we deal with the problem where we have to find an axis-parallel box of maximum volume that has its lower left corner at the origin and contains no points from a given set P. We will call these boxes *anchored boxes*, or *subintervals*, or *corners*, or simply *anchors*, and the set of all anchored boxes inside the unit cube will be denoted as $\mathcal{A}$. In the next two sections, we will use the term box as shorthand for *anchored box*, and rectangle for *anchored* rectangle, only emphasizing it here and there to remind the reader.

The parameterized version of the decision problem is defined as follows:

Definition 20. d-Largest-Empty-Anchor
Input: *A set P of points in I^d, and $\delta \in \mathbb{Q}$*
Parameter: d
Question: *Is there an anchored box $A \in \mathcal{A}$ with $A \cap P = \emptyset$ and $\mathrm{vol}(A) \geq \delta$?*

The Largest-Empty-Anchor problem was recently shown to be NP-hard by Gnewuch et al. [53]. Here, besides showing its W[1]-hardness, our construction yields that it is even $W[1]$-hard to approximate this problem within a factor of $2^{\mathrm{poly}(|P|)}$.

3.1.1 The Construction

We now come to describe the construction. Again, we reduce from k-Clique. Let $G = ([n], E)$ denote the graph for which we want to decide whether it has a clique of size k. The main idea is as follows: we again consider $\mathbb{R}^{2k}$ as product of k orthogonal two-dimensional planes, denoted as $\mathcal{S}_i$, $1 \leq i \leq k$. In these, we place points to allow only a certain set of large rectangles in each $\mathcal{S}_i$—namely one for each vertex. Again, the product of the respective $\mathcal{S}_i$ is used to place additional blocking points which encode the edge set.

We will proceed as follows: First, we determine where the upper right corners of the n large rectangles in each plane have to be. From this, we will determine the blocking points (which are the analog of the red points above) that are needed to encode the edges.

In each plane, there will be n *large* rectangles to choose from, corresponding to the n vertices of G. It will only be possible to choose large rectangles from two different planes if the corresponding vertices are connected in G. Observing that the volume of a box is the product of the areas of the respective $\mathcal{S}_i$, this yields a one-to-one correspondence between "large" empty boxes and cliques of size k.

Let $\gamma > 1$ be a parameter to be specified later. For now, we can just set $\gamma := 2$, but later we will use the freedom we have in choosing this parameter in order to show the other results. One possibility to determine the upper right corners of the rectangles, each having area $C := \frac{1}{\gamma^{n-1}}$ in one $\mathcal{S}_i$, is as follows: for each $u \in V$, we set

$$c_i(u) := \left(C\gamma^{u-1}, \frac{1}{\gamma^{u-1}}\right)$$

and define

$$C_i(u) := \{c_i(u) \mid 1 \leq u \leq n\}.$$

Observe that every rectangle in $\mathcal{S}_i$ whose upper right corner is at $c_i(v)$, for some v, has an area of $C\gamma^{v-1} \cdot \frac{1}{\gamma^{v-1}} = C$.

We now place points such that any maximal empty (open) rectangle, i.e., a rectangle supported by two points, has its upper right corner at one $c_i(u)$. This can be realized by the following set: Define

$$p_i(u) := \left(C\gamma^{u-1}, \frac{1}{\gamma^{u}}\right) \in \mathcal{S}_i$$

and set

$$P_i^{\text{scaffold}}(u) := \{p_i(u) \mid 0 \le u \le n\}.$$

Finally, set

$$P^{\text{scaffold}} := \biguplus_{1 \le i \le k} P_i^{\text{scaffold}}.$$

Thus, in each $\mathcal{S}_i$, we have n choices for the upper right corner of the rectangles: the points $c_i(u)$, $1 \le u \le n$. If a rectangle has its upper right point somewhere else on $(x, C/x)$ or above, it contains a point from P^{scaffold}, and any other feasible rectangle has smaller size.

Consequently, choosing a large rectangle in each of the $\mathcal{S}_i$ yields an empty box of total volume C^k, and it again corresponds to a choice of an ordered k-tuple of the vertices. See Figure 3.1 for an example.

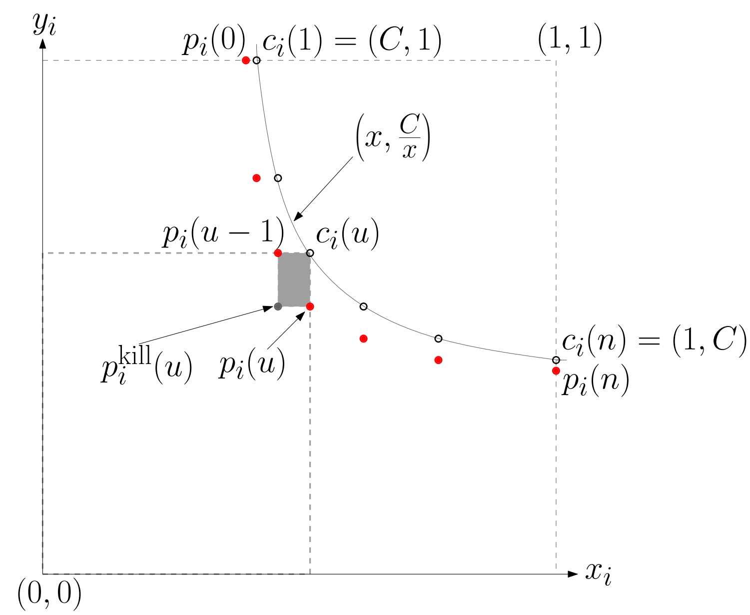


Figure 3.1: The plane $\mathcal{S}_i$. A rectangle selecting vertex u and the region $F_i(u)$ indicated.

3.1.2 Encoding the Edges

As above, if the vertices corresponding to two different large rectangles in the planes $\mathcal{S}_i$ and $\mathcal{S}_i$ are not connected, we will add a point in the product $\mathcal{S}_{ij}$ that forbids these two rectangles to be chosen at the same time. To this end, we define points

$$p_i^{\text{kill}}(u) := \left(C\gamma^{u-2}, \frac{1}{\gamma^u}\right) \in \mathcal{S}_i.$$

These points are themselves not added to the set P, but we use them to define the actual killing points:

$$p_{ij}^{\text{kill}}(uv) := p_i^{\text{kill}}(u) + p_j^{\text{kill}}(v) \in \mathcal{S}_{ij}.$$

We then add all the killing points corresponding to non-edges in the graph:

$$P^{\mathrm{E}} := \{p_{ij}^{\mathrm{kill}}(uv) \mid i \neq j, uv \notin E\}.$$

Recall that this also includes all points of the form $p_{ij}^{\mathrm{kill}}(uu)$, as there are no loops in the graph. Finally, we define the set of all points as

$$P := P^{\mathrm{E}} \cup P^{\mathrm{scaffold}}.$$

The size of P is $O(n^2k^2)$, and if we set $\gamma := 2$ and $C := 1/\gamma^{n-1} = 1/2^{n-1}$, all coordinates have an encoding size polynomial in the size of the input. Clearly, the construction can be performed in time polynomial in k and n.

We can now prove the correctness of the construction. To this end, let $F_i(u)$ denote the open region with corners $p_i(u-1)$, $c_i(u)$, $p_i(u)$, $p_i^{\mathrm{kill}}(u)$, as indicated in Figure 3.1.

Lemma 4. *Any feasible rectangle in $\mathcal{S}_i$ that does not intersect any region $F_i(u)$, $1 \leq u \leq n$, has size at most $C/\gamma = 1/\gamma^n$.*

Proof. Such a rectangle has its upper right point below the graph $(x, \frac{C}{x\gamma})$ going through the points $p_i(u)$, $1 \leq u \leq n$. Thus, its area is at most $x\frac{C}{x\gamma} = C/\gamma$. □

We use this to prove the main lemma, the continuous analog of Lemma 1 from the previous chapter:

Lemma 5. *G has a k-clique if and only if there is an empty anchored box of size $\delta = C^k$. Further, if G does not have a k-clique, the largest empty anchored box has volume at most C^k/γ.*

Proof. "⇒" Let $v_1, \ldots v_k$ be a clique in G. In each $\mathcal{S}_i$, $1 \leq i \leq k$, choose the open rectangle with upper right corner $(C\gamma^{v_i-1}, \frac{1}{\gamma^{v_i-1}})$. Define the box A to be the product if these rectangles, which has volume $\delta = C^k$. By definition, it does not contain any point from one of the $\mathcal{S}_i$. Further, because the vertices $v_1, \ldots v_k$ are pairwise connected, no point $p_{ij}(v_iv_j)$ is contained in P, and thus A does not contain any killing point.

"⇐" Assume there is no k-clique. Let A be a box of volume δ. The intersection with each $\mathcal{S}_i$ is of area exactly C, for otherwise, A had to contain one of the points from $P_i^{\mathrm{scaffold}}(u)$. Thus, each intersection with an $\mathcal{S}_i$ corresponds to a vertex in V. But among k vertices from V, at least two have to be not connected, say u and v. Without loss of generality, assume that u corresponds to a rectangle defined in $\mathcal{S}_u$ and v corresponds to a rectangle defined in $\mathcal{S}_v$. Then A also contains the point $p_{uv}^{\mathrm{kill}}(uv) \in P$, and thus the box cannot be empty.

In order for a box to not contain any points, we thus have to choose a rectangle in at least one of the $\mathcal{S}_i$ whose upper right corner is below $(x, \frac{C}{x\gamma})$. But any such rectangle has area at most C/γ, and thus the total volume of any empty box is at most C^k/γ.

□

This finishes the proof of the following theorem.

Theorem 4. *The problem d-Largest-Empty-Anchor is W[1]-hard.*

3.1.3 An Inapproximability Result

Clearly, as seen in Lemma 5, by setting $\gamma = 2$, we derive that it is already W[1]-hard to get a 2-approximation for the optimization problem. We will now use the freedom we have in choosing γ (and C) in order to prove a much stronger result. Recall that Lemma 5 shows that the fraction between a positive solution and a negative solution is $C^k/(C^k/\gamma) = \gamma$.

We now take advantage of this fact. The largest occurring coordinate, in terms of bit size, is γ^n, which has encoding size $\log \gamma^n$. This means that the total bit size of the reduction is $O(n \log \gamma n^2 k^2)$.

Consequently, when choosing $\gamma = 2^{\mathrm{poly}(n)}$ (and C accordingly) in our reduction, we still have a polynomial time reduction. A graph with a k-clique is then transformed to an instance with an empty anchor of volume C^k, whereas by Lemma 5 in a negative instance, the largest empty anchor is of volume $C^k/\gamma = C^k/2^{\mathrm{poly}(n)}$. That is, we can exponentially blow up the gap between a positive and a negative instance—giving us a very strong inapproximability result, which leads to the following stronger version of Theorem 4.

Theorem 5. *For any polynomially growing function f, the problem d-*Largest-Empty-Anchor *is W[1]-hard to approximate by a factor of $2^{f(|P|)}$.*

3.2 The Star Discrepancy Problem

After having considered the problem where we were not allowed to put *any* points into the box, we now tackle the problem of computing the maximum Lebesgue discrepancy for the range space of anchored boxes, the *star discrepancy*. For a point set P, we set $D^*(P) = \bar{D}_{\mathcal{A}}(P)$. The decision problem parameterized by the dimension is then defined as:

Definition 21. d-Star-Discrepancy
Input: *A set P of points in I^d, and $\delta \in \mathbb{Q}$*
Parameter: d
Question: *Is $D^*(P) \geq \delta$?*

Even though the restriction to anchored boxes seems somewhat artificial, the star discrepancy still is interesting in its own right. Moreover, the star discrepancy and the discrepancy for boxes are closely related: in any fixed dimension, one is bounded by a constant multiple of the other, see for example Matoušek [70, Observation 1.4]. Thus, an algorithm for computing the star discrepancy would give an algorithm for approximating the discrepancy for boxes.

The Star-Discrepancy problem has been shown to be NP-hard by Gnewuch et al. [53]. An exact algorithm that runs in time $O(n^{1+d/2})$ was presented in Dobkin, Edelsbrunner, and Mitchell [37]. Thiémard [93] has given an approximation algorithm that achieves additive error and runs in fpt time with respect to the error and the dimension. However, as Gnewuch, Srivastav, and Winzen [53] also noted, by setting the error to the same order as the optimum (to achieve a constant factor approximation), the running time of any algorithm following Thiémards approach becomes $n^{\Theta(d)}$.

We show that computing the star discrepancy of a point set inside the unit cube is $W[1]$-hard with respect to the dimension.

There are two reasons why the previous reduction does not give us the hardness-result for this problem right away, similar to the obstacles we encountered in Section 2.2:

- First, the maximum discrepancy can be attained by either a large box with few points inside or by *a small box with many points inside*. For example, in our construction from Section 3.1, large point sets lie in a box with affine dimension $d-1$ and thus a volume of 0.
- Second, even if the maximum is attained for a large box, it might still contain *some* points, in which case our construction would fail.

However, we can get rid of both problems by simply choosing the right value for γ, and thus, C^k, the size of the largest empty box.

For a graph G, let N be the total number of points in our construction from the previous section. Recall that $N \in O(k^2n^2)$.

Corollary 3. *If there is a box A with $|N \cdot \mathrm{vol}(A) - |A \cap P|| > N-1$, then A is empty (i.e., $A \cap P = \emptyset$).*

Proof. Assume that A contains at least one point.

If it contains less than N points, the discrepancy can be at most $N-1$: on the one hand, the discrepancy is at most the number of points inside the box, which is $N-1$. On the other hand, the volume is at most 1, and thus the discrepancy is at most $N-1$.

If it contains all N points, then its volume must be 1, as there are points with $x_i = 1$ and $y_i = 1$ for all $1 \leq i \leq k$, and thus its discrepancy is 0. □

To use this lemma, we need to ensure that large empty boxes in a positive instance have higher discrepancy than every box containing points, i.e.,

$$N \cdot C^k = N \cdot \left(\frac{1}{\gamma^{n-1}}\right)^k > N-1.$$

Geometrically, with decreasing γ towards 1, we push the points towards the upper right corner of the unit cube. We thus choose γ so that

$$1 < \gamma < \left(\frac{N}{N-1}\right)^{\frac{1}{k(n-1)}}.$$

To make sure that γ requires only polynomially (in k and n) many bits, observe the following.

Lemma 6. *For $\gamma = 1 + \frac{1}{t}$ with $t = 2knN$, it holds that $\gamma^{k(n-1)} < \frac{N}{N-1}$.*

Proof. Write $\frac{N}{N-1} = \sum_{i=0}^{\infty} \left(\frac{1}{N}\right)^i$. Then

$$\begin{aligned}\gamma^{k(n-1)} &< \gamma^{kn} = \left(1+\frac{1}{t}\right)^{kn} = \sum_{i=0}^{kn} \binom{kn}{i} \frac{1}{t^i} \leq \sum_{i=0}^{kn} (kn)^i \frac{1}{t^i} \leq \sum_{i=0}^{\infty} \left(\frac{kn}{t}\right)^i \\ &< \sum_{i=0}^{\infty} \left(\frac{1}{2N}\right)^i < \sum_{i=0}^{\infty} \left(\frac{1}{N}\right)^i = \frac{N}{N-1}.\end{aligned}$$

□

We now construct the set P with this value for γ. Observe that, as γ has a bit complexity polynomial in the input, the respective value $C = 1/\gamma^{n-1}$ in binary representation is at most n times as long, and thus is still polynomial in the input. We can then prove the main lemma.

Lemma 7. *G has a clique of size k if and only if $D^*(P) = N \cdot C^k$.*

Proof. "⇒" If there is a clique of size k, then by Lemma 5 there is an empty box with volume C^k that does not contain any point. Thus, the discrepancy of that box is $N \cdot C^k$, as desired.

"⇐" By Corollary 3, if there is a box A with discrepancy $N \cdot C^k > N - 1$, it contains no point from P. By Lemma 5, such a box corresponds to a k-clique in G. □

This shows the main theorem of this section.

Theorem 6. *The problem d-Star-Discrepancy is W[1]-hard.*

3.3 The Largest Empty Box Problem

So far, we always assumed that the boxes be anchored, i.e., contain the origin. In what follows, we drop this constraint in order to extend our result to the more natural problem of Largest-Empty-Box and Box-Discrepancy. That means our range space is now $\mathcal{B}$, the set of all axis-parallel boxes inside the unit cube. We start with the case where we have to find a box of large volume not containing any points.

Definition 22. d-Largest-Empty-Box
Input: *A set P of points in I^d, and $\delta \in \mathbb{Q}$*
Parameter: *d*
Question: *Is there a box $B \in \mathcal{B}$ with $B \cap P = \emptyset$ and $\mathrm{vol}(B) \geq \delta$?*

The Largest-Empty-Box problem has been studied extensively in the planar case, see for example Agarwal and Suri [2] and references therein. When the dimension is part of the input, the problem has only recently been shown to be NP-hard by Backer and Keil [9], who also give the fastest exact algorithm, which runs in time $O(n^d \log^{d-2} n)$. Also recently, Dumitrescu and Jiang [40] gave an $O((8ed\varepsilon^{-2})^d \cdot n \log^d n)$-time $(1-\varepsilon)$-approximation algorithm for this problem. Note that, since $(\log n)^k < n + f(k)$ for some $f(k)$, this counts as fpt time in parameters $1/\epsilon$ and d, in contrast to our results for Largest-Empty-Anchor.

What makes the previous construction fail is the following: as the box does not have to contain the origin, now the $\mathcal{S}_i$ cannot be considered separately any more. This kills our construction from the previous section: the open box $(0,1)^{2k}$ does not contain any points from P, but has volume 1.

The plan is to re-establish this dependence, so that we can use the same reasoning as above. For this, we will now put the points closer to the center of the d-dimensional unit cube. This will result in a point set for which similar properties as in Corollary 1 apply.

To this end, we apply a simple trick, which we call *lifting*: from a graph G, we first construct the set P as in Section 3.1 with the constant C satisfying $C^k = 2/3$, and thus $\gamma = (2/3)^{1/k(n-1)}$. Then, we define the function lift: $\mathbb{R}^{2k} \to \mathbb{R}^{2k}$ as follows:

$$\text{lift}(x_1, \ldots, x_{2k}) = (x'_1, \ldots, x'_{2k}) \text{ with } x'_i = \begin{cases} x_i & \text{if } x_i \neq 0 \\ x'_i = 1/2 & \text{otherwise.} \end{cases}$$

Now we apply the function lift to all points in the set P to get a set $P' = \text{lift}(P)$. For a lifted point $x \in P'$, we call the $\mathcal{S}_i$ that the point was lifted from the *corresponding* $\mathcal{S}_i$. This gives the following analog of Observation 1:

Lemma 8. *Any box $B \in \mathcal{B}$ with volume at least $2/3$ contains a point $x \in P'$ if and only if the projection onto the corresponding $\mathcal{S}_i$ contains the projection of x.*

Proof. Recall that the volume of the box is the product of the areas of the corresponding projections onto the $\mathcal{S}_i$. The maximal area of a projection is 1. Thus, as the box has volume at least $2/3$, each of its projections onto any of the $\mathcal{S}_i$ has an area of at least $2/3$. As any rectangle in some $\mathcal{S}_i$ that does not contain the point $(1/2, 1/2)$ has area at most $1/2$, this point is contained in the projection of B onto $\mathcal{S}_i$, for all $1 \leq j \leq k$.

Further, recall that a point x is contained in B if and only if for all $1 \leq i \leq k$ the projection of x onto $\mathcal{S}_i$ is contained in the projection of B onto $\mathcal{S}_i$.

If a point x is lifted from $\mathcal{S}_i$, all coordinates $j \neq i$ are $1/2$. By the previous observation, a box with volume at least $2/3$ thus contains all projections of x onto $\mathcal{S}_j$, for $j \neq i$. This means that if the projection of x onto $\mathcal{S}_i$ is contained in the projection of B onto $\mathcal{S}_i$, then x is contained in B, and vice versa. This finishes the proof. □

Further, any box of volume $2/3$ without loss of generality has its lower left endpoint inside $[0, 1/2)^{2k}$: As all the points of P' lie inside $[1/2, 1]^{2k}$, we can extend any large empty box until its lower left corner is the origin.

After these modifications we can use the same arguments as in the previous sections.

Lemma 9. *G has a clique of size k if and only if there is an empty box B with volume C^k.*

Proof. The claim follows from Lemma 5 and Lemma 8.

□

Theorem 7. *The problem d-Largest-Empty-Box is W[1]-hard.*

3.4 The Box Discrepancy Problem

Our efforts now culminate in showing hardness of computing the discrepancy for axis-parallel boxes $\mathcal{B}$, *box discrepancy* for short. The decision problem parameterized by the dimension, derived from Definition 11, is defined as follows:

Definition 23. d-Box-Discrepancy
Input: *A set P of points in I^d, and $\delta \in \mathbb{Q}$*
Parameter: d
Question: *Is $\bar{D}_{\mathcal{B}}(P) \geq \delta$?*

For the box discrepancy, no hardness results were known so far. Here, we will establish W[1]-hardness with respect to the dimension, which also implies NP-hardness of the problem, as our reduction is polynomial in both n and k.

In order for our proof to work for this case, we will combine the ideas of the previous sections. To this end, we start with the construction from Section 3.2. Recall that in this construction, we shrunk the point set to lie close to the upper right corner of the unit cube, so that there always will be a huge empty box. To this point set, we now apply the lifting transform from Section 3.3. This ensures that for any such box, whether or not a point is contained in the box will only depend on the projection of the box onto the corresponding $\mathcal{S}_i$, as stated by Lemma 8. This completes the construction.

Lemma 10. *G has a clique of size k if and only if $\bar{D}_{\mathcal{B}}(P) \geq NC^k$.*

Proof. "⇒" If G has a clique of size k, we can pick a box according to Lemma 9, which is empty and has volume C^k. Thus, the discrepancy is NC^k.

"⇐" By the choice of C, and by the same arguments as in Corollary 3, a box with discrepancy NC^k cannot contain any points from P. By Lemma 9, such a box corresponds to a k-clique in G. □

This shows the main theorem for the d-Box-Discrepancy problem.

Theorem 8. *The problem d-Box-Discrepancy is W[1]-hard.*

3.5 General Position

Along the lines of Section 2.3, we now show that the constructions can be put in general position, while maintaining the main properties. To this end, we simply perturb each point by a random value of at most α in each direction. We have to be careful to not push any points outside the unit cube, and in this case simply perturb them towards the inside. As in Section 2.3, for a set of points P, we denote such a set as an α-perturbation of P.

As an analog of Lemma 3, we claim that the size of a largest empty box does not change too much if α is small.

Lemma 11. *Let P be a set of n points in I^d, and let $\tilde{P}$ denote an α-perturbation of P. Let B be a largest empty box inside I^d with respect to P, and let B' denote the size of a largest empty box inside I^d for $\tilde{P}$. Then*

$$\mathrm{vol}(B') - 2d\alpha \leq \mathrm{vol}(B) \leq vol(B') + 2d\alpha.$$

Proof. Let B be a largest empty box for P, $B := \Pi_{i=1}^{d}(x_i, y_i)$. Define

$$B' := \Pi_{i=1}^{d}(x_i + \alpha, y_i - \alpha).$$

Clearly, B' has volume at least $\mathrm{vol}(B) - 2d\alpha$. Further, as every point from P was perturbed by at most α in each direction, and B did not contain any points, B' also does not contain any points.

The reverse inequality then follows from interchanging the roles of P and $\tilde{P}$. □

We can now use this lemma to show that the problems considered in Chapter 3 are also hard for point sets in general position. For each of the reductions, the absolute difference for the largest empty box between a positive and a negative instance was at least some value α^*, with a bit complexity polynomially bounded in the size of the input. Thus, for an instance in $2d$ dimensions, choosing a random perturbation with $2d\alpha < \alpha^*/2$ for each of the points creates a point set in general position. Further, by Lemma 11, the gap between a positive and a negative instance instance is still at least $\alpha^* - 2 \cdot 2d\alpha > 0$.

From this, we derive the following theorem:

Theorem 9. *Theorems 4, 6, 7, and 8 also hold if we require the point sets to be in general position position.*

Chapter 4

Conclusion

Finally, we summarize our results and state the hardness results with respect to several other range spaces. We also elaborate the connection to a recent result on (relative) computational lower bounds and state the necessary limitations for our parameterized hardness proofs, which stem from discrepancy theory itself. As an application, we discuss the relation to the problem of verification of ε-nets.

4.1 Other Geometric Range Spaces

In the previous sections, we have considered as ranges only axis-parallel boxes, sometimes required to contain the origin. Similar questions can of course be asked when the ranges are determined by other (geometric) objects, and, perhaps not surprisingly, many of these will lead to similar hardness results. To not bore the reader, we will not present all variants of continuous/combinatorial discrepancy and all possible ranges. As the volume formulas are rather tedious to elaborate, we concentrate on the discrete cases where the ranges are given by the set of all half-spaces. We denote the corresponding decision problems parameterized by the dimension as d-Bichromatic-Halfspace and d-Halfspace-Discrepancy.

Figure 4.1 shows the construction for half-spaces in a single $\mathcal{S}_i$. The killing points are then placed on the *segments* between two points to be blocked, i.e.,

$$p_{ij}^{\text{kill}}(uv) = \frac{1}{2}\left(b_i(u) + b_j(v)\right).$$

Now clearly any half-space that does not contain any red points but at least one blue point has to contain the origin, as can be seen from Figure 4.1. Further, every half-space can contain at most one blue point in each $\mathcal{S}_i$, and by the definition of the killing points, two points can only be chosen if the corresponding vertices are connected in the graph.

By following Lemma 1, we can thus derive the following:

Lemma 12. *There is a half-space that contains k blue points and no red points if and only if there is a k-clique in G.*

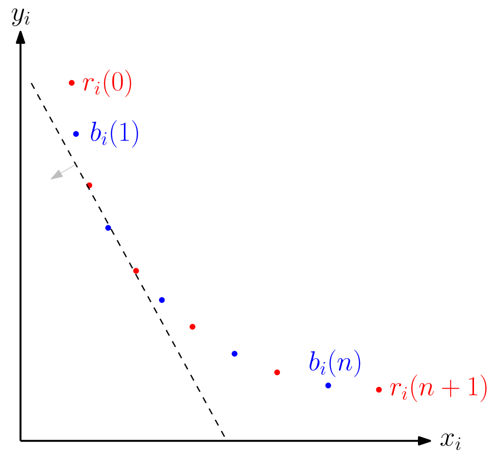


Figure 4.1: The construction for half-spaces in a single $\mathcal{S}_i$

To adapt the proof to the case of discrepancy, let P_r denote the set of red points in the construction. Again, by adding $|P_r|$ points close to the origin, analog to Section 2.2, we can derive the same result for the combinatorial discrepancy with respect to half-spaces.

Lemma 13. *There is a half-space with discrepancy $|P_r|+k$ if and only if there is a k-clique in G.*

This shows that the problems are also hard with respect to half-spaces as ranges.

Theorem 10. *The problems* d-BICHROMATIC-HALFSPACE *and* d-HALFSPACE-DISCREPANCY *are W[1]-hard.*

It is easily seen that the same construction also works for simplices, as the intersection of a half-space with the positive orthant defines a simplex.

We are positive that simple adaptations of the construction lead to similar results for other ranges. In the case of convex sets, this is not worth the effort however: As we will see in Part III, for this case we are able to show a significant strengthening of the result.

4.2 Why Only One Parameter?

For the BICHROMATIC-RECTANGLE problem we have shown that it is W[1]-hard with respect to both the dimension d and the size of the solution k. This, under standard complexity theoretic assumptions, excludes an algorithm with a running time of $O\left(f(d,k)n^c\right)$, for constant c. A similar result was obtained for the problem of MAXIMUM-EMPTY-ANCHOR. This raises the question whether we can strengthen the result for the other continuous problems as well. Can we exclude algorithms with a running time of $O\left(f(d,\delta)n^c\right)$ for STAR-DISCREPANCY and BOX-DISCREPANCY?

The answer is no, and the reason for this is found in discrepancy theory itself: It is known that *every* set of n points in $[0,1]^d$ has box discrepancy at least $\Omega(\log^{(d-1)/2} n)$, see

[70, Chapter 6], and by the aforementioned relation to star discrepancy, this also holds for the latter. This leads to a trivial fpt algorithm as follows.

Let P, $n := |P|$, be a set of points and $\delta > 0$. We want to decide whether the red-blue discrepancy of P is at least δ.

If $\delta < \log n$, we can simply accept the answer—by the combinatorial lower bound, any set of size n has discrepancy at least $\log n$, so our answer is correct. If $\delta \geq \log n$, we solve the problem brute force, which takes at most $n^d \leq \left(2^\delta\right)^d \in O\left(f(d,\delta)n\right)$ time.

In any case, the running time is bounded by a function f that depends on δ (and maybe d), but not on n. Thus, the resulting algorithm indeed runs in fpt time.

This is a phenomenon common to Ramsey-type problems in parameterized complexity, which has led to the study of so called small kernels for parameterized problems, see Kratsch [66] for a recent result. We will discover a similar result in Section 11.2: because of a theorem that *guarantees* the existence of certain sets, we immediately get a trivial fixed-parameter tractable algorithm for *finding* such sets.

4.3 A Slightly Stronger Formulation of the Results

By proving that the problems considered are W[1]-hard with respect to the dimension d, we have shown that they most likely cannot be solved in time $O\left(f(d)n^c\right)$ for *any* computable function f. However, this does not exclude the possibility of algorithms whose running time is bounded by, say, $O(n^{\log d})$.

Thus, we now present a stronger formulation of the results, by combining W[1]-hardness of a problem with recent results on (relative) computational lower bounds presented by Chen et al. [28]. In Theorem 5.3 they state that, unless Sat can be solved in time $2^{o(n)}$, k-Clique cannot be solved in time $f(k)n^{o(k)}$. The former, that Sat on n variables cannot be solved in time $2^{o(n)}$, is the Exponential Time Hypothesis (ETH) and is considered highly plausible in computational complexity.

We can use this in the following way for our problems: as in our reductions, the dimension is kept linear in the parameter k, an $f(d)n^{o(d)}$ algorithm for one of our problems would admit an $f(k)n^{o(k)}$ algorithm for k-clique. This in turn would lead to a $2^{o(n)}$ algorithm for Sat, contradicting the ETH.

Thus, we can state the following stronger theorem:

Theorem 11. *The problems considered in Chapters 2 and 3 cannot be solved in time $f(d)n^{o(d)}$, where d is the dimension and n is the number of points, unless the Exponential Time Hypothesis is false.*

For the problem of finding the largest monochromatic half-space, Aronov and Har-Peled [7] give an $(1-\varepsilon)$ approximation that runs in time $O(n^{\lfloor d/2\rfloor}(\varepsilon^{-2}\log n)^{\lceil d/2\rceil})$. As we have shown that the problem is W[1]-hard with respect to both the size of the solution and the dimension (see Sections 2.1 and 4.1), and as both parameters stay linear in the reduction, this implies that it is not possible to lower the running time significantly by lowering the exponent of n to, say, $\sqrt{d}$—if we believe in the ETH, it has to stay linear in d.

4.4 Implications on Verification of ε-Nets

As mentioned in the introduction, it is known that random samples of size $O(d/\varepsilon \log 1/\varepsilon)$ are ε-nets with high probability in any fixed dimension. Thus, it would be nice to have an algorithm that, given a set of points P and a sample $S \subseteq P$, decides whether S is indeed an ε-net. That is, we want to solve the following problem.

Definition 24. d-$\mathcal{R}$-Epsilon-Net-Verification
Input: *A set P in $\mathbb{R}^d$, a set $S \subseteq P$, and $\varepsilon > 0$*
Parameter: *d*
Question: *Is S an ε-net for P with respect to $\mathcal{R}$?*

The relation between d-Bichromatic-Rectangle (see Section 2.1) and ε-net verification problems is intuitively clear: if there is a large monochromatic set, then the complementary color class will *not* form an ε-net for the entire set, and vice versa.

To state the hardness result, recall that for a class of problems $\mathcal{C}$, co-$\mathcal{C}$ denotes the problems whose complements are in $\mathcal{C}$. Now we take as a starting point the construction from Section 2.1. Let $P = P(G, k) = P_r(G, k) \cup P_b(G, k)$ be the set of all points and $S := P_r(G, k)$ be the set of red points. Further, let $\varepsilon := (k + 1)/|P|$. From Lemma 1, we then read off the following.

Lemma 14. *S is* not *an ε-net for P with respect to boxes if and only if G has a k-clique.*

Proof. If G has a k-clique, then there is a set containing $k + 1 = \varepsilon|P|$ blue points and no red point. Thus, S is not an ε-net for P.

On the other hand, if G has no k-clique, then there is no box with $k + 1$ points which does not contain any red point. Thus, the set of red points intersects all rectangles that contain at least $k + 1$ blue points. This means that S is an ε-net for P. □

This shows that the problem is also hard with respect to the dimension.

Theorem 12. *The problem d-*Box-Epsilon-Net-Verification *is co-W[1]-hard.*

In order to strengthen this result, we show that the problem is even W[1]-hard for *constant* ε. To this end, we now replace the single blue point at the origin by α blue points close by, for

$$\alpha = |P| - 2k.$$

This creates a set P' with a total of $|P'| = 2|P| - 2k$ points. Further, we set $\varepsilon := 1/2$. Then, analog to Lemma 1, a box with

$$\alpha + k = \frac{|P| - k}{|P'|}|P'| = \frac{1}{2}|P'| = \varepsilon|P'|$$

blue points and no red points corresponds to a k-clique in the input graph G.

This shows that the problem is W[1]-hard with respect to the dimension even for constant ε. In the light of Theorem 11, we can formulate this in a stronger form:

Theorem 13. *The problem d-Box-Epsilon-Net-Verification for a set of n points in $\mathbb{R}^d$ cannot be solved in time $f(d, 1/\varepsilon)n^{g(1/\varepsilon)o(d)}$ for any computable functions f and g, unless the Exponential Time Hypothesis is false.*

In order to strengthen this result even further, it remains is to find a reduction where both $1/\varepsilon$ as well as the size of the sample are bounded by a function in d only. Apart from this, the problem of verifying ε-approximations is still open. In order to use our construction from Section 2.2, we somehow have to manage to show hardness for the *relative* combinatorial discrepancy. This means that we have to ensure that there are no boxes with a lot more red than blue point, and at the moment, we do not see how to do this.

4.5 Outlook

We have shown that many problems related to computing the discrepancy of point sets in d dimensions are W[1]-hard for the most common range spaces. By combining our hardness reductions with deep results on computational lower bounds, our reductions show that all these problems most likely cannot be solved in time $f(d)n^{o(d)}$.

As an application, we have shown that for a set of n points in $\mathbb{R}^d$ it is not possible to check in time $f(d, 1/\varepsilon)n^{g(1/\varepsilon)o(d)}$ whether a given set is an ε-net with respect to another set, for an implicitly given set of ranges such as boxes, half-spaces, and simplices.

The main open problem thus is the (in-)approximability of Box-Discrepancy and Star-Discrepancy. More specifically, given a point set P in I^d, can we find a box that has discrepancy at least $\alpha \cdot \mathrm{OPT}$ in time $O(f(d)n^c)$, for some constant $\alpha > 0$? Also, can we show that Red-Blue-Discrepancy is W[1]-hard with respect to both parameters, the dimension d and the discrepancy δ?

Part II

Ham-Sandwich Cuts and the Power of d-Sum Reductions

Chapter 5

Introduction and Motivation

In this part, we investigate the complexity of several decision problems in higher dimensions with respect to the d-Sum problem. We will see how to "embed" instances of the latter into $\mathbb{R}^{d+1}$. Depending on the problem we investigate, solutions to the d-Sum problem will correspond to certain geometric objects in $\mathbb{R}^{d+1}$. These include small size Helly sets, small size Carathéodory sets, linear ham-sandwich cuts, and points of high Tverberg depth.

To this end, we develop a technique similar to Erickson [46] who shows that detecting affine degeneracies (i.e., $d+2$ points on a hyperplane in $\mathbb{R}^d$) is d-Sum-hard, and that testing convex hull simplicity in $\mathbb{R}^d$ is $\lfloor d/2 \rfloor$-Sum-hard. Our technique has the additional advantage of producing highly symmetric point sets, allowing us to prove results not achievable with the old technique.

5.1 Some Discrete Geometry

Basic problems

Many basic theorems from combinatorial geometry are of the following type: If a set of n objects in $\mathbb{R}^d$ has a certain property, then there is already a subset of size $d+1$ that has this property. Two examples of this are Caratheodory's theorem [18] and Helly's theorem [58] (see also Matoušek [68] or Ziegler [97]).

Theorem (Carathéodory's theorem)**.** *Let P be a set of points in $\mathbb{R}^d$ with $\mathbf{0} \in \operatorname{conv}(P)$. Then there is a set $Q \subseteq P$ of size at most $d+1$ with $\mathbf{0} \in \operatorname{conv}(Q)$.*

We will call a minimal set containing $\mathbf{0}$ in the convex hull a *Carathéodory set*. Another related theorem is the following.

Theorem (Helly's theorem)**.** *Let C be a set of convex objects in $\mathbb{R}^d$ such that any $d+1$ of them intersect. Then they all intersect in at least one point.*

To see the close relation to Carathéodory's theorem, we can state it as the contrapositive.

Theorem (Helly's theorem, alternative formulation). *Let C be a set of convex objects in $\mathbb{R}^d$ with empty intersection. Then there is already a set $C' \subseteq C$ of size at most $d+1$ with empty intersection.*

For all these theorems, several computational problems come to mind. Computing a point in the common intersection of convex sets can be done by linear programming, if the sets are given by linear inequalities, and with the same method, one can decide whether their common intersection is empty. Further, given a set of n points in $\mathbb{R}^d$ that contain the origin in their convex hull, one can find a set of at most $d+1$ points containing the origin in their convex hull by Gaussian elimination (cf. Part IV). Both can be done in (weakly) polynomial time.

The variants that we are considering here are related to the second problem: can we compute a set of points of *minimum cardinality* that contains the origin in its convex hull? And: can we decide whether, for a given set of convex objects in $\mathbb{R}^d$ with empty intersection, there are already d with an empty intersection? It turns out that both problems are equivalent to computing affine degeneracies of a point set.

Ham-sandwich cuts

A problem of a different spirit is the so called ham-sandwich cut problem for point sets in $\mathbb{R}^d$. Let h^+ and h^- denote the positive and negative open half-space defined by a hyperplane, respectively. Further, for a set of points P, let h_P^+ denote the points of P that lie strictly on the positive side of h, and analogously h_P^-. A hyperplane h is then said to *bisect* a set P if $|h_P^+| \leq \left\lfloor \frac{|P|}{2} \right\rfloor$ and $|h_P^-| \leq \left\lfloor \frac{|P|}{2} \right\rfloor$, i.e., if it does not contain more than half of the points in either open half-space. A *ham-sandwich cut* for d point sets $P_1, \ldots, P_d$ in $\mathbb{R}^d$ is then a hyperplane h that bisects each of the sets simultaneously. In particular, if the number of points in each set is odd, the hyperplane has to pass through at least one of the points from each set.

The ham-sandwich theorem states that such a cut always exists.

Theorem (Ham-sandwich theorem). *Let $P_1, \ldots P_d$ be point sets in $\mathbb{R}^d$. Then there is a hyperplane that bisects each of the sets simultaneously.*

The proof is a simple application of the Borsuk-Ulam theorem (see, e.g., Matoušek [69]). Computing a ham-sandwich cut efficiently is an important problem and has been studied extensively (see Edelsbrunner and Waupotitsch [43], Matoušek, Lo, and Steiger [71], Yu [96]). For general dimension, the fastest known algorithm [71] runs in time roughly $O\left(n^{d-1}\right)$.

For well-separated sets, i.e., set systems where each subset of the sets can be separated from the remaining sets by a hyperplane, better algorithms are known. Based on a result by Bárány, Hubard, and Jéronimo [10] and Steiger and Zhao [91], Bereg [12] showed that in this case ham-sandwich cuts can be computed in linear time in any *fixed* dimension. This is a big improvement over the general case. However, the method does not generalize to arbitrary point sets.

Particularly in the light of recent developments related to the Polynomial Partitioning Technique, it is an important open problem whether ham-sandwich cuts can be computed in time polynomial in both the number of points as well as the dimension, see Guth and Katz [55] and Kaplan, Matoušek, and Sharir [62]. A more modest goal would be to give an fpt time algorithm for the problem, i.e., to give an algorithm that finds a cut in time $f(d)n^c$. Of course, here also approximate solutions (apart from the standard approach using ε-approximations) are of interest.

The ham-sandwich problem is not a decision problem, as, given an instance, we know that there always exists a solution. However, there are many natural ways to state it as such. The variant we are going to consider in this part is the following: how hard is it to decide whether there is a ham-sandwich cut through a given point? Or: how hard is it to decide whether there is a ham-sandwich cut through $d+1$ given sets in $\mathbb{R}^d$?

Tverberg depth

Yet another unrelated problem that we can tackle with our technique is derived from the following theorem.

Theorem (Tverberg's theorem)**.** *Any set $P \subseteq \mathbb{R}^d$ with $n = (r-1)(d+1)+1$ points can be partitioned into r sets $P_1, \ldots, P_r$ such that $\bigcap_{i=1}^r \operatorname{conv}(P_i) \neq \emptyset$.*

We will not describe this in any more detail, but instead refer the reader to Part IV, where a more concise introduction to this and related problems is given. Here, the decision problem we derive from this is, for example, the following: given a point set P and a point q, can we partition P into r sets such that each one contains q in its convex hull?

5.2 3-Sum and d-Sum

3-Sum is the following problem: Given sets a set of numbers S, do any three of them sum up to zero? Intuitively, this problem should take $\Omega(n^2)$ time, and this can be used to show the suspected lower bound for other problems as well. As Gajentaan and Overmars [50] state in their seminal paper, many geometric problems for which the fastest known algorithm runs in time $\Omega(n^2)$ turn out to be 3-Sum-hard, which justifies this lower bound on the running time. The reasoning goes as follows: if one can reduce an instance of 3-Sum to a linear size instance of a problem L in subquadratic time, then a subquadratic algorithm for L implies a subquadratic algorithm for 3-Sum. Consequently, L should take time $\Omega(n^2)$ as well. Showing 3-Sum-hardness for geometric problems is considered a routine task today and has been applied to many low dimensional problems.

The generalization of the problem is d-Sum, the parameterized version of the NP-complete Subset-Sum problem:

Definition 25. d-Sum
Input: *A set of integers S, and $d \in \mathbb{N}$*

Parameter: d
Question: *Is there a set $S' \subseteq S$ of d numbers such that $\sum_{s\in S'} s = 0$?*

As a generalization of 3-Sum, it is believed that this problem should require time $\Omega(n^{\lceil d/2\rceil}\text{polylog}(n))$. Moreover, recent results by Pătraşcu and Williams [86] imply that d-Sum should require $n^{\Omega(d)}$ time, if the Exponential Time Hypothesis holds (cf. Section 4.3). Their paper also shows, without explicitly mentioning it, that the problem d-Sum is W[1]-hard with respect to d. While this was known before (see Fellows and Koblitz [48]), from their reduction one can derive a stronger relative lower bound. This remains true if the numbers are allowed to be chosen more than once in a solution, i.e., if we are allowed to pick a multi-set of size d of the numbers in S.

Using d-Sum as a base problem for reductions to problems in $\mathbb{R}^d$ is a very natural approach. Surprisingly—apart from Erickson's work—this technique has not been used to show W[1]-hardness of other geometric problems in $\mathbb{R}^d$.

5.3 Our Contribution

We will first present our technique for several simple problems (and their duals) from combinatorial geometry and combinatorial optimization mentioned in the introduction. These can (and some are) also be shown to be hard by Erickson's technique [46], but we still find it instructive to present them here because they will make it easier to understand the proofs in later chapters. Also, many of them have not been explicitly stated. The problems we are considering in Chapter 6 are all very similar in spirit. They are related to affine degeneracy detection, computing minimum size sets containing the origin in the convex hull, and computing minimal sets of half-spaces having an empty intersection. We show that all these problems are d-Sum hard in $\mathbb{R}^{d+1}$.

In Chapter 7, we then use the full strength of our method to investigate the complexity of certain decision problems related to the ham-sandwich theorem and Tverberg's theorem. Among others, we show that deciding whether $d+2$ sets in $\mathbb{R}^{d+1}$ can simultaneously be bisected by a common hyperplane is d-Sum hard. We then modify the construction to show similar hardness results for computing the Tverberg depth of a given point, or a given point set.

As the d-Sum problem is W[1]-hard, all the results will be stated in a parameterized complexity sense, in that they are W[1]-hard with respect to the dimension, excluding algorithms with a running time of $O(f(d)n^c)$. Using the result by Pătraşcu and Williams [86], in the conclusion we are able to strengthen these results to yield even better relative lower bounds for the problems.

Chapter 6

Basic Problems

In this chapter, we illustrate our reduction technique on the basic examples mentioned in the introduction. Not until later, in Chapter 7, will we need the full strength of our new technique.

6.1 The Affine Containment Problem

We start with a problem for which the hardness proof is the most straightforward. This proof will subsequently be modified to show the main theorems. The following problem and its different guises are variations of the generic Affine-Degeneracy problem.

A point $x \in \mathbb{R}^d$ is said to be in the affine hull of $\{p_1, \ldots, p_n\} \subset \mathbb{R}^d$, if there exist $\alpha_1, \ldots, \alpha_n$ such that $\sum \alpha_i = 1$ and $\sum \alpha_i p_i = x$.

Definition 26. d-Affine-Containment
Input: *A set of points P in $\mathbb{R}^d$*
Parameter: d
Question: *Is the origin contained in the affine hull of any d points?*

We will now show how to reduce the d-Sum problem to the $(d+1)$-Affine-Containment problem by a polynomial time parameterized reduction. To this end, for a given set $S = \{s_1, \ldots, s_n\}$, we will create a point-set in $\mathbb{R}^{d+1}$ in which $d+1$ points span an affine plane through the origin if and only d of these numbers sum up to 0.

Let e_i denote the i-th unit vector. Set

$$p_i^j := \frac{1}{s_i} e_j + e_{d+1} = \left(0, \ldots, \frac{1}{s_i}, \ldots, 0, 1\right)^T \in \mathbb{R}^{d+1}$$

and $q := -\sum_{i=1}^{d} e_i$.

We then define our point set $P \subset \mathbb{R}^{d+1}$ as

$$P := \left\{ p_i^j \mid 1 \le j \le d, 1 \le i \le n \right\} \cup \{q\}.$$

The size of the point set is $nd + 1$, and clearly all coordinates are polynomially bounded in the input. Further, for this set P, the following lemma holds, which appears in all subsequent constructions with slight variations, depending on specific characteristics the of the construction.

Lemma 15. *There are d elements that sum up to 0 if and only if there are $d+1$ points in $P \subset \mathbb{R}^{d+1}$ whose affine hull contains the origin.*

Proof. "$\Rightarrow$" Let $\sum_{j=1}^{d} s_{i_j} = 0$. We choose points $z_j := p_{i_j}^j$, $1 \le j \le d$ and $z_{d+1} := q$. Let $\alpha_j := s_{i_j}$ and $\alpha_{d+1} := 1$. Then

$$\sum_{j=1}^{d+1} \alpha_j z_j = \sum_{j=1}^{d} s_{i_j} p_{i_j}^j + q = \sum_{j=1}^{d} e_j + \left(\sum_{j=1}^{d} s_{i_j} \right) e_{d+1} - \sum_{j=1}^{d} e_j = \mathbf{0}$$

and

$$\sum_{j=1}^{d+1} \alpha_j = \sum_{j=1}^{d} s_{i_j} + \alpha_{d+1} = 1.$$

That means that $\mathbf{0}$ is in $\operatorname{aff}\left(\{p_{i_1}^1, \dots p_{i_d}^d, q\}\right) = \operatorname{aff}(\{z_1, \dots, z_{d+1}\})$.

"$\Leftarrow$" Let $\mathbf{0} \in \operatorname{aff}(\{z_1, \dots, z_{d+1}\})$, with $\sum_{j=1}^{d+1} \alpha_j z_j = \mathbf{0}$ and $\sum \alpha_j = 1$. As all points but q lie on the hyperplane $x_{d+1} = 1$, one of the points, without loss of generality z_{d+1}, is q. For a point p, let $(p)_j := \operatorname{pr}_j(p)$ denote its j-th coordinate. Because $(q)_{d+1} = 0$, and $(z)_{d+1} = 1$ for all $z \neq q$, by computing the $(d+1)$-st coordinate we get

$$0 = \sum_{j=1}^{d} (\alpha_j z_j)_{d+1} = \sum_{j=1}^{d} \alpha_j (z_j)_{d+1} = \sum_{j=1}^{d} \alpha_j \tag{6.1}$$

and thus $\alpha_{d+1} = 1 - \sum_{j=1}^{d} \alpha_j = 1$.

Further, as $\sum_{j=1}^{d+1} \alpha_j z_j = \mathbf{0}$, it holds that

$$\sum_{j=1}^{d} \alpha_j z_j = -\alpha_{d+1} q = \sum_{j=1}^{d} e_j.$$

Every z_j is non-zero for only one other coordinate except the $(d+1)$-st, and as $(q)_j = -1$ for all $j < d+1$, for each j there is at least one point that is nonzero at coordinate j (in particular, also $\alpha_j \neq 0$). Thus, there are exactly d such points. Without loss of generality assume that z_j is the point that has nonzero j-th coordinate, so $(z_j)_j = \frac{1}{s_{i_j}}$ for some i_j. This means that $\alpha_j \frac{1}{s_{i_j}} - 1 = 0$, and thus $\alpha_j = s_{i_j} \in S$, which implies (Equation 6.1) that we have d elements in S summing up to 0, as desired. □

This completes the proof of the following theorem.

Theorem 14. *The problem d-AFFINE-CONTAINMENT is W[1]-hard.*

6.2 The Carathéodory Set Problem

As seen in the introduction, Carathéodory's theorem states that whenever the origin is contained in the convex hull of a set $P \subset \mathbb{R}^d$, it is already contained in the convex hull of a subset $P' \subseteq P'$ of size $d+1$. Thus, when given a set P, we can easily decide whether the origin is contained in a subset of size $d+1$ by simply testing it for the entire set. But what if we want to know whether it is already contained in the convex hull of only d points? This is a much harder harder problem, as we will see now.

To this end, we define the decision problem parameterized by the dimension as follows.

Definition 27. d-Carathéodory
Input: *A set of points P in $\mathbb{R}^d$ containing the origin in their convex hull*
Parameter: *d*
Question: *Is the origin contained in the convex hull of any d points?*

This looks very similar to the d-Affine-Containment problem, and indeed our previous construction almost works. We have to modify it such that all coefficients can be chosen positive to replace the notion of an affine hull by that of a convex hull.

To this end, we set

$$p_i^j := \frac{1}{|s_i|}e_j + \operatorname{sign}(s_i)e_{d+1} \in \mathbb{R}^{d+1}$$

where

$$\operatorname{sign}(x) := \begin{cases} 1 & x \geq 0 \\ -1 & x < 0. \end{cases}$$

Let q be defined as above. The set P then consists of all the points $p_i^j, 1 \leq j \leq d, 1 \leq i \leq n$ and q.

Lemma 16. *There are d elements in S that sum up to 0 if and only if the origin lies in the convex hull of $d+1$ points of P.*

Proof. "$\Rightarrow$" Observe that $\mathbf{0} \in \operatorname{conv}(P)$ if and only if $\mathbf{0} = \sum_{p\in P} \alpha_p p$ for any $\alpha_p \geq 0$, $\sum \alpha_p > 0$. Let $\sum_{j=1}^d s_{i_j} = 0$. Setting $\alpha_j := |s_{i_j}| > 0$, $z_j := p_{i_j}^j$ for $1 \leq j \leq d$, and $\alpha_{d+1} := 1$, $z_{d+1} := q$ yields

$$\sum_{j=1}^{d+1} \alpha_j z_j = \sum_{j=1}^{d} |s_{i_j}| p_{i_j}^j + q = \sum_{j=1}^{d} e_j + \left(\sum_{j=1}^{d} \operatorname{sign}(s_{i_j})|s_{i_j}|\right) e_{d+1} - \sum_{j=1}^{d} e_j = \mathbf{0}.$$

"$\Leftarrow$" Let $\sum_{j=1}^{d+1} \alpha_j z_j = \mathbf{0}, \alpha_j \geq 0$. The point q is one of the points of the convex combination, as all other points lie in the positive half-space $\sum_{j=1}^d x_j > 0$. We can assume $z_{d+1} = q$ and $\alpha_{d+1} = 1$. Further, by the same argument as in Lemma 15, there are at least d other points for the total sum to become $\mathbf{0}$. Again, without loss of generality let

$(z_j)_j \neq 0$. As $(q)_j = -1$ for all $1 \leq j \leq d$, this means that $\alpha_j \frac{1}{|s_{i_j}|} = 1$ for some i_j, and thus $\alpha_j = |s_{i_j}|$. Further, because of the $(d+1)$-st coordinate, we get

$$0 = \sum_{j=1}^{d} \alpha_j \mathrm{sign}(s_{i_j}) = \sum_{j=1}^{d} \mathrm{sign}(s_{i_j}) \left|s_{i_j}\right| = \sum_{j=1}^{d} s_{i_j}$$

and thus we have d elements summing up to 0. □

This implies the following theorem.

Theorem 15. *The problem d-Carathéodory is W[1]-hard.*

6.3 The Helly Set Problem

Starting from the result in the previous section, we now show how to prove hardness for the problem of computing Helly sets of minimum size.

Definition 28. d-Helly
Input: *A set of convex sets $\mathcal{C}$ in $\mathbb{R}^d$ with an empty intersection*
Parameter: *d*
Question: *Do any d sets from $\mathcal{C}$ have an empty intersection?*

Observe that, by Helly's theorem, the question is again easy to answer if we ask for $d+1$ convex sets.

Proving the hardness of this problem is just a standard application of a duality transform; a similar application (which is used to prove Carathéodory's theorem from Helly's theorem) can be found in [44, Chapter 2.3]. For a given set P in $\mathbb{R}^d$, we will construct a set of convex sets (that are actually half-spaces) such that d of them have an empty intersection if and only if there are d points in P that contain the origin in their convex hull.

Consider a set P of points $p_1, \ldots, p_n \in \mathbb{R}^d$ whose convex hull contains the origin. For each point $p \in P$, define the half-space

$$p^* := \left\{ x \mid p^T x \geq 1 \right\}.$$

Define P^* to be the set of all these half-spaces corresponding to the points in P. We show that any Carathéodory set of P (for the origin) corresponds to a Helly set (a set of half-spaces with empty intersection) of P^* of the same size. Since checking if the minimum Carathéodory set has cardinality at most d is W[1]-hard, it then follows that checking if the minimum Helly set is of cardinality at most d is also $W[1]$-hard.

Let $Q \subseteq P$ and let V be a $d \times |Q|$ matrix whose columns represent the vectors in Q. Further, let cone(Q) denote the conic hull of the vectors, i.e., the set $\left\{ \sum_{q \in Q} \alpha_q q \mid \alpha_q \geq 0 \right\}$. A cone is called *pointed* if it does not contain a line.

We can now show the main lemma of this section, which is a variant of Gordan's Theorem, see Dantzig and Thapa [34, Theorem 2.13]:

Lemma 17. *Let Q be a set of points in $\mathbb{R}^d$ and let V be a $d \times |Q|$ matrix whose columns represent the vectors in Q. Then $\mathbf{0} \in \operatorname{conv}(Q)$ if and only if the system of inequalities $V^T x \geq \mathbf{1}$ is infeasible.*

Proof. Observe that that cone(Q) is pointed if and only if $V^T x \leq \mathbf{0}$ is a full-dimensional cone.

"⇒" Suppose that $V^T x \geq \mathbf{1}$ is feasible. Then there exists a vector $\alpha \in \mathbb{R}^d$ such that $V^T \alpha \leq -\mathbf{1}$. That is, $V^T \alpha < \mathbf{0}$ and thus $V^T x \leq \mathbf{0}$ is a full-dimensional cone. Therefore, cone(Q) is pointed. But this means that $\mathbf{0} \notin \operatorname{conv}(Q)$.

"⇐" Now suppose $\mathbf{0} \notin \operatorname{conv}(Q)$, then cone($Q$) is pointed and therefore $V^T x \leq \mathbf{0}$ is a full-dimensional cone. Thus, there exists $\alpha \in \mathbb{R}^d$ such that $V^T \alpha < \mathbf{0}$, and so for a large enough $\lambda > 0$, it holds that $V^T(-\lambda\alpha) > \mathbf{1}$. Hence $V^T x \geq \mathbf{1}$ is feasible. □

Thus, any set $Q \subseteq P$ of points whose convex hull contains the origin corresponds to a set $Q^* \subseteq P^*$ of convex sets (inequalities) of the same size that has an empty intersection, and vice versa. This finishes the proof of the following theorem.

Theorem 16. *The problem d-HELLY is W[1]-hard.*

6.4 The Minimum Infeasible Subsystem Problem

These decision versions of Carathéodory's and Helly's theorem have not explicitly been considered in the literature so far. This is quite surprising, as they are interesting to people from computational as well as discrete geometry. However, similar problems arise in the context of linear programming, most notably the following:

Definition 29. d-MIN-IRREDUCIBLE-INFEASIBLE-SUBSYSTEM
Input: *An infeasible linear program in $\mathbb{R}^d$ with n constraints*
Parameter: *d*
Question: *Is there a subset of d constraints that is infeasible?*

The problem d-MIN-IIS has been studied before, mainly because of its connection to the NP-complete MAXIMUM-FEASIBLE-SUBSYSTEM problem, where one is given an infeasible linear program and one has to find a feasible subsets of constraints of maximum size. Amaldi and Kann [5] show that d-MIN-IIS is NP-hard by a reduction from DOMINATING-SET. However, the dimension depends on the size of the graph, so it does not reveal anything with respect to the parameter d. Further, based on this result, Amaldi, Pfetsch, and Trotter [6] show that the problem is even hard to approximate by a factor $O(2^{\log^{1-\varepsilon} n})$, where n is the number of constraints. Still, it is now clear how this implies anything about the parameterized complexity of the problem. However, as the convex sets created in our hardness proof for d-HELLY are even half-spaces, we can derive the analog result regarding the problem d-MIN-IIS.

Corollary 4. *The problem d-MIN-IIS is W[1]-hard.*

Chapter 7

Advanced Applications

In this section, we explore the full strength of our new technique to show that certain decision versions of the ham-sandwich problem and of computing the Tverberg depth are W[1]-hard. To the best of our knowledge, this is the first hardness result regarding ham-sandwich cuts in higher dimension.

7.1 The Ham-Sandwich Cut Problem

The ham-sandwich problem is not a classical decision problem, as, given an instance, we know that there always exists a solution. The only decision version for the ham-sandwich problem in the plane that we are aware of has been studied by Chien and Steiger [30]: decide whether there is more than one cut. They provide an $\Omega(n \log n)$ lower bound, and as a cut in the plane can be computed in linear time, this shows that finding an object can be easier than deciding whether that object is unique.

Here, we will show that a natural "incremental" approach for computing the ham-sandwich cut will not work unless $W[1] = P$: one way to find a ham-sandwich cut incrementally could be to take any point, decide whether there is some ham-sandwich cut through it, and perform a dimension reduction until the hyperplane is determined. This gives rise to the following decision problem:

Definition 30. d-Ham-Sandwich
Input: *Sets of points $P_1, \ldots, P_d$ and a point z in $\mathbb{R}^d$*
Parameter: d
Question: *Is there a ham-sandwich cut through z?*

Using the construction from Section 6.1, we will now prove that this problem is W[1]-hard. Definition 30 asks whether there is a cut that goes through a given point z, and via translation we can assume z to be the origin. We call such a cut a *linear ham-sandwich cut.*

We will create $d+1$ sets $P_1, \ldots, P_{d+1}$ in $\mathbb{R}^{d+1}$. The set P_{d+1} will consist of the single point $q := \sum_{j=1}^{d} e_j$. The sets P_j will be the union of two sets R_j and B_j. R_j contains all points of the form p_i^j, defined exactly as in Section 6.1, i.e.,

$$R_j := \left\{ p_i^j \mid 1 \leq i \leq n \right\}$$

for $p_i^j = \frac{1}{s_i} e_j + e_{d+1}$. If we choose a linear hyperplane through one of these points, the number of points on each side will (most likely) not be the same. So in addition to these, for each of these sets we now add $n-1$ *balancing* points B_j to ensure that any linear hyperplane passing through any of these points has equally many points of P_j on both sides.

7.1.1 Construction of the Balancing-Set

The idea is to add a point set similar to the mirror image of the original set R_j. This way any linear hyperplane that has *many* of the original points on the positive side will contain *few* of the mirrored points on the positive side, and vice versa.

By making the total number of points in each set P_j odd, we will ensure that any ham-sandwich cut must pass through one of the points from P_j. Further, by the construction of the balancing set, it will not be possible to choose a linear cut through q that also goes through any of these balancing points.

For this, we will choose the mirror-image of a set of $n-1$ points that lie *between two successive points* in R_j (recall that in the construction from Section 6.1, all points from R_j lie on a line). Let S be indexed such that $1/s_i < 1/s_j$ for $i < j$.

Then, let $\varepsilon_j = \frac{1}{2^j}$ and

$$b_i^j := -\left(\frac{1}{s_i - \varepsilon_j} \right) e_j - e_{d+1}.$$

This the mirror image of a point slightly to the right of p_i^j, for $1 \leq i < n$; see Figure 7.1. Let B_j consist of all balancing points of the form b_i^j and set

$$P_j := R_j \cup B_j$$

for $1 \leq j \leq d$. Then the set $P = \bigcup P_j$ is of size $d\,(2n-1)+1$.

7.1.2 The Main Lemma

Now we come to prove the main lemma, namely that the point set allows a linear ham-sandwich cut if and only if there are d elements that sum up to 0, based on the following two lemmas. The first one states that any linear ham-sandwich cut intersects exactly one point from each set P_j, whereas the second one guarantees that any linear hyperplane that contains a point from R_j will bisect P_j.

Lemma 18. *Any linear ham-sandwich cut intersects exactly one point from each P_j, $1 \leq j \leq d+1$.*

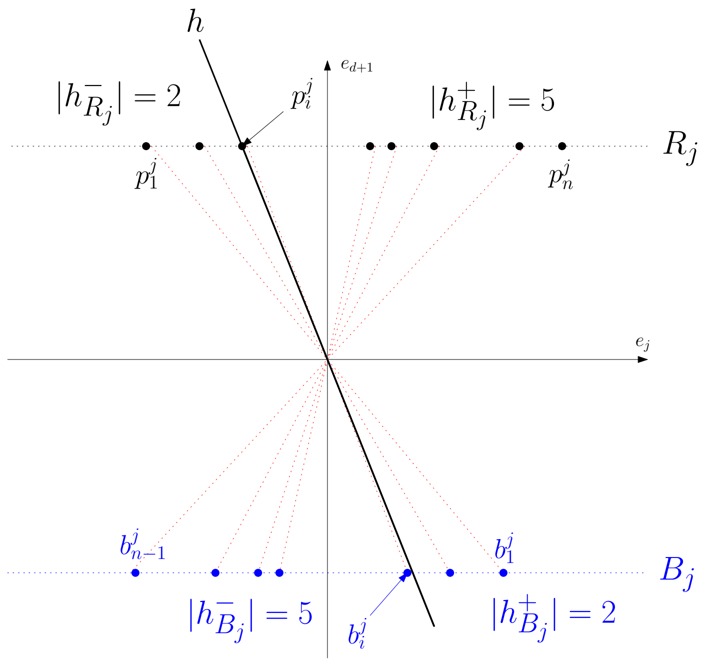


Figure 7.1: The "balanced" set P_j

Proof. For $P_{d+1} = \{q\}$ this is clear. We show that for any linear ham-sandwich cut $h = (h_1, \ldots, h_{d+1})$, we have $h_i \neq 0$ for all i: First, if h_{d+1} were 0, we would have $h_j = 0$ for all j, because the cut must pass through at least one point from each P_j. Thus, $h_{d+1} \neq 0$. Further, as $h_j(p^j)_j = -h_{d+1}(p^j)_{d+1} \neq 0$ for some $p^j \in P_j$, also $h_j \neq 0$ for all j.

Thus, no cut can pass through more than one point of any set P_j: If

$$h_j(p)_j + h_{d+1}(p)_{d+1} = hp = 0 = hp' = h_j(p')_j + h_{d+1}(p')_{d+1}$$

for two points $p, p' \in P_j$, then $p = p'$ or $h_j = 0$, a contradiction.

□

Lemma 19. *Any linear hyperplane intersecting a single point from R_j bisects the set P_j.*

Proof. Let $hp^j_i = 0$ and without loss of generality $hp^j_k < 0$ for all $1 \leq k < i$. Then also $h(-b^j_k) < 0$ and thus $hb^j_k > 0$ for all $1 \leq k < i$. Further, $hp^j_k > 0$ for all $k > i$ and $hb^j_k < 0$ for $k \geq i$. So

$$|h^-_{P_j}| = |h^-_{R_j}| + |h^-_{B_j}| = i - 1 + n - i = \left\lfloor \frac{|P_j|}{2} \right\rfloor = |h^+_{P_j}|.$$

□

Lemma 20. *There are d elements in S that sum up to 0 if and only if there is a linear ham-sandwich cut.*

Proof. "⇒" Let $\sum_{j=1}^{d} s_{i_j} = 0$. We have to find a linear hyperplane $hx = 0$ such that for each set P_j it holds that $|h^+_{P_j}|, |h^-_{P_j}| \leq \left\lfloor \frac{|P_j|}{2} \right\rfloor$. Choose $h_j := s_{i_j}$ for $1 \leq j \leq d$ and

$h_{d+1} := -1$. Because $\sum_{j=1}^{d} s_{i_j} = 0$, we have $hq = \sum_{j=1}^{d} s_{i_j} = 0$ (so the one element set P_{d+1} is bisected). Further, it holds that

$$hp^j_{i_j} = h_j \frac{1}{s_{i_j}} + h_{d+1} = 1 - 1 = 0.$$

Because of Lemma 19, this means that all sets are bisected, and thus we have a linear ham-sandwich cut.

"$\Leftarrow$" Let h be a linear ham-sandwich cut. By Lemma 18, all h_j are nonzero, so we can assume $h_{d+1} = -1$. For each j, we have $hp^j = 0$ for exactly one point $p^j \in P_j$. This means that

$$0 = hp^j = h_j(p^j)_j + h_{d+1}(p^j)_{d+1} = h_j(p^j)_j - 1(p^j)_{d+1} = h_j(p^j)_j - 1,$$

and so either $h_j = s_{i_j}$ or $h_j = s_{i_j} - \varepsilon_j$ for some i_j. Because for any $\emptyset \neq J \subseteq \{1, \ldots, d\}$ we have $0 < \sum_{j \in J} \varepsilon_j < 1$ and S is a set of integers, if one (or more) of the h_j were of the latter form, the total sum can never be an integer, and in particular not 0. But this is required for q to lie on h.

Thus, $h_j = s_{i_j} \in S$ for some i_j, and as q also lies on the hyperplane, we get

$$0 = hq = \sum_{j=1}^{d} h_j = \sum_{j=1}^{d} s_{i_j},$$

i.e., there are d elements in S that sum up to 0. $\square$

This shows the following.

Theorem 17. *The problem d-Ham-Sandwich is W[1]-hard.*

7.1.3 Remarks.

In this construction, the origin (i.e., the point for which we want to solve the decision version) is not part of any of the sets. However, as we know that a ham-sandwich cut must pass through at least one point from each set, it would be more natural for this point to be part of one of the sets as well. This is easily taken care of: set $P_{d+1} = \{\mathbf{0}, q/2, q\}$. Then any ham-sandwich cut through $\mathbf{0}$ also has to go through the other two points (otherwise there would be too many points on either side). Thus it also contains q. On the other hand, whenever there are no such d elements that sum up to 0, all ham-sandwich cuts are (truly) affine hyperplanes through $q/2$. This gives a slightly stronger result:

Corollary 5. *The following problem is W[1]-hard with respect to the dimension: Given d point sets in $\mathbb{R}^d$ and a point $z \in \bigcup P_i$, is there a ham-sandwich cut through z?*

In a different formulation, we can also state the result as follows. For a given family of $d+1$ sets in $\mathbb{R}^d$ we are not guaranteed that there is a cut that bisects all the sets simultaneously. By adding the origin as a single set, the previous shows that deciding whether there is still such a cut is also a computationally hard question:

Corollary 6. *The following problem is W[1]-hard with respect to the dimension: Given $d+1$ point sets in $\mathbb{R}^d$, is there a hyperplane that bisects all sets?*

7.2 The Tverberg Depth Problem

Finally, we consider two problems related to the computation of the Tverberg depth of a point set.

Definition 31. *Let P be a set of of points in $\mathbb{R}^d$. The* Tverberg depth of a point c *with respect to P is the the maximum number of sets that P can be partitioned into such that each set contains c in its convex hull. The* Tverberg depth of a point set P *is the highest number of sets that P can be partitioned into such that the intersection of their convex hulls is non-empty.*

A detailed discussion of complexity theoretic status regarding the computation of points of high Tverberg depth in high dimension will be given in Part IV.

Computing the Tverberg depth of a given point $c \in \mathbb{R}^d$ with respect to a point set P is an NP-hard problem if the dimension is unbounded, as shown by Teng [92]. In this section, we strengthen the result to show that the problem is even W[1]-hard with respect to the dimension.

Definition 32. d-TVERBERG-DEPTH
Input: *A set $P \subset \mathbb{R}^d$ of points, a point $c \in \mathbb{R}^d$, and $r \in \mathbb{N}$*
Parameter: d
Question: *Is the Tverberg depth of c with respect to P at least r?*

That is, we want to decide whether a given set P can be partitioned into r sets, each of which contains c in its convex hull.

We start with the construction from Section 6.2. Recall that the instances were created in $\hat{d} := d + 1$ dimensions. We will use the above idea to construct a set of points with the following property:

- The origin will be contained in the convex hull of many disjoint sets of size $\hat{d} + 1$.
- If there are d numbers adding up to 0, then the origin will also be contained in the convex hull of at least one smaller set.

Let $K := \sum_{i=1}^{n} |s_i| + 1$. We define additional points r and r' as

$$r := -\sum_{i=1}^{d} e_i + K e_{d+1}$$

and

$$r' := -e_{d+1}.$$

In addition to the points added in the construction of Section 6.2, we now add $n-1$ points r and $n-1$ points r' to the set. If we do not want to allow multiple points, we simply add scalar multiples of each. Thus, the point set now consists of $N := dn + 1 + 2(n-1) = (d+2)n - 1 = (\hat{d}+1)n - 1$ points. Now the main claim is the following.

Lemma 21. *The origin has Tverberg depth n if and only if the are d numbers that sum up to 0.*

Proof. "$\Leftarrow$" Assume that $\sum_{l=1}^{d} s_{i_l} = 0$. We choose $P_1 := \left\{p_{i_l}^l \mid 1 \le l \le d\right\} \cup \{q\}$, and split the rest of the sets as follows: form an arbitrary partition of the remaining points such that each set contains exactly one point from each P_j, and add one r and one r' to each set. This makes a total of n sets. We claim that each set contains the origin in its convex hull. Indeed, this is true for P_1: choosing $\alpha_l := |s_{i_l}|$ and $\alpha_{d+1} := 1$ yields $\sum_{l=1}^{d} \alpha_l p_{i_l}^l + \alpha_{d+1} q = \mathbf{0}$. For every other set $Q = \left\{p_{k_1}^1, \ldots, p_{k_d}^d, r, r'\right\}$, choose $\alpha_l := |s_{k_l}|$, $\alpha_r := 1$, and $\alpha_{r'} := K + \sum_{l=1}^{d} s_{k_l} > 0$. Then

$$\sum_{l=1}^{d} \alpha_l p_{k_l}^l + \alpha_r r + \alpha_{r'} r' = \mathbf{0},$$

and thus, the origin is contained in the convex hull of each of the sets.

"$\Rightarrow$" Let $P = P_1 \uplus \cdots \uplus P_n$ be a partition of P into n parts, each of which contains the origin on its convex hull.

First, we show that no set of size less than $d+1$ can contain the origin in its convex hull. Let us say that a set contains a point, if the point is part of the set, and the corresponding coefficient is larger than 0.

As all points of the form $|s_i|e_j + \operatorname{sign}(s_i)e_{d+1}$ lie in the positive half-space $\sum_{i=1}^{d} x_i > 0$, every such set must contain a point r, r', or q. If, among these, it only contains r', it must also contain at least one point of the form $|s_i|e_j + \operatorname{sign}(s_i)e_{d+1}$ in order for the last coordinate to become 0. But this means that it has to also contain either r or q, for otherwise the first d coordinates cannot all be 0. But if it contains r or q, then it also has to contain d points of the form $|s_i|e_j + \operatorname{sign}(s_i)e_{d+1}$ in order for the first d coordinates to become 0.

This means that if the origin has Tverberg depth n, at least one of the sets contains *exactly* $d+1$ points. By the same reasoning as in Lemma 16, such a set corresponds to d numbers in S that sum up to 0.

□

This shows the following theorem.

Theorem 18. *The problem d-TVERBERG-DEPTH is W[1]-hard.*

Still it might be possible to find a point of highest depth—without having to decide the depth of a particular point. This leads to the related problem of computing the Tverberg depth of a point set.

Definition 33. d-SET-TVERBERG-DEPTH
Input: *A set $P \subset \mathbb{R}^d$ of points, and $r \in \mathbb{N}$*
Parameter: d
Question: *Is the Tverberg depth of P at least r?*

We will show that it is W[1]-hard to decide this problem. To this end, we need to make sure that the point of highest Tverberg depth is the origin—so far, there might be many points with much higher Tverberg depth in our construction.

To achieve this, we put β copies of our construction into β orthogonal subspaces of dimension $d+1$, creating an set of points in $\mathbb{R}^{\beta(d+1)}$. No point apart from $\mathbf{0}$ can have a Tverberg depth higher than $N = (d+2)n - 1$, which is the total number of points in a single $\mathbb{R}^{d+1}$. In any case, by Lemma 21, the origin has depth $\beta(n-1)$ in a negative instance, whereas for a positive instance it will be of depth βn. Choosing $\beta > (d+2)$ thus ensures that the origin is the point of highest Tverberg depth. Still, the dimension $\beta(d+1)$ depends only on d, and thus it stays a parameterized reduction (though the dimension is squared). This shows the following theorem.

Theorem 19. *The problem d-*Set-Tverberg-Depth *is W[1]-hard.*

Observe that our idea also shows how to reduce the depth computation for a set with respect to a certain point to the depth computation for a set only, though in general we will need more orthogonal copies.

This concludes the section on computing the Tverberg depth of a point set. We will come back to this problem in Part IV.

Chapter 8

Conclusion

We have developed a technique to show that many problems arising from basic theorems in combinatorial geometry are d-SUM hard in dimension $d+1$. These include several versions of the affine degeneracy problem, as well as problems regarding the computation of ham-sandwich cuts and the Tverberg depth of a point set.

Combining our reductions with a result of Pătraşcu and Williams [86], Theorems 15, 16,17, and 18 immediately give:

Corollary 7. *The problems d-HAM-SANDWICH, d-TVERBERG-DEPTH, d-CARATHÉODORY, and d-HELLY cannot be solved in time $n^{o(d)}$ (where n is the size of the input), unless the Exponential Time Hypothesis (cf. Section 4.3) is false.*

Several aspects of our construction are still a bit unsatisfactory. First of all, we would like to know whether the point sets we create can be put in general position, in particular for the case of ham-sandwich cuts. That is, does the problem of deciding whether there is a ham-sandwich cut through a certain point remain W[1]-hard if the points are required to be in general position? Also, how do the hardness results about the decision problems for ham-sandwich cuts and Tverberg points exactly relate to the hardness of *computing* such a cut? It is an intriguing question to examine this relation more closely—if there is any at all.

Part III

Computational Aspects of Erdős-Szekeres in $\mathbb{R}^3$

Chapter 9

Introduction and Motivation

Whereas the previous problems we considered allow algorithms that run in polynomial time in every fixed dimension, we now turn to a problem that becomes NP-hard already in 3 dimensions. As the problem is solvable in polynomial time in the plane, this shows a strong dichotomy between the planar and the higher dimensional case.

9.1 The Erdős-Szekeres Theorem and Points in Convex Position

Let P be a set of points. A set $P' \subseteq P$ is in *convex position*, if none of the points $p \in P'$ is contained in $\text{conv}(P' - \{p\})$. It is said to be in *empty convex position*, if it is in convex position, and further does not contain any other point of P in its convex hull. See Figure 9.1.

The Erdős-Szekeres theorem is one of the major theorems from combinatorial geometry and one of the earliest results in geometric Ramsey theory. It states that every large enough set contains a large subset of points in convex position.

Theorem. *[Erdős and Szekeres; [45]] For every k there is a number n_k such that every planar set of n_k points in general position contains k points in convex position. Moreover, if n_k^* denotes the smallest such number, it holds that*

$$2^{k-2} + 1 \leq n_k^* \leq \binom{2k-4}{k-2} + 1 \leq 4^k.$$

Does the theorem remain true if we ask for large point sets in *empty* convex position? That this is not the case was shown by Horton [59]: in the plane there are arbitrarily large sets which do not contain empty 7-gons. Nicolás [81] and Gerken [51] independently gave a positive answer to the long standing open problem whether or not there is always an empty 6-gon. See Matoušek [68, Chapter 3] for further details and references.

Both these questions generalize to dimension larger than 2 in the obvious way, and clearly the numbers n_k^* do not increase when the dimension gets larger (proof: project to

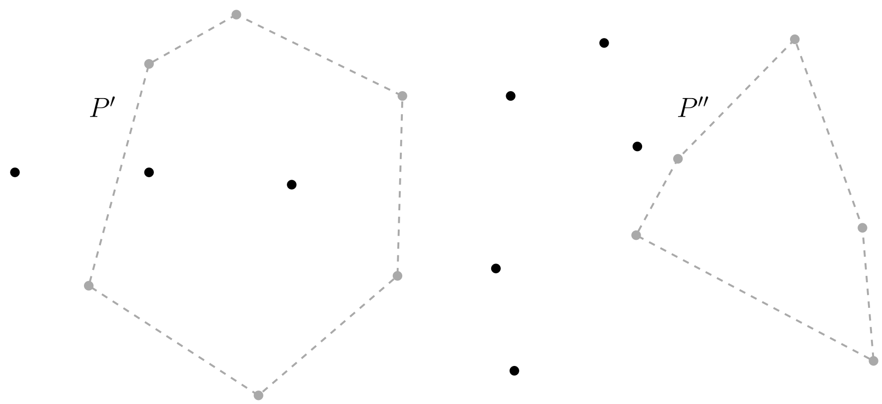


Figure 9.1: P' is in convex position, but not in empty convex position. P'' is in empty convex position.

some plane). Determining the exact values for n_k^* is still a vivid area of research, also for higher dimensions, and we refer the reader to the surveys by Bárány and Károlyi [11] or Morris and Soltan [79] for an overview of the status and (more or less) recent progress on the subject.

9.2 Previous Results

The corresponding computational problems have also received a lot of attention in the past. For the planar case, Chvátal and Klincsek [31] give an $O(n^3)$ algorithm for the problem of finding a largest convex set. This algorithm was then used by Avis and Rappaport [8] for finding the largest *empty* convex set. Several years later, Dobkin, Edelsbrunner, and Overmars [36] improved this algorithm to run in time $O\left(\gamma_3(P)\right)$, where $\gamma_3(P)$ is the number of empty triangles in the set P, which lies between n^2 and n^3. These algorithm are all based on dynamic programming and do not generalize to higher dimensions. Another approach that even enumerates all empty convex sets of size k in polynomial time in both n and k was given by Mitchell et al. [78], after a series of papers by differing authors.

In higher dimensions, the only computational result appears in [36], where it is shown that all sets of size r in empty convex position in $\mathbb{R}^3$ can be found in time $O\left(r! n \log^3 n\right)$ per set. As there can be as many as n^r of such sets, this is (at best) a small improvement over the trivial $O(n^{r+1})$ algorithm. Consequently, the question was raised whether it is possible to determine the largest set in (empty) convex position $\mathbb{R}^3$ in polynomial time.

9.3 Our Contribution

Here, we consider the computational problems of finding a largest *convex subset*, or a largest *empty convex subset*, in three dimension.

Using the reduction technique from Part I, it is an easy exercise to show that both problems are NP-hard if the dimension is not fixed, and even W[1]-hard with respect to the dimension. This means that it is very unlikely to admit an algorithm with running time $O\left(f(d)n^c\right)$ for *any* computable function f and constant c.

Still, this does not exclude the possibility that in every fixed dimension, the problem can be solved with a running time of, say, $O(n^{d+1})$. We show that this cannot be the case (under standard complexity theoretic assumptions). In particular, we will show that both problems become NP-hard already in $\mathbb{R}^3$.

First, in Section 10.1, we show NP-hardness of finding a largest empty convex set. In Section 10.2, the proof is then adapted to the actual Erdős-Szekeres problem. In the conclusion, we derive a similar result for testing weak ε-nets and red-blue discrepancy and make several suggestions for further research on the subject.

Chapter 10

The Reduction

We now present the reduction to show the hardness of the two decision problems arising from the aforementioned questions. The reduction uses a lifting transform to the elliptic paraboloid well known in computational geometry. Still, the only other application of this transform in an NP-hardness proof that we are aware of is due to Buchin et al. [15] for the problem of approximating polyhedral objects by spherical caps.

10.1 The Largest Empty Convex Set Problem

First, we will consider the following decision problem.

Definition 34 (Largest-Empty-Convex-Set, LECS)**.** *Let P be a set of points in $\mathbb{R}^d$ and $k \in \mathbb{N}$. Is there a set $Q \subseteq P$ of k points in empty convex position?*

We will show that the problems is NP-hard by a reduction from a slight modification of the following problem:

Definition 35 (Independent-Set-of-Non-overlapping-Unit-Disks, ISNUD)**.** *Let P be a set of pairwise non-overlapping unit disks in $\mathbb{R}^2$ and $k \in \mathbb{N}$. Are there k disks such that no two of them touch?*

Here, non-overlapping means that the interiors of the disks are pairwise disjoint. The intersection graphs of non-intersecting unit disks are also called *penny graphs*. As shown by Cerioli et al. [19], the problem ISNUD is NP-hard by a simple reduction from a variant of the Vertex-Cover problem. However, the reduction has a little flaw, because some of the centers of the disks created have irrational coordinates.

We overcome this obstacle by perturbing each point a little by at most some small ε in order to have a rational center, and enlarging the diameter of each disk by 2ε. This way, we get an instance of unit disks that *almost* forms a penny graph—some of the disks might now overlap a little. Moreover, in the original construction, the angle between any two intersections along a circle is always at least $\pi/2$. Thus, by choosing ε appropriately, after the perturbation and enlargement of the circles, the angle between the two closest points of two circles intersecting a common circle is still $\pi/2 \pm \alpha^*$, for some small $\alpha^* < \pi/100$. Moreover, the minimum distance between two centers is still at least $\delta^* = 2 - 2\varepsilon$. In particular, instances that arise this way do not induce any new incidences, and thus there is an independent set of size k among the perturbed disks if and only if there is a set of k independent disks in the original instance. Observe that the value of this ε does not depend on the input, and thus can be chosen as some arbitrarily small constant, say $\varepsilon = 10^{-6}$.

We call instances of unit disks that arise this way *quasi non-intersecting*. Combining this with the reduction from [19] then shows the following corollary:

Corollary 8. *The following problem is NP-hard: Given a set of quasi non-intersecting unit disks and $k \in \mathbb{N}$, decide whether there are k disks such that no two of them intersect.*

We will now first reduce this problem to LECS and afterwards show how to adapt it to the problem of finding a largest (not necessarily empty) convex set in the next section.

For a given instance $\mathcal{D}$ of unit disks in the plane, we will create a set of points in $\mathbb{R}^3$. All these points will lie close to the elliptic paraboloid, in a sense to be made precise later.

For a point $x = (x_1, x_2) \in \mathbb{R}^2$, let

$$\text{lift}\colon (x_1, x_2) \mapsto (x_1, x_2, x_1^2 + x_2^2)$$

denote the standard lifting transform to the paraboloid. Let D_c denote the n centers of the disks in $\mathcal{D}$, and let L denote the set of all points $\hat{x} := \text{lift}(x)$, for $x \in D_c$.

We now want to forbid certain pairs of points to lie in empty convex positions, namely those for which the corresponding disks intersect. Thus, for a pair of intersecting disks d, d' and their centers $c_d, c_{d'}$, we add a blocking point

$$b_{dd'} := \frac{1}{2}\left(\text{lift}(c_{d'}) + \text{lift}(c_d)\right).$$

The set B is then defined as

$$B := \{b_{dd'} \mid d \cap d' \neq \emptyset\},$$

and thus lies slightly above the paraboloid. Finally, we set $P := L \cup B$.

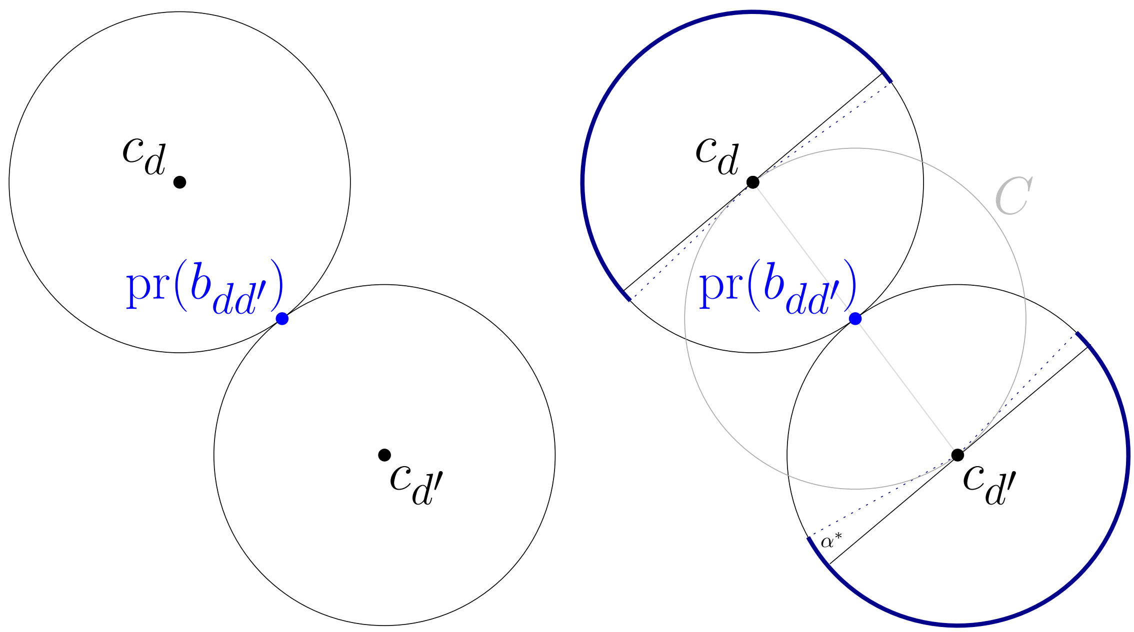
Thus, we have created $O(|\mathcal{D}|)$ points, and as the underlying geometric graph is planar, the size of the reduction is linear in the input size. The main property of the reduction is captured by the following lemma.

Lemma 22. *A blocking point $b_{dd'}$ is contained in the convex hull of a set $Q \subseteq L$ if and only if both $\hat{c}_d$ and $\hat{c}_{d'}$ are contained in Q.*

Proof. "$\Leftarrow$" by definition

"$\Rightarrow$" We show that there is a plane that contains $b_{dd'}$, $\hat{c}_d$, and $\hat{c}_{d'}$ and has all other points strictly on the positive side. Here we will make use of the fact that our instance consists of quasi non-intersecting unit disks—otherwise, the claim would not hold.

Let C be the circle whose center is the projection of $b_{dd'}$ to the first two coordinates that passes through c_d and $c_{d'}$. Because the centers of the disks have a distance of at least



(a) Two intersecting disks and the projected blocking point

(b) Because all intersections with other disks lie at the bold arcs, no projected blocking points lies inside C

Figure 10.1: Finding an empty circle

$\delta^* > 3/2$, and every point in C is at most at distance $\sqrt{2+2\varepsilon} < 3/2$ from either c_d or $c_{d'}$, this circle does not contain any other points from D_c. Further, because all intersection points are at least an angle of α^* apart, the circle does not contain any (projection of) a blocking point. See Figure 10.1. Now we define h to be the unique plane whose intersection with the paraboloid projects to the circle C. This plane contains all three points, and because C does not contain any other points, all other points from P lie strictly above h. Thus, $b_{dd'}$ is contained in conv(Q) if and only if $b_{dd'} \in \text{conv}(Q \cap h) = \text{conv}(\{\hat{c}_d, \hat{c}_{d'}\})$, and the claim follows.

□

The following states that whether or not a set is in empty convex position will depend only on which points we choose from L. The set B can always be added without destroying this property.

Lemma 23. *The sets L and B each are in empty convex position. Further, it holds that* $\text{conv}(L) = \text{conv}(L \cup B)$.

Proof. By construction, all points of L lie on the paraboloid. The points from B can be separated from each other by the plane defined in the previous proof. As all of them are convex combinations of points in L, we have $\text{conv}(B) \subseteq \text{conv}(L)$. □

Combining Lemmas 22 and 23, we get the following.

Corollary 9. *A set $L' \cup B' \subseteq P$ is in empty convex position if and only if no point of $B' \subseteq B$ is contained in the convex hull of $L' \subseteq L$.*

Now we are ready to prove the main property of our construction.

Lemma 24. *There is an independent set of size k among the unit disks if and only if there are $k + |B|$ points in empty convex position.*

Proof. "⇒" Let I, $|I| = k$, be an independent set among the set of disks. Let $\hat{I} \subseteq L$ denote the corresponding lifted centers. We claim that $S = \hat{I} \cup B$ is in empty convex position. Indeed, by Corollary 9, no point of $L - \hat{I}$ is in the convex hull of S. Further, by Lemma 22, if some point $b \in B$ was in $\text{conv}(S)$, this would mean that there are two points in $\hat{I}$ that contained b in their convex hull. Thus, by Lemma 22, the corresponding disks would touch, and I would not be an independent set. This means that there are $k + |B|$ points in empty convex position.
"⇐" Now assume that there is no independent set of size k. This means that for any choice of k disks, two of them touch. Now take any set S of $k + |B|$ points. As there are only $|L| + |B|$ points in total, this must contain at least k points from L. Thus, at least two of them belong to disks that intersect. By Lemma 22, their convex hull contains a point of B. Thus, S is not in empty convex position. □

This shows the following theorem.

Theorem 20. *The problem* LARGEST-EMPTY-CONVEX-SET *is NP-hard in* $\mathbb{R}^3$.

10.2 The Erdős-Szekeres Problem

We now show how this reduction can be applied to the following decision problem, arising from the Erdős-Szekeres theorem.

Definition 36 (ERDŐS-SZEKERES). *Let P be a set of points in $\mathbb{R}^d$ and $k \in \mathbb{N}$. Is there a set $Q \subseteq P$ of k points in convex position?*

In order to show NP-hardness of this problem, we use the exact same construction as in the previous section. We only need an analog of Lemma 24.

Lemma 25. *There is an independent set of size k among the unit disks if and only if there are $k + |B|$ points in convex position.*

Proof. "⇒" By Lemma 24, the existence of an independent set of size k corresponds to an empty convex set of size $k + |B|$, and an empty convex set is convex.

"⇐" We show that every set of points in convex position can be modified to yield a set in empty convex position.

Let S be a set of $k+|B|$ points in convex position with $|S \cap B| < |B|$, and let $I = S \cap L$. Let D_I denote the corresponding set of disks. Observe that, if $|S \cap B| < |B|$, then $|I| > k$, and thus if all disks from D_I are independent, we are done.

Otherwise, we show how to construct a set S' in convex position of the same size such that $|S' \cap B| = |S \cap B| + 1$. Let d and d' be two disks from D_I that intersect. The point $b_{dd'}$ cannot be part of S, for otherwise S would not be in convex position. If we thus set $S' = I - \{\hat{d}\} \cup B \cup \{b_{dd'}\}$, by Corollary 9 the set is still in convex position, and we have $|S'| = |S|$ and $|S' \cap B| = |S \cap B| + 1$.

Thus, after finitely many steps we end up with a set of $k + |B|$ points in convex position which contains all points from B. In particular, no point of B is contained in the convex hull of $S \cap L$. By Lemma 24, this means that the disks corresponding to the k points from L do not intersect. Thus, we have an independent set of size k among the disks. □

This finishes the proof of the following theorem.

Theorem 21. *The problem* Erdős-Szekeres *is NP-hard in* $\mathbb{R}^3$.

Chapter 11

Conclusion

11.1 Testing Weak ε-Nets and Red-Blue Discrepancy

Here, we shortly mention that the hardness proofs also show hardness for two closely related problems. From Section 1.1, recall that a range space is a pair $(X, \mathcal{R})$, where $\mathcal{R} \subseteq 2^X$. If X is a set of points in $\mathbb{R}^d$ and $\mathcal{R}$ is the set of all convex sets determined by them, in general this range space does not allow ε-nets of small size. Thus, the notion of *weak* ε-nets was introduced, where the net is not required to consist of points of X, but can instead be chosen arbitrarily. A canonical decision problem regarding weak ε-nets is as follows:

Definition 37 (Weak-ε-Net-Verification)**.** *Let $P \subset \mathbb{R}^d$, $S \subset \mathbb{R}^d$ and $\varepsilon > 0$. Is S a weak ε-net for P with respect to all convex sets?*

Though this problem is not as interesting as the verification of *strong* ε-nets—simply due to the lack of a random sampling methods for constructing weak ε-nets—Chazelle et al. [26] give an algorithm with running time $O(n^3)$ for the problem in the plane (in a slightly different but equivalent formulation). Moreover, they state the question whether it is solvable in polynomial time in $\mathbb{R}^3$.

Recall from Section 1.1 that a closely related concept is that of red-blue discrepancy. For a set P_r of red and a set P_b of blue points, the *discrepancy* of a range R is defined as

$$D_R(P_r, P_b) := ||R \cap P_r| - |R \cap P_b||\,.$$

The discrepancy of a bicolored set $P = P_r \cup P_b$ with respect to a set of ranges $\mathcal{R}$ is then defined as $D(P_r, P_b) := \max_{R \in \mathcal{R}} D_R(P_r, P_b)$. The corresponding decision problem Red-Blue-Discrepancy asks whether the discrepancy of a given set is at least some given value $k \in \mathbb{N}$.

As in Section 4.4, the relation between large empty convex sets and verification of weak ε-nets is straightforward: the set of blocking points B determines a (k/n)-net for the set of n lifted points L if and only if there is *no* independent set of size k among the disks. A similar argument holds for Red-Blue-Discrepancy. Our reduction thus also shows the following:

Theorem 22. *The problem* Weak-ε-Net-Verification *is co-NP-hard in* $\mathbb{R}^3$ *and* Red-Blue-Discrepancy *is NP-hard in* $\mathbb{R}^3$.

This shows that, whereas (weak) ε-nets for boxes can be verified in $n^{O(d)}$ time in d dimensions, for the range space of convex sets, this is not possible. Further, it is easily seen that again this result even holds for *constant* ε.

11.2 Conclusion and Open Problems

The point set we created is slightly degenerate due to the existence of triples of points on a line. This is easily taken care of by "pushing" the blocking points yet a little further inside the paraboloid. We then add a large set of points in convex position high above the construction, so that they have to be picked in any maximal empty convex set. This way, the combinatorial properties of the construction are maintained, and the point set is in general position.

The major open question is to find an approximation algorithm for the problems Erdős-Szekeres and Largest-Empty-Convex-Set. The obvious approach (projecting to $\mathbb{R}^2$ and solving the problem there) does not work very well: as pointed out by Rote [88], it was shown by Chazelle et al. [27] that there are polytopes on n vertices whose projection in any direction has no empty convex subset larger than $\Theta(\log n / \log\log n)$. Thus, the question for a more intelligent (probably constant-factor) approximation algorithm remains and seems to be very challenging.

In addition to this, the most interesting question is the following: Is Largest-Empty-Convex-Set in $\mathbb{R}^3$ fixed parameter tractable with respect to the size of the solution? That is, can we decide whether there are k points in empty convex position in time $O\left(f(k)n^c\right)$ for some computable function f and constant c? More generally, given a point set P in $\mathbb{R}^d$, can we decide whether there is an empty convex set of size k in time $O\left(f(k)n^{O(d)}\right)$?

Observe that due to the Erdős-Szekeres theorem itself, the problem Erdős-Szekeres is trivially fixed-parameter tractable: it states that any set of at least 4^k points admits a convex subset of size k. Thus, given a point set P and a $k \in \mathbb{N}$, if $n := |P| \leq 4^k$, we use a brute force algorithm, i.e., simply try all subsets of size k. This takes time $\binom{n}{k} \approx n^k \leq (4^k)^k$. If $n > 4^k$, we simply answer yes. In any case, the running time is bounded by $4^{k^2}n$, and thus we have an algorithm with running time $O\left(f(k)n\right)$. Still, from a parameterized complexity point of view, the question for a polynomial size problem kernel is of interest. That is, can we preprocess the input in polynomial time to yield an instance (P', k') of size bounded by poly(k) and a parameter $k' \leq k$ with the following property: P' admits a convex set of size k' if and only if P admits a convex sets of size k. Applying the brute-force algorithm on the kernel would then improve the running time of the fpt algorithm significantly from $O(4^{k^2}n^c)$ to $4^{O(k\log k)}n^{c'}$.

Part IV

Approximation of Tverberg- and Centerpoints in $\mathbb{R}^d$

Chapter 12

Introduction and Motivation

Finally, we come to describe a positive result for the computational complexity of a higher dimensional problem. We present a novel lifting approach for approximating high dimensional Tverberg (and center-) points. Our technique is very generic and leads to several different new algorithms. Simple algorithms are obtained by using our technique separately, while we are later able to get stronger results by combining our approach with other known methods, most notably a recent result by Miller and Sheehy [77].

12.1 Tverberg's Theorem and Its Relatives

In many applications (such as statistical analysis or finding sparse geometric separators in meshes) we would like to have a way to generalize the one-dimensional notion of a median to higher dimensions. A very natural means to accomplish this is the notion of *half-space depth* (or *Tukey depth*) of a point set.

Definition 38. *Let P be a set of points in $\mathbb{R}^d$. The* half-space depth of a point $c \in \mathbb{R}^d$ with respect to P *is defined as*

$$\min_{\text{half-space } h, c \in h} |h \cap P|.$$

The half-space depth of a point set P *is then defined as the half-space depth of a point with maximum depth.*

The following theorem states that every point set admits a point of high half-space depth (see Danzer et al. [35] and Rado [87]).

Theorem (Centerpoint theorem)**.** *Let P be a set of n points in $\mathbb{R}^d$. Then there exists a point c of half-space depth at least $\lceil n/(d+1) \rceil$.*

Another classic fact about convexity, with a close connection to centerpoints that is not obvious right away, is the following.

Theorem (Radon's theorem). *For any* $P \subseteq \mathbb{R}^d$ *with* $d+2$ *points there exists a partition* (P_1, P_2) *of* P *such that* $\operatorname{conv}(P_1) \cap \operatorname{conv}(P_2) \neq \emptyset$.

Tverberg [94] generalized this theorem for larger point sets.

Theorem (Tverberg's theorem). *Any set* $P \subseteq \mathbb{R}^d$ *with* $n = (r-1)(d+1)+1$ *points can be partitioned into* r *sets* $P_1, \ldots, P_r$ *such that* $\bigcap_{i=1}^r \operatorname{conv}(P_i) \neq \emptyset$.

As mentioned in Section 7.2, in general for any partition $P_1, \ldots, P_{r'}$ of P into r' sets and any point c that is contained in the convex hull of each of the sets, we say that c has *Tverberg depth* at least r' with respect to P. Consequently, c is called an *approximate Tverberg point* (of depth r'). Tverberg's theorem thus states that, for any set P in $\mathbb{R}^d$, there is a point of depth at least $\lfloor (n-1)/(d+1)+1 \rfloor = \lceil n/(d+1) \rceil$. See Figure 12.1.

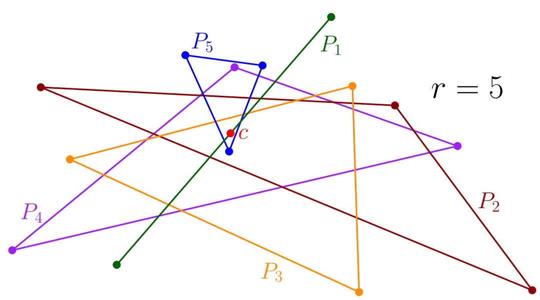


Figure 12.1: c is a point of Tverberg depth $r = 5$.

Clearly, the Tverberg depth is a lower bound on the half-space depth, and as such, Tverberg's theorem implies the centerpoint theorem. Consequently, any method of finding a point of high Tverberg depth at the same time returns a point of high half-space depth.

12.2 Previous Approaches

If we actually would like to compute a centerpoint for a given point set, the situation becomes more involved.

For lower dimensions, the situation is still well understood. In two dimensions, a centerpoint can be computed in linear time, see Jadhav and Mukhopadhyay [60]. As in $\mathbb{R}^2$, any centerpoint is also a Tverberg point (for n a multiple of 3), this algorithm also computes a Tverberg point.

For higher dimensions, Helly's theorem implies that the set of all centerpoints is given by the intersection of $O(n^d)$ half-spaces (see Edelsbrunner [42]), so we can find a centerpoint in $O(n^d)$ time through linear programming. Chan [22] shows how to improve this running time to $O(n^{d-1})$ steps in expectation. He actually solves the harder problem of finding a point with *maximum* half-space depth. Moreover, as mentioned by Agarwal, Sharir, and Welzl [1], by adapting Tverberg's original proof one is able to find a Tverberg point or n points in d dimensions in time $n^{O(d^2)}$. We will refer to this algorithm as the *exact algorithm*.

However, a running time of $n^{\Omega(d)}$ is not feasible for large d, so it makes sense to look for faster approximate solutions. A classic approach uses ε-approximations (cf. Section 1.1): in order to obtain a point of Tukey depth $n(1/(d+1)-\varepsilon)$, take a random sample $A \subseteq P$ of size $O((d/\varepsilon^2)\log(d/\varepsilon))$ and compute a centerpoint for A, using the linear-programming method. This gives the desired approximation with constant probability, and the resulting running time after the sampling step is constant. What more could we possibly wish for? For one, the algorithm is Monte-Carlo: with a certain probability, the reported point fails to be a centerpoint, and we know of no fast algorithm to check its validity (cf. Section 7.2). This problem can be solved by constructing the ε-approximation deterministically, see Chazelle [25], at the expense of a more complicated algorithm. Nonetheless, in either case the resulting running time, though constant, still grows exponentially with d, an undesirable feature for large dimensions.

This situation motivated Clarkson et al. [32] to look for more efficient randomized algorithms for approximate centerpoints. They give a simple probabilistic algorithm that computes a point of Tukey depth $O(n/(d+1)^2)$ in time $O(d^2(d\log n+\log(1/\delta))^{\log(d+2)})$, where δ is the error probability. They also describe a more sophisticated algorithm that finds such a point in time polynomial in n, d, and $\log(1/\delta)$. Both algorithms are based on a repeated algorithmic application of Radon's theorem. Unfortunately, there remains a probability of δ that the result is not correct, and we do not know how to detect a failure efficiently. This is no surprise: if the dimension is not fixed, a result by Teng [92] shows that it is co-NP-hard to check whether a given point is indeed a centerpoint.

Thus, more than ten years later, Miller and Sheehy [77] launched a new attack at the problem. Their goal is to develop a deterministic algorithm for approximating centerpoints whose running time is subexponential in the dimension. For this, they use a different proof of the centerpoint theorem that is based on the aforementioned result by Tverberg, and the observation that any point of Tverberg depth at least $n/(d+1)$ is a center point. Their main result reads as follows.

Theorem (Miller-Sheehy [77])**.** *Let P be a set of points in $\mathbb{R}^d$. Then in time $n^{O(\log d)}$ one can find a partition of P into $r=\left\lceil\frac{n}{2(d+1)^2}\right\rceil$ sets $Q_1,\dots,Q_r$ and a point c, such that $c\in\operatorname{conv}(Q_i)$ for all $1\le i\le r$.*

Hence, c constitutes an approximate centerpoint for P. At the same time, it has the advantage that also finds an approximate *Tverberg partition* of P—and as such provides us with a *certificate* for the half-space and Tverberg depth of the returned point. The algorithm is deterministic and runs in time $n^{O(\log d)}$. While this running time is subexponential in d, unfortunately the exponent of n still increases with the dimension.

12.3 Our Contribution

In this part, we show that the running time for finding approximate Tverberg partitions (and hence approximate centerpoints) can be improved. In particular, we show how to find a Tverberg partition of size $\lceil n/4(d+1)^3 \rceil$ for a set of n points in deterministic time $d^{O(\log d)} n$. This is linear in n for any fixed dimension, and the dependence on d is only quasipolynomial.

First, in Section 13.1, we present a simple lifting argument which leads to an easy Tverberg approximation algorithm. While this does not yet give a good approximation ratio (though constant for any fixed d), it is a very natural approach to the problem: it computes a higher dimensional Tverberg point via successive median partitions—just as a Tverberg point is a higher dimensional generalization of the 1-dimensional median.

By collecting several low-depth points and afterwards applying the brute-force algorithm on small point sets, we then obtain a polynomial approximation factor for any *fixed* dimension, still achieving a linear running time in n.

With a more general version of our lifting argument, in Section 13.2 we show how our technique can be combined with any previous method for computing high depth points. We then apply this idea to the approach of Miller and Sheehy, thereby improving our algorithm to yield a running time quasipolynomial in d.

Finally, we compare these results to the Miller-Sheehy algorithm and its extensions and give a short outlook on future approaches to the problem.

Chapter 13

The Algorithm

In this chapter, we present the different versions of our new algorithm. We start with the simplest variant, which we will subsequently improve both in terms of running time as well as approximation factor. From now on we will use the term *depth* as shorthand for Tverberg depth. As a model of computation we assume the uniform cost arithmetic model that carries out elementary arithmetic operations in $O(1)$ time.

13.1 A Simple Fixed-Parameter Algorithm

First, we present a simple algorithm that runs in linear time for any fixed dimension and computes a point of depth $\lceil n/2^d \rceil$. For this, we show how to compute a Tverberg point by recursion on the dimension. As a byproduct, we obtain a quick proof of a weaker version of Tverberg's theorem.

13.1.1 Basic Operations

In order to avoid linear programming, alongside with the partition of the point set, for each set we will implicitly save a convex combination of the respective point. As all points considered during our algorithms arise iteratively from by taking convex combinations of our input points, we implicitly use the following observation in order to maintain this invariant.

Observation 3. *If $x_i = \sum_{p \in P_i} \alpha_p p$ and $y = \sum_i \beta_i x_i$ are convex combinations, then*

$$\sum_i \sum_{p \in P_i} \beta_i \alpha_p p$$

is a convex combination of the points in $\bigcup P_i$ for y.

Further, by Carathéodory's theorem, in order to describe a Tverberg partition of depth r, we only need $r(d+1)$ points from P (cf. Chapter 6). In order for our algorithms to run in linear time in n, we need the following observation, also used by Miller and Sheehy [77].

Lemma 26. *Let Q be a set of $m > d+1$ points in $\mathbb{R}^d$ whose convex hull contains a point $c \in \mathbb{R}^d$, and suppose we know a convex combination of Q for c. Then we can find a set $Q' \subset Q$ of $d+1$ points that still contains c in its convex hull, together with a corresponding convex combination, in time $O(d^3m)$.*

Proof. Miller and Sheehy observe that replacing $d+2$ by $d+1$ points can be done in $O(d^3)$ time by finding an affine dependency using Gaussian elimination, see Grötschel, Lovasz, Shrijver [54, Chapter 1]. As the choice of affine dependencies does not matter, we can thus take any subset of points of size $d+2$ and eliminate one of the affine dependencies and adapting the convex combination accordingly. Repeating this process, we can replace m points by $d+1$ points in time $(m-(d+1))O(d^3) \in O(d^3m)$. □

We denote the process of replacing larger sets by sets of size $d+1$ as *pruning*, and denote a partition of a d-dimensional point set where each of the sets has size at most $d+1$ as a *pruned partition*. This pruning process later enables us to bound the cost of many operations in terms of the dimension d, instead of the total number of points n.

13.1.2 The Lifting Argument and a Simple Algorithm

Let P be a d-dimensional point set. As a Tverberg point is a higher dimensional version of the median, a natural way to compute a Tverberg point for P is to first project P to some lower-dimensional space, then to recursively compute a good Tverberg point for this projection, and use this point to find a solution in the higher-dimensional space. Surprisingly, we are not aware of any argument along these lines having appeared in the literature so far.

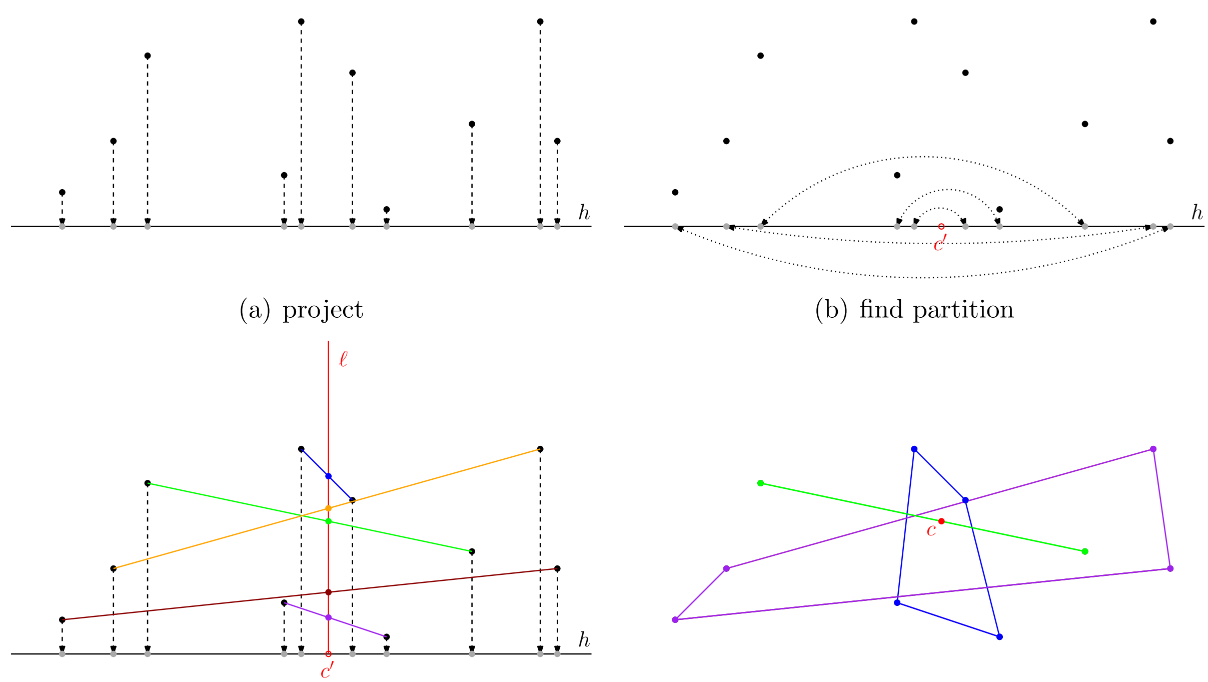
In what follows, we will describe how to *lift* a lower-dimensional Tverberg point into some higher dimension. Unfortunately, this process will come at the cost of a decreased depth for the lifted Tverberg point. For clarity of presentation, we first explain the lifting lemma in its simplest form. In Section 13.2.1, we then state the lemma in its full generality.

Lemma 27. *Let P be a set of n points in $\mathbb{R}^d$, and let h be a hyperplane in $\mathbb{R}^d$. Let $c' \in h$ be a Tverberg point of depth r for the projection of P onto h, with pruned partition $P_1, \dots, P_r$. Then we can find a Tverberg point $c \in \mathbb{R}^d$ of depth $\lceil r/2 \rceil$ for P and a corresponding Tverberg partition in time $O(n)$.*

Proof. For every point $p \in P$, let $\mathrm{pr}(p)$ denote the projection of p onto h, and for every $Q \subseteq P$, let $\mathrm{pr}(Q)$ be the projections of all the points in Q. Let $P_1, \dots, P_r \subseteq P$ such that $\mathrm{pr}(P_1), \dots, \mathrm{pr}(P_r)$ is a Tverberg partition for $\mathrm{pr}(P)$ with Tverberg point c'. Let ℓ be the line orthogonal to h that passes through c'.

Since our assumption implies $c' \in \mathrm{conv}(\mathrm{pr}(P_i))$ for $i = 1, \dots, r$, it follows that ℓ intersects each $\mathrm{conv}(P_i)$ at some point $x_i \in \mathbb{R}^d$. In particular, as we have a convex combination of the $\mathrm{pr}(P_i)$ for c', i.e., $c' = \sum_{p \in P_i} \alpha_p \,\mathrm{pr}(p)$, this intersection point is simply $x_i = \sum_{p \in P_i} \alpha_p p$.

Assuming an appropriate numbering, let $\widehat{Q}_i = \{x_{2i-1}, x_{2i}\}$, $i = 1, \dots, \lceil r/2 \rceil$, be a Tverberg partition of $x_1, \dots, x_r$. (If r is odd, one of the sets contains only one point, the



(a) project

(b) find partition

(c) intersect hulls of the sets with orthogonal complement

(d) find median of intersections and combine

Figure 13.1: Illustrating the lifting lemma in the plane: we project the point set P to the line h and find a Tverberg partition and a Tverberg point c' for the projection. Then, we construct the line ℓ through c' that is perpendicular to h, and we take the intersection with the lifted convex hulls of the Tverberg partition. We then find the median c and the corresponding partition for the intersections along ℓ. Finally, we group the points according to this partition.

median.) Since the points x_i lie on the line ℓ, such a Tverberg partition exists and can be computed in time $O(r)$ by finding the median c, i.e., the element of rank $\lceil r/2 \rceil$, according to the order along ℓ (see Cormen et al. [33]).

We claim that c is a Tverberg point for P of depth $\lceil r/2 \rceil$. Indeed, we have

$$c \in \operatorname{conv}(\widehat{Q}_i) = \operatorname{conv}(\{x_{2i-1}, x_{2i}\}) \subseteq \operatorname{conv}(P_{2i-1} \cup P_{2i}),$$

for $1 \leq i \leq \lceil r/2 \rceil$. Thus, if we set $Q_i := P_{2i-1} \cup P_{2i}$, then $Q_1, \ldots, Q_{\lceil r/2 \rceil}$ is a Tverberg partition for the point c.

Thus, the total time to find c and the Q_i is $O(n)$, as claimed. See Figure 13.1 for a two-dimensional illustration of the lifting argument. □

The proof of the next theorem is now a direct consequence of Lemma 27.

Theorem 23. *Let P be a set of n points in $\mathbb{R}^d$. One can compute a Tverberg point of depth $\lceil n/2^d \rceil$ for P and the corresponding pruned partition in time $d^{O(1)}n$.*

Proof. If $d = 1$, we can immediately find a Tverberg point and a corresponding partition by finding the median c of P [33] and pairing each point to the left of the median with exactly one point to the right of the median.

If $d > 1$, we project the points onto the hyperplane defined by $x_d = 0$. This results in an n-point set $P' \subseteq \mathbb{R}^{d-1}$. By induction, we recursively find a point and a corresponding pruned partition of depth $r' = \lceil n/2^{d-1} \rceil$ for P'. On this set, we then apply Lemma 27. Thus, we obtain a point $c \in \mathbb{R}^d$ of depth $\lceil \lceil n/2^{d-1} \rceil /2 \rceil \geq \lceil n/2^d \rceil$ and a partition $\hat{Q}_1, \dots, \hat{Q}_r$. Each of the sets consists of at most $2d$ points, so by Lemma 26, we can prune each of the sets in time $O(d^4)$.

This yields a total running time of $T_d(n) \leq T_{d-1}(n) + d^{O(1)}n$, which implies the result. □

In particular, we obtain a weak version of Tverberg's theorem with a very elementary proof.

Corollary 10 (Weak Tverberg theorem)**.** *Let P be a set of n points in $\mathbb{R}^d$. Then P can be partitioned into $\lceil n/2^d \rceil$ sets $P_1, \dots, P_{\lceil n/2^d \rceil}$ such that*

$$\bigcap_{i=1}^{\lceil n/2^d \rceil} \operatorname{conv}(P_i) \neq \emptyset.$$

13.1.3 An Improved Approximation Factor

In order to improve the approximation factor, we will now use an easy lemma to bootstrap the Tverberg depth. The idea is that, because of Carathéodory's theorem, we only need $(d+1)r$ points in $\mathbb{R}^d$ in order to describe a Tverberg partition of depth r. Thus, if our algorithm finds a point of depth $n/2^d$, we are still left with $n\left(1 - (d+1)/2^d\right)$ points which are not used at all. We will now show how to use these points in order to achieve an even higher Tverberg depth.

Lemma 28. *Suppose for any m-point set $Q \subseteq \mathbb{R}^d$ we can compute a point of Tverberg depth $\lceil m/\rho \rceil$ and a corresponding pruned Tverberg partition in time $q(m,d)$. Let $P \subseteq \mathbb{R}^d$ with $|P| = n$, and let $c \in [2, n/\rho]$ be a constant. Define the target depth as $\delta := \lceil n/c\rho \rceil$. Then we can find $\alpha := \lceil \frac{n(1-1/c)}{\delta(d+1)} \rceil$ disjoint subsets $Q_1, \dots, Q_\alpha$ of P and a Tverberg point of depth δ together with a pruned partition $\mathcal{P}_i$ in each.*

This takes total time

$$O\Big(\frac{(c-1)\rho}{d+1}(p(n,d) + q(n,d))\Big),$$

where $p(n,d)$ is the time for the pruning phase.

Proof. Let $P_1 := P$. We take an arbitrary subset $P_1' \subseteq P_1$ with $\lceil n/c \rceil$ points and find a Tverberg point c_1 of depth δ and a corresponding Tverberg partition $\mathcal{P}_1'$ for P_1'. Then we prune $\mathcal{P}_1'$ to get a Tverberg partition $\mathcal{P}_1$. Note that this takes time $p(n,d) + q(n,d)$ and that $Q_1 := \bigcup_{Z \in \mathcal{P}_1} Z$ contains at most $\delta(d+1)$ points. Set $P_2 := P_1 \setminus Q_1$ and continue.

Each partition $\mathcal{P}_i$ partitions a set $Q_i \subseteq P$ such that the Q_i are pairwise disjoint. We can repeat this process until

$$n - i\delta(d+1) < \frac{n}{c},$$

which so solves to

$$\alpha \geq i > \left\lceil \frac{n(1-1/c)}{\delta(d+1)} \right\rceil.$$

Thus, we obtain α points $c_1, \ldots, c_\alpha$ with corresponding Tverberg partitions $\mathcal{P}_1, \ldots, \mathcal{P}_\alpha$, each of depth at least $\lceil n/c\rho \rceil$, as desired.

□

For example, by Theorem 23 we can find a point of depth $\lceil n/2^d \rceil$ and a corresponding pruned partition in time $d^{O(1)}n$. Thus, by applying Lemma 28 with $c = 2$, $\rho = 2^d$, we can also find $\lceil n/(2\lceil n/2^{d+1} \rceil(d+1)) \rceil \approx 2^d/(d+1)$ points of depth $\lceil n/2^{d+1} \rceil$ in linear time in any fixed dimension.

In order to make use of Lemma 28, we will also need a lemma that describes how we can combine these points in order to increase the total depth. This generalizes a similar lemma by Miller and Sheehy [77, Lemma 4.1].

Lemma 29. *Let P be a set of n points in $\mathbb{R}^d$, and let $P = \biguplus_{i=1}^{\alpha} P_i$ be a partition of P. Furthermore, suppose that for each P_i we have a Tverberg point $c_i \in \mathbb{R}^d$ of depth r, together with a corresponding pruned Tverberg partition $\mathcal{P}_i$. Let $C := \{c_i \mid 1 \leq i \leq \alpha\}$ and c be a point of depth r' for C, with corresponding pruned Tverberg partition $\mathcal{C}$. Then c is a point of depth rr' for P. Furthermore, we can find a corresponding pruned Tverberg partition in time $d^{O(1)}n$.*

Proof. For $i = 1, \ldots, \alpha$, write $\mathcal{P}_i = \{Q_{i1}, \ldots, Q_{ir}\}$, and write $\mathcal{C} = \{D_1, \ldots, D_{r'}\}$. For $a = 1, \ldots, r'$, $b = 1, \ldots, r$, we define sets Z_{ab} as

$$Z_{ab} := \bigcup_{c_i \in D_a} Q_{ib}.$$

We claim that the set $\mathcal{Z} := \{Z_{ab} \mid a = 1, \ldots, r'; b = 1, \ldots, r\}$ is a Tverberg partition of depth rr' for P with Tverberg point c. Clearly, by definition $\mathcal{Z}$ is a partition with the appropriate number of elements. It only remains to check that $c \in \operatorname{conv}(Z_{ab})$ for each Z_{ab}. Indeed, we have

$$c \in \operatorname{conv}(D_a) = \operatorname{conv}\left(\bigcup_{c_i \in D_a} \{c_i\}\right) \subseteq \operatorname{conv}\Big(\bigcup_{c_i \in D_a} \operatorname{conv}(Q_{ib})\Big) = \operatorname{conv}\Big(\bigcup_{c_i \in D_a} Q_{ib}\Big) = \operatorname{conv}(Z_{ab}),$$

for $a = 1 \ldots r', b = 1 \ldots r$.

As the partitions $\mathcal{P}_i$ and $\mathcal{C}$ were pruned, each Z_{ab} consists of at most $(d+1)^2$ points. Thus, by Lemma 26, each Z_{ab} can be pruned in time $O(d^5)$. Since certainly $|\mathcal{Z}| \leq n$, the lemma follows. □

Combining Lemmas 28 and 29, we are now ready to prove a better Tverberg approximation.

Theorem 24. *Let P be a set of n points in $\mathbb{R}^d$. Then one can compute a Tverberg point of depth $\lceil n/2(d+1)^2 \rceil$ and a corresponding partition in time $f_d(2^{d+1}) + d^{O(1)}n$, where $f_d(m) \in m^{O(d^2)}$ is the time for computing a Tverberg point of depth $\lceil m/(d+1) \rceil$ for m using the exact algorithm.*

Proof. If $n \leq 2^{d+1}$, we use the exact algorithm. This takes at most $f_d(2^{d+1})$ time.

Otherwise, we apply Lemma 28 with $c = 2$ and $\rho = 2^d$ to obtain a set C of

$$|C| = \left\lceil \frac{n}{2\lceil n/2^{d+1} \rceil (d+1)} \right\rceil$$

points for P of depth $\lceil n/2^{d+1} \rceil$ with corresponding pruned partitions in time $d^{O(1)}n$ time. We then use the exact algorithm to get a Tverberg point for C with depth $\lceil |C|/(d+1) \rceil$ with a corresponding partition, in time $f_d(|C|)$. Finally, we apply Lemma 29 to obtain a Tverberg point and corresponding partition in time $d^{O(1)}n$. Repetitive application of $\lceil a \rceil \lceil b \rceil \geq \lceil a \lceil b \rceil \rceil$ yields that the resulting depth is

$$\lceil n/2^{d+1} \rceil \cdot \lceil |C|/(d+1) \rceil \geq \left\lceil \left\lceil \frac{n}{2^{d+1}} \right\rceil \frac{n}{2\lceil n/2^{d+1} \rceil (d+1)^2} \right\rceil = \left\lceil \frac{n}{2(d+1)^2} \right\rceil,$$

and the total running time is $f_d(2^d) + d^{O(1)}n$, as desired.

□

Alternatively, instead of the exact algorithm, we can use the algorithm by Miller and Sheehy to find a point among the deep points, while slightly reducing the running time.

Theorem 25. *Let P be a set of n points in $\mathbb{R}^d$. Then one can compute a Tverberg point of depth $\lceil n/4(d+1)^3 \rceil$ and a corresponding partition in time $2^{O(d \log d)} + d^{O(1)}n$.*

13.2 An Improved Running Time

The algorithm in the previous section runs in linear time for any fixed dimension, but the constants are huge. Thus, finally we show how to speed up our approach through an improved recursion and obtain an algorithm with running time $d^{O(\log d)}n$, while losing a depth factor of $1/2(d+1)$.

13.2.1 A More General Version of the Lifting Argument

We first present a more general version of the lifting argument in Lemma 27. For this we need some more notation. Let $P \subseteq \mathbb{R}^d$. A k-dimensional *flat* $F \subseteq \mathbb{R}^d$ (often abbreviated as *k-flat*) is defined as a k-dimensional affine subspace of $\mathbb{R}^d$ (or, equivalently, as the affine hull of $k+1$ affinely independent points in $\mathbb{R}^d$). Generalizing the notion of a Tverberg point, we call a k-dimensional flat $F \subseteq \mathbb{R}^d$ a *Tverberg k-flat of depth r for P*, if there is a partition of P into sets $P_1, \ldots, P_r$ such that $\mathrm{conv}(P_i) \cap F \neq \emptyset$ for all $i = 1, \ldots, r$.

Lemma 30. *Let P be a set of n points in $\mathbb{R}^d$, and let $h \subseteq \mathbb{R}^d$ be a k-flat. Suppose we have a Tverberg point $c \in h$ of depth r for $\mathrm{pr}(P) := \mathrm{pr}_h(P)$, as well as a corresponding Tverberg partition. Let $h_c^\perp$ be the $(d-k)$-flat orthogonal to h that passes through c. Then $h_c^\perp$ is a Tverberg $(d-k)$-flat for P of depth r, with the same Tverberg partition.*

Proof. Let $\mathrm{pr}(P_1), \ldots, \mathrm{pr}(P_r)$ be the Tverberg partition for the projection $\mathrm{pr}(P)$. It suffices to show that $\mathrm{conv}(P_i)$ intersects $h_c^\perp$ for $i = 1, \ldots, r$. Indeed, for $P_i = \{p_{i1}, \ldots, p_{il_i}\}$ let $c = \sum_{j=1}^{l_i} \lambda_j \,\mathrm{pr}(p_{ij})$ be a convex combination that witnesses $c \in \mathrm{conv}(\mathrm{pr}(P_i))$. We now write each $p_{ij} = \mathrm{pr}(p_{ij}) + \mathrm{pr}^\perp(p_{ij})$, where $\mathrm{pr}^\perp(\cdot)$ denotes the projection onto the orthogonal complement $h^\perp$ of h. Then,

$$\sum_{j=1}^{l_i} \lambda_j p_{ij} = \sum_{j=1}^{l_i} \lambda_j \,\mathrm{pr}(p_{ij}) + \sum_{j=1}^{l_i} \lambda_j \,\mathrm{pr}^\perp(p_{ij}) \in c + h^\perp = h_c^\perp,$$

as claimed. □

First of all, this shows how a good algorithm for *any fixed dimension* improves the general case:

Lemma 31. *Let $\delta \geq 1$ be a fixed integer. Suppose we have an algorithm $\mathcal{A}$ with the following property: for every point set $Q \subseteq \mathbb{R}^\delta$, the algorithm $\mathcal{A}$ constructs a Tverberg point of depth $\lceil |Q|/\rho \rceil$ for Q as well as a corresponding pruned Tverberg partition in time $f(|Q|)$.*

Then, for any n-point set $P \subseteq \mathbb{R}^d$ and for any $d \geq \delta$, we can find a Tverberg point of depth $n/\rho^{\lceil d/\delta \rceil}$ and a corresponding pruned partition in time $\lceil d/\delta \rceil f(n) + d^{O(1)} n$.

Proof. We use induction on $k := \lceil d/\delta \rceil$ to show that such an algorithm exists with running time $k(f(n) + d^{O(1)} n)$. If $k = 1$, we can just use algorithm $\mathcal{A}$ and there is nothing to show.

Now suppose $k > 1$. Let $h \subseteq \mathbb{R}^d$ be a δ-flat in $\mathbb{R}^d$, and let $\mathrm{pr}(P)$ be the projection of P onto h. We use algorithm $\mathcal{A}$ to find a Tverberg point c of depth $\lceil n/\rho \rceil$ for $\mathrm{pr}(P)$ as well as a corresponding pruned partition $\mathrm{pr}(P_1), \ldots, \mathrm{pr}(P_{\lceil n/\rho \rceil})$. This takes time $f(n)$. By Lemma 30, the $(d-\delta)$-flat $h_c^\perp$ is a Tverberg flat of depth $\lceil n/\rho \rceil$ for P, with corresponding pruned partition $P_1, \ldots, P_{\lceil n/\rho \rceil}$. For each i, we can thus find a point q_i in $\mathrm{conv}(P_i) \cap h_c^\perp$ in time $d^{O(1)}$.

Now consider the point set $Q = \{q_1, \ldots, q_{\lceil n/\rho \rceil}\} \subseteq h_c^\perp$. The set Q is a $(d-\delta)$-dimensional point set. Since we have $\lceil (d-\delta)/\delta \rceil = k - 1$, by induction we can find a Tverberg point c'

for Q of depth $|Q|/\rho^{\lceil d/\delta\rceil-1} = n/\rho^{\lceil d/\delta\rceil}$ and a corresponding pruned Tverberg partition $\mathcal{Q}$ in total time $(k-1)(f(n)+d^{O(1)}n)$. Now, c' is a Tverberg point of depth $n/\rho^{\lceil d/\delta\rceil}$ for P: a corresponding Tverberg partition is obtained by replacing each point q_i in the partition $\mathcal{Q}$ by the corresponding subset P_i. The resulting partition can be pruned in time $d^{O(1)}n$.

Thus, the total running time is

$$(k-1)(f(n)+d^{O(1)}n)+f(n)+d^{O(1)}n = k(f(n)+d^{O(1)}n),$$

and since $k = O(d)$, the claim follows.

□

For example, we can compute a point of depth $\lceil n/2(k+1)^2\rceil$ in k dimensions in time $n^{O(\log k)}$. Thus, we can compute a point of depth $\approx n/2(k+1)^{2d/k}$ in time $d^{O(1)}n^{O(\log k)}$.

13.2.2 An Improved Algorithm

We will now show how to combine the above techniques for an algorithm with a better running time. The idea of the new algorithm is as follows: using Lemma 31, we reduce solving a d-dimensional instance to solving two instances of dimension $d/2$. This can be done recursively, but unfortunately, it reduces the depth of the partition. To fix this, we apply Lemmas 28, 29 and the Miller-Sheehy algorithm to increase the depth again.

Theorem 26. *Let P be a set of n points in $\mathbb{R}^d$. Then one can compute a Tverberg point of depth $\lceil n/4(d+1)^3\rceil$ and a corresponding pruned partition in time $d^{O(\log d)}n$.*

Proof. We prove the theorem by induction on d. As stated before, for $d = 1$ the claim is immediate, as in this case the problem reduces to a median computation.

Thus, suppose that $d > 1$. By induction, for any at most $\lceil d/2\rceil$-dimensional point set $Q \subseteq \mathbb{R}^{\lceil d/2\rceil}$ there is an algorithm that returns a Tverberg point of depth $\lceil |Q|/4(\lceil d/2\rceil+1)^3\rceil$ and a corresponding pruned Tverberg partition in time $d^{\alpha\log(d/2)}n$, for some sufficiently large constant $\alpha > 0$.

Thus, by Lemma 31 (with $\delta = \lceil d/2\rceil$), there exists an algorithm that can compute a Tverberg point for P of depth $\lceil n/16(\lceil d/2\rceil+1)^6\rceil$ and a corresponding pruned Tverberg partition in total time $2d^{\alpha\log(d/2)} + d^{O(1)}n$.

Now we apply Lemma 28 with $c = 2$ and $\rho = \lceil n/32(\lceil d/2\rceil+1)^6\rceil$. The lemma shows that we can compute $\lceil 16(\lceil d/2\rceil+1)^6/(d+1)\rceil$ points of depth δ and corresponding (disjoint) pruned partitions in time $d^{\alpha\log(d/2)+O(1)}n$.

Let C be the set of these Tverberg points. Applying the Miller-Sheehy algorithm, we can find a Tverberg point for C of depth $\lceil |C|/2(d+1)^2\rceil$ and a corresponding pruned Tverberg partition in time $|C|^{O(\log d)}$. Now, Lemma 29 shows that in additional $d^{O(1)}n$ time, we obtain a Tverberg point and a corresponding pruned Tverberg partition for P of size

$$\left\lceil \frac{n}{2\cdot 16(\lceil d/2\rceil+1)^6}\right\rceil \left\lceil \frac{16(\lceil d/2\rceil+1)^6}{2(d+1)^2(d+1)}\right\rceil = \left\lceil \frac{n}{4(d+1)^3}\right\rceil,$$

as desired.

It remains to analyze the running time. Adding the various terms, we end up with a time bound of

$$T(n,d) = d^{\alpha \log(d/2)+O(1)}n + |C|^{O(\log d)} + d^{O(1)}n.$$

Since $|C| = d^{O(1)}$, we get

$$\begin{aligned} T(n,d) &\leq d^{\alpha \log(d/2)+O(1)}n + d^{O(\log d)}n \\ &\leq d^{\alpha \log d - \alpha/2}n + d^{\beta \log d}n, \end{aligned}$$

for α large enough and some $\beta > 0$, independent of d. Hence, it follows that for large enough α we have

$$T(n,d) \leq d^{\alpha \log d}n = d^{O(\log d)}n,$$

as claimed. This completes the proof. □

Thus, we can compute a polynomial approximation to a Tverberg point in time pseudopolynomial in d and linear in n.

Chapter 14

Conclusion

Finally, we compare our approach and its several variations to that of Miller and Sheehy, and give some ideas for further work on the subject.

14.1 Comparison to Miller-Sheehy

In the table below, we compare our algorithm in more detail to the Miller-Sheehy algorithm and its extensions. They give a generalization of their approach that shows that by computing higher order Tverberg points of depth r by the brute-force algorithm, the running time can be improved for small d. This comes with the loss of factor r in the output. No exact values are given, but as far as we can tell, one can achieve a polynomial $O(f(d)n^2)$ running time for fixed d by setting the parameter $r = (d+1)$, while losing a factor of $(d+1)$ in the approximation. Further, even though it is not explicitly mentioned in the paper, we think that it is possible to also bootstrap their own algorithm (for a better running time in terms of d, while losing another factor of $(d+1)$ in the output). Table 14.1 shows a rough comparison (ceilings omitted) of the different approaches. Again, $f(m) = f_d(m) := m^{O(d^2)}$ denotes the running time of the algorithm derived from the original proof of Tverberg's theorem (cf. Section 12.2).

Algorithm	**Running time**	**Depth**
Theorem 23	$O(n)$	$n/2^d$
Miller-Sheehy	$n^{O(\log d)}$	$n/2(d+1)^2$
Theorem 24	$O\left(f(2^d) + d^{O(1)}n\right)$	$n/2(d+1)^2$
Miller-Sheehy generalized ($r = d+1$)	$O\left(f(d)n^2\right)$	$\approx n/2(d+1)^3$
Theorem 25	$O\left(2^{O(d \log d)} + n\right)$	$n/4(d+1)^3$
Miller-Sheehy bootstrapped	$d^{O(\log d)}n^3$	$\approx n/2(d+1)^4$
Theorem 26	$d^{O(\log d)}n$	$n/4(d+1)^3$

We should emphasize that for all dimensions d with $2^d \leq 2(d+1)^2$, which solves to $d \leq 8$, our simplest algorithm outperforms every other approximation algorithm in both running time and approximation ratio. For example, it gives a 1/2-approximate Tverberg point in 3 dimensions in linear time.

14.2 Conclusion and Outlook

We have presented a very simple algorithm for finding an approximate Tverberg point, which runs in linear time for any fixed dimension. Using more sophisticated methods and combining our methods with known results, we managed to improve the running time to $d^{O(\log d)}n$, while getting within a factor of $1/4(d+1)^2$ of the guaranteed optimum.

Unfortunately, the resulting running time is still quasipolynomial in d, and we still do not know whether there exists a polynomial algorithm (in n and d) for finding an approximate Tverberg point. However, we are hopeful that our techniques constitute a further step towards a truly polynomial time algorithm and that such an algorithm will eventually be discovered—maybe even by a more clever combination of our algorithm with that of Miller and Sheehy.

A common issue with Tverberg (and center-) point algorithms in high dimensions, also pointed out in [32], is that the coefficient arising during the algorithm might become exponentially large. While this is not a problem in our uniform cost model, for implementations of the algorithm it seems necessary to bound these. In particular, it would be interesting to investigate the bit complexity of the intermediate solutions arising during the pruning process. In order to strengthen our result to work in a weaker model where the numbers are bounded by a polynomial in the input, it remains to check that the coefficients that arise in the process of combining points and applying Gaussian eliminations are not too large. If this is not possible, one might have to perturb the points in the process, thereby lowering the order of the coefficients.

In addition to this, an alternative algorithmic approach to computing Tverberg points that one might want to pursue stems from the most beautiful proof of Tverberg's theorem. It is due to Sarkaria and can be found in Matousek's book [68, Chapter 8]. It uses the colorful Carathéodory theorem:

Theorem (Colorful Carathéodory)**.** *Let $P = C_1 \uplus \cdots \uplus C_{d+1}$ be sets of points in $\mathbb{R}^d$, such that for each i, $1 \leq i \leq d+1$, it holds that $\mathbf{0} \in \operatorname{conv}(C_i)$. Then there is a set C with $\mathbf{0} \in \operatorname{conv}(C)$ and $|C_i \cap C| = 1$.*

Sarkaria's proof transforms a d-dimensional instance of n points of the Tverberg point problem to a colorful Carathéodory problem in approximately dn dimensions. The question now is whether such a colorful simplex can be found in time polynomial in both d and n, which would lead to a polynomial time algorithm for computing a Tverberg point.

The simplest proof of the colorful Carathéodory theorem leads directly to an algorithm for finding such a colorful simplex. It works as follows: Take a random colorful simplex. If the origin is not contained in it, delete the farthest color and take a point of that color that together with the other points induces a simplex that is closer to the origin. However, it is unknown whether this procedure runs in polynomial time for both d and n, and settling this question would be a big progress on the problem.

Outlook

In this thesis, we have investigated several problems from discrete geometry in higher dimensions. While in the plane, many of them are well understood and can be solved efficiently, we have seen that in higher dimensions, many of them become considerably harder to solve.

In addition to the open problems stated at the end of the respective parts, we suggest two different approaches in coping with them here.

First, we strongly encourage people to try to find algorithms that are fpt with respect to the dimension. Apart from the results mentioned in the introduction and our Tverberg point approximation, there are very few positive results in this direction. Still, we think that there is a huge variety of interesting problems waiting to be attacked. In particular, in all cases where a truly polynomial time algorithm is out of reach due to NP-hardness of the problem, the search for and algorithm that is fpt with respect to the dimension seems like a most suitable approach.

In addition to the problems considered in this thesis, one particularly interesting example is the well-known problem of deciding whether two sets of n points in $\mathbb{R}^d$ are congruent. The fastest algorithm for this problem is due to Brass and Knauer [13] and runs in time $O(n^{\lceil n/3 \rceil} \log n)$. Here, an algorithm that runs in time polynomial in both n and d would imply a polynomial time algorithm for Graph-Isomorphism. While this problem is most likely not NP-complete, it would still be a major breakthrough if such an algorithm existed. However, a more modest approach that solves the problem in time $O(f(d) n \log n)$ does not seem so far fetched. It would run in optimal time in any *fixed* dimension, but at the same time it would have no implications on the complexity of Graph-Isomorphism.

Another suitable approach is to show hardness in a different sense for these problem. Many problems we considered are known to always admit a solution, but still we do not know how to find them efficiently. Thus, classical complexity theory based on decision problems cannot completely capture their hardness. In this thesis, we instead looked at a modifications of the problems in order to make it a (more classical) decision problem. For example, for the ham-sandwich problem, we instead considered problem of deciding whether there is a cut through a certain point. However, this does not really settle the complexity of *finding* such a cut.

Complexity classes more suitable for such problems were suggested by Papadimitriou [85], most notably the class PPAD ("Polynomial Parity Arguments on Directed graphs"). A lot of effort has been put into investigating problems with respect to these classes, and many turn out to be PPAD-complete. These include, for example, Borsuk-Ulam theorem and the like (see [85]) and computing 2-player Nash-equilibria (see [29]). We refer the reader to Kitali [63] for a longer list. We will not give an introduction into the large area of these problems here, but instead refer the reader to Papadimitriou [85] and Johnson [61] for details.

Surprisingly, for geometric problems we are not aware of any results in this direction, even though there is a wide variety of problems suitable for such an approach. These include computing ham-sandwich cuts, computing Tverberg points, computing centerpoints, or finding a set of a certain size in convex position in $\mathbb{R}^3$, just to mention a few. In fact, most theorems from discrete geometry give rise to such a problem. Thus, we think that there is a lot of potential in investigating these problems with respect to this complexity class, and this would require a thesis of its own.

Bibliography

[1] P. K. Agarwal, M. Sharir, and E. Welzl. Algorithms for center and Tverberg points. *ACM Transactions on Algorithms*, 5(1):1–20, 2008.

[2] A. Aggarwal and S. Suri. Fast algorithms for computing the largest empty rectangle. In *Proceedings of the 3rd Symposium on Computational Geometry (SoCG)*, pages 278–290. ACM, 1987.

[3] T. Akutsu. On determining the congruence of point sets in d dimensions. *Computational Geometry - Theory and Applications*, 9(4):247 – 256, 1998.

[4] N. Alon. A non-linear lower bound for planar epsilon-nets. *Discrete & Computational Geometry*, 47(2):235–244, 2012.

[5] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147(1-2):181–210, 1995.

[6] E. Amaldi, M. E. Pfetsch, and L. E. Trotter Jr. On the maximum feasible subsystem problem, IISs and IIS-hypergraphs. *Mathematical Programming*, 95(3):533–554, 2003.

[7] B. Aronov and S. Har-Peled. On approximating the depth and related problems. *SIAM Journal on Computing*, 38(3):899–921, 2008.

[8] D. Avis and D. Rappaport. Computing the largest empty convex subset of a set of points. In *Proceedings of the 1st Symposium on Computational Geometry (SoCG)*. ACM, 1985.

[9] J. Backer and J. M. Keil. The mono- and bichromatic empty rectangle and square problems in all dimensions. In *Proceedings of the 9th Latin American Theoretical Informatics Symposium (LATIN)*, volume 6034 of *LNCS*, pages 14–25, 2010.

[10] I. Bárány, A. Hubard, and J. Jerónimo. Slicing convex sets and measures by a hyperplane. *Discrete & Computational Geometry*, 39(1):67–75, 2008.

[11] I. Bárány and G. Károlyi. Problems and results around the Erdős-Szekeres convex polygon theorem. In *Discrete & Computational Geometry*, volume 2098 of *LNCS*, pages 91–105. Springer, 2001.

[12] S. Bereg. Computing generalized ham-sandwich cuts. *Information Processing Letters*, 112(13):532–534, 2012.

[13] P. Braß and C. Knauer. Testing the congruence of d-dimensional point sets. *International Journal of Computational Geometry and Applications*, 12(1-2):115–124, 2002.

[14] H. Brönnimann and M. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete & Computational Geometry*, 14(1):463–479, 1995.

[15] K. Buchin, S. Plantinga, G. Rote, A. Sturm, and G. Vegter. Convex approximation by spherical patches. In *Proceedings of the 23rd European Workshop on Computational Geometry (EuroCG)*, 2007.

[16] S. Cabello, P. Giannopoulos, and C. Knauer. On the parameterized complexity of d-dimensional point set pattern matching. In *Proceedings of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 4169 of *LNCS*, pages 175–183, 2006.

[17] S. Cabello, P. Giannopoulos, C. Knauer, D. Marx, and G. Rote. Geometric clustering: fixed-parameter tractability and lower bounds with respect to the dimension. *ACM Transactions on Algorithms*, 7(4):Article 43, 27 pages, 2011.

[18] C. Carathéodory. über den Variabilitätsbereich der Fourierschen Konstanten von positiven harmonischen Funktionen. *Rendiconti del Circolo Matematico di Palermo*, 32(1):193–217, 1911.

[19] M. Cerioli, L. Faria, T. Ferreira, and F. Protti. On minimum clique partition and maximum independent set on unit disk graphs and penny graphs: complexity and approximation. *Electronic Notes in Discrete Mathematics*, 18:7379, 2004.

[20] M. Cesati and L. Trevisan. On the efficiency of polynomial time approximation schemes. *Information Processing Letters*, 64(4):165–171, 1997.

[21] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4):361–368, 1996.

[22] T. M. Chan. An optimal randomized algorithm for maximum Tukey depth. In *Proceedings of the 15th Symposium on Discrete Algorithms (SODA)*, pages 430–436. ACM-SIAM, 2004.

[23] T. M. Chan. A (slightly) faster algorithm for Klee's measure problem. *Computational Geometry - Theory and Applications*, 43(3):243–250, 2010.

[24] T. M. Chan, E. Grant, J. Könemann, and M. Sharpe. Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In *Proceedings of the 23rd Symposium on Discrete Algorithms (SODA)*, pages 1576–1585. ACM-SIAM, 2012.

[25] B. Chazelle. *The discrepancy method: randomness and complexity.* Cambridge University Press, 2000.

[26] B. Chazelle, H. Edelsbrunner, D. Eppstein, M. Grigni, L. Guibas, M. Sharir, and E. Welzl. Algorithms for weak epsilon-nets. Manuscript, 1995.

[27] B. Chazelle, H. Edelsbrunner, and L. J. Guibas. The complexity of cutting complexes. *Discrete & Computational Geometry*, 4(1):139–181, 1989.

[28] J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. A. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. *Information and Computation*, 201:216–231, 2005.

[29] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3), 2009.

[30] H. Chien and W. Steiger. Some geometric lower bounds. In *Algorithms and Computation*, volume 1004 of *LNCS*, pages 72–81. Springer, 1995.

[31] V. Chvátal and G. Klincsek. Finding largest convex subsets. *Congresus Numeratium*, 29:453–460, 1980.

[32] K. L. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S.-H. Teng. Approximating center points with iterative Radon points. *International Journal of Computational Geometry and Applications*, 6(3):357–377, 1996.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, third edition, 2009.

[34] G. B. Dantzig and M. N. Thapa. *Linear Programming 2: Theory and Extensions.* Springer, 2003.

[35] L. Danzer, B. Grünbaum, and V. Klee. Helly's theorem and its relatives. In *Proceedings of the 7th Symposium in Pure Mathematics*, pages 101–180, 1963.

[36] D. Dobkin, H. Edelsbrunner, and M. Overmars. Searching for empty convex polygons. *Algorithmica*, 5(1-4):561–571, 1990.

[37] D. P. Dobkin, D. Eppstein, and D. P. Mitchell. Computing the discrepancy with applications to supersampling patterns. *ACM Transactions on Graphics*, 15(4):354–376, 1996.

[38] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer, 1999.

[39] M. Drmota and R. F. Tichy. *Sequences, Discrepancies and Applications*, volume 1651 of *Lecture Notes in Mathematics*. Springer, 1997.

[40] A. Dumitrescu and M. Jiang. On the largest empty axis-parallel box amidst n points. *Algorithmica*, 2012. to appear.

[41] J. Eckstein, P. L. Hammer, Y. Liu, M. Nediak, and B. Simeone. The maximum box problem and its application to data analysis. *Computational Geometry - Theory and Applications*, 23(3):285–298, 2002.

[42] H. Edelsbrunner. *Algorithms in combinatorial geometry*. Springer, 1987.

[43] H. Edelsbrunner and R. Waupotitsch. Computing a ham-sandwich cut in two dimensions. *Journal of Symbolic Computation*, 2(2):171–178, 1986.

[44] H. G. Eggleston. *Convexity*. Cambridge University Press, 1963.

[45] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.

[46] J. Erickson. New lower bounds for convex hull problems in odd dimensions. *SIAM Journal on Computing*, 28(4):1198–1214, 1999.

[47] G. Even, D. Rawitz, and S. Shahar. Hitting sets when the VC-dimension is small. *Information Processing Letters*, 95:358–362, 2005.

[48] M. R. Fellows and N. Koblitz. Fixed-parameter complexity and cryptography. In *Proceedings of the 10th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC)*, pages 121–131, 1993.

[49] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. Springer, 2006.

[50] A. Gajentaan and M. H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Computational Geometry - Theory and Applications*, 5(3):165–185, 1995.

[51] T. Gerken. Empty convex hexagons in planar point sets. *Discrete & Computational Geometry*, 39(1-3):239–272, 2008.

[52] P. Giannopoulos, C. Knauer, M. Wahlström, and D. Werner. Hardness of discrepancy computation and ε-net verification in high dimension. *Journal of Complexity*, 28(2):162–176, 2012.

[53] M. Gnewuch, A. Srivastav, and C. Winzen. Finding optimal volume subintervals with k points and calculating the star discrepancy are NP-hard problems. *Journal of Complexity*, 25:115–127, 2009.

[54] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer, second corrected edition edition, 1993.

[55] L. Guth and N. H. Katz. On the Erdős distinct distance problem in the plane. *Computing Research Repository (CoRR)*, 1011.4105, 2010.

[56] S. Har-Peled. *Geometric Approximation Algorithms*. Mathematical surveys and monographs. American Mathematical Society, 2011.

[57] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete & Computational Geometry*, 2:127–151, 1987.

[58] E. Helly. über Mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresbericht - Deutsche Mathematiker-Vereinigung*, 32:175–176, 1923.

[59] J. D. Horton. Sets with no empty convex 7-gons. *Canadian Mathematical Bulletin*, 26(4):482484, 1983.

[60] S. Jadhav and A. Mukhopadhyay. Computing a centerpoint of a finite planar set of points in linear time. In *Proceedings of the 9th Symposium on Computational Geometry (SoCG)*, pages 83–90. ACM, 1993.

[61] D. S. Johnson. The NP-completeness column: Finding needles in haystacks. *ACM Transactions on Algorithms*, 3(2):1–21, 2007.

[62] H. Kaplan, J. Matoušek, and M. Sharir. Simple proofs of classical theorems in discrete geometry via the Guth–Katz polynomial partitioning technique. *Computing Research Repository (CoRR)*, 1102.5391, 2011.

[63] S. Kintali. A compendium of PPAD-complete problems. http://www.cs.princeton.edu/ kintali/ppad.html.

[64] C. Knauer, H. R. Tiwary, and D. Werner. On the computational complexity of ham-sandwich cuts, Helly sets, and related problems. In *Proceedings of the 28th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 9, pages 649–660. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011.

[65] C. Knauer and D. Werner. Erdős-Szekeres is NP-hard in 3 dimensions — and what now? In *Proceedings of the 28th European Workshop on Computational Geometry (EuroCG)*, pages 61–64, 2012.

[66] S. Kratsch. Co-nondeterminism in compositions: a kernelization lower bound for a Ramsey-type problem. In *Proceedings of the 23rd Symposium on Discrete Algorithms (SODA)*, pages 114–122. ACM-SIAM, 2012.

[67] S. Langerman and P. Morin. Covering things with things. *Discrete & Computational Geometry*, 33(4):717–729, 2005.

[68] J. Matoušek. *Lectures on Discrete Geometry*. Springer, 2002.

[69] J. Matoušek. *Using the Borsuk-Ulam Theorem - Lectures on Topological Methods in Combinatorics and Geometry*. Springer, 2003.

[70] J. Matoušek. *Geometric Discrepancy: An Illustrated Guide*. Springer, 2010.

[71] J. Matoušek, C.-Y. Lo, and W. Steiger. Ham-sandwich cuts in R^d. In *Proceedings of the 24th Symposium on Theory of Computing (STOC)*, pages 539–545. ACM, 1992.

[72] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4-5):498–516, 1996.

[73] N. Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the ACM*, 31(1):114–127, 1984.

[74] N. Megiddo. On the complexity of some geometric problems in unbounded dimension. *Journal of Symbolic Computation*, 10(3-4):327–334, 1990.

[75] N. Megiddo and A. Tamir. On the complexity of locating linear facilities in the plane. *Operations Research Letters*, 1(5):194–197, 1982.

[76] K. Mehlhorn. *Data Structures and Algorithms 2: Graph Algorithms and NP-Completeness*, volume 2 of *Monographs in Theoretical Computer Science. An EATCS Series*. Springer, 1984.

[77] G. L. Miller and D. R. Sheehy. Approximate centerpoints with proofs. *Computational Geometry - Theory and Applications*, 43(8):647–654, 2010.

[78] J. S. B. Mitchell, G. Rote, G. Sundaram, and G. Woeginger. Counting convex polygons in planar point sets. *Information Processing Letters*, 56(1):4549, 1995.

[79] W. Morris and V. Soltan. The Erdős-Szekeres problem on points in convex position — a survey. *Bulletin of the American Mathematical Society*, 37(4):437–458, 2000.

[80] W. Mulzer and D. Werner. Approximating Tverberg points in linear time for any fixed dimension. In *Proceedings of the 28th Symposium on Computational Geometry (SoCG)*. ACM, 2012.

[81] C. M. Nicolás. The empty hexagon theorem. *Discrete & Computational Geometry*, 38(2):389–397, 2007.

[82] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[83] H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*. Society for Industrial and Applied Mathematics, 1992.

[84] J. Pach and G. Tardos. Tight lower bounds for the size of epsilon-nets. In *Proceedings of the 27th Symposium on Computational Geometry (SoCG)*, pages 458–463. ACM, 2011.

[85] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[86] M. Pătraşcu and R. Williams. On the possibility of faster SAT algorithms. In *Proceedings of the 21st Symposium on Discrete Algorithms (SODA)*, pages 1065–1075. ACM-SIAM, 2010.

[87] R. Rado. A theorem on general measure. *Journal of the London Mathematical Society*, 21:291–300, 1946.

[88] G. Rote. Personal communication.

[89] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete & Computational Geometry*, 6(5):423–434, 1991.

[90] M. Sharir and P. K. Agarwal. *Davenport-Schinzel sequences and their geometric applications*. Cambridge University Press, 1996.

[91] W. Steiger and J. Zhao. Generalized ham-sandwich cuts. *Discrete & Computational Geometry*, 44(3):535–545, 2010.

[92] S.-H. Teng. *Points, spheres, and separators: a unified geometric approach to graph partitioning*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1992.

[93] E. Thiémard. An algorithm to compute bounds for the star discrepancy. *Journal of Complexity*, 17(4):850–880, 2001.

[94] H. Tverberg. A generalization of Radon's theorem. *Journal of the London Mathematical Society*, 41:123–128, 1966.

[95] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[96] F. Yu. On the complexity of the pancake problem. *Electronic Notes in Theoretical Computer Science*, 167:95–115, 2007.

[97] G. M. Ziegler. *Lectures on Polytopes*. Springer, 1998.