COMPUTATIONAL AND GEOMETRIC ASPECTS OF LINEAR OPTIMIZATION

Computational and Geometric Aspects of Linear Optimization

Zhongyan Guan
Department of Computing and Software
McMaster University

May 27, 2021

Abstract

Linear optimization is concerned with maximizing, or minimizing, a linear objective function over a feasible region defined as the intersection of a finite number of hyperplanes. Pivot-based simplex methods and central-path following interior point methods are the computationally most efficient algorithms to solve linear optimization instances. Discrete optimization further assumes that some of the variables are integer-valued. This dissertation focuses on the geometric properties of the feasible region under some structural assumptions. In the first part, we consider lattice (d,k)-polytopes; that is, the convex hull of a set of points drawn from $\{0,1\dots,k\}^d$, and study the largest possible diameter, $\delta(d,k)$, that a lattice (d,k)-polytope can achieve. We present novel properties and an enumeration algorithm to determine previously unknown values of $\delta(d,k)$. In particular, we determine the values for $\delta(3,6)$ and $\delta(5,3)$, and enumerate all the lattice $(3,3)$-polytopes achieving $\delta(3,3)$. In the second part, we consider the convex hull of all the 2^{2^d-1} subsums of the 2^d-1 nonzero $\{0,1\}$-valued vectors of length d, and denote by $a(d)$ the number of its vertices. The value of $a(d)$ has been determined until $d=8$ as well as asymptotically tight lower and upper bounds for $\log a(d)$. This convex hull forms a so-called primitive zonotope that is dual to the resonance hyperplane arrangement and belongs to a family that is conjectured to include lattice polytopes achieving the largest possible diameter over all lattice (d,k)-polytopes. We propose an algorithm exploiting the combinatorial and geometric properties of the input and present preliminary computational results.

Acknowledgements

First and foremost, I would like to thank Dr. Antoine Deza and Dr. Franya Franek for their advice, guidance, and support throughout this entire journey. I could not have had a better pair of advisors to help me during my Ph.D. Their continued presence, unlimited wisdom, and timely feedback were critical to the success of my research.

Much appreciation to my Ph.D. supervisory committee (Dr. Kai Huang, Dr. Reza Samavi, and Dr. Chris Swartz) and Dr. Tamon Stephen for their expertise, critical questions and insightful feedback.

I would also like to thank my family for their continued support and patience in allowing me to pursue my interests. My wife, my parents, and my brother are the main sources of motivation for me to push through difficulties and why I strive to be successful.

A big thanks goes out to all the members of the Advanced Optimization Laboratory (AdvOL), who have been like a second family. They embraced and welcomed me from the start and provided a comfortable and professional environment where many life-long friendships were made.

Contents

List of Figures

List of Tables

List of Symbols

d	:	dimension
k	:	range of integer coordinates
P	:	polytope
F	:	facet of a polytope P
$d(u,v)$	:	distance between vertices u and v in the edge-graph of P
$\delta(P)$	:	diameter of the edge-graph of the polytope P
$\delta(d,k)$	:	largest diameter achieved by a lattice (d,k)-polytope
$\gamma_i^-(P), \gamma_i^+(P)$	:	min $\{x_i : x \in P\}$, max $\{x_i : x \in P\}$
$F_i^-(P), F_i^+(P)$	:	$\{x \in P : x = \gamma_i^-(P)\}$, $\{x \in P : x = \gamma_i^+(P)\}$
$d(u,F)$	:	min $d(u,v)$ for $v \in F$
$\mathscr{F}_{d,k}^*$	:	set of lattice (d,k)-polytopes achieving $\delta(d,k)$
$\mathscr{V}_{d,k,g}$	:	the set formed by all vertices of all lattice (d,k)-polytopes of diameter at least $\delta(d,k)-g$
$\mathscr{P}_{d,k,g}$	:	set of all points with integer coordinates in the intersection of all lattice (d,k)-polytopes of diameter at least $\delta(d,k)-g$
Γ	:	graph of currently known vertices and edges
$\mathscr{C}_{d,k,g}^{\Gamma}$	:	the convex hull of all points in Γ and implicitly known points
$d_\Gamma(u,v)$	:	distance between u and v in the partial shelling, Γ
$\widetilde{d}(u,F_i^-)$	:	upper bound for $d(u,F)$
$d_\circ(u,v)$	:	upper bound for $d(u,v)$
$\delta_z(d,k)$	:	largest diameter of all primitive (d,k)-zonotopes

G_d	:	generators of dimension d
H_d	:	convex hull of all possible $2^{2^{d-1}}$ nonzero vectors with $\{0,1\}$-valued coordinates
$a(d)$	:	the number of vertices of H_d
σ_d	:	the center of symmetry of H_d

Chapter 1

Introduction

1.1 Polytopes

We start by recalling some basic definitions and properties of polytopes. For additional properties, we refer to Ziegler [46] and references therein.

A *convex polyhedron* P is the, possible empty, intersection of finite number n of half-spaces in $\mathbb{R}^d$. In other words, P can be defined as the set of solutions to a system of linear inequalities: $P = \{x \in \mathbb{R}^d : Ax \leq b\}$ where A is a n by d matrix.

A *polytope* is a bounded convex polytope; that is, there exists a scalar M such that P is inside the d dimensional sphere of radius M. A polytope can also be defined as the convex hull of a finite set of points in $\mathbb{R}^d$ [9]. A *lattice polytope* is a polytope such that all its vertices are integer-valued. A lattice $(d,k)-$polytope is a polytope in dimension d whose vertices are drawn from $\{0,1,\ldots,k\}^d$.

We consider the face of P whose vertices' i^{th} coordinate is minimized over P. When such face is of dimension $(d-1)$, it is denoted by F_i^-; we also refer to F_i^- as the facet of P defined as $F_i^- = \{x \in P : x_i = \gamma_i^-\}$ where $\gamma_i^- = \text{argmin}\{x_i :$

$x \in P\}$. Similarly, we use F_i^+ to denote a $(d-1)$ dimensional face of P whose vertices' i^{th} coordinate is maximized over P; that is, F_i^+ is the facet of P defined as $F_i^+ = \{x \in P : x_i = \gamma_i^+\}$ where $\gamma_i^+ = \text{argmax}\{x_i : x \in P\}$. In particular, when P is a lattice (d,k)-polytope intersecting each of the $2d$ facets of the cube $[0,k]^d$ over a facet of P, then F_i^-, respectively F_i^+, is the intersection of P with the hyperplane $\{x \in P : x_i = 0\}$, respectively $\{x \in P : x_i = k\}$.

A 1-dimensional face of P is referred to as an *edge*. The graph consisting of all vertices and edges of P is also known as the *edge-graph* of P.

1.2 Shortest paths in the edge-graph and diameter

A *path* between two vertices (u,v) of a polytope P is a sequence of edges along the edge-graph of P connecting the vertices u and v. A *shortest path* is a path connecting u and v consisting of the smallest number of edges. The *distance*, $d(u,v)$, is the length of a shortest path between u and v. The *diameter* of a polytope P, denoted by $\delta(P)$, is the smallest value such that the distance of any pair of vertices of P does not exceed it, that is the largest value of the shortest path between any pair of vertices of P.

The distance from a vertex u to facet F, $d(u,F)$, is defined as the minimum distance between u and v over all vertices v in F; that is, $d(u,F) = \text{argmin}\{d(u,v) : v \in F\}$.

1.3 Relation to optimization

Our research objectives relates to the following two related question in the area of combinatorial optimization. The overarching objective is to understand the structure of polytopes achieving a high diameter with respect to given parameters such as the dimension d, the number of inequalities (or facets; that is, facet-inducing inequalities) n, or the range k for lattice polytopes. The diameter is a lower bound for the worst-case number of iterations required for pivot-based linear optimization algorithms. The first part of the thesis deals with the determination of the largest possible diameter over all lattice (d,k)-polytopes for small d and k. The second part deals with a specific lattice polytope that belongs to a family of polytopes with large diameters. This family called primitive lattice zonotopes, includes polytopes with applications ranging from convex matroid optimization [34, 38] to quantum field theory [19].

1.3.1 Simplex method

Dantzig's simplex method [12] is one of the most widely used algorithms to solve linear optimization instances. It was the first practical algorithm that exploits the combinatorial properties of polyhedra. Originally introduced in 1947 by George Dantzig, the algorithm derives its name from the concept of a simplex, i.e. a generalization of a triangle in an arbitrary dimension. The method is pivot-based and purely combinatorial. Starting from an initial vertex, found using a surrogate formulation, the simplex method stays on the boundary of the feasible region until reaching, in a finite number of iterations, a vertex maximizing a linear function. Assuming for clarity of the exposition that every vertex of the feasible region is simple; that is, satisfies with equality exactly d inequalities, the simplex method travels from a vertex to an adjacent vertex using an edge whose scalar product with the objective function is nonnegative (if we are maximizing). In the dual setting, this corresponds to pivoting from a simplex to an adjacent simplex sharing a face of dimension $d-1$, hence the simplex method name. The set of inequalities satisfied with equalities are associated to the basic variables and the other inequalities to the nonbasic variables.

More specifically, the simplex algorithm is usually applied to linear programs which are in standard form:

$$\begin{aligned}&\max c^T x\\&Ax \leq b\\&\text{and } x \geq 0\end{aligned}$$

where $x = (x_1, \ldots, x_d)$ are the decision variables, $c = (c_1, \ldots, c_d)$ the coefficients of the objective function, and a set of constraints defined by an $n \times d$ matrix A and $b = (b_1, \ldots, b_n)$ being the non-negative right hand side of the n inequalities.

Each constraint defines a half-space in d dimensions which is convex. The intersection of the n half-spaces forms the set of all feasible solutions which is convex as the intersection of n convex sets. If the feasible region is bounded, it is also equal to the convex hull of its vertices. Assuming the instance is feasible and bounded in the direction of the objective function, to solve the linear optimization problem is to find a feasible solution maximizing the objective function. Since the set of maximizers form a face of the feasible region, the optimal solution must occur at minimum at one vertex. This means that neighbouring vertices of an optimal vertex cannot have a strictly larger objective value.

The simplex algorithm can be presented in a geometric fashion as a path traversal problem along the exterior of the polytope representing the set of feasible solutions, see Figure 1.1. The algorithm starts at a vertex of the polytope and checks adjacent vertices for a larger objective value. If a neighbour has a larger objective value, then the algorithms moves to that vertex according to a given pivoting rule and continues. The algorithm traverses along the exterior of the polytope, going from vertex to vertex, until it reaches a vertex with no neighbour having a better objective value.

When there are multiple neighbours with a better solution, a pivot rule is used to decide which vertex to go to. There are several choices for the pivot rule such as picking the largest value of the scalar product between the objective function and the edge, an analogue of a steepest gradient, or some lexicographical ordering. The chosen pivot rule can significantly affect the running time of the algorithm. While the simplex algorithm is quite efficient in practice, a worst case instance leading to an exponential number of iterations, or even cycling, is known for nearly all existing pivoting rules. Achieving a deeper understanding of the combinatorial and geometric properties of polytopes achieving a large diameter would contribute to finding for novel pivoting rules that are more efficient.

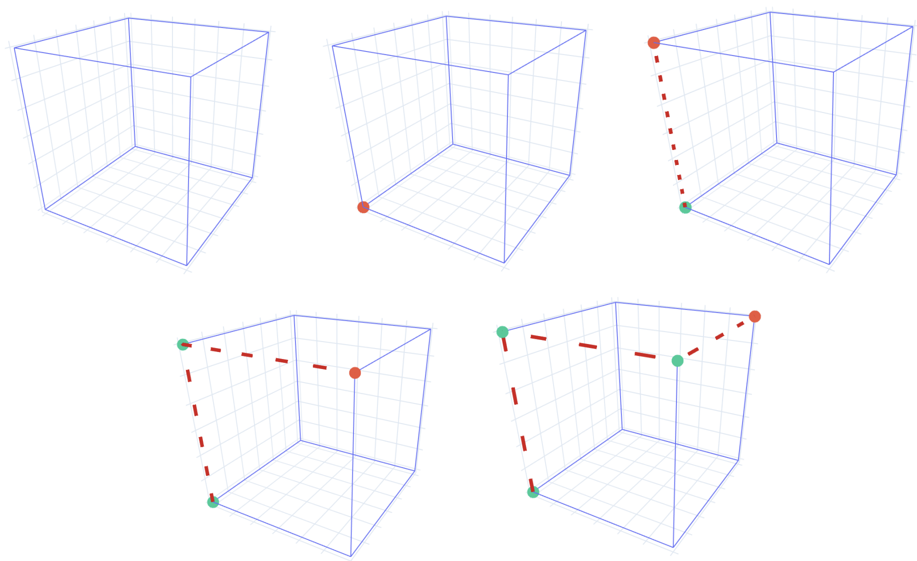

Figure 1.1: Iterations of the Simplex method, where the red point represents the current basis and the green point represents previous bases.

1.4 Research objectives

As a step towards achieving a deeper understanding of the combinatorial and geometric properties of polytopes achieving a large diameter, our first goal is to determine the maximal diameter of a lattice polytope in dimension d in an integer grid ranging from 0 to k for small entries of d and k. We discuss the structural properties, our proposed algorithm, and the obtained results in Chapter 2.

In Section 2.1, we describe the problem in more detail and discuss the structural properties in Section 2.2. In Section 2.3, we present a novel algorithm to determine the largest diameter of a lattice (d,k)-polytope. We show results obtained using the algorithm including the results for $(d,k) = (3,6)$ and $(d,k) = (5,3)$ in Section 2.4. Our algorithm is not only able to determine the maximal diameter for a given d and k, but also to enumerate all the polytopes, up to symmetry, achieving a target diameter. We provide an example of this for $(d,k) = (3,3)$ and $(d,k) = (3,2)$ in the results section. The provided results are published in [14].

The second part of the thesis deals with a family of lattice (d,k)-polytopes with large diameter that is conjectured to include polytopes achieving the largest possible diameter over all possible lattice (d,k)-polytopes. This family consists of so-called primitive zonotopes, that is, Minkowski sums of primitive segments. One such primitive zonotope arises in a number of different contexts. This zonotope, also called the white whale [4], is the Minkowski sum of all the $2^d - 1$ nonzero $\{0,1\}$-valued vectors of length d and thus dual to the resonance hyperplane arrangement, see [22] and references therein.

In Chapter 3, we present recent results concerning the vertices of the white whale including lower and upper bounds for their numbers. Then, we delve into the problem of enumerating all the vertices, propose an algorithm exploiting the combinatorial and geometric properties of the input, and present preliminary computational results.

Chapter 2

Lattice polytopes with large diameter

In this chapter, we examine the problem of determining the largest possible diameter over all lattice (d,k)-polytopes, hereinafter referred to as $\delta(d,k)$. In Section 2.2, we investigate the structural properties of lattice (d,k)-polytopes that achieve the largest possible diameter in order to exploit the symmetries, extremal properties, and reductions to smaller search spaces. We introduce a variable g which can be considered as an integer slack variable that estimates a target diameter given d,k, and $\delta(d-1,k)$. In Section 2.3, we describe in detail a novel algorithm to determine $\delta(d,k)$. We break the algorithm down into key steps, provide certificates of non-existence of polytopes achieving the target diameter and show how the algorithm allows for the enumeration of all lattice (d,k)-polytope achieving a target diameter. Finally, in Section 2.4, we present newly discovered results including the entires for $\delta(3,6)$ and $\delta(5,3)$, and all lattice $(3,3)$-polytopes achieving $\delta(3,3)$ as well as list a specific subset of all lattice $(3,2)$-polytopes achieving $\delta(3,2)$.

2.1 Introduction

In this section, we review some open research questions and results that motivate our work and present contributions by various researchers. We start by recalling the Hirsch conjecture which provided the original motivation for this work, review the progress made on the upper and lower bounds by researchers in the second half of the 20^{th} century, and look at more recent results obtained in the past few years.

The Hirsch conjecture, originally formulated in 1957 by Warren Hirsch and reported in [11], stated that the diameter of a convex polytope in dimension d with n facets is bounded by $n-d$. This has since been proven false with a counterexample presented by Santos [40] in 2012, but nonetheless, the conjecture opened up a new realm of research into the diameter of polytopes with many open questions related to the diameter of polytopes and more generally with the combinatorial, geometric, and algorithmic aspects of linear optimization. In particular, the existence of a polynomial upper bound is still an open question. Note that the initial counterexample of Santos is of dimension 43 while further investigation lead to a counterexample of dimension 20 [33, 41].

The problem of finding the maximal diameter of a polytope is inherently related to the worst case complexity of the simplex method. Since the diameter is a lower bound for the worst case complexity, exhibiting a polytope with a diameter exponential in (d,n) would imply that any pivot-based simplex method can not be polynomial in (d,n). On the other end, being able to prove the existence of a polynomial upper bound for the diameter is not enough to obtain a polynomial simplex method, as being able to exhibit such a path might be intractable. Note that since the simplex method is purely combinatorial, obtaining a polynomial simplex method would yield a strongly polynomial algorithm for linear optimization, and thus solve Smale's 9^{th} problem [42] which asks whether linear optimization can be solved with a strongly polynomial algorithm. Concerning this still open computational optimization we refer to Allamigeon et al. [1] who showed that primal-dual log-barrier interior point methods are not strongly polynomial.

The search for an upper bound on the largest diameter $\Delta(d,n)$ over all polytopes in dimension d having n facets dates back to 1967 with the work of Klee and Walkup [28] and the work of Larman [31] in 1970 who provided an upper bound that was further improved by Barnette [3] to $\Delta(d,n) \leq n2^d/6$. Note that this bound, for fixed dimension d, is linear in n. In 1992, Kalai and Kleitman [25] provided a bound of $\Delta(d,n) \leq n^{\log d+2}$. This bound has since been tightened by Todd [44] and Sukegawa [43] to $\Delta(d,n) \leq (n-d)^{\log(d-1)}$. For additional results, we refer the reader to [7, 8] and references therein.

In the case of lattice (d,k)-polytopes, k can be used an alternative parameter to n and the value of the largest diameter $\delta(d,k)$ over all lattice (d,k)-polytopes has been investigated by Naddef [37] in 1989 who showed that $\delta(d,1) = d$, and consequently that lattice $(d,1)$-polytopes, that is 0-1 polytopes, satisfy the Hirsch conjecture. A few years later, Naddef's result was generalized to any dimension by Kleinschmidt and Onn [29] who proved that $\delta(d,k) \leq kd$. Del Pia and Michini [39] were able to strengthen this bound to $\delta(d,k) \leq kd - \lceil d/2 \rceil$ when $k \geq 2$, and they showed that $\delta(d,2) = \lfloor 3d/2 \rfloor$. The bound was further strengthened by Deza and Pournin [16] to $\delta(d,k) \leq kd\lceil 2d/3 \rceil - (k-3)$ when $k \geq 3$.

In 2017, Deza, Manoussakis, and Onn [15] introduced a lower bound that is achieved by a family of lattice zonotopes, referred to as primitive zonotopes. They proved that $\delta(d,k) \geq \lfloor (k+1)d/2 \rfloor$ for $k \leq 2d-1$. Furthermore, they proposed conjecture 2.1.1.

Conjecture 2.1.1. *For any d and k, $\delta(d,k)$ is achieved, up to translation, by a Minkowski sum of lattice vectors. In particular, $\delta(d,k) = \lfloor (k+1)d/2 \rfloor$ for $k \leq 2d-1$.*

Following this line of research, Chadder and Deza [10] developed a framework to show computationally that the conjecture holds for $(d,k) = (3,4)$ and $(d,k) = (3,5)$, that is, $\delta(3,4) = 7$ and $\delta(3,5) = 9$. Our research results further substantiates this conjecture.

Table 2.1 shows the latest results for the maximal diameter of lattice polytope, $\delta(d,k)$ with our contributions in bold.

d \ k	1	2	3	4	5	6	7	8	9	...
1	1	1	1	1	1	1	1	1	1	...
2	2	3	4	4	5	6	6	7	8	...
3	3	4	6	7	9	**10**				
4	4	6	8							
5	5	7	**10**							
$\vdots$	$\vdots$	$\vdots$								
d	d	$\lfloor \frac{3}{2}d \rfloor$								

Table 2.1: The largest possible diameter $\delta(d,k)$ of a lattice (d,k)-polytope

Considering the largest diameter over all lattice (d,k)-zonotopes, $\delta_z(d,k)$, Deza, Pournin, and Sukegawa [18] showed that, up to an explicit multiplicative constant, $\delta_z(d,k)$ grows like $k^{d/d-1}$ when d is fixed and k goes to infinity. Since $\delta_z(d,k) \leq \delta(d,k)$, this result provides a new lower bound for $\delta(d,k)$.

2.2 Structural properties of lattice (d,k)-polytopes with large diameter

In this section, we introduce structural properties and lemmas that provide the foundations for the algorithm described in section 2.3.

We start by defining some symbols and abbreviations. For u and v to be vertices of a polytope P, let $d(u,v)$ be the shortest distance between u and v on the edge-graph of P. Let F be a facet of P, $d(u,F) = \min\{d(u,v) : v \in F\}$ is the shortest distance from u to any vertex that lies on the facet F. Let $\delta(P)$ denote the diameter of the edge-graph of P, which is the longest shortest path between any pair of vertices of P. The coordinates of a vector $x \in \mathbb{R}^d$ are denoted by x_1 to x_d, and its scalar product with a vector $y \in \mathbb{R}^d$ by $x{\cdot}y$.

We restate Lemma 2.2.1 originally introduced by Del Pia and Michini. This lemma provides an upper bound for $d(u,F)$ where u is a vertex of P and F a facet of P. We build off of this lemma to introduce additional structural properties.

Lemma 2.2.1 ([39]). *Consider a lattice (d,k)-polytope P. If u is a vertex of P and $c \in \mathbb{R}^d$ is a vector with integer coordinates, then $d(u,F) \leq c{\cdot}u - \gamma$ where $\gamma = \min\{c{\cdot}x : x \in P\}$ and $F = \{x \in P : c{\cdot}x = \gamma\}$.*

Assuming in Lemma 2.2.1 that $c = \pm c^i$ where c^i is the vector whose coordinates are all equal to 0 except for the i-th coordinate that is equal to 1, we consider the following objects. Let $\gamma_i^-(P) = \min\{x_i : x \in P\}$ and $F_i^-(P) = \{x \in P : x_i = \gamma_i^-(P)\}$. Similarly, let $\gamma_i^+(P) = \max\{x_i : x \in P\}$ and $F_i^+(P) = \{x \in P : x_i = \gamma_i^+(P)\}$. $F_i^-(P)$, and $F_i^+(P)$ will be denoted by F_i^- and F_i^+, when there is no ambiguity. For paths connecting u to v that go through $F_i^-(P)$ or $F_i^+(P)$, $d(u,v)$ can be bounded as follows. Note that if $d(u,v) \geq d(u,F_i^-) + d(v,F_i^-) + \delta(d-1,k)$ for some pair of vertices (u,v) of P then F_i^- must be a facet of P, that is a face of dimension $d-1$. Similarly, if $d(u,v) \geq d(u,F_i^+) + d(v,F_i^+) + \delta(d-1,k)$ then F_i^+ must be a facet of P.

$$\begin{aligned} d(u,v) \leq \min_{i=1,\ldots,d} \min\{&\delta(F_i^-) + d(u,F_i^-) + d(v,F_i^-), \\ &\delta(F_i^+) + d(u,F_i^+) + d(v,F_i^+)\}. \end{aligned} \tag{2.1}$$

Since $c = \pm c^i$, inequality (2.1) can be rewritten as Corollary 2.2.2 which is a key component to show by induction that $\delta(d,k) \leq kd$.

Corollary 2.2.2. *Let u and v be two vertices of a lattice (d,k)-polytope, then*

$$d(u,v) \leq \min_{i=1,\dots,d} \min\{\delta(F_i^-)+u_i+v_i, \delta(F_i^+)+2k-u_i-v_i\}.$$

We use proposition 2.2.3, borrowed from [21], see Corollary 12.2 and Proposition 12.4 therein, to prove Lemma 2.2.4.

Proposition 2.2.3. *Let P^1 and P^2 be two polytopes in $\mathbb{R}^d$ and $P = P^1 + P^2$ their Minkowski sum. Let $v = v^1 + v^2$, such that $v^1 \in P^1$ and $v^2 \in P^2$. Then v is a vertex of P if and only if (i) v^1 and v^2 are vertices of P^1 and P^2, respectively; and (ii) there exists an objective function $c \in \mathbb{R}^d$ that is uniquely minimized at v^1 in P^1 and at v^2 in P^2. Moreover, if u and v are adjacent vertices of P with Minkowski decompositions $u = u^1 + u^2$ and $v = v^1 + v^2$, respectively, then u^i and v^i are either adjacent vertices of P^i, or they coincide, for $i = 1,2$.*

Lemma 2.2.4. *For any lattice (d,k)-polytope Q, there exists a lattice (d,k)-polytope P of diameter at least $\delta(Q)$ satisfying $\gamma_i^-(P) = 0$ and $\gamma_i^+(P) = k$ for $i = 1,\dots,d$.*

Proof. Assume that, for some i, $\gamma_i^+(Q) - \gamma_i^-(Q) < k$. Up to translation, we can assume that $\gamma_i^-(Q) = 0$. Consider the segment $\sigma^i = \text{conv}\{0, (k-\gamma_i^+(Q))c^i\}$ where c^i is the point whose coordinates are all equal to 0 except for the i^{th} coordinate that is equal to 1. By construction, $Q+\sigma^i$ is a lattice (d,k)-polytope such that $\gamma_i^-(Q+\sigma^i) = 0$ and $\gamma_i^+(Q+\sigma^i) = k$. Let u and v be two vertices of Q such that $d(u,v) = \delta(Q)$. By Proposition 2.2.3, with $c = c^i$, there exist two vertices u' and v' of $Q+\sigma^i$ obtained as the Minkowski sums of u and v, respectively with two (possibly identical) vertices of σ^i. Moreover, for any path of length l between u' and v' in the edge-graph of $Q+\sigma^i$, there exists a path of length at most l between u and v in the edge-graph of Q. Consequently, the distance of u and v in Q is at most the distance of u' and v' in $Q+\sigma^i$. Thus, $\delta(Q) \leq \delta(Q+\sigma^i)$. If $\gamma_j^+(Q+\sigma^i) - \gamma_j^-(Q+\sigma^i) < k$ for some $j \neq i$, the above procedure can be repeated until no such coordinate remains. □

Next, we introduce a slack variable, g, to quantify the gap between the trivial upper bound, $\delta(d,k) \leq \delta(d-1,k)+k$ and a target diameter. Lemma 2.2.5 shows how g can be used.

Lemma 2.2.5. *Assume that $\delta(d,k) = \delta(d-1,k)+k-g$ for an integer g with $0 \leq g \leq k$.*

- (i) *If u and v are two vertices of a lattice (d,k)-polytope such that $d(u,v) = \delta(d,k)$, then $|u_i+v_i-k| \leq g$ for $i = 1,\ldots,d$.*
- (ii) *There exists a lattice (d,k)-polytope P of diameter $\delta(d,k)$ such that the intersection of P with each facet of the hypercube $[0,k]^d$ is, up to an affine transformation, a lattice $(d-1,k)$-polytope of diameter at least $\delta(d-1,k)-2g$.*

Proof. Setting $d(u,v) = \delta(d-1,k)+k-g$ in Corollary 2.2.2 yields:

$$\delta(d-1,k)+k-g \leq \delta(F_i^-)+(u_i+v_i) \quad \text{for } i = 1,\ldots,d, \tag{2.2}$$

$$\delta(d-1,k)+k-g \leq \delta(F_i^+)+2k-(u_i+v_i) \quad \text{for } i = 1,\ldots,d. \tag{2.3}$$

Thus,

$$k-g \leq u_i+v_i+\delta(F_i^-)-\delta(d-1,k) \quad \text{for } i = 1,\ldots,d, \tag{2.4}$$

$$k+g \geq u_i+v_i+\delta(d-1,k)-\delta(F_i^+) \quad \text{for } i = 1,\ldots,d. \tag{2.5}$$

Hence, since both $\delta(F_i^-)$ and $\delta(F_i^+)$ are at most $\delta(d-1,k)$, the inequality $k-g \leq u_i+v_i \leq k+g$ holds for $i = 1,\ldots,d$; that is, item (i) holds.

By Lemma 2.2.4, there exists a lattice (d,k)-polytope P of diameter $\delta(d-1,k)+k-g$ such that the intersection of P with each facet of the hypercube $[0,k]^d$ is nonempty. Let u and v be two vertices of P such that $d(u,v) = \delta(P)$. Inequalities (2.4) and (2.5) can be rewritten as:

$$\delta(F_i^-) \geq \delta(d-1,k) - g + k - (u_i + v_i) \quad \text{for } i = 1,\ldots,d, \tag{2.6}$$
$$\delta(F_i^+) \geq \delta(d-1,k) - g - k + (u_i + v_i) \quad \text{for } i = 1,\ldots,d. \tag{2.7}$$

Thus, since $k - g \leq u_i + v_i \leq k + g$ for $i = 1,\ldots,d$ by item (i), $\delta(F_i^-)$ and $\delta(F_i^+)$ are at least $\delta(d-1,k) - 2g$ for $i = 1,\ldots,d$; that is, item (ii) holds. □

We recall that the bounds obtained by Del Pia and Michini [39] and Deza and Pournin [15] hold in general for lattice polytopes inscribed in rectangular boxes.

Corollary 2.2.6 (Remark 4.1 in [16])**.** *Let $\delta(k_1,\ldots,k_d)$ denote the largest possible diameter of a polytope whose vertices have their i-th coordinate in $\{0,\ldots,k_i\}$ for $i = 1,\ldots,d$ and, up to relabeling, $k_1 \leq k_2 \leq \cdots \leq k_d$. The following inequalities hold:*

1. $\delta(k_1,\ldots,k_d) \leq k_2 + k_3 + \cdots + k_d - \lceil d/2 \rceil + 2$ *when* $k_1 \geq 2$*,*

2. $\delta(k_1,\ldots,k_d) \leq k_2 + k_3 + \cdots + k_d - \lceil 2d/3 \rceil + 3$ *when* $k_1 \geq 3$.

Observe that the statement of Remark 4.1 in [16] contains a typographical incorrectness as k_1 and k_d were interchanged in (i) and in (ii). Conjecture 2.1.1 can also be stated for lattice polytopes inscribed in rectangular boxes; that is, $\delta(k_1,\ldots,k_d)$ is at most $\lfloor (k_1 + k_2 + \cdots + k_d + d)/2 \rfloor$, and is achieved, up to translation, by a Minkowski sum of lattice vectors. Note that this generalization of Conjecture 2.1.1 holds for $d = 2$ and for $(k_1,k_2,k_3) = (2,2,3)$ and $(2,3,3)$. Moreover, $\delta(k_1,k_2) = \delta(k_1,k_1)$, and $\delta(2,2,3) = \delta(2,3,3) = 5$.

2.3 Computational framework to determine $\delta(d,k)$

To generate all lattice (d,k)-polytopes is quickly intractable even for relatively small d. A brute force generation would require considering the convex hull of all $2^{(k+1)^d}$ subsets of the $(k+1)^d$ integer points in the cube $[0,k]^d$. Note that for $k=1$, the number of lattice $(d,1)$-polytopes grows asymptotically like 2^{2^d} when d goes to infinity.

We present a computationally efficient enumeration approach which takes advantage of the combinatorial structure of the problem by breaking it down into a tree structure. Each branch is analyzed to determine if a solution can be found. If not, then the branch can be immediately eliminated from consideration. Otherwise, the problem is broken down into further smaller scenarios. A crucial component of the algorithm is the convex hull computation for a set of points. For this component, we employ a library of the double description method implemented in *C* by Komei Fukuda [20] and described in more details in 2.3.1. Our implementation is described in more detail in 2.3.7.

2.3.1 Double description method for convex hull computations

An important component of the main algorithm is to generate the convex hull for a given set of points to determine whether each point is a vertex or not. The module used to compute the convex hull for a given set of points is the CDD package developed by Komei Fukuda [20]. The module is written in C and uses the double description method to compute the convex hull, hence its name. The double description method was originally developed by Motzkin et al. [36] in the 1950s. The initial algorithm proposed by Motzkin et al. has been highly enhanced by Komei Fukuda to make it one of the most efficient tools to compute general dimension instances.

The double description algorithm is an incremental algorithm that, up to a normalization via a pointed cone reformulation, takes the set of inequalities as input called the hyperplane representation, also referred to as the $H-representation$, and outputs a vertex representation, referred to as $V-representation$, consisting of all vertices and extreme rays. Since we only consider lattice (d,k)-polytopes, our output is free of extreme rays. The algorithm first selects a set of $d+1$ inequalities in general position and then adds remaining inequalities one by one to construct the corresponding vertex representation. After each step, some vertices are truncated from the current convex hull by the entering inequality and some are added by the intersection of some edges with the entering inequality. The algorithm terminates when all inequalities have been considered. The efficiency of the algorithm can be greatly affected by the order in which the inequalities are considered, see Avis, Bremner, and Seidel [2], and redundancy can also be an issue.

2.3.2 The main algorithm

The main algorithm is based off the classical branch and bound algorithm coined by Little et al. [32]. Branch and bound is an approach where a systematic enumeration of the search space is performed. A rooted tree represents the set of candidate solutions and each branch contains a subset of the solution set. A current optimal solution is maintained and before enumerating a branch, the branch is checked against this solution. If it cannot produce a better solution than the bound, then it is discarded and not considered. An important component of this algorithm is to have efficient estimation of the bounds for each branch, otherwise, the algorithm can degenerate into exhaustive brute-force.

For finding the diameter of a polytope, there are existing lower and upper bounds providing a range of possible values. For cases where the difference is small, it is reasonable to show that only one value is possible by eliminating all other possibilities. Here, we introduce a slack variable g to account for this gap in the bounds. This variable will allow us to set different objective diameters. By starting from upper bound, each potential value for $\delta(d,k)$ is explored and eliminated when we can guarantee that no lattice (d,k)-polytopes exist with a diameter of the target value. By this process of elimination, we eventually reach a value for $\delta(d,k)$ where we can generate a valid lattice (d,k)-polytope, and hence find what the actual value is. We modify the classical branch and bound approach by considering the target diameter our current optimal solution and by using heuristics to find the best solution a branch can produce. So any branch which has an optimal solution worse than the target diameter can be eliminated from consideration.

The main algorithm has three main components. First, we generate all possible (u,v) pairs of vertices that can achieve the target diameter, $\delta(d-1,k)+k-g$. Then, for each pair of vertices, we create a sub-skeleton of a valid polytope by considering each intersection with the $[0,k]^d$ hypercube, described in more detail in Section 2.3.4. This procedure is called the *shelling* step, as we are attemping to create a shell of a polytope by adding once facet at a time. A number of certificates of validity are checked after each new facet is added to ensure that the current sub-skeleton can still achieve the target diameter. A set of choices for the $2d$ intersections with the facets of the hypercube $[0,k]^d$ that is obtained by the shelling step is called a *shelling*. If there are no valid sub-skeletons generated by the shelling step, then we can conclude that $\delta(d,k) < \delta(d-1,k)+k-g$. Finally, for each shelling, we consider all remaining interior points, with integer coordinates ranging from 1 to $k-1$, as candidate vertices. The output of this final step consists of a list of all polytopes achieving the target diameter containing the given (u,v) pair. Polytopes which are duplicate up to the symmetries of the $[0,k]^d$ hypercube are removed. This process is called the *inner step* and is described in more detail in section 2.3.5.

The idea to consider each intersection of the $[0,k]^d$ hypercube was originally introduced by Chadder et al. [10], but it can only determine whether $\delta(d,k)$ is equal to $\delta(d-1,k)+k$. In terms of 2.2.5, this is equivalent to assuming $g=0$. This case is significantly easier than $g>0$ since both $\delta(F_i^-)$ and $\delta(F_i^+)$ must be faces that have the maximal diameter, $\delta(d-1,k)$ and therefore, greatly reduces the number of possibilities for the choice of each facet.

2.3.3 Generating potential (u,v) pairs

The first step of the algorithm is to generate the set of potential (u,v) pairs of vertices of a lattice (d,k)-polytope that satisfies $d(u,v) = \delta(d-1,k)+k-g$. Reducing the number of potential (u,v) pairs that are considered is a crucial step in the algorithm. The number of (u,v) pairs determines the number of search trees, hence, by having as few candidate pairs as possible, the overall search space is greatly reduced. We present several key ideas used to accomplish this.

First, one can notice that many aspects of symmetry can be applied to the $[0,k]^d$ hypercube. Take for example a polytope in dimension d with a vertex at $(0,0,\ldots,0)$, this polytope can be rotated in a way that the same vertex lies on $(k,k,...,k)$ instead, or $(0,k,\ldots,k)$, or $(0,0,k,\ldots,k)$. Hence, we make the assumption that the coordinates of u satisfies:

$$u_i \leq u_{i+1} \leq \lfloor k/2 \rfloor \qquad \text{for } i = 1,\ldots,d-1.$$

Furthermore, by item (i) of 2.2.5, the coordinates of u and v are constricted even more to satisfy:

$$k-g \leq u_i + v_i \leq k+g \qquad \text{for } i = 1,\ldots,d.$$

We can also use the symmetries of the $[0,k]^d$ hypercube acting on the pair (u,v) and assume that all u are generated in lexicographical order (denoted by $\prec$ in the following). The coordinates of u and v can be assumed to satisfy the following conditions where $\tilde{w}$ is the point consisting of the coordinates of w reordered lexicographically:

$$\begin{aligned} \{v_i \leq v_{i+1} \text{ if } u_i = u_{i+1}\} &\qquad \text{for } i = 1,\ldots,d-1, \\ u \prec \tilde{w} \text{ where } w = (k,\ldots,k) - v &\qquad \text{if } \{v_i \geq \lceil k/2 \rceil \text{ for } i = 1,\ldots,d\}. \end{aligned}$$

The final idea that is used to reduce the number of potential (u,v) pairs is to consider the fact that the intersections of the polytope with the facets of $[0,k]^d$ must be of sufficiently large diameter. Let $\mathscr{V}_{d,k,g}$ denote the set formed by all the vertices of all the lattice (d,k)-polytopes of diameter at least $\delta(d,k)-g$ and let $\bar{v}_i$ denote the point in $\mathbb{R}^{d-1}$ consisting of all coordinates of v except v_i. We then define $g_i^- = g+u_i+v_i-k$ and $g_i^+ = g+k-(u_i+v_i)$. For u and v to be vertices of a lattice (d,k)-polytope such that $d(u,v)$ is at least $\delta(d-1,k)+k-g$, the following conditions must be met:

$$\begin{aligned} &\{\bar{u}_i \in \mathscr{V}_{d-1,k,g_i^-} \text{ if } u_i = 0\} \qquad \text{for } i=1,\ldots,d, \\ &\{\bar{v}_i \in \mathscr{V}_{d-1,k,g_i^+} \text{ if } v_i = k\} \qquad \text{for } i=1,\ldots,d. \end{aligned}$$

Let $\mathscr{P}_{d,k,g}$ denote the set of all the points with integer coordinates that belong to the intersection of all the lattice (d,k)-polytopes of diameter at least $\delta(d,k)-g$. Let $\mathscr{C}^{u,v}_{d,k,g}$ denote the convex hull of u, v, and the following set of points:

$$\left[\bigcup_{i=1}^{d}\{x\in\mathbb{R}^d : x_i = 0 \text{ and } \bar{x}_i \in \mathscr{P}_{d-1,k,g_i^-}\}\right] \cup \left[\bigcup_{i=1}^{d}\{x\in\mathbb{R}^d : x_i = k \text{ and } \bar{x}_i \in \mathscr{P}_{d-1,k,g_i^+}\}\right].$$

The following condition is necessary for u and v to be vertices of a lattice (d,k)-polytope such that $d(u,v)$ is at least $\delta(d-1,k)+k-g$:

$$u \text{ and } v \text{ are vertices of } \mathscr{C}^{u,v}_{d,k,g}.$$

Potential (u,v) pairs for $(d,k,g)=(3,6,0)$

In this section, we will use an example to illustrate the conditions that must be met for a (u,v) pair capable of achieving $d(u,v)=\delta(d-1,k)+k-g$. We will examine the case of $(d,k,g)=(3,6,0)$. In this case, we are assuming that u and v are vertices of a lattice $(3,6)$-polytope of diameter 12. Since $g=0$, we can assume that:

$$
\begin{aligned}
u_1 \le u_2 \le u_3 &\le 3, \\
k - g \le u_i + v_i \le k + g \implies u_i + v_i &= (6,6,6) \qquad \text{for } i = 1,2,3 \\
\{\bar{u}_i \in \mathscr{V}_{2,6,0} \text{ if } u_i &= 0\} \qquad \text{for } i = 1,2,3, \\
u \text{ is a vertex of } \mathscr{C}^{u,v}_{3,6,0} \text{ if } u_1 &\ne 0.
\end{aligned}
$$

There are a total of 20 points satisfying $u_1 \le u_2 \le u_3 \le 3$. Out of these points, only $(0,0,1)$, $(0,0,2)$, $(0,0,3)$, $(0,1,1)$, and $(0,1,2)$ satisfy the condition of $u_1 = 0$ and $(u_2, u_3) \in \mathscr{V}_{2,6,0}$. There is no point where $u_1 \ne 0$ and is also a vertex of $\mathscr{C}^{u,v}_{3,6,0}$. Thus, since $u + v = (6,6,6)$, we need to consider only 5 pairs of vertices (u, v). The 5 pairs of vertices considered for $(3,6,0)$ are shown in table 2.2.

u	v
(0,0,1)	(3,3,2)
(0,0,2)	(3,3,1)
(0,0,3)	(3,3,0)
(0,1,1)	(3,2,2)
(0,1,2)	(3,2,1)

Table 2.2: All considered pairs $\{u, v\}$ for $(d,k,g) = (3,6,0)$

The sets $\mathscr{V}_{2,6,0}$ and $\mathscr{P}_{2,6,0}$ are illustrated in Figure 2.1. Note that both $\mathscr{V}_{d,k,g}$ and $\mathscr{P}_{d,k,g}$ are invariant under the symmetries of $[0,k]^d$.

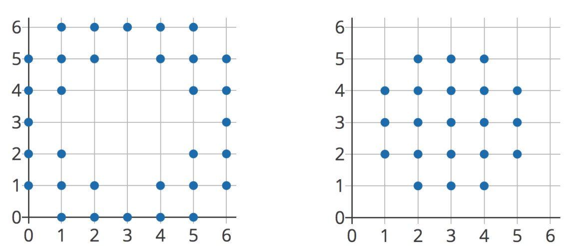

Figure 2.1: The sets $\mathscr{V}_{2,6,0}$ and $\mathscr{P}_{2,6,0}$

2.3.4 Main procedure to check whether polytopes achieving the target diameter exists

The main procedure of the algorithm is the shelling step. This module takes as input all (u,v) pairs generated by the step described in Section 2.3.3 and performs the necessary calculations to determine whether there exists a lattice (d,k)-polytope which contain u and v as vertices such that $d(u,v) = \delta(d-1,k)+k-g$. We consider all possible choices for the $2d$ intersections with the facets of the hypercube $[0,k]^d$. For each intersection, we try all possible facets of dimension $d-1$ with a diameter greater or equal to $\delta(d-1,k)-g_i$. Since there can be an extensive number of polytopes achieving this diameter, reducing the search space is an important factor.

Key ideas to reduce search space

The first key idea is to prioritize the sequencing of the intersections with the $[0,k]^d$ hypercube. The order in which intersections are considered is crucial in reducing the number of candidate branches to check. Intersections may have a different number of candidate facets. Therefore, different orderings will result in different search trees. We want to prioritize nodes with fewer branches as this will result in more candidates being eliminated when a branch is pruned.

We use the facet gap parameter, g_i^- and g_i^+, as a score to determine the priority of the facets, $F_0^-,\ldots,F_d^+$. These facet gap parameters are calculated as follows:

$$g_i^- = g+u_i+v_i-k$$
$$g_i^+ = g+k-(u_i+v_i)$$

A lower score is equivalent to a higher priority. The scores, g_i^- and g_i^+, are often the same for different intersections, so we use the following tiebreaker rules to determine the optimal sequencing.

1. The facet gap parameter, g_i^- and g_i^+. Lower value takes precedence.

2. The number of currently known vertices of P belonging to the intersection. An intersection containing more known points takes precedence.

3. An intersection containing u or v has higher priority over an intersection containing neither.

4. Finally, a default order of $F_1^-,...,F_d^-,F_1^+,...,F_d^+$ is the last tiebreaker.

The facet gap parameter is used as the first tiebreaker since it is directly tied to the number of candidate facets to be considered. Smaller values for g_i means that candidate polytopes in $d-1$ have a larger diameter, and hence, fewer such candidates exist.

Known vertices in an intersection also present a way to reduce candidate facets from consideration. Suppose, we start with (u,v) equal to $(0,0,0)$ and $(3,3,3)$, and we are considering F_0^-. Any validate candidate must contain $(0,0)$ as a vertex, otherwise, it would invalidate our previously assumed vertex $(0,0,0)$. Hence, more known vertices means more restrictions on candidate facets, and therefore fewer candidate facets exists.

The final two rules provide a deterministic way to select the sequencing of facets.

Another crucial step in reducing the search space is to identify paths between u and v induced by the shelling step as early as possible. These are the estimation of bounds that will determine when a branch of the search tree can be eliminated. When a path, an upper bound for the maximal distance, is found from u to v that is strictly less than $\delta(d-1,k)+k-g$, then this guarantees that there cannot exist a candidate in the current branch capable of achieving the target diameter.

We introduce two certificates that show no lattice (d,k)-polytope with vertices u and v can exist such that $d(u,v) = \delta(d-1,k)+k-g$.

Certificate 1: Shortest path in shelling is less than $\delta(d-1,k)+k-g$

Let Γ denote the graph defined by the currently known edges and vertices of a shelling. Initially, Γ only contains $\{u,v\}$, a pair of potential vertices. Let $d_\Gamma(x,y)$ denote the distance in Γ between two vertices x and y. The following are upper bounds for the distance between u or v and the intersection with a facet of the $[0,k]^d$:

$$\begin{aligned}
\widetilde{d}(u,F_i^-) &= \min_{w\in\Gamma}\{d_\Gamma(u,w)+w_i\} \qquad \text{for } i=1,\ldots,d,\\
\widetilde{d}(u,F_i^+) &= \min_{w\in\Gamma}\{d_\Gamma(u,w)+k-w_i\} \qquad \text{for } i=1,\ldots,d,\\
\widetilde{d}(v,F_i^-) &= \min_{w\in\Gamma}\{d_\Gamma(v,w)+w_i\} \qquad \text{for } i=1,\ldots,d,\\
\widetilde{d}(v,F_i^+) &= \min_{w\in\Gamma}\{d_\Gamma(v,w)+k-w_i\} \qquad \text{for } i=1,\ldots,d.
\end{aligned}$$

One can notice that setting w to u or v gives $\widetilde{d}(u,F_i^-)\le u_i$, $\widetilde{d}(v,F_i^-)\le v_i$, $\widetilde{d}(u,F_i^+)\le k-u_i$, and $\widetilde{d}(v,F_i^+)\le k-v_i$. This intuitively makes sense since a distance from a vertex to an intersection with the hypercube, F_i^- and F_i^+, cannot exceed the minimum of the value of the coordinate in said dimension, u_i or v_i, and $k-u_i$ or $k-v_i$.

The following quantity $d_\circ(u,v)$, where both $\delta(F_i^-)$ and $\delta(F_i^+)$ are bounded from above by $\delta(d-1,k)$, is an upper bound for $d(u,v)$ by inequality (2.1):

$$d_\circ(u,v)=\min_{i=1,\ldots,d}\{\min\{\widetilde{d}(u,F_i^-)+\widetilde{d}(v,F_i^-)+\delta(F_i^-),\widetilde{d}(u,F_i^+)+\widetilde{d}(v,F_i^+)+\delta(F_i^+)\}\}.$$

The value of $d_\circ(u,v)$ is updated every time a choice for the intersection with a facet of the hypercube $[0,k]^d$ is selected. Similarly, Γ can be considered a subgraph of the edge-graph of the final polytope, all vertices and edges of Γ will be vertices and edges in the graph of the final polytope. Therefore, $d_\Gamma(u,v)$ is another upper bound for $d(u,v)$. Thus, we define the following nonnegative parameter γ:

$$\gamma=\delta(d-1,k)+k-g-\min\{d_\Gamma(u,v),d_\circ(u,v)\}.$$

Hence, $\gamma > 0$ is a certificate that no lattice (d,k)-polytope with vertices u and v such that $d(u,v) = \delta(d-1,k) + k - g$ exists.

Certificate 2: u or v is not a vertex of $\mathscr{C}^{\Gamma}_{d,k,g}$

Every time a selection is made for an intersection with the $[0,k]^d$ hypercube, the vertex set of Γ is updated. Using $\mathscr{P}_{d,k,g}$, defined previously as the set of all integer points that belong to the intersection of all lattice (d,k)-polytopes of diameter at least $\delta(d,k) - g$, we can create a convex hull to determine whether u and v remain vertices after a selection has been made. Let $\mathscr{C}^{\Gamma}_{d,k,g}$ be the convex hull of the vertex set of Γ and the following set of points:

$$\left[\bigcup_{i=1}^{d}\{x \in \mathbb{R}^d : x_i = 0 \text{ and } \bar{x}_i \in \mathscr{P}_{d-1,k,g_i^-}\}\right] \cup \left[\bigcup_{i=1}^{d}\{x \in \mathbb{R}^d : x_i = k \text{ and } \bar{x}_i \in \mathscr{P}_{d-1,k,g_i^+}\}\right].$$

A certificate that there does not exist a lattice (d,k)-polytope with vertices u and v such that $d(u,v) = \delta(d-1,k) + k - g$ is:

$$u \text{ or } v \text{ is not a vertex of } \mathscr{C}^{\Gamma}_{d,k,g}.$$

Shelling step

Using a branch and bound approach and the key ideas described in Section 2.3.4, the shelling step attempts to build a polytope one facet at a time. For each facet, all $d-1$ faces are considered and eliminated based upon a set of rules. The output of this step is a set of shellings, sub-skeletons of polytopes, that achieve a target diameter. As input, this step accepts a triple, (d,k,g), and a pair of vertices, (u,v). The dimension is determined by d and the integer grid ranges from 0 to k in every dimension. The slack variable g determines the target diameter to look for, $\delta(d-1,k) + k - g$. Finally, the pair of vertices (u,v) provides a set of starting vertices that can potentially achieve the target distance in a lattice (d,k)-polytope.

First, each facet's score, g_i^- and g_i^+ is calculated. The ordering of intersections to be considered is determined using the rules described in Section 2.3.4. For each facet, F_i^- or F_i^+, the set of all lattice $(d-1,k)$-polytopes of diameter at least $\delta(d-1,k)-g_i^-$ or $\delta(d-1,k)-g_i^+$ respectively are generated. These polytopes are iteratively selected as a candidate for the intersection. We define $\bar{x}_i$ as the point in $d-1$ excluding the coordinate in the i^{th} dimension of the intersection being considered. For a polytope to be a valid candidate, it must contain vertices which are currently vertices of the shelling; that is, $\bar{x}_i$ must be a vertex if x is a vertex of P and $x_i = 0$ for F_i^- or $x_i = k$ for F_i^+.

After one such $(d-1,k)$-polytope is chosen, its vertices and edges are added to Γ. The values of γ, $\mathscr{C}^{\Gamma}_{d,k,g}$, and the scores, g_i^-, and g_i^+, of not yet considered intersections are all updated accordingly, to $g_i^- = g + \widetilde{d}(u,F_i^-) + \widetilde{d}(v,F_i^-) - k$ and $g_i^+ = g + \widetilde{d}(u,F_i^+) + \widetilde{d}(v,F_i^+) - k$.

The certificates, described in Section 2.3.4, are checked to see if the current subtree can contain an optimal solution, a full shelling containing u and v such that $d(u,v) = \delta(d-1,k) + k - g$. If either certificate of non-existence is fulfilled, then the subtree can be pruned and the search moves on to the next candidate for the current intersection. If neither certificate is fulfilled, then the algorithm moves on to the next intersection with the highest priority. Typically, the first chosen intersection will yield a certificate of non-existence.

At the conclusion of the algorithm, the output contains all full shellings that contain u and v such that $d(u,v) = \delta(d-1,k) + k - g$. These are valid polytopes that can achieve the target diameter, but may not be the complete set, up to symmetry, of all polytopes achieving this diameter. In order to determine the full set, up to symmetry, of polytopes with a non-empty intersection with the $[0,k]^d$ hypercube, we required the inner step. Should the output of the shelling step be the empty set, then it can be concluded that $\delta(d,k) < \delta(d-1,k) + k - g$.

Shelling process for $(d,k,g) = (3,6,0)$

An example is provided here to illustrate the shelling process for $(d,k,g) = (3,6,0)$. As discussed in Section 2.3.3, there are a total of 5 (u,v) pairs to consider in the shelling process. Since $g = 0$, all the intersections have a score of 0. The only currently known vertices are u and v and $u_1 = 0$ for all pairs. According to the tiebreaker rules established in Section 2.3.4, the first considered intersection is therefore F_1^-.

For each (u,v) pair, the set of lattice $(2,6)$-polytopes of diameter 6 containing (u_2,u_3) as a vertex are generated. There are only 4 different lattice $(2,6)$-polytopes, and it can be easily checked that $\gamma > 0$ for each such choice for all (u,v) pairs. Hence, since the certificate of non-existence is satisfied, the shelling step completes after considering all possibilities for F_1^- and it can be concluded that there are no lattice (d,k)-polytopes that have a diameter of 12 and therefore, $\delta(3,6) < 12$.

2.3.5 Inner step

The *inner step* function is a post shelling step that is performed to generate all lattice polytopes that achieve the desired diameter. This step is important since the main algorithm uses lattice $(d-1,k)$-polytopes as facets for the intersection of the hypercubes, and these facets are generated by applying the algorithm on the lattice $(d-1,k)$-polytopes.

As input, the function takes in the set of shellings generated by the shelling step. Then, for each shelling, all inner points (points with integer coordinates that lie inside of they hypercube and outside of the shelling), p such that $p_i \in \{1,...,k-1\}$ for $i = 1,...,d$ and $p \notin$ vertex set of Γ, are considered individually. Let $\mathscr{C}_{d,k,g}^{\Gamma \cup p}$ denote the convex hull of $p \cup \mathscr{C}_{d,k,g}^{\Gamma}$, then a necessary condition for p to be a vertex of a lattice (d,k)-polytope of diameter $\delta(d-1,k)+k-g$ is p is a vertex of $\mathscr{C}_{d,k,g}^{\Gamma \cup p}$.

Generated lattice (d,k)-polytopes whose diameter is at most $\delta(d-1,k)+k-g-1$ as well as duplicates, up to the symmetries of the hypercube $[0,k]^d$, are removed.

If the output of the inner step is empty, we can conclude that $\delta(d,k)$ is strictly less than $\delta(d-1,k)+k-g$. Otherwise, we can conclude that $\delta(d,k)=\delta(d-1,k)+k-g$, and the output of the inner step provides, up to the symmetries of the hypercube $[0,k]^d$, all lattice (d,k)-polytopes of diameter $\delta(d-1,k)+k-g$ whose intersection with each facet of the hypercube $[0,k]^d$ is nonempty. Further computations allow to determine all lattice (d,k)-polytopes of diameter $\delta(d-1,k)+k-g$ with an empty intersection with at least one facet of the hypercube $[0,k]^d$, as detailed in Section 2.3.6.

Inner step for $(d,k,g)=(3,4,2)$

Here is an example to illustrate the inner step for $(d,k,g)=(3,4,2)$ and the pair $\{u,v\}=\{(0,0,0),(4,4,4)\}$. This is equivalent to considering $u=(0,0,0)$ and $v=(4,4,4)$ as vertices of a lattice $(3,4)$-polytope such that $d(u,v)=6$.

First, the shelling step is performed to generate valid shellings. Since the scores satisfy $g_1^-=g_2^-=g_3^-=g_1^+=g_2^+=g_3^+=2$, the 6 intersections with the facets of $[0,4]^3$ are of diameter at least 2.

At the conclusion of the shelling step, one of the output candidates is a shelling consisting of the 6 identical facets as depicted in Figure 2.2 where the edges of the shelling are shown in blue. Each of these facets is, up to an affine transformation, equal to the square $[0,1]^2$. Out of the 27 points whose coordinates are $\{1,2,3\}$-valued, 15 are contained in the convex hull of this shelling.

Thus, the inner step must consider the remaining 12 inner points as possible vertices to be added to this shelling. There are a total of 214 lattice $(3,4)$-polytopes,

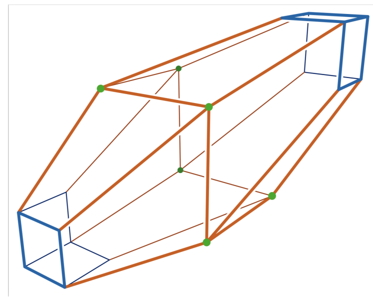

Figure 2.2: A polytope considered by the inner step for $(d,k,g) = (3,4,2)$ and $\{u,v\} = \{(0,0,0),(4,4,4)\}$

up to the symmetries of $[0,4]^3$, and all of them have a diameter of at most 5.

There are exactly 8 polytopes with a diameter equivalent to 5. One of these lattice $(3,4)$-polytopes of diameter 5 is represented in Figure 2.3 where the 6 added vertices are show in green and the edges of the intersections with the facets of $[0,4]^3$ are shown in blue. Note that since polytopes of diameter at most $\delta(d-1,k)+k-g-1 = 4+4-2-1 = 5$ are removed, none of the 214 lattice $(3,4)$-polytopes generated by this shelling are part of the output of the inner step for $(d,k,g) = (3,4,2)$.

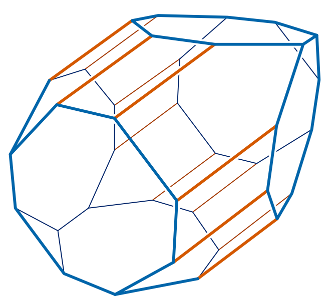

Figure 2.3: The unique shelling generated for $(d,k,g) = (3,4,0)$

2.3.6 Generation of all lattice (d,k)-polytopes of diameter at least $\delta(d-1,k)+k-g$

Running the main procedure for (d,k,g) allows us to determine, up to the symmetries of $[0,k]^d$, the set of all the lattice (d,k)-polytopes with diameter at least $\delta(d-1,k)+k-g$ whose intersection with each facet of $[0,k]^d$ is nonempty. In this section, we outline how the ones with an empty intersection with at least one facet of $[0,k]^d$ can be derived from this set.

The main idea behind generating all lattice (d,k)-polytopes with diameter at least $\delta(d-1,k)+k-g$ whose intersection with one of the facets of $[0,k]^d$ is empty is to use the idea from Lemma 2.2.4. We will attempt to expand the polytope in a direction, s, to check the possibility of contracting P in the direction of $-s$. If a possible contraction exists, we can then check the diameter of the resulting polytope to see if a new valid polytope of diameter at least $\delta(d-1,k)+k-g$ with an empty intersection with one of the facets of $[0,k]^d$.

Let $I(Q)$ denote the set of the coordinates i such that $\gamma_i^+(Q)-\gamma_i^-(Q)<k$. Consider a lattice (d,k)-polytope Q of diameter at least $\delta(d-1,k)+k-g$ such that $I(Q)\neq\emptyset$. For all $i\in I(Q)$, we can assume, up to translation, that $\gamma_i^-(Q)=0$ and consider the segment $\sigma^i=\operatorname{conv}\{0,(k-\gamma_i^+(Q))c^i\}$. Let S denote the Minkowski sum of all σ^i for $i\in I(Q)$. As shown in the proof of Lemma 2.2.4, $Q+S$ is a lattice (d,k)-polytope of diameter at least $\delta(Q)$ satisfying $I(Q+S)=\emptyset$. In other words, $Q+S$ is, up to the symmetries of $[0,k]^d$, in the output of the algorithm ran for (d,k,g). Note that setting $P^1=Q$ and $P^2=[0,s]$ where $s_i\geq 0$ for all i in Proposition 2.2.3 gives Remark 2.3.1.

Remark 2.3.1. *Consider a segment $\sigma=[0,s]$; a point v' is a vertex of $Q+\sigma$ if and only if there exists an objective function $c\in\mathbb{R}^d$ that is uniquely minimized at v in Q and (i) $v'=v$ and c is uniquely minimized at 0 in σ, or (ii) $v'=v+s$ and c is uniquely minimized at s in σ. Moreover, if u' and v' are adjacent vertices of*

$Q+\sigma$, then either (u',v') is equal to (u,v) or to $(u+s,v+s)$ where u and v are adjacent vertices of Q, or it is equal to $(u,u+s)$ where u is a vertex of Q.

Consequently, up to translation and up to the symmetries of the hypercube $[0,k]^d$, the set of the lattice (d,k)-polytopes Q of diameter at least $\delta(d-1,k)+k-g$ such that $I(Q) \neq \emptyset$ can be generated as follows:

1. for each lattice (d,k)-polytope P in the output of the algorithm for (d,k,g), check whether $P = Q+\sigma$ where Q is a lattice (d,k)-polytope and σ a lattice segment. By Remark 2.3.1, this can be done by checking whether P and $P+\sigma$ have the same number of vertices.

2. for each P such that $P = Q+\sigma$ found at step (i), determine Q and check whether $\delta(Q) \geq \delta(d-1,k)+k-g$.

As for the shelling and inner steps, the symmetries of the hypercube $[0,k]^d$ are used to remove duplicates generated within steps (i) and (ii). The set of lattice segments σ considered in step (i) can be limited to a few segments whose coordinates are relatively prime and used iteratively. For an illustration, we consider the case $(d,k,g) = (3,3,1)$. As discussed in Section 2.4.3, the output of the algorithm consists in 9 lattice $(3,3)$-polytopes of diameter 6 whose intersection with each facet of $[0,3]^3$ is nonempty. One can check that, in order to perform step (i), it is enough to consider for σ, iteratively, the 3 unit vectors $([0,0,1],[0,1,0],[1,0,0])$ and the unique sums of pairs of unit vectors $([0,1,1],[1,0,1],[1,1,0])$. All the 9 considered lattice $(3,3)$-polytopes of diameter 6 can be written as $Q+\sigma$. Performing step (ii), one can check that $\delta(Q) = 5$ for each such Q. Thus, there is no lattice $(3,3)$-polytope Q of diameter 6 such that $I(Q) \neq \emptyset$.

2.3.7 Implementation details

In this section, we describe the implementation (written in *C#*) of our algorithm in more detail. All experiments were run on a MacBook Pro with a 2.8GHz i7 processor and 16GB of RAM. The implementation adheres to Object Oriented

Programming principles by applying proper encapsulation, abstraction, and polymorphism as well as ensuring loose coupling and high cohesion.

Data Structures

The main data structures we use are a Point class and a Graph class.

The Point class represents an individual point (or vertex) and contains a list of integer values corresponding to its coordinates. It contains the necessary accessors and setters in addition to helper methods to perform actions including: comparing with another Point, increasing/decreasing the dimension, incrementing the value of a coordinate, and checking whether it shares a facet with another Point.

The Graph class is used to represent facets, shellings, and polytopes. It consists two fields, a list of points representing its vertices, and a dictionary object containing the adjacency list. A few key methods of this class included adding a new graph to the current object (used to add facets during the shelling step), shallow and deep clones of the current graph, and increasing the dimension of the current graph. During each of these steps, we ensure consistency and accuracy by removing duplicate points and edges, ensuring the adjacency list remains current, and updating all instances of a Point when it is modified.

Classes

Following a loose coupling design, we create a number of helper classes that each perform a set of highly cohesive actions. These classes include a FileIO class, a UVGeneration class, a Shelling class, and a CDD wrapper class.

The FileIO class provides a number of functionality related to reading and writing various objects (graphs, vertices, adjacency lists, etc.) from/to a file. These methods ensure that objects are represented in a standardized and consistent format.

The UVGeneration class implements the ideas described in Section 2.3.3. Logging is enabled to verify that all candidate pairs are checked as well as the reason for each candidate's elimination. Candidate pairs are iteratively checked and eliminated if they meet any of the following criteria:

- u and v are the same point
- the reflection of v has already been considered
- $\{\bar{u}_i \notin \mathscr{V}_{d-1,k,g_i^-}$ if $u_i = 0\}$ or $\{\bar{v}_i \notin \mathscr{V}_{d-1,k,g_i^+}$ if $v_i = k\}$

The Shelling class contains all functionality related to the main shelling algorithm including the inner step. Traversal of the search tree is implemented as a combination of iterative and recursive steps. The intersections with the $[0,k]^d$ hypercube are considered recursively, and for each intersection, candidate facets of dimension $(d-1)$ are considered iteratively. The initial implementation included generation of lattice $(d-1,k)$-polytopes, but this was revised in later versions to use a pre-built index of polytopes for performance purposes.

The CDD wrapper classes is used to connect with the CDD library [20] to perform convex hull calculations. As an argument, a list of vertices is accepted and converted to the $V-representation$ format expected by the CDD library. By starting a new process of the compiled CDD executable and passing in the $V-representation$ list of vertices, the CDD library calculates the convex hull and returns the list of vertices and adjacency list. This can then be compared to the input list of vertices to determine whether there are redundant vertices in the initial list. This part is not optimized for performance since each time this method is called, a new process is spawned and scalability may be an issue.

Additionally, a global config file is used to store parameters including d, k, and g, as well as other tunable settings (e.g. custom (u,v) pair).

2.4 Results

In this section, we detail the results that were obtained using the algorithm described in section 2.3. The main result is theorem 2.4.1 which provided new values for $\delta(5,3)$ and $\delta(3,6)$.

Theorem 2.4.1. *$\delta(5,3)$ and $\delta(3,6)$ are equal to* 10*.*

Theorem 2.4.1 is obtained by computationally verifying that the output of the inner step is empty for $(d,k,g) = (3,6,1)$ and $(5,3,0)$. Thus, $\delta(3,6) < 11$ and $\delta(5,3) < 11$; and since we know that the lower bound for both $\delta(3,6)$ and $\delta(5,3)$ is 10, therefore, $\delta(3,6) = \delta(5,3) = 10$. Running the algorithm for $(5,3,0)$ requires the determination of all lattice $(3,3)$-polytopes of diameter 5 or 6 and all lattice $(4,3)$-polytopes of diameter 8.

2.4.1 Determination of $\delta(3,6)$

As mentioned in Section 2.3.4, the output of the shelling step is empty for $(d,k,g) = (3,6,0)$ and thus we can conclude that $\delta(3,6) < 12$. Running the algorithm for $(d,k,g) = (3,6,1)$ is computationally efficient because of two key properties.

First, there are only 4 lattice $(2,6)$-polytopes of diameter 6, see Figure 2.4 for an illustration.

Second, for $d = 2$, there are only 8 lattice edges σ such that $|\sigma_1| + |\sigma_2| \leq 2$. Thus, any lattice $(2,6)$-polytope of diameter 5 or 6 includes at least 2 edges such that $|\sigma_1|$ or $|\sigma_2|$ is at least 2.

Consequently, unless both u and v are inner points, the update of the scores $g_i^- = g + \widetilde{d}(u,F_i^-) + \widetilde{d}(v,F_i^-) - k$, respectively of $g_i^+ = g + \widetilde{d}(u,F_i^+) + \widetilde{d}(v,F_i^+) - k$, implies that g_i^- or g_i^+ is updated to zero for some i after the first intersection with a facet of $[0,6]^3$ is considered in the shelling step. As $g_i^- = 0$ or $g_i^+ = 0$ implies that $\delta(F_i^-) = 6$ or $\delta(F_i^+) = 6$, respectively, there are at most 4 lattice $(2,6)$-polytopes to consider for the next intersection with a facet of $[0,6]^3$, and so forth.

As an illustration, consider the pair $\{u,v\} = \{(0,0,0),(6,6,6)\}$. Initially, the

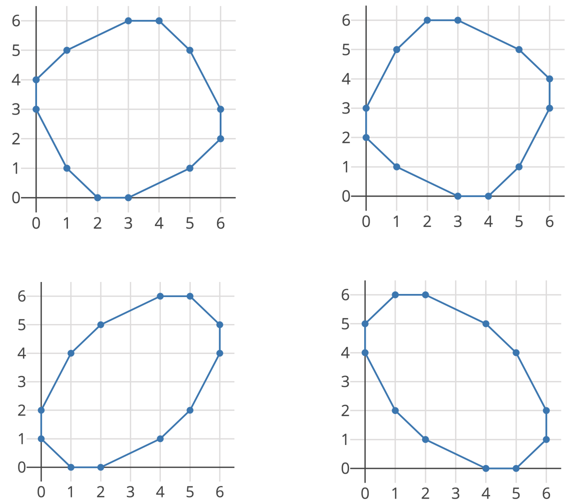

Figure 2.4: All lattice $(2,6)$-polytopes of diameter 6

scores satisfy $g_1^- = g_2^- = g_3^- = g_1^+ = g_2^+ = g_3^+ = 1$ and the shelling step starts by considering a lattice $(2,6)$-polytope of diameter at least 5 for F_1^-.

For example, assume that F_1^- is, up to an affine transformation, the lattice $(2,5)$-polytope obtained as the Minkowski sum of $(1,0),(2,1),(1,1),(1,2)$, and $(0,1)$. Before the next intersection with a facet of $[0,6]^3$ is considered, $\tilde{d}(u,F_2^+)$ is updated to 5 as $d(u,u')=2$ and $u'_2=3$, see Figure 2.5 where the vertex u' is coloured black while u and v are coloured red. The second edge on the path from u to u' satisfies $e_2 \geq 2$. Consequently, g_2^+ is updated to $g+\widetilde{d}(u,F_2^+)+\widetilde{d}(v,F_2^+)-6 = 1+5+0-6=0$. Thus, $\delta(F_2^+)=6$ which is impossible since $\bar{v}_2=(6,6) \notin \mathscr{V}_{2,6,0}$; that is, there is no shelling with the current choice of F_1^-.

The same holds for any choice of F_1^- since any lattice $(2,6)$-polytope of diameter at least 5 includes at least one edge σ such that $|\sigma_1|$ or $|\sigma_2|$ is at least 2. Consequently, there is no shelling for $\{u,v\}=\{(0,0,0),(6,6,6)\}$.

Table 2.3 lists the 69 considered pairs $\{u,v\}$ of vertices of a lattice $(3,6)$-polytope P such that $d(u,v)=11$ where P is assumed to have a nonempty intersection with each facet of $[0,6]^3$.

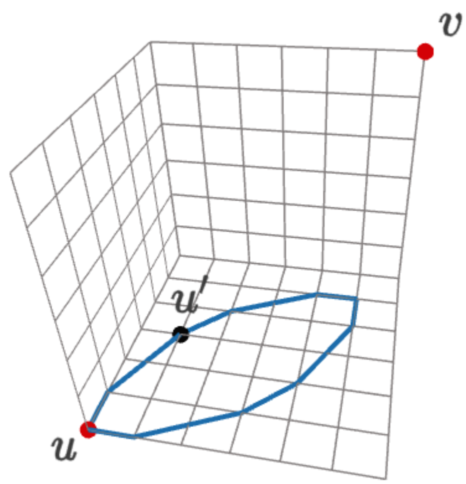

Figure 2.5: Initial iteration of the shelling step for $(d,k,g) = (3,6,1)$ and $\{u,v\} = \{(0,0,0),(6,6,6)\}$

u	v
(0,0,0)	(6,6,6)
(0,0,1)	(5,5,4), (5,5,5), (5,5,6), (5,6,4), (5,6,5), (5,6,6), (6,6,4), (6,6,5)
(0,0,2)	(5,5,3), (5,5,4), (5,5,5), (5,6,3), (5,6,4), (5,6,5), (6,6,3), (6,6,4)
(0,0,3)	(5,5,2), (5,5,3), (5,5,4), (5,6,2), (5,6,3), (5,6,4), (6,6,2), (6,6,3)
(0,1,1)	(5,4,4), (5,4,5), (5,4,6), (5,5,5), (5,5,6), (6,4,4), (6,4,5), (6,5,5)
(0,1,2)	(5,4,3), (5,4,4), (5,4,5), (5,5,3), (5,5,4), (5,5,5), (5,6,4), (6,4,3), (6,4,4), (6,4,5), (6,5,3), (6,5,4)
(0,1,3)	(6,4,2), (6,4,3), (6,4,4), (6,5,2), (6,5,3), (6,6,2)
(0,2,2)	(6,3,4), (6,4,4)
(0,2,3)	(6,3,2), (6,3,4), (6,4,2), (6,4,3), (6,5,2)
(1,1,1)	(4,5,5), (5,5,5)
(1,1,2)	(5,5,3), (5,5,4)
(1,1,3)	(5,5,2), (5,5,3), (5,6,2), (6,6,2)
(1,2,2)	(5,4,4)
(1,2,3)	(6,5,2)
(2,2,3)	(4,5,2)

Table 2.3: All considered pairs $\{u,v\}$ for $(d,k,g) = (3,6,1)$

2.4.2 Determination of $\delta(5,3)$

The determination of $\delta(5,3)$ requires the list of all lattice $(4,3)$-polytopes of diameter 8 up to the symmetries of $[0,3]^4$. In order to obtain all lattice $(4,3)$-polytopes of diameter 8, we first determine all lattice $(4,3)$-polytopes of diameter 8 with a nonempty intersection with each facet of $[0,3]^4$ by running the algorithm for $(d,k,g) = (4,3,1)$. Then, using the procedure described in Section 2.3.6, we can use the output of the algorithm for $(d,k,g) = (4,3,1)$ to determine all the lattice $(4,3)$-polytopes of diameter 8 with an empty intersection with at least one facet of $[0,3]^4$.

Note that running the algorithm for $(d,k,g) = (4,3,1)$ requires the list of all the lattice $(3,3)$-polytopes of diameter 5 or 6. This is achieved by running the algorithm for $(d,k,g) = (3,3,2)$ and using the procedure described in Section 2.3.6.

Table 2.4 lists the 6 considered pairs $\{u,v\}$ of vertices of a lattice $(3,3)$-polytope P such that $d(u,v) = 6$ where P is assumed to have a non-empty intersection with each facet of $[0,3]^3$.

u	v
(0,0,0)	(3,3,3)
(0,0,1)	(2,3,2), (2,3,3), (3,3,1), (3,3,2)
(0,1,1)	(3,2,2)

Table 2.4: All considered pairs $\{u,v\}$ for $(d,k,g) = (3,3,1)$

2.4.3 All lattice $(3,3)$-polytopes of diameter 6

As discussed in section 2.3.6, we provide a method to generate all lattice (d,k)-polytopes achieving at least a given diameter, up to symmetry. In this section, we present all 9 lattice $(3,3)$-polytopes of diameter 6, shown in Figure 2.6, up to the symmetries of $[0,3]^3$. In this figure, the edges of the intersections with the facets of $[0,3]^3$ are shown in blue. Using the procedure described in Section 2.3.6, one can check that there is no lattice $(3,3)$-polytope of diameter 6 with an empty intersection with at least one facet of $[0,3]^3$. In other words, any lattice $(3,3)$-polytope of diameter 6 is, up to the symmetries of $[0,3]^3$, one of the 9 polytopes depicted in Figure 2.6.

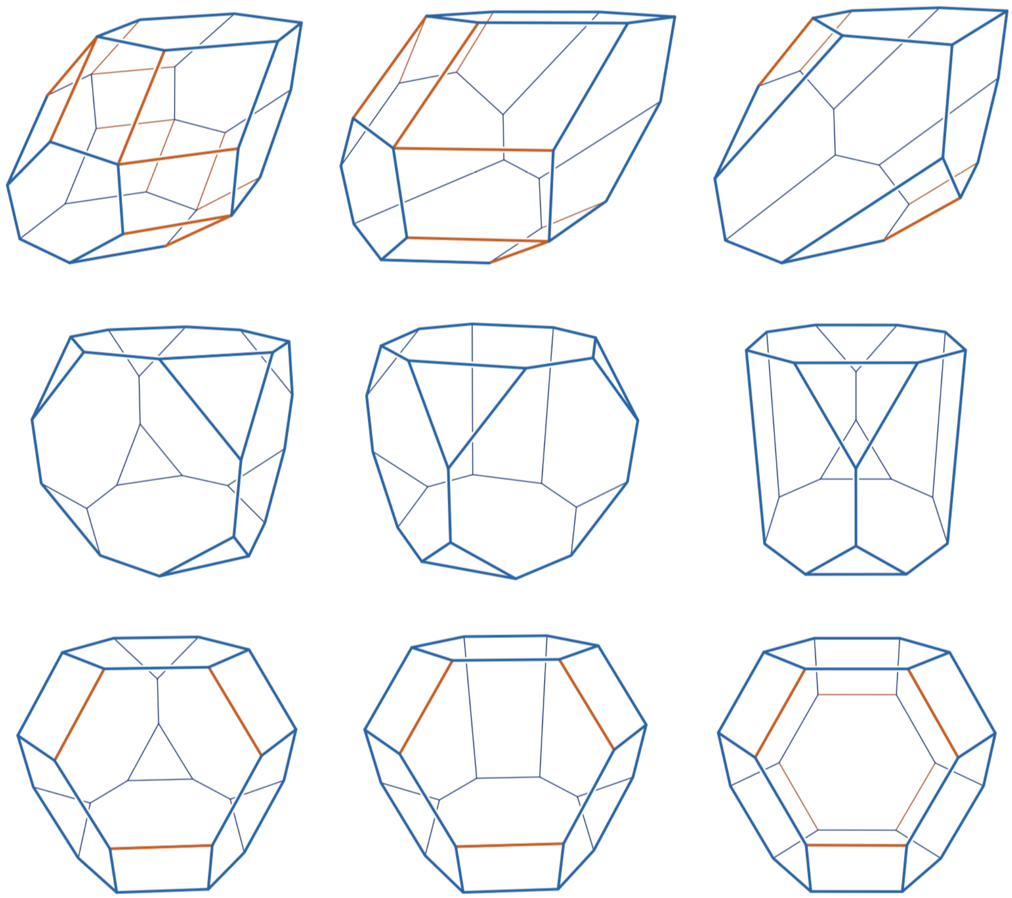

Figure 2.6: All, up to the symmetries of $[0,3]^3$, lattice $(3,3)$-polytopes of diameter 6

Table 2.5 provides the numbers $f_0(P)$ and $f_2(P)$ of vertices and facets of the 9 polytopes represented in Figure 2.6. The breakdown by incidence is also indicated.

For example, $24\{3\}$ and $8\{3\}+6\{8\}$ indicates that the truncated cube P_4 has 24 vertices, all belonging to 3 facets, and 14 facets consisting in 8 triangles and 6 octagons.

Polytope	$f_0(P)$	Vertex incidence	$f_2(P)$	Facet incidence
P_1	26	20{3}+6{4}	18	12{4}+6{6}
P_2	23	20{3}+3{4}	15	9{4}+6{6}
P_3	20	20{3}	12	6{4}+6{6}
P_4	24	24{3}	14	8{3}+6{8}
P_5	22	22{3}	13	6{3}+1{4}+2{6}+4{8}
P_6	20	20{3}	12	4{3}+2{4}+4{6}+2{8}
P_7	24	24{3}	14	4{3}+3{4}+4{6}+3{8}
P_8	22	22{3}	13	2{3}+4{4}+6{6}+1{8}
P_9	24	24{3}	14	6{4}+8{6}

Table 2.5: Some combinatorial properties of the lattice $(3,3)$-polytopes with maximal diameter.

2.4.4 All lattice $(3,2)$-polytopes of diameter 4

Note first that a lattice $(3,2)$-polytope with an empty intersection with at least one of the facet of $[0,2]^3$ is either a hexagonal prism or a lattice $(3,2)$-polytope of diameter at most 3. Thus, the hexagonal prism depicted in Figure 2.7 in the middle of the top row is, up to the symmetries of $[0,2]^3$, the unique lattice $(3,2)$-polytope of diameter 4 with an empty intersection with at least one facet of $[0,2]^3$.

The set $\mathscr{V}_{3,2}$ of all the vertices of all the lattice $(3,2)$-polytopes of diameter 4 consists of all $\{0,1,2\}$-valued points except $(1,1,1)$. This point forms the intersection of all lattice $(3,2)$-polytopes of diameter 4; that is, $\mathscr{P}_{3,2} = \{(1,1,1)\}$.

There are 3, up to the symmetries of $[0,2]^3$, lattice $(3,2)$-polytopes of diameter 4 with 15 vertices which are depicted in the bottom row of Figure 2.7 where the edges of the intersections with the facets of $[0,2]^3$ are shown in blue. The unique, up to the symmetries of $[0,2]^3$, lattice $(3,2)$-polytope of diameter 4 with 11, respectively 16, vertices is represented leftmost, respectively rightmost, in the top row.

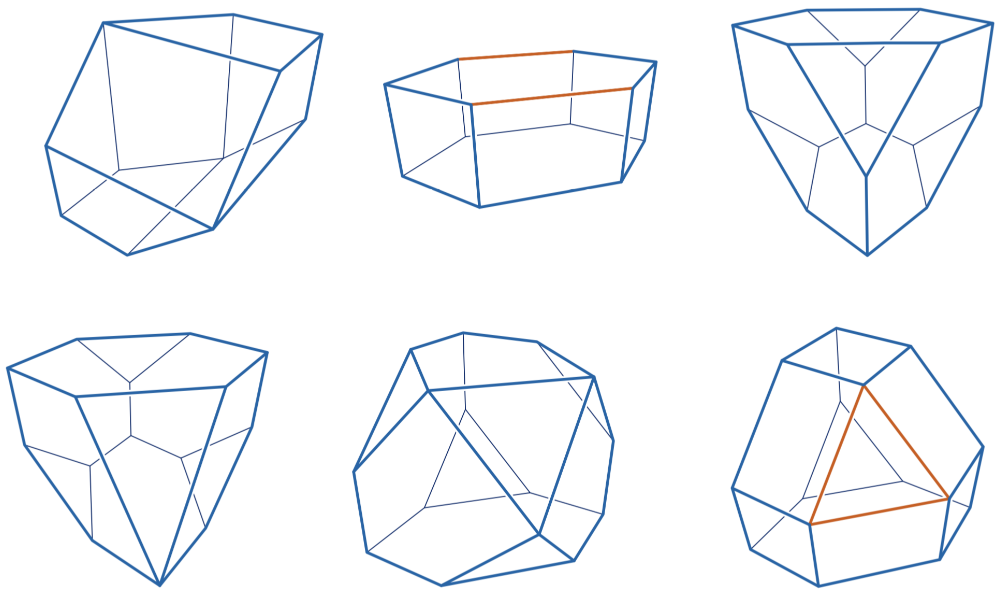

Figure 2.7: All, up to the symmetries of $[0,2]^3$, lattice $(3,2)$-polytopes of diameter 4 with 11, 15, or 16 vertices, or with an empty intersection with at least one facet of $[0,2]^3$

2.5 Future Work

There are still several directions that this research can go into to obtain more results.

From a theoretical perspective, using our proposed algorithm, all polytopes achieving the maximal diameter can be found. Hence, more analysis could be performed to explore the possibility of attributes or characteristics that are common to all of these polytopes. Should these unique properties exist, then this could lead to more results tightening the bounds as well as more efficient computational algorithms.

Another research direction would be to adapt our framework to get polytopes with large diameter with respect to n under the optimistic assumption that counterexamples to the Hirsch conjecture exist for relatively small d and k. Exploring such directions would involve considering all 3 parameters (d,k,n) together.

Due to the exponential nature of the search space, the problem quickly becomes intractable as we increase d and k. One area of future work is to determine additional properties that can be used to create heuristics to further increase the scalability of the algorithm in order to solve for larger values.

On the computational side, parallelization and cloud computing can be leveraged to handle the various levels of concurrency in this algorithm and improve on the current implementation. This could significantly increase the performance and enable the solving of instances with larger values of d and k. Since the algorithm is based on a branch and bound approach, branches performing computations of sub-problems (i.e. considering different facets for an intersection with the $[0,k]^d$ hypercube) are independent of each other and can be run on different threads/cores. There would be minimal communication between these processes so the overhead involved should not be a significant bottleneck.

Chapter 3

Vertex enumeration algorithm for primitive zonotopes

This chapter focus on a specific case of lattice polytopes called primitive zonotopes; that is, Minkowski sums of segments with integer-valued vertices which are primitives. A point or a vector is called *primitive* if the greatest common divisor of the coordinates is equal to 1. The zonotope Z associated to the set of line-segments G, also called the set of *generators* of Z, is defined as the convex hull of all the possible $2^{|G|}$ subsums of element of G.

For example, let $G = \{g^1, g^2, g^3\} = \{[0,1],[1,0],[1,1]\}$, then we have we consider the $2^3 = 8$ possible subsums $\varepsilon_1 g^1 + \varepsilon_2 g^2 + \varepsilon_3 g^3$ with $\varepsilon_i \in \{0,1\}$ for $i = 1,2$ and 3. The 8 subsums yield the following 7 points

$$\{(0,0),(0,1),(1,0),(1,1),(1,2),(2,1),(2,2)\}$$

as $(1,1)$ is repeated since it is realized with both $(\varepsilon_1, \varepsilon_2, \varepsilon_3) = (1,1,0)$ and $(\varepsilon_1, \varepsilon_2, \varepsilon_3) = (0,0,1)$. Taking the convex hull of these 7 points yields the following 6-gon as

$(1,1)$ is not a vertex since $(1,1)$ is the midpoint between $(0,0)$ and $(2,2)$

$$\{(0,0),(0,1),(1,0),(1,2),(2,1),(2,2)\}.$$

Note that the diameter of a Minkowski sum is equal to the number of its pairwise linearly independent generators. Thus, if we assume without loss of generality that the first nonzero coordinate of any generator is positive, the diameter of a primitive zonotope is equal to the number of its generators. Primitive (d,k)-zonotopes are conjectured to achieve the largest diameter over all lattice (d,k)-polytopes, that is, $\delta(d,k) = \delta_z(d,k)$. Note that the conjecture holds for all known entries of $\delta(d,k)$ and for $(2,k)$ for all k, $(d,1)$ and $(d,2)$ for all d, $(3,k)$ for $k \leq 6$, and $(d,3)$ for $d \leq 5$. The value of $\delta_z(d,k)$ was determined for any (d,k) by Deza, Pournin, and Sukegawa [18] who showed that, up to an explicit multiplicative constant, $\delta_z(d,k)$ grows like $k^{d/d-1}$ when d is fixed and k goes to infinity. Since $\delta_z(d,k) \leq \delta(d,k)$, this result provides a new lower bound for $\delta(d,k)$.

One of the well-studied primitive zonotope is the so-called *white whale* that corresponds to the Minkowski sum of all the the 2^d-1 nonzero vectors with $\{0,1\}$-valued coordinates. Let H_d denotes the convex hull of all the possible 2^{2^d-1} subsums of these 2^d-1 nonzero vectors with $\{0,1\}$-valued coordinates. The number of vertices of H_d is called $a(d)$. For instance, the 6-gon describes above corresponds to H_2 and its number of vertices is $a(2) = 6$, see Figure 3.1 for an illustration. Note that H_d is a centrally symmetric around $\sigma_d = 2^{d-2}(1,1,\dots,1)$, and that H_d is a lattice $(d,2^{d-1})$-polytope of diameter 2^d-1.

In this chapter, we present our contributions to the effort of determining the value of $a(d)$ for small d by developing a computational framework that exploit the combinatorial and geometric structure of the input. In Section 3.1, we recall a few properties and results concerning $H(d)$ and $a(d)$. Section 3.2 describes different areas where this zonotope arises including quantum field theory, psychometrics, and combinatorics. Section 3.3 describes our contributions which include defining new properties, designing new heuristics, and presenting a new algorithm that is used to efficiently verify existing results. While the framework is currently not able to determine new values for $a(d)$, previously known entries were efficiently recomputed until $d = 8$. Section 3.4 lists the results that have been obtained. We conclude this chapter with on-going and future research directions in Section 3.5.

3.1 Sizing the white whale

The quantity $a(d)$ arises in several contexts, see [5, 6, 19, 26, 35]. For instance, it appears in quantum field theory as the number of generalized retarded functions on $d+1$ variables [19] and in combinatorics as the number of maximal unbalanced families of subsets of $\{1,2,...,d+1\}$ [5]. The values of $a(d)$ have been computed up to $d = 8$ [19, 26, 45], and can be found in the Online Encyclopedia of Integer Sequences as sequence A034997. We report them in Table 3.1.

d	$a(d)$
1	2
2	6
3	32
4	370
5	11 292
6	1 066 044
7	347 326 352
8	419 172 756 930

Table 3.1: Number of vertices of H_d for d at most 8

It is shown in [5] that

$$\prod_{i=0}^{d-1}\left(2^i+1\right) \leq a(d) < 2^{d^2}.$$

The lower bound on $a(d)$; that is the size of the H_d which is also called the white whale by Billera [4], has been significantly improved by Gutekunst, Mészáros, and Petersen [22] who showed that the upper bound is the right asymptotic estimate. Note that the $a(d)$ is the sum of the Betti numbers of its dual hyperplane arrangement. The first two non-trivial of these Betti numbers have been determined by Kühne [30]. The upper bound was slightly improved by Deza, Pournin, and Rakotonarivo [17].

Consequently the current best upper and lower bounds are :

$$\frac{(d+1)}{2^{d+1}}2^{d^2(1-\frac{10}{\ln d})} \leq a(d) < \frac{2(d+4)}{2^{3d-2}}2^{d^2}.$$

Note that, same as the number of vertices of any zonotope, $a(d)$ is even and that, in addition, $a(d)$ is a multiple of $d+1$, see [5]. In the dual setting, H_d corresponds to the central hyperplane arrangement consisting of all the $2^d - 1$ hyperplanes which normal vector is a nonzero $\{0,1\}$-valued vector. This arrangement is called the resonance arrangement and its regions are in a one-to-one relation with the vertices of H_d. For instance, in dimension $d = 2$, the 3 hyperplanes defined by the equalities $x_1 = 0, x_2 = 0$ and $x_1 + x_2 = 0$ form $a(2) = 6$ regions.

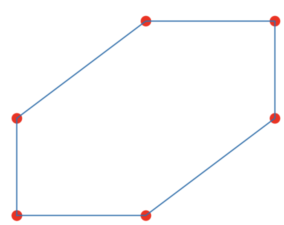

Figure 3.1: The 2-dimensional white whale H_2

3.2 A few appearances of the white whale

The problem that we are investigating is concurrently being studied in several related areas. Here, we review examples from three different fields including: quantum field theory, psychometrics, and combinatorics.

• QUANTUM FIELD THEORY

Quantum field theory is an important subfield of theoretical physics focusing on the understanding of the microscopic world. It combines ideas from classical field theory, special relativity, and quantum mechanics to construct physical models of subatomic particles. Particles are treated as excited states (also called quanta) of their underlying fields, which are more fundamental than the basic particles. Interactions between particles are described by interaction terms in the Lagrangian involving their corresponding fields. Each interaction can be visually represented by Feynman diagrams, which are formal computational tools, in the process of relativistic perturbation theory. For a more detailed description of the area, refer to [24].

Within quantum field theory, there are a set of methods used to calculated expectation values of physical observables at a finite temperature. This is the subfield of thermal quantum field theory. Green functions are key tools in quantum field theory. They are generated by taking functional derivatives with respect to the sources of the generating functional and then setting unphysical sources to 0. The retarded and advanced Green functions are decompositions of the Green function. They appear in the linear response approximation which tries to describe small perturbations to a plasma. The retarded functions are usually written as expectation values of multiple commutators, with theta functions. The number of generalized retarded functions on $d+1$ variables [19] is equal to $a(d)$.

• PSYCHOMETRICS

Psychometrics is the field of study concerning the theory and measurement of mental capacities and processes. Introduced originally in the field of psychometrics, the unfolding model is used for preference ranking. Since its introduction, the model has been used in many other fields including marketing science [13] and voting theory [23].

Suppose there are m unique items labeled $1,2,\ldots,m$ and an individual is ranking these objects based on their preference. In the unfolding model, these m objects are represented by the points $\mu_1, \mu_2, \ldots$ in Euclidean space $\mathbb{R}^n$. The individual is represented by the point y in the same space, commonly referred to as the joint space. The individual, y, prefers object i to object j if and only if y is closer to μ_i than to μ_j. Hence, for a given y, a ranking of $(i_1, i_2, \ldots, i_m)$ is produced, where i_1 denotes the most preferred item for the individual and i_m, the least preferred.

In general, there are $m!$ rankings among m objects, but in the unfolding model, some rankings can not be generated. Admissible rankings are rankings for which there exists a y in the joint space with that ranking. Inadmissible rankings are those where there does not exist a y in the joint space with that ranking. Take for example, the case where $m = 3$ and $n = 1$. Suppose $\mu_1 = 0$, $\mu_2 = 4$, and $\mu_3 = 7$. If $y < 2$, then the ranking is $(1,2,3)$. If $2 < y < 3.5$, then the ranking is $(2,1,3)$. If $3.5 < y < 5.5$, the ranking is $(2,3,1)$. The final scenario is when $y > 5.5$, in which case the ranking is $(3,2,1)$. So the admissible rankings are $(1,2,3),(2,1,3),(2,3,1),(3,2,1)$ and the inadmissible rankings are $(1,3,2),(3,1,2)$.

The number of possible sets of rankings is called the ranking pattern. In general, depending on the choices of $\mu_1,...,\mu_m$, the resulting ranking pattern will differ. The problem is to determine how many distinct ranking patterns there are for a particular m and n. If an additional constraint is added of the unfolding model being of co-dimension one, i.e. $n = m+2$, then the result is equivalent to $a(d)$, see [27].

• COMBINATORICS

In the area of combinatorics, a family of subsets of natural numbers up to n is considered to be balanced if there exists a convex combination of their characteristic functions which is constant. Otherwise, they are an unbalanced family. A family is considered to be maximally unbalanced if in addition to being unbalanced, any other family that is not a proper superset of it is balanced. Let χF denote the characteristic function of F. Suppose $n = 3$, then the family $\{F_1 = \{1\}, F_2 = \{1,2\}, F_3 = \{1,3\}\}$ is maximally unbalanced where $\chi F_1 =$

$(1,0,0), \chi F_2 = (1,1,0)$, and $\chi F_3 = (1,0,1)$. The family $\{F_1 = \{1\}, F_2 = \{1,2\}\}$ is unbalanced, but not maximally unbalanced, since it is a subset of the previous family. The number of maximal unbalanced families of subsets of a d element set is equal to $a(d)$, see [4].

3.3 Orbitwise enumeration of the vertices of H_d

In this section, we present a computational approach to generate, up to symmetries, all the vertices of H_d and thus to determine $a(d)$. The input to the algorithm is the set of all generators, $G_d = \{g^1, g^2, \ldots, g^{2^d-1}\}$ and the list of all the canonical vertices of H_d and, by action of the symmetry group on these canonical vertices, the value of $a(d)$.

Note that the origin is always a vertex of H_d. Starting from the origin, the algorithm visit all possible neighbours of the currently know canonical vertices obtained by adding a generator. Elementary checks are used to remove points from the list of potential vertices. Then a certificate sufficient to show that a point is not a vertex is used to further reduce the search space. A certificate sufficient to show that a point is vertex to obtain the final list. Symmetries and combinatorial properties are used achieve higher computational efficiency.

We recall a few elementary properties of H_d:

Lemma 3.3.1.

(i) *H_d is a centrally symmetric lattice $(d, 2^{d-1})$-polytope.*

(ii) *H_d is invariant under the symmetries induced by the permutations of the coordinates and by the reflection centered on $\sigma_d = 2^{d-2}(1,1,\ldots,1)$.*

(iii) *The intersection of H_d with any facet of the cube $[0, 2^{d-1}]^d$ is, up to symmetry and translation, equal to H_{d-1}.*

Proof. Recall that any zonotope is centrally symmetric around the midpoint between the origin and the point corresponding to the sum of all generator. Since $\sum_{i=1}^{2^d-1} g^i = 2^{d-1}(1,1,\ldots,1)$, the center of symmetry of H_d is $\sigma_d = 2^{d-2}(1,1,\ldots,1)$ and H_d is a lattice $(d,2^{d-1})$-polytope. Since the set of generators, G_d, is invariant under the symmetries induced by the permutations of the coordinates, the same applies to H_d. Since, up to truncating the last coordinate, the $2^{d-1}-1$ generators of G_d having zero as last coordinate form G_{d-1}, the intersection of H_d with the hyperplane $x_d = 0$ is, up to truncating the last coordinate, equal to H_{d-1}. The same holds for any other intersection of the cube $[0,2^{d-1}]^d$ by the action of the symmetries of H_d. $\square$

• CANONICAL VERTICES OF H_d

The vertices of H_d can be partitioned into equivalence classes, or orbits, by the action of the group of symmetries. By item (i) and (ii) of Lemma 3.3.1, each class can be represented by following the canonical vertex satisfying:

- $x_i \leq x_{i+1}$ for $i = 1,\ldots,d-1$,
- $\sum_{i=1}^{d} x_i \leq d2^{d-2}$, and if $\sum_{i=1}^{d} x_i = d2^{d-2}$, the number of coordinates larger than 2^{d-2} is at most the number of coordinates smaller than 2^{d-2}.

• LIFTING OPERATION

By item (iii) of Lemma 3.3.1, a point of H_d with zero as last coordinate is a vertex of H_d if and only if truncating the last coordinate yields a vertex of H_{d-1}. Consequently, we can define the 0-lifting of a vertex $v = (v_1,v_2,...,v_{d-1})$ of H_{d-1} as the vertex $v = (v_1,v_2,...,v_{d-1},0)$ of H_d. Consequently, assuming that the vertices H_{d-1} have been determined, our objective is to determine all the vertices of H_d with coordinates satisfying $0 < x_i < 2^{d-1}$ for $i = 1,\ldots,d$.

• CERTIFICATES OF NON-VERTEXHOOD

By convexity p, a point of a polytope P is a vertex of P if and only if

$$\{p = \frac{p^1+p^2}{2} \text{ with } p^1 \in P \text{ and } p^2 \in P\} \Longrightarrow \{p^1 = p^2 = p\}$$

Using a few points of H_d known to be vertices of H_d for any d, we can consider a few simplices that belong to H_d and membership to this cone can be easily computed. Any point that belong to such simplex without being one of its vertices can not be a vertex H_d. Consider the following example: the origin and $2\sigma(d) = 2^{d-1}(1,\dots,1)$ are obvious symmetric vertices of H_d, thus, $2^{d-2}(1,\dots,1,0)$ is also a vertex of H_d as the 0-lifting of the vertex $2\sigma(d-1)$ of H_{d-1}. Thus, the simplex with $(0,\dots,0)$ as apex and the d permutations of $2^{d-2}(1,\dots,1,0)$ belongs to H_d. This simplex has $d+1$ facets and thus is defined as the set of points satisfying the following $d+1$ inequalities :

$$\sum_{j=1}^{d} x_j \leq (d-1)2^{d-2}$$

$$(d-2)x_i - \sum_{j=1, j\neq i}^{d} x_j \leq 0 \text{ for } i = 1,\dots,d$$

Membership to this simplex amounts to checking whether any of the $d+1$ inequalities is violated, and thus is computationally cheap. For example consider $d = 4$, the simplex is defined by:

$$x_1 + x_2 + x_3 + x_4 \leq 12$$

$$2x_1 - x_2 - x_3 - x_4 \leq 0$$

$$-x_1 + 2x_2 - x_3 - x_4 \leq 0$$

$$-x_1 - x_2 + 2x_3 - x_4 \leq 0$$

$$-x_1 - x_2 - x_3 + 2x_4 \leq 0$$

Thus, the point $(2,2,3,4)$ can not be a vertex of H_4 as it violated the last inequality.

While this type is certificate is valid for any polytope, we can consider additional certificate that are applicable to Minkowski sums. In particular, any face of a

Minkowski can be a written as the unique sum of a subset of generators. We illustrate this general property of Minkowski sum with the following construction that is specific to H_d.

Assume that a point p of H_d admits two decomposition, that is, there exist two distinct subsets S and T of G_d such that

$$p = \sum_{g^i \in S} g^i = \sum_{g^i \in T} g^i$$

Then one can check that p is the midpoint of two distinct points of H_d and thus can not be a vertex of H_d as

$$2p = \sum_{g^i \in S \cap T} g^i + \sum_{g^i \in S \cup T} g^i$$

For example, any $\{0,1\}$-valued point of H except the origin and the unit vector admit two decompositions and thus can not be vertices of H_d. Similarly, $(1,\ldots,1,x)$ where $x \in \{2,3,\ldots,d-1\}$ admits two decompositions and thus can not be a vertex of of H_d.

While checking whether a point admits two decomposition can be computationally challenging, we can run fast heuristics that are able to find a second decomposition by elementary flips when such decomposition exists which is the case for most lattice points of H_d. The heuristic automatically stops after a predetermined time if no second decomposition is found.

• CERTIFICATE OF VERTEXHOOD

Besides the 0-lifting of H_{d-1}, some points can be determined to be vertices of H_d for any d. While for general polytopes, checking whether a point is a vertex amounts to checking the feasibility of a potentially large linear optimization instance, the same check can be performed for Minkowski sums via a linear optimization instance of size equal to the number of generators.

Let S be a subset of G_d and p the point defined as $p = \sum_{g^i \in S} g^i$. Consider the following linear optimization feasibility problem where c is the variable and of size $d \times 2^d - 1$.

$$-c^T g^i \leq -1 \text{ for } g^i \in S$$

$$c^T g^i \leq -1 \text{ for } g^i \notin S$$

This problem is feasible if and only of there exists a nonzero cost vector c such that $p = \text{ argmax}\{c^T x : x \in H_d\}$. Since $d = 9$ is the smallest d such that H_d is undetermined, solving such a linear optimization problem is computationally cheap. The key issue is to be able to first remove most of the non-vertices with elementary certificate of non-vertexhood and to run the linear optimization based certificate of vertexhood on as few candidates as possible.

As illustration, let us consider the point $p = (1, 1, \ldots, 1, d)$, that is, $p = \sum_{g^i \in S} g^i$ where $S = \{[1, 0, \ldots, 0, 1], [0, 1, 0 \ldots, 0, 1], \ldots, [0, \ldots, 0, 1, 1], [0, \ldots, 0, 1]\}$. To check whether this Minkowski sum of d generators is a vertex of H_d, we need to check whether there exists a c such that:

$$c_i + c_d \geq 1 \text{ for } i = 1, \ldots, d-1$$

$$c_d \geq 1$$

$$c^T g^i \leq -1 \text{ for } g^i \notin S$$

One can check that $c = (-2, \ldots, -2, 3)$ is feasible and thus $p = (1, 1, \ldots, 1, d)$ is always a vertex of H_d.

We list a few known vertices of H_d for any d

- $(0, \ldots, 0)$ and its symmetric $2\sigma(d) = 2^{d-1}(1, \ldots, 1)$
- $(0, \ldots, 0, 1)$ and its symmetric $(2^{d-1} - 1, 2^{d-1}, \ldots, 2^{d-1})$
- $2^{d-2}(0, 1, \ldots, 1)$

- $(1,\ldots,1,d)$
- $(1,2^{d-2}+1,\ldots,2^{d-2}+1)$

We list a few simplices that belong to H_d for any d:

- Apex $(0,\ldots,0)$, base formed by the d permutation of $(1,\ldots,1,d)$
- Apex $(0,\ldots,0)$, base formed by the d permutation of $2^{d-2}(0,1,\ldots,1)$
- Apex $(0,\ldots,0)$, base formed by the d permutation of $(1,2^{d-2}+1,\ldots,2^{d-2}+1)$

As the algorithm discovers new vertices, new similar simplices can be added along the way.

3.3.1 Main steps of the orbitwise enumeration of the vertices of H_d

Assuming that the canonical vertices of H_{d-1} are known, the list of known canonical vertices of H_d is initialized with, up to permutation, the 0-liftings of the canonical vertices of H_{d-1} and of the symmetric in H_{d-1} of the canonical vertices of H_{d-1}. Canonical vertices of H_d that are already known and not already in the list, such as $(1,\ldots,1,d)$ are added to the list.

Known or determined vertices of H_d are entered as their Minkowski sum of generators. For instance $(1,1,\ldots,1,d)$ is entered as

$$S=\{[1,0,\ldots,0,1],[0,1,0\ldots,0,1],\ldots,[0,\ldots,0,1]\}.$$

A canonical vertex is marked as *computed* once all its orbitwise forward neighbourhood has been determined. The forward neighbourhood of a vertex v consist of all, up to symmetry, vertices of w of H_d such that $w-v$ is a generator of H_d; that is, $w-v$ is a nonzero $\{0,1\}$-valued vector of length d. Note that the set of all edges of a Minkowski sum form, up to translation, the set of generators.

To determine forward neighbourhood of a vertex v, we consider, up to symmetry, adding one generator g^i that was not used to generate v. The algorithm first checks whether $v+g^i$ is a non-vertex using the convexity and double decomposition certificates presented earlier. If yes, this point is disregarded, otherwise the algorithm checks whether $v+g^i$ is a vertex using the linear optimization based certificate presented earlier. If yes, the point is added to the list of canonical vertices, assuming this vertex was not found earlier. Otherwise, this point is disregarded.

Once all vertices in the list of canonical vertices are marked as computed, the algorithm terminates as this list contains all orbitwise vertices of H_d.

Note that the algorithm does not perform any convex hull computation.

3.4 Computational experiments for small d

• DETERMINING $a(3)$

We illustrate how the algorithm works to determine all the canonical vertices of H_3 and to compute $a(3) = 72$.

The list of canonical vertices of H_2 is $\{(0,0),(0,1)\}$.

The list of canonical vertices of H_3 is therefore initialized with

$$\{(0,0,0),(0,0,1),(0,1,2),(0,2,2),(1,1,3)\}$$

as the 4 0-liftings of the canonical vertices of H_2 and their symmetric counterparts by $\sigma_2 = (1,1)$ we can add known vertex $(1,1,3)$.

The set G_3 of generators of H_3 is

$$\{[0,0,1],[0,1,0],[1,0,0],[0,1,1],[1,0,1],[1,1,0],[1,1,1]\}.$$

Consider first the canonical vertex $v^0 = (0,0,0)$, up to symmetry, we can consider the following generators to be added to v^0: $[0,0,1],[0,1,1]$, and $[1,1,1]$. Out of these 3 Minkowski sums, the only one that is a vertex of H_3 is $v^1 = (0,0,1)$ as all are potential 0-liftings of vertices of H_2 since one of the coordinates is 0. Consequently v^0 is marked as computed, and v^1 should be added to the list unless it was already there – which is the case.

Consider the next canonical vertex $v^1 = (0,0,1)$. Since the generator $[0,0,1]$ is already used, we can consider, up to symmetry, the following generators to be added to v^1: $[0,1,0],[1,0,1],[1,1,0]$ and $[1,1,1]$. This leads to the following 4 canonical points: $(0,1,1)$ which not a vertex since $(1,1)$ is not a vertex of H_2, $v^2 = (0,1,2)$ which is a known vertex, $(1,1,1)$ which is known non-vertex, and $(1,1,2)$ which also a known non-vertex. Consequently v^1 is marked as computed, and v^2 should be added to the list unless it was already there – which is the case.

Consider the next canonical vertex $v^2 = (0,1,2)$. Since the generators $[0,0,1]$ and $[0,1,1]$ are already used, we can consider, up to symmetry, the following generators to be added to v^2: $[1,0,0],[0,1,0],[1,0,1],[1,1,0]$ and $[1,1,1]$. This leads to the following 5 canonical points: $(1,1,2)$ which is not a vertex since $(1,1,2)$ admits a second decomposition as $[1,0,1]+[0,1,1]$, $v^3 = (0,2,2)$ which a known vertex, $v^4 = (1,1,3)$ which is a known vertex, and $(1,2,3)$ which is not a vertex as $(1,2,3)$ admits a second decomposition as $[0,0,1]+[0,1,1]+[0,1,0]+[1,0,1]$. Consequently v^2 is marked as computed, and v^3 and v^4 should be added to the list unless it was already there – which is the case.

Consider the next canonical vertex $v^3 = (0,2,2)$. Since canonical vertices of H_3 must satisfy $x_1 + x_2 + x_3 \leq 6$, we can consider, up to symmetry, the following generators to be added to v^3: $[1,0,0]$ and $[1,0,1]$. This leads to the following 2 canonical points: $(1,2,2)$ which is not a vertex since $(1,2,2)$ admits a second decomposition as $[1,1,1]+[0,1,1]$, and $(1,2,3)$ that was previously identified as a non-vertex. Thus the forward neighbourhood of v^3 is empty and v^3 can be marked as computed.

Finally, we consider $v^4 = (1,1,3)$. The condition $x_1 + x_2 + x_3 \leq 6$ and the fact that $[0,1,1]$ is already being used yields, up to symmetry, that only the generator $(0,1,0)$ can be added to v^4. This leads to the point $(1,2,3)$ that was previously identified as a non-vertex. Consequently the forward neighbourhood of v^4 is empty and v^4 can be marked as computed.

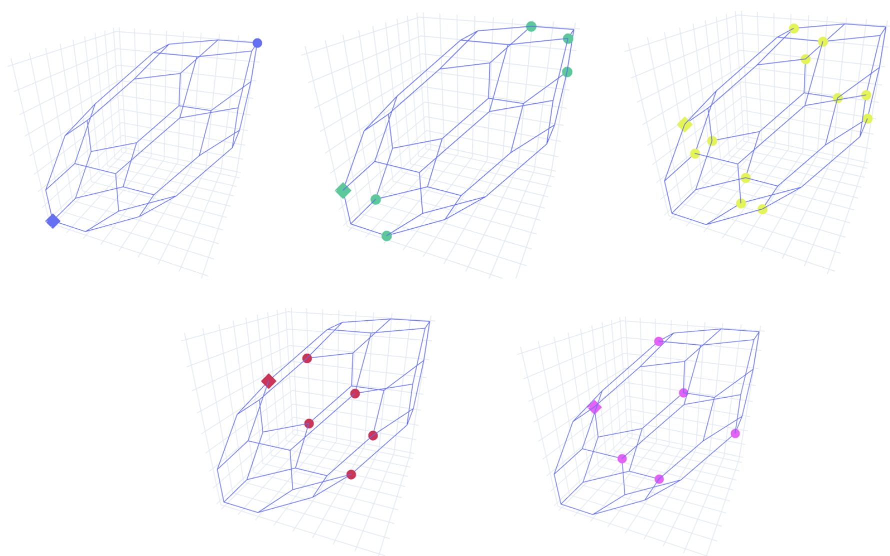

Figure 3.2: Canonical vertices of H_3 and their orbits

As all the vertices in the list have been marked as computed, the description is complete. In other words, the vertices of H_3 can be partitioned into 5 orbits as indicated in Table 3.2. The action of the symmetry group on each canonical vertex gives the size of its orbit. The sum of the orbit sizes is equal to $a(3)$. The canonical vertices of H_3 and their corresponding orbits are shown in Figure 3.2.

The 3-dimensional white whale, H_3, is represented in Figure 3.3 with its 5 canonical vertices displayed in hollow circles, and the vertices of their orbits in the corresponding colour.

Canonical Vertex	Method Obtained	Orbit size
(0,0,0)	Always a vertex	2
(0,0,1)	Always a vertex	6
(0,1,2)	0-lifting of a vertex from a lower dimension	12
(0,2,2)	0-lifting of a vertex from a lower dimension	6
(1,1,3)	Always a vertex	6
a(3)		32

Table 3.2: Canonical vertices of $a(3)$

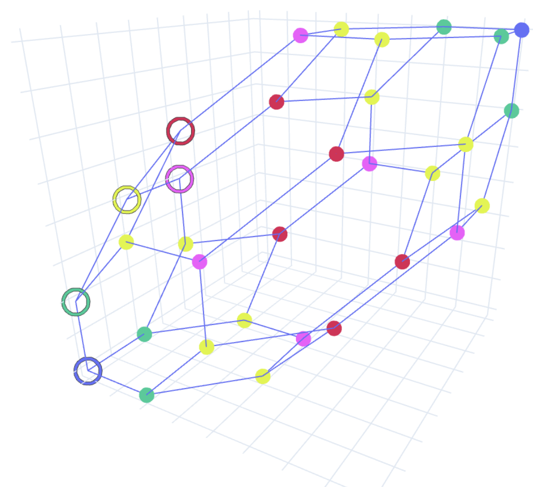

Figure 3.3: The 3-dimensional white whale H_3 with canonical vertices shown in hollow circles and their corresponding orbits

• DETERMINING $a(4)$

The list of canonical vertices, the method used to obtained them, and their respective orbit sizes for $d = 4$ are listed in Table 3.3

Canonical Vertex	Method Obtained	Orbit size
(0,0,0,0)	Always a vertex	2
(0,0,0,1)	Always a vertex	8
(0,0,1,2)	0-lifting of a vertex from a lower dimension	24
(0,0,2,2)	0-lifting of a vertex from a lower dimension	12
(0,1,1,3)	0-lifting of a vertex from a lower dimension	24
(0,1,3,3)	0-lifting of a vertex from a lower dimension	24
(0,2,2,4)	0-lifting of a vertex from a lower dimension	24
(0,2,3,4)	0-lifting of a vertex from a lower dimension	48
(0,3,4,4)	0-lifting of a vertex from a lower dimension	24
(0,4,4,4)	0-lifting of a vertex from a lower dimension	8
(1,1,1,4)	Always a vertex	8
(1,1,4,4)	Obtained from algorithm	12
(1,2,2,5)	Obtained from algorithm	24
(1,2,4,5)	Obtained from algorithm	48
(1,3,5,5)	Obtained from algorithm	24
(2,2,3,6)	Obtained from algorithm	24
(2,2,4,6)	Obtained from algorithm	24
(3,3,3,7)	Always a vertex	8
a(4)		370

Table 3.3: Canonical vertices of $a(4)$

• DETERMINING $a(5)$

The list of canonical vertices, the method used to obtained them, and their respective orbit sizes for $d = 5$ are listed in Table 3.4

Table 3.4: Canonical vertices of $a(5)$

Canonical Vertex	Method Obtained	Orbit size
(0,0,0,0,0)	Always a vertex	2
(0,0,0,0,1)	Always a vertex	10
(0,0,0,1,2)	0-lifting of a vertex from a lower dimension	40
(0,0,0,2,2)	0-lifting of a vertex from a lower dimension	20
(0,0,1,1,3)	0-lifting of a vertex from a lower dimension	60

(0,0,1,3,3)	0-lifting of a vertex from a lower dimension	60
(0,0,2,2,4)	0-lifting of a vertex from a lower dimension	60
(0,0,2,3,4)	0-lifting of a vertex from a lower dimension	120
(0,0,3,4,4)	0-lifting of a vertex from a lower dimension	60
(0,0,4,4,4)	0-lifting of a vertex from a lower dimension	20
(0,1,1,1,4)	0-lifting of a vertex from a lower dimension	40
(0,1,1,4,4)	0-lifting of a vertex from a lower dimension	60
(0,1,2,2,5)	0-lifting of a vertex from a lower dimension	120
(0,1,2,4,5)	0-lifting of a vertex from a lower dimension	240
(0,1,3,5,5)	0-lifting of a vertex from a lower dimension	120
(0,1,5,5,5)	0-lifting of a vertex from a lower dimension	40
(0,2,2,3,6)	0-lifting of a vertex from a lower dimension	120
(0,2,2,4,6)	0-lifting of a vertex from a lower dimension	120
(0,2,4,6,6)	0-lifting of a vertex from a lower dimension	120
(0,2,5,6,6)	0-lifting of a vertex from a lower dimension	120
(0,3,3,3,7)	0-lifting of a vertex from a lower dimension	40
(0,3,3,5,7)	0-lifting of a vertex from a lower dimension	120
(0,3,4,6,7)	0-lifting of a vertex from a lower dimension	240
(0,3,6,6,7)	0-lifting of a vertex from a lower dimension	120
(0,4,4,4,8)	0-lifting of a vertex from a lower dimension	40
(0,4,4,5,8)	0-lifting of a vertex from a lower dimension	120
(0,4,4,7,7)	0-lifting of a vertex from a lower dimension	60
(0,4,5,6,8)	0-lifting of a vertex from a lower dimension	240
(0,4,6,6,8)	0-lifting of a vertex from a lower dimension	120
(0,4,7,7,7)	0-lifting of a vertex from a lower dimension	40
(0,5,5,7,8)	0-lifting of a vertex from a lower dimension	120
(0,5,7,7,8)	0-lifting of a vertex from a lower dimension	120
(0,6,6,8,8)	0-lifting of a vertex from a lower dimension	60
(0,6,7,8,8)	0-lifting of a vertex from a lower dimension	120
(0,7,8,8,8)	0-lifting of a vertex from a lower dimension	40

(0,8,8,8,8)	0-lifting of a vertex from a lower dimension	10
(1,1,1,1,5)	Always a vertex	10
(1,1,1,5,5)	Obtained from algorithm	20
(1,1,2,2,6)	Obtained from algorithm	60
(1,1,2,5,6)	Obtained from algorithm	120
(1,1,3,6,6)	Obtained from algorithm	60
(1,1,6,6,6)	Obtained from algorithm	20
(1,2,2,3,7)	Obtained from algorithm	120
(1,2,2,5,7)	Obtained from algorithm	120
(1,2,4,7,7)	Obtained from algorithm	120
(1,2,6,7,7)	Obtained from algorithm	120
(1,3,3,3,8)	Obtained from algorithm	40
(1,3,3,6,8)	Obtained from algorithm	120
(1,3,4,7,8)	Obtained from algorithm	240
(1,3,7,7,8)	Obtained from algorithm	120
(1,4,4,4,9)	Obtained from algorithm	40
(1,4,4,6,9)	Obtained from algorithm	120
(1,4,4,8,8)	Obtained from algorithm	60
(1,4,5,7,9)	Obtained from algorithm	240
(1,4,7,7,9)	Obtained from algorithm	120
(1,4,8,8,8)	Obtained from algorithm	40
(1,5,5,8,9)	Obtained from algorithm	120
(1,5,8,8,9)	Obtained from algorithm	120
(1,6,6,9,9)	Obtained from algorithm	60
(1,6,8,9,9)	Obtained from algorithm	120
(1,7,9,9,9)	Obtained from algorithm	40
(2,2,2,4,8)	Obtained from algorithm	40
(2,2,2,5,8)	Obtained from algorithm	40
(2,2,5,8,8)	Obtained from algorithm	60
(2,2,6,8,8)	Obtained from algorithm	60

(2,3,3,4,9)	Obtained from algorithm	120
(2,3,3,6,9)	Obtained from algorithm	120
(2,3,5,8,9)	Obtained from algorithm	240
(2,3,7,8,9)	Obtained from algorithm	240
(2,4,4,5,10)	Obtained from algorithm	120
(2,4,4,6,10)	Obtained from algorithm	120
(2,4,5,9,9)	Obtained from algorithm	120
(2,4,6,8,10)	Obtained from algorithm	240
(2,4,7,8,10)	Obtained from algorithm	240
(2,4,8,9,9)	Obtained from algorithm	120
(2,5,6,9,10)	Obtained from algorithm	240
(2,5,8,9,10)	Obtained from algorithm	240
(2,6,7,10,10)	Obtained from algorithm	120
(2,6,8,10,10)	Obtained from algorithm	120
(3,3,4,4,10)	Obtained from algorithm	60
(3,3,4,7,10)	Obtained from algorithm	120
(3,3,5,8,10)	Obtained from algorithm	120
(3,3,8,8,10)	Obtained from algorithm	60
(3,4,5,5,11)	Obtained from algorithm	120
(3,4,5,7,11)	Obtained from algorithm	240
(3,4,6,8,11)	Obtained from algorithm	240
(3,4,8,8,11)	Obtained from algorithm	120
(3,4,9,9,10)	Obtained from algorithm	120
(3,5,5,10,10)	Obtained from algorithm	60
(3,5,9,9,11)	Obtained from algorithm	120
(3,6,6,10,11)	Obtained from algorithm	120
(3,7,7,11,11)	Obtained from algorithm	60
(4,4,4,4,11)	Obtained from algorithm	10
(4,4,4,8,11)	Obtained from algorithm	40
(4,4,5,9,11)	Obtained from algorithm	120

(4,4,6,6,12)	Obtained from algorithm	60
(4,4,6,7,12)	Obtained from algorithm	120
(4,4,7,8,12)	Obtained from algorithm	120
(4,4,8,8,12)	Obtained from algorithm	60
(4,4,10,10,10)	Obtained from algorithm	20
(4,5,5,5,12)	Obtained from algorithm	40
(4,5,5,8,12)	Obtained from algorithm	120
(4,5,5,10,11)	Obtained from algorithm	120
(4,5,6,9,12)	Obtained from algorithm	240
(4,6,6,10,12)	Obtained from algorithm	120
(5,5,5,11,11)	Obtained from algorithm	20
(5,5,6,6,13)	Obtained from algorithm	60
(5,5,6,8,13)	Obtained from algorithm	120
(5,5,7,9,13)	Obtained from algorithm	120
(6,6,6,7,14)	Obtained from algorithm	40
(6,6,6,8,14)	Obtained from algorithm	40
(7,7,7,7,15)	Always a vertex	10
a(5)		11 292

3.5 Future Work

While the proposed computational framework allows for an efficient orbitwise enumeration of the vertices of H_d, further work is required to determine $a(9)$ that is still currently intractable.

One area to continue exploring is to improve existing or add new certificates of detecting a vertex or non-vertex based on new structural properties. The 9-dimensional white whale may have around 2^{30} orbits and thus the certificate for not being a vertex must be further enhanced to allow for even more efficient pruning. The linear optimization based certificate for being a vertex is difficult to improve.

From a computational perspective, there are a few directions to try. Since many tasks can be done in parallel, leveraging distributed computing resoures such as using a cluster might allow for a significant speedup. Since there are significant amounts of orbits, finding more efficient ways to store and manipulate these lists of numbers can improve the performance.

Finally, solving theoretical questions that remain currently open may allow for a dramatic speed-up. In particular, we wish to highlight the following fundamental question dealing with Minkowski sums and lying at the boundary of convexity and combinatorics.

As discussed earlier, for any Minkowski sum, if a sum of generators admits a second decomposition, this sum of generators cannot be a vertex of the Minkowski sum. Specifically for H_d, does the reverse implication hold, that is, is a sum of generators of H_d a vertex of H_d if no other sum of generators gives the same point?

Bibliography

[1] Xavier Allamigeon, Pascal Benchimol, Stéphane Gaubert, and Michael Joswig. Log-barrier interior point methods are not strongly polynomial. *SIAM Journal on Applied Algebra and Geometry*, 2:140–178, 2018.

[2] David Avis, David Bremner, and Raimund Seidel. How good are convex hull algorithms? *Computational Geometry*, 7:265–301, 1997.

[3] David Barnette. An upper bound for the diameter of a polytope. *Discrete Mathematics*, 10:9–13, 1974.

[4] Louis J. Billera. In pursuit of a white whale. *Berlin Mathematical School Colloquium*, 2018.

[5] Louis J. Billera, Justin T. Moore, Costandino D. Moraites, Yipu Wang, and Kameryn Williams. Maximal unbalanced families. *arXiv:* 1209.2309, 2012.

[6] Jonathan Block and Shmuel Weinberger. Aperiodic tilings, positive scalar curvature, and amenability of spaces. *Journal of the American Mathematical Society*, 5:907–918, 1992.

[7] Nicolas Bonifas, Marco Di Summa, Friedrich Eisenbrand, Nicolai Hähnle, and Martin Niemeier. On sub-determinants and the diameter of polyhedra. *Discrete & Computational Geometry*, 52:102–115, 2014.

[8] Steffen Borgwardt, Jesús A. De Loera, and Elisabeth Finhold. The diameters of network-flow polytopes satisfy the hirsch conjecture. *Mathematical Programming*, 171:283–309, 2018.

[9] Grünbaum Branko. *Convex Polytopes*. Preprated by Volker Kaibel, Victor Klee, and Günter M. Ziegler, 2nd edition, 2003.

[10] Nathan Chadder and Antoine Deza. Computational determination of the largest lattice polytope diameter. *Electronic Notes in Discrete Mathematics*, 62:105–110, 2017.

[11] George B. Dantzig. *Linear programming and extensions*. Princeton University Press, 1963.

[12] George B. Dantzig, Alexander Orden, and Philip Wolfe. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific Journal of Mathematics*, 5:183–195, 1955.

[13] Wayne S. DeSarbo and Donna L. Hoffman. Constructing mds joint spaces from binary choice data: A multidimensional unfolding threshold model for marketing research. *Journal of Marketing Research*, 24:40–54, 1987.

[14] Anna Deza, Antoine Deza, Zhongyan Guan, and Lionel Pournin. Distances between vertices of lattice polytopes. *Optimization Letters*, 14:309–326, 2020.

[15] Antoine Deza, George Manoussakis, and Shmuel Onn. Primitive zonotopes. *Discrete and Computational Geometry*, 60:27–39, 2018.

[16] Antoine Deza and Lionel Pournin. Improved bounds on the diameter of lattice polytopes. *Acta Mathematica Hungarica*, 154:457–469, 2018.

[17] Antoine Deza, Lionel Pournin, and Rado Rakotonarivo. The vertices of primitive zonotopes. *Contemporary Mathematics*, 764:71–82, 2021.

[18] Antoine Deza, Lionel Pournin, and Noriyoshi Sukegawa. The diameter of lattice zonotopes. *Proceedings of the American Mathematical Society*, 148(8):3507–3516, 2020.

[19] Timothy S. Evans. What is being calculated with thermal field theory? *9th Lake Louise Winter Institute: Particle Physics and Cosmology*, pages 343–352, 1994.

[20] Komei Fukuda. CDD and CDDplus homepage. `https://people.inf.ethz.ch/fukudak/cdd_home/index.html`, 2018.

[21] Komei Fukuda. *Polyhedral computation*. ETH research collection. Zürich, Switzerland, 2020. `https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/426218/PolyCompBook20200829.pdf`.

[22] Samuel C. Gutekunst, Karola Mészáros, and T. Kyle Petersen. Root cones and the resonance arrangement. *The Electronic Journal of Combinatorics*, 28(1), 2021.

[23] Melvin J. Hinich and Michael C. Munger. *Analytical Politics*. Cambridge University Press, 1997.

[24] Claude Itzykson and Jean-Bernard Zuber. *Quantum field theory*. Courier Corporation, 2012.

[25] Gil Kalai and Daniel J. Kleitman. A quasi-polynomial bound for the diameter of graphs of polyhedra. *Bulletin of the American Mathematical Society*, 26(2):315–316, 1992.

[26] Hidehiko Kamiya, Akimichi Takemura, and Hiroaki Terao. Ranking patterns of unfolding models of codimension one. *Advances in Applied Mathematics*, 47:379–400, 2011.

[27] Hidehiko Kamiya, Akimichi Takemura, and Hiroaki Terao. Ranking patterns of unfolding models of codimension one. *Advances in Applied Mathematics*, 47(2):379–400, 2011.

[28] Victor Klee and David D. Walkup. The d-step conjecture for polyhedra of dimension $d < 6$. *Acta Mathematics*, 117:53 – 78, 1967.

[29] Peter Kleinschmidt and Shmuel Onn. On the diameter of convex polytopes. *Discrete Mathematics*, 102:75–77, 1992.

[30] Lukas Kühne. The universality of the resonance arrangement and its Betti numbers. *arXiv:* 2008.10553, 2020.

[31] David G. Larman. Paths on polytopes. *Proceedings of the London Mathematical Society*, 3:161–178, 1970.

[32] John D. C. Little, Katta G. Murty, Dura W. Sweeney, and Caroline Karel. An algorithm for the traveling salesman problem. *Operations Research*, 11:972–989, 1963.

[33] Benjamin Matschke, Francisco Santos, and Christophe Weibel. The width of 5-dimensional prismatoids. *Proceedings of the London Mathematical Society*, 110:647–672, 2015.

[34] Michal Melamed and Shmuel Onn. Convex integer optimization by constantly many linear counterparts. *Linear Algebra and its Applications*, 447:88–109, 2014.

[35] Justin Tatch Moore. Amenability and ramsey theory. *Fundamenta Mathematicae*, 220:263–280, 2013.

[36] Theodore S. Motzkin, Howard Raiffa, Gerald L. Thompson, and Robert M. Thrall. The double description method. *Contributions to the Theory of Games*, 2:51–73, 1953.

[37] Dennis Naddef. The Hirsch conjecture is true for (0,1)-polytopes. *Mathematical Programmming*, 45:109–110, 1989.

[38] Shmuel Onn and Uriel G. Rothblum. Convex combinatorial optimization. *Discrete & Computational Geometry*, 32:549–566, 2004.

[39] Alberto Del Pia and Carla Michini. On the diameter of lattice polytopes. *Discrete & Computational Geometry*, 55:681–687, 2016.

[40] Francisco Santos. A counterexample to the Hirsch conjecture. *Annals of Mathematics*, 176:383–412, 2012.

[41] Francisco Santos, Tamon Stephen, and Hugh Thomas. Embedding a pair of graphs in a surface, and the width of 4-dimensional prismatoids. *Discrete & Computational Geometry*, 47:569–576, 2012.

[42] Steve Smale. Mathematical problems for the next century. *American Mathematical Society*, Mathematics: Frontiers and Perspective:271–294, 2000.

[43] Noriyoshi Sukegawa. Improving bounds on the diameter of a polyhedron in high dimensions. *Discrete Mathematics*, 340(9):2134 – 2142, 2017.

[44] Michael J. Todd. An improved Kalai-Kleitman bound for the diameter of a polyhedron. *SIAM Journal on Discrete Mathematics*, 28:1944–1947, 2014.

[45] Michel A. van Eijck. *Thermal field theory and the finite-temperature renormalization group*. PhD thesis, University of Amsterdam, 1995.

[46] Günter M. Ziegler. *Lectures on polytopes*, volume 152. Springer, 1995.