

Experimental Investigations on New Algorithms for Polytopes

By

KARLA PATRICIA LANZAS

SENIOR THESIS

Submitted in partial satisfaction of the requirements for Highest Honors for the degree of

BACHELOR OF SCIENCE

in

MATHEMATICS

in the

COLLEGE OF LETTERS AND SCIENCE

of the

UNIVERSITY OF CALIFORNIA,

DAVIS

Approved:

Jesús A. De Loera

June 2011

Introduction.

In this senior thesis, we present two different studies of polytopes. In the first section of this thesis, we will focus on the estimation and approximation of integer lattice points within a polyhedra. Through this section we will explore two distinct methods for estimation. The first a sampling algorithm and the second the Gaussian heuristic. We will also test the accuracy and efficiency of these algorithms. The second portion of this thesis will focus on the 53 year-old Hirsch Conjecture which has been recently disproved by Francisco Santos. We will analyze the counterexample and attempt to come up with a less complex counterexample (with less facets, lower dimension, ... etc) than the one that Santos presented in his paper. Before the presentation of both topics the reader will receive a brief reminder of terms that will be commonly used throughout this thesis.

Contents

Chapter 1. Introduction	1
1.1. Basic Definitions of Polytopes	1
1.2. Lattice Points and Applications	3
1.3. The Simplex Method and the Hirsch Conjecture	4
Chapter 2. Estimating the Number of Lattice Points Inside Polyhedra	19
2.1. Estimation and Approximate Counting	19
2.2. Details on Experiments	24
Chapter 3. Analysis of the Recent Counterexample of the Hirsch Conjecture	33
3.1. Hirsch Conjecture	33
3.2. Santos' Counterexample	35
3.3. The Search for a Computationally Smaller Counterexample	36
Appendix. Bibliography	43

CHAPTER 1

Introduction

1.1. Basic Definitions of Polytopes

In order to motivate the research in this senior thesis, we begin with some basic definitions. The basic concept of the research involves understanding what exactly a polytope is. Therefore, we provide the following definitions.

DEFINITION 1. (definition 2.1.2 in [22]) The set of solutions of a system of linear inequalities is called a polyhedron. In its general form, a polyhedron is a set of the type $P = \{x \in \mathbb{R}^d : \langle c_i, x \rangle \leq \beta_i\}$ for some non-zero vectors c_i in $\mathbb{R}^d$ and some real numbers β_i .

In other words a *polytope* is a geometric shape with flat sides that exist in any general number of dimensions. For instance, a polygon is a polytope in two dimensions and a polyhedron is a polytope in three dimensions.

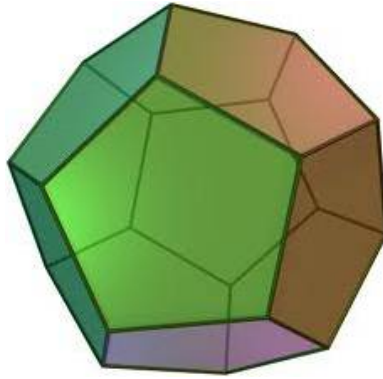


FIGURE 1. Three dimensional polytope also known as a dodecahedron

DEFINITION 2. A *hyperplane* in $\mathbb{R}^n$ is

$$H = \{x \mid a_1x_1 + \cdots + a_nx_n = b\},$$

where not all $a_i = 0$. By further analyzing this definition we see that $\dim(H) = n - 1$, since if, say, $a_1 \neq 0$, H can be written as $[b/a_1 \ 0 \ 0 \ \dots \ 0]^T$ plus the nullspace of $1 \times n$ matrix $[a_1 \ a_2 \ \dots \ a_n]$.

COROLLARY 1. (Corollary 2.3.5 in [22]) Let P be a d -dimensional polytope in $\mathbb{R}^d$. Then the following is true:

- (1) The intersection of P with a hyperplane is a polytope. If the hyperplane passes through a point in the relative interior of P then the intersection is a $(d - 1)$ -polytope.
- (2) Every projection of P is a polytope. More generally, the image of P under a linear map is another polytope.

For information about the proof please refer to [22].

DEFINITION 3. A *halfspace* is the set

$$H = \{x \mid a_1x_1 + \cdots + a_nx_n \geq b\}$$

DEFINITION 4. A *polyhedron* is the intersection of finitely many halfspaces. A bounded nonempty polyhedron is called a *polytope*.

For the purpose of this senior thesis we will focus on bounded *convex polytopes*. A *convex* polytope is a special classification of polytopes, in which it is also a convex set of points. A *convex set* is a set in which any point on a line segment connecting any two points in the set is also in the set.

Formally, if we let $K \subset \mathbb{R}^n$ and $x_1, x_2 \in K$ then K is *convex* if and only if all the points of the line segment with endpoints x_1, x_2 belong to K . Furthermore, K is convex if and only if $(1 - \lambda)x_1 + \lambda x_2 \in K$ for all $0 \leq \lambda \leq 1$ and whenever $x_1, x_2 \in K$.

The *convex hull* of $A \subset \mathbb{R}^n$, denoted $\text{conv}(A)$, is defined to be the set of all convex combinations of finitely many elements of A . That is, $\text{conv}(A) = \{\lambda_1x_1 + \cdots + \lambda_kx_k : k \text{ is a positive integer, } x_1, \dots, x_k \in A, \sum_{i=1}^k \lambda_i = 1, \text{ and } \lambda_i \geq 0, i = 1, \dots, k\}$. If A is a finite set, k can be taken to be the number of elements of A . However, if A is infinite, k is arbitrary.

EXAMPLE 1. In $\mathbb{R}^3$ let $\vec{x}_1 = (1, 0, 0), \vec{x}_2 = (0, 1, 0), \vec{x}_3 = (0, 0, 1)$. Then the convex combinations fill out the closed triangle with the vertices $\vec{x}_1, \vec{x}_2, \vec{x}_3$

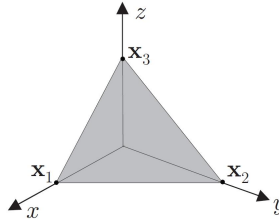


FIGURE 2. The set of convex combination of $\vec{x}_1, \vec{x}_2, \vec{x}_3$

THEOREM 1. **Weyl-Minkowski Theorem-** *Every polytope is a polyhedron. Every bounded polyhedron is a polytope.*

This theorem allows us to work with two distinct representations of the inequalities or the vertex representation.

A convex polytope is the convex hull of a finite set of points. In $\mathbb{R}^2$, a *convex polytope* is called a convex polygon. The faces of a convex polytope are convex polytopes of lower dimensions. The 0-dimensional face is called a vertex. The 1-dimensional faces are called edges and the $(n-1)$ -dimensional faces are called facets. The $(n-2)$ -dimensional faces of an n -dimensional polytope are often referred to as sub-facets.

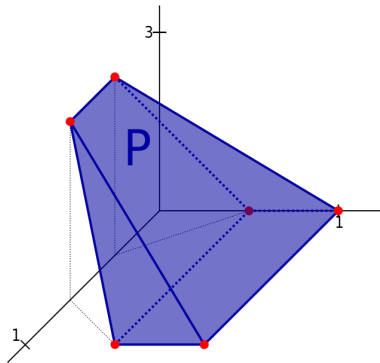


FIGURE 3. A three dimensional convex polytope

We will assume that the empty set is also convex. Additionally the intersection of convex sets is convex too.

1.2. Lattice Points and Applications

Now that we have recalled some of the basic definitions. The importances of lattice points, will be discussed later in this thesis. Given a polytope $P = \{x : Ax = b, x \geq 0\}$, specified by a $d \times n$ integral matrix A and an integral d -vector b , our goal is to *count or at least estimate how many lattice points are inside P* . In some applications we are in fact interested in the *counting function*,

$$\phi_A(b) = \#\{x : Ax = b, x \geq 0, x \text{ integral}\},$$

where b can be taken as a vector of parameters. This is the function that counts the lattice points inside a convex polytope. In pure mathematics, one is in fact interested in formulas for $\phi_A(b)$ in terms of the variables $b_1, \dots, b_d$. Areas of application where these algorithmic questions often appear include mathematical programming, probability and statistics, and theoretical computer science. We illustrate with several examples the many uses of computing or evaluating $\phi_A(b)$.

Counting lattice points of polytopes is at the foundation of the theory of mathematical optimization. There is a very close algorithmic relationship between optimization and counting (this is made explicit in [31, 21, 23]). The number of feasible lattice points in a polyhedron is a measure of progress of a branch-and-bound algorithm for solving integer optimization problems. For integer programs in some applications, it is desirable to find all optimal solutions [17]. For example, in network flow problems regarding stability or security, one may be interested in knowing the number of maximal integral-valued flows or cuts [2]. This is the same as counting the number of lattice points of a *flow polytope* (see [3] and references therein).

In statistics, counting and sampling lattice points becomes useful to make inferences about the statistical models imposed in tabular data (contingency tables). For example, to decide whether

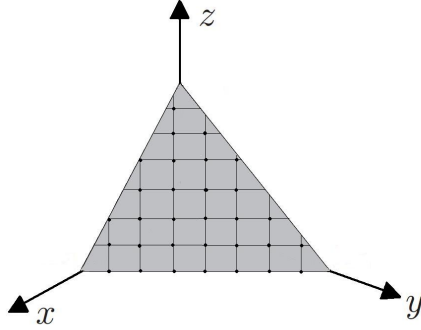


FIGURE 4. The Lattice points inside a polytope

two of the traits of the statistic, say, income and race, are independent of each other, several tests of independence depend on counting the integral tables satisfying some marginal conditions (see [25, 29] and its references). Statisticians have developed ad hoc sampling-based estimation methods and, in fact, counting is closely related to the problem of sampling a point uniformly at random or with respect to other probability distribution models (see [10, 11, 12, 37] and references therein). Of course for an arbitrary polyhedron this is an unrealistic assumption since it would require intractable computation to even jump start such methods. Counting is also related to statistical mechanics models such as the Ising and Potts models (see e.g., [35, 42]).

In computer science, coding theory and cryptography have always been important areas of application for lattices and lattice point counting [41, 53]. An emerging new application of lattice point counting is computer program verification and code optimization. The systematic reasoning about a program's runtime behavior, performance, and execution requires specific knowledge of the number of operations and resource allocation within the program. This is important knowledge for the sake of checking correctness as well as for automatically detecting run-time errors, buffer overflows, null-pointer dereferences or memory leaks. This can be modeled as problem of counting points in parametrized regions (see [15, 30, 43, 50] and references therein)

Further discussion of lattice points can be found in chapter two.

1.3. The Simplex Method and the Hirsch Conjecture

1.3.1. Linear Programing. A *linear programing* problem may be defined as the problem of maximizing or minimizing a linear function subject to linear constraints. The constraints may be equalities or inequalities. Linear programing plays a vital in solving scheduling problem throughout a wide variety of industries. Examples of scheduling problems occur: for airline agencies to schedule their crews, for factory managers to compute the mix of raw materials which produces the most profitable blend of output products, or for utilities to plan their oil and gas purchases. In all these situations, one must set up and solve problems with many variables and constraints.

Thus in linear programing the purpose is to maximize or minimize some linear function of these decision variables. $\zeta = c_1x_1 + c_2x_2 + \cdots + c_nx_n$ this is the optimal function. The variables whose values have to be decided up on by the optimal function are called the *decision variables* usually

noted as x_j , $j = 1, 2, \dots, n$. In addition to the optimal function, linear programming questions also have constraints of the form:

$$a_1x_1 + a_2x_2 + \dots + a_nx_n \begin{cases} \leq \\ = \\ \geq \end{cases} b.$$

There are two cases of linear programs

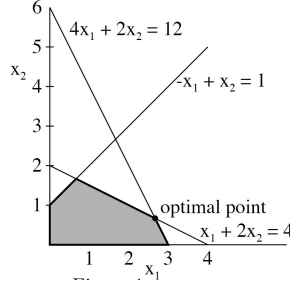


FIGURE 5. example of a graphed optimal solution

Standard Form:

$$\begin{aligned} \min \quad & c^T x, \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

In which every variable is sign-constrained and only equality constraints are allowed.

Canonical Form:

$$\begin{aligned} \min \quad & c^T x, \\ \text{s.t.} \quad & A \geq b \\ & x \geq 0 \end{aligned}$$

allows inequalities constraints and in which all variables are sign-constrained. Both forms are equivalent, either one may be manipulated into the other. This is demonstrated in [32] The linear programming problem we will study can be formulated as follows:

$$\begin{aligned} \text{maximize} \quad & c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\ & x_1, x_2, \dots, x_n \geq 0. \end{aligned}$$

Which is an expanded version of the *standard form*. Where m is used to describe the number of constraints and n is the number of decision variables. A solution are proposed values for the decision variables, if a solution $(x_1, x_2, \dots, x_n)$ meets all the constraints then the solution is said to be feasible. It becomes an optimal solution if it contains the desired maximum.

DEFINITION 5. (from [32]) Consider the linear program

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where A is of rank m and is $m \times n$. If columns $j_1 < j_2 < \dots < j_m$ are a basis, the corresponding *basic solution* x is defined as follows and ignores c . Let B be the nonsingular matrix consisting of columns $j_1, j_2, \dots, j_m$ of A . Any variable x_l where $l \notin \{j_1, j_2, \dots, j_m\}$ is the set to 0 and is called *nonbasic*. For $k = 1, \dots, m$, x_{j_k} is the set to the k th component of $B^{-1}b$ and is known as *basic*; the basic variables may or may not be 0 (The basic solution x is the unique vector satisfying $Ax = b$ subject to the condition that $x_l = 0$ for $l \notin \{j_1, \dots, j_m\}$.) If x satisfies the sign constraints $x \geq 0$, x is a *basic feasible solution* or *bfs*.

EXAMPLE 2. (from [32])

$$\begin{aligned} x_1 + 2x_2 - 3x_3 + x_4 &= 7 \\ 2x_1 + 0x_2 - 7x_3 + x_4 &= 9 \\ x_j &\geq 0 \end{aligned}$$

So, by definition $A =$

$$\begin{bmatrix} 1 & 2 & -3 & 1 \\ 2 & 0 & -7 & 1 \end{bmatrix}.$$

We can obtain a basis from column 3 and column 4 denoted by $\{3, 4\}$ which is the following:

$$\left\{ \begin{bmatrix} -3 \\ -7 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}.$$

So let $B =$

$$\begin{bmatrix} -3 & 1 \\ -7 & 1 \end{bmatrix}.$$

$B^{-1} =$

$$\begin{bmatrix} -3 & 1 \\ -7 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{4} & -\frac{1}{4} \\ \frac{7}{4} & -\frac{3}{4} \end{bmatrix}.$$

Since B^{-1} exist we know that B is a basis. So, the basic solution corresponding to basis $\{3, 4\}$ is the vector

$$\begin{bmatrix} 0 \\ 0 \\ \frac{-1}{2} \\ \frac{11}{2} \end{bmatrix}.$$

As $x_3 < 0$, it is infeasible.

If $m \times n$ matrix A has rank m , then

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned}$$

always has a basic solution. If the feasible region is nonempty, we will see that it must have a bfs as well.

Notation: Let LPS denote:

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where A is of rank m and is $m \times n$. Let F denote the feasible set $\{x \mid Ax = b, x \geq 0\}$

THEOREM 2. (*Theorem 6 in [32]*) *In LPS, suppose that $w \in \mathbb{R}^n$. Then w is a vertex of the feasible set F if and only if w is a basic feasible solution of the LPS instances.*

COROLLARY 2. (Corollary 7 in [32]) There are only finitely many vertices.

PROOF. Each is a basic feasible solution, which is determined by m -elements subset of the columns. The number of basic feasible solutions is at most $\binom{n}{m}$. $\square$

DEFINITION 6. A basic feasible solution is *degenerate* if it has more than $m - n$ zeros.

EXAMPLE 3. (from 1.4 in [32])

$$\begin{aligned} x_1 + x_2 - 3x_3 &= 5 \\ 2x_1 - x_2 + 3x_3 &= 10, \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

We can convert this into the following matrix

$$A = \begin{bmatrix} 1 & 1 & -3 \\ 2 & -1 & 3 \end{bmatrix},$$

Since this is a 2×3 matrix we know that $m = 3$ and $n = 2$ hence we get $n - m = 3 - 2 = 1$. Thus the basic feasible solution corresponding to basis $\{1,2\}$ is

$$\begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix},$$

which has more than $n - m$ zeros, since it also corresponds to the basis $\{1,3\}$.

THEOREM 3. *If two different bases correspond to a single bfs v , then v is degenerate.*

Note that the converse does not always hold true.

For details on the proof please refer to [32]

LEMMA 1. *Suppose $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ is a polyhedron. Then v is an extreme point of P if and only if there are n linearly independent constraints among the constraints $Ax \leq b$ that are tight at v .*

Proof is omitted but may be seen in [32]

THEOREM 4. *In LPS, suppose that $p \in F$. Then either the LPS instance is unbounded or there is a vertex v of F satisfying $c^T v \leq c^T p$.*

COROLLARY 3. If in LPS $F \neq \emptyset$ and $c^T x \geq B$ for all $x \in F$, then there is an optimal vertex. Provided that the cost is bounded below, linear programming is a finite problem!

THEOREM 5. *If $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ is a polytope, then every $x \in P$ can be written as a convex combination of at most $n + 1$ vertices.*

1.3.2. Simplex Method. In 1947, George Dantzig designed the Simplex Algorithm, which has become one of the most widely used methods for solving linear programs. The Simplex Algorithm is fast in practice; problems with thousands of variables and constraints can be solved by this algorithm. The algorithm begins from an arbitrary vertex of the feasible region and (ideally) finds a better neighboring vertex, a neighboring vertex being a different basic feasible solution whose corresponding basis differs from that on the first vertex, in exactly one column. If no other vertex has a cheaper cost, then this vertex may be proven to be the optimal solution.

EXAMPLE 4. (from 2.1 in [32])

$$\begin{array}{ll} \text{Minimize} & 2x_2 + x_4 + 5x_6 \\ \text{such that} & 2x_1 + x_2 + 2x_3 + 2x_4 + x_5 = 16 \\ & x_1 + 4x_3 + 2x_5 = 16 \\ & -x_1 + 2x_2 + x_3 + 5x_4 - 5x_5 + 2x_6 = 34 \\ & 3x_2 - x_3 + x_4 + 4x_5 + 4x_6 = 38 \\ & x_i \geq 0 \end{array}$$

Now re-write this in matrix notation,

$$A = \begin{bmatrix} 2 & 1 & 2 & 2 & 1 & 0 \\ 1 & 0 & 4 & 0 & 2 & 0 \\ -1 & 2 & 1 & -5 & 5 & 2 \\ 0 & 3 & -1 & 1 & 4 & 4 \end{bmatrix}, b = \begin{bmatrix} 16 \\ 16 \\ 34 \\ 38 \end{bmatrix}.$$

We can see the corresponding basis $\{1,2,5,6\}$ has the basic feasible solution

$$\begin{bmatrix} 4 \\ 2 \\ 0 \\ 0 \\ 6 \\ 2 \end{bmatrix};$$

and therefore

$$4A_1 + 2A_2 + 6A_5 + 2A_6 = b = \begin{bmatrix} 16 \\ 16 \\ 34 \\ 38 \end{bmatrix}.$$

Nonbasic column

$$A_4 = \begin{bmatrix} 2 \\ 0 \\ -5 \\ 1 \end{bmatrix}$$

can be written as this linear combination of the basic columns A_1, A_2, A_5, A_6 :

$$(1.1) \quad A_4 = 2A_1 - 1A_2 - 1A_5 + 2A_6.$$

Now consider increasing the value of nonbasic x_4 from 0 to some value θ , while maintaining feasibility and leaving nonbasic x_3 at 0. Let x'_1, x'_2, x'_5 , and x'_4 be the new values, respectively, of x_1, x_2, x_5 , and x_6 . We want

$$(1.2) \quad (x'_1 A_1 + x'_2 A_2 + x'_5 A_5 + x'_6 A_6) + \theta A_4 = \begin{bmatrix} 16 \\ 16 \\ 34 \\ 38 \end{bmatrix}.$$

Using (1.1) we can rewrite (1.2) as follows:

$$(x'_1 + 2\theta)A_1 + (x'_2 - \theta)A_2 + (x'_5 - \theta)A_5 + (x'_6 + 2\theta)A_6 = \begin{bmatrix} 16 \\ 16 \\ 34 \\ 38 \end{bmatrix}.$$

This is clearly satisfied if

$$(1.3) \quad (x'_1 + 2\theta) = x_1 \quad \Rightarrow x'_1 = 4 - 2\theta$$

$$(1.4) \quad (x'_2 - \theta) = x_2 \quad \Rightarrow x'_2 = 2 + \theta$$

$$(1.5) \quad (x'_5 - \theta) = x_5 \quad \Rightarrow x'_5 = 6 + \theta$$

$$(1.6) \quad (x'_6 + 2\theta) = x_6 \quad \Rightarrow x'_6 = 2 - 2\theta$$

Any θ will satisfy the equality constraints but to remain feasible, we have to be careful with the sign constraints. Hence:

$$\text{From(1.3)} \quad 4 - 2\theta \geq 0 \quad \Rightarrow \theta \leq 2$$

$$\text{From(1.4)} \quad 2 + \theta \geq 0 \quad \Rightarrow \theta \geq -2$$

$$\text{From(1.5)} \quad 6 + \theta \geq 0 \quad \Rightarrow \theta \geq -6$$

$$\text{From(1.6)} \quad 2 - 2\theta \geq 0 \quad \Rightarrow \theta \leq 1$$

So $-2 \geq \theta \geq 1$. Take $\theta = 1$, the largest value of θ that will maintain feasibility. The new point x' is

$$\begin{bmatrix} 2 \\ 3 \\ 0 \\ 1 \\ 7 \\ 0 \end{bmatrix}$$

and is a basic feasible solution; column 4 enters the basis and column 6 leaves. We see that by increasing a nonbasic variable as much as possible (and bumping into sign constraints), we will reach a new basic feasible solution. This process is called *pivot*. We continue to do this until we have reached an optimal solution, which is done when we can no longer increase a non basic variable.

To recap this procedure ([32]), let x_0 be a basic feasible solution corresponding to ordered basis $B = B(1), B(2), \dots, B(m)$. Let x_{i_0} denote the $B(i)$ th component of x_0 , $1 \leq i \leq m$. Then $Ax_0 = b$

implies $\sum_{i=1}^m x_{i0} A_{B(i)} = b$. Because $\mathcal{B}$ is a basis, we can write each nonbasic column j as a linear combination of the basic columns for certain reals x_{ij} :

$$(1.7) \quad A_j = \sum_{i=1}^m x_{ij} A_{B(i)}$$

By increasing the values of the j th entry from x_0 from 0 to θ ,

$$\begin{aligned} \theta A_j &= \sum_{i=1}^m x_{ij} \theta A_{B(i)} \\ \sum_{i=1}^m x_{ij} \theta A_{B(i)} - \theta A_j &= 0 \end{aligned}$$

and therefore we have

$$\begin{aligned} \sum_{i=1}^m x_{i0} A_{B(i)} - \sum_{i=1}^m x_{ij} \theta A_{B(i)} - \theta A_j &= b \\ (1.8) \quad \sum_{i=1}^m (x_{i0} - \theta x_{ij}) A_{B(i)} - \theta A_j &= b \end{aligned}$$

Now an n -vector can be built as follows:

- Its j th component is θ .
- For each i , its $B(i)$ th component $x_{i0} - x_{ij}\theta$.
- Its other components are 0.

Then (1.8) show that x'_0 satisfies the equality constraints $Ax = b$. If in addition $x_{i0} - x_{ij}\theta \geq 0$ for $i = 1, \dots, m$ then x'_0 also satisfies the sign constraints. We always choose

$$\theta = \theta_0 = \min_{i: x_{ij} > 0} \frac{x_{i0}}{x_{ij}},$$

the largest value of θ that will maintain feasibility.

THEOREM 6. *Given ordered basis $\mathcal{B} = B(1), B(2), \dots, B(m)$ and the basic feasible solution x_0 corresponding to $\mathcal{B}$ (whose $B(i)$ th component is x_{i0} $1 \leq i \leq m$), let $j \notin \mathcal{B}$. Write $A_j = \sum_{i=1}^m x_{ij} A_{B(i)}$. Suppose $\{i \mid x_{ij} > 0\}$ is nonempty. Let θ_0 denote $\min_{i: x_{ij} > 0} \frac{x_{i0}}{x_{ij}}$ and suppose $\theta_0 = \frac{x_{l0}}{x_{lj}}$ (with $x_{lj} > 0$). Let*

$$\mathcal{B}'(i) = \begin{cases} B(i) & i \neq l; \\ j & i = l. \end{cases}$$

Then $\mathcal{B}' = B'(1), B'(2), \dots, B'(m)$ is the basis corresponding to the basic feasible solution x'_0 whose $B'(i)$ th component is $x_{i0} - \theta_0 x_{ij}$, if $i \neq l$, and whose j th component is θ_0 .

Now we will discuss the concept of tableaux. If the columns $A_{B(1)}, \dots, A_{B(m)}$ constituted an identity matrix, pivoting would be easy. In that case, we would use a non basic column A_j as $\sum_{i=1}^m x_{ij} A_{B(i)}$ trivially:

$$A_j = \sum_{i=1}^m a_{ij} A_{B(i)}$$

(where a_{ij} is the (i, j) entry of A). We will force the basic columns to be an identity matrix by executing elementary row operations.

EXAMPLE 5. Suppose we have the system

$$\begin{aligned} 6 &= x_1 + x_2 + 4x_6 \\ 14 &= -2x_1 + 2x_2 + x_3 - x_4 + x_6 \\ -11 &= x_1 - 2x_2 + x_4 + 2x_6 \\ 7 &= x_1 - 3x_4 + x_5 - 5x_6 \\ x_j &\geq 0 \quad \forall j \end{aligned}.$$

Represent it as an $m \times (n + 1)$ matrix, a *tableau*:

	x_1	x_2	x_3	x_4	x_5	x_6
6	1	1	0	0	0	4
14	-2	2	1	-1	0	1
-11	1	-2	0	1	0	2
7	1	0	0	-3	1	-5

The leftmost column is known as column 0. Suppose we choose $B(1) = 2$, $B(2) = 3$, $B(3) = 4$, $B(4) = 5$. Though these four columns are linearly independent, they do not contain an identity matrix. Without changing the solution space, we can put e_i in $A_{B(i)}$ for each i by executing elementary row operations (on all $n + 1$ columns), obtaining:

	x_1	x_2	x_3	x_4	x_5	x_6
6	1	1	0	0	0	4
3	-1	0	1	0	0	3
1	3	0	0	1	0	10
10	10	0	0	0	1	25

By virtue of having the identity matrix in columns $A_{B(1)}, A_{B(2)}, \dots, A_{B(m)}$, the i th entry in the column 0 gives the $B(i)$ th component in the current basic feasible solutions! That is, if X is the name of this matrix, the i th entry in X 's 0th column equals x_{i0} . Furthermore, the (i, j) entry of X is x_{ij} .

Having renamed this tableau A , bring nonbasic column 1 into the basis.

$$A_1 = 1A_2 - A_3 + 3A_4 + 10A_5 = \sum_{i=1}^m x_{i1} A_{B(i)}$$

we compute

$$\theta_0 = \min_{i: x_{i1} > 0} \frac{x_{i0}}{x_{i1}} = \min \left\{ \frac{6}{1}, \frac{1}{3}, \frac{10}{10} \right\} = \frac{1}{3} = \frac{x_{30}}{x_{31}}$$

We now want to do two things, by executing elementary row operations:

- (1) Update column 0, so that it contains the basic entries of the new basic feasible solutions
- (2) Put e_i in column $A_{B'(i)}$, for every i . Of course, it suffices to put e_l into A_j , since the remaining e 's are already where we want them.

Hence we get the following:

	x_1	x_2	x_3	x_4	x_5	x_6
$\frac{17}{3}$	0	1	0	$-\frac{1}{3}$	0	$\frac{2}{3}$
$\frac{10}{3}$	0	0	1	$\frac{1}{3}$	0	$\frac{19}{3}$
$\frac{1}{3}$	1	0	0	$\frac{1}{3}$	0	$\frac{10}{3}$
$\frac{20}{3}$	0	0	0	$-\frac{10}{3}$	1	$-\frac{25}{3}$

The new ordered basis is $B'(1) = 2, B'(2) = 3, B'(3) = 1, B'(4) = 5$, column 1 having replaced column 4 in the basis. The beauty of the tableaux is that doing (2) also does (1). But this is easy to see: if $A_{B'(i)} = e_i$ for $1 \leq i \leq m$, then since we haven't changed the solution space by executing elementary row operations, i th entry of column 0 must be the $B'(i)$ th entry of the basic feasible solution corresponding to B' .

LEMMA 2. *Let X be a tableau (without column 0) after s pivots, where A was the original matrix. Let columns $B(1), B(2), \dots, B(m)$ be the current ordered basis where column $B(i)$ of X is e_i . Then B denote the $m \times m$ matrix whose i th column is $A_{B(i)}$. Then $X = B^{-1}A$. Furthermore, the 0th column is exactly $B^{-1}b$*

Let z be an n -vector whose j th component is $z_j = \sum_{i=1}^m x_{ij}c_{B(i)}$ where c_B is the m -vector whose i th component is $c_{B(i)}$; $(c_B)_i = c_{Bi}$. In this notation, $z^T = c_B^T X = c_B^T B^{-1}A$

In order to pivot we need to find a negative entry $\bar{c}_j$, but if $\bar{c} \geq 0$ then the current basic feasible solution is optimal.

THEOREM 7. *If $\bar{c} = c - z \geq 0$ at bfs x_0 , then x_0 is optimal.*

LEMMA 3. *For all i , $\bar{c}_{B(i)} = 0$, i.e., the modified cost of a basic column is 0.*

The modified cost of the $\bar{c}$ is important for this reason we carry them along in the computation. This is done in row 0 of the tableau, as follows:

- (1) Initially we write $\bar{c}_j$ in the $(0, j)$ position, and place a 0 in cell $(0, 0)$.
- (2) Let $B(1), B(2), \dots, B(m)$ be a basis whose basic solution is feasible. Place e_i in column $B(i)$ for each i by performing elementary row operations.
- (3) As yet, we still have c in row 0, not $\bar{c}$. If j is basic, $\bar{c}_j$ is 0. So we place 0s above the basic columns, by executing $(\text{row } 0) \leftarrow (\text{row } 0) - c_{B(i)}(\text{row } i)$ for $i = 1, \dots, m$. (Do this to all of row 0 including column 0).
- (4) The j th entry of row 0 ($j \geq 1$) is now $c_j - \sum_{i=1}^m x_{ij}c_{B(i)} = \bar{c}_j$, the j th modified cost. In column 0 we now have $-\sum_{i=1}^m x_{i0}c_{B(i)} - (\text{cost of current basic feasible solutions})$.

LEMMA 4. *(Lemma 17 in [32]) The $(0, 0)$ entry will always hold the negative of the cost of the current bfs, and the $(0, j)$ entry will always hold $\bar{c}_j$ (for $j \geq 1$), if whenever we pivot we perform a row operation so as to zero out the entry atop the entering column.*

THEOREM 8. *If $\bar{c}_j < 0$ and $x_{ij} \leq 0$ for all i , then the LPS instance is unbounded.*

To tie this all together, consider the following example:

EXAMPLE 6.

$$\begin{array}{ll}
 \text{Minimize} & 19x_1 - 13x_3 + 3x_6 \\
 \text{such that} & 3x_1 - 4x_3 + x_4 + 4x_5 = 6 \\
 & -4x_1 + x_2 + x_3 + 2x_5 = 1 \\
 & 5x_1 - 2x_3 - x_5 + x_6 = 1 \\
 & x_i \geq 0 \quad \forall i
 \end{array}$$

Represent it as a *tableau*:

0	19	0	-13	1	0	3
6	3	0	-4	1	4	0
1	-4	1	1	0	2	0
1	5	0	-2	0	-1	1

For the ordered basis $B(1) = 4, B(2) = 2$, and $B(3) = 6$, we already have e_i in column $B(i)$ and a 0 above column 2, but not above columns 4 or 6. We subtract 1(row 1) and then 3(row 3) from row 0 in order to zero out the fourth and sixth entries in row 0.

-9	1	0	-3	0	-1	0
6	3	0	-4	1	4	0
1	-4	1	1	0	2	0
1	5	0	-2	0	-1	1

Now we choose $j = 4$ and then $l = 2$. We pivot on the 2 and zero out the fifth entry of row 0.

$-\frac{17}{2}$	-1	$\frac{1}{2}$	$-\frac{5}{2}$	0	0	0
4	11	-2	-6	1	0	0
$\frac{1}{2}$	-2	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0
$\frac{3}{2}$	3	$\frac{1}{2}$	$-\frac{3}{2}$	0	0	1

Choose $j=3$ and then $l=2$, we now pivot on the $(2,3)$ entry.

-6	-11	3	0	0	5	0
10	-13	4	0	1	12	0
1	-4	1	1	0	2	0
3	-3	2	0	0	3	1

Now choose $j = 1$. We conclude from the negative $\bar{c}_j$ and the non positive numbers in column j that the objective functions can be made arbitrarily small. Hence,

$$\begin{bmatrix} \theta \\ 0 \\ 1 + 4\theta \\ 10 + 13\theta \\ 0 \\ 3 + 3\theta \end{bmatrix}$$

is feasible for all $\theta \geq 0$ and of cost $6 - 11\theta$.

In order to ensure that the Simplex Method will terminate, it is enough to ensure that we do not reuse a basis. There are many ways to choose the entering and leaving variables. Thus there exist particular rules, *pivoting rules*, that make these choices ambiguous. There are many different types of pivoting rules. *Bland's Rule* gives us a way to select the entering and exiting columns in such a way that the Simplex Method must terminate.

Bland's Rule Choose the smallest numbered column as possible to enter the basis, and then, the smallest possible numbered column to leave it. More precisely, always choose j minimal such that $\bar{c}_j < 0$, and afterward choose an l for which $B(l)$ is as small as possible, from among all l satisfying $x_{lj} > 0$ and $\frac{x_{l0}}{x_{lj}} = \theta_0$. (Be careful. Choose l so that $B(l)$ is minimum, not l itself.)

THEOREM 9. *Simplex terminates as long as the pivot row and column are chosen by Bland's Rule.*

So far we have discussed how the Simplex Method operates, however, we have only looked at cases where we have non-negative solutions. This ensured that we had a feasible solution. The best case scenario is when we start with

$$\begin{aligned} \min \quad & c^T x \\ & Ax \leq b \\ & x \geq 0 \end{aligned}$$

and $b \geq 0$. All we have to do is convert the problem into standard form by adding n nonnegative slack variables. They form a basic feasible solution and the corresponding columns are a basis.

Now, let's discuss what to do if no bfs is evident. We can use the *artificial-variable* or the *two-phase* method. Let

$$\begin{aligned} \min \quad & c^T x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

be our initial problem. By negating some constraints we may assume without loss of generality that $b \geq 0$.

Two Phase Method First reduce $\boxed{A|b}$ two-echelon form. If the rank of A is r yet the i th row of A for some $i > r$ has a nonzero right-hand side (and all-zero left-hand side), report infeasibility and halt. If no such rows exist, eliminate the all-zero rows. Call this new matrix A and let b be the new right-hand side. If A is $m \times n$, $\text{rank}(A) = m$.

(Phase I) Add m new "artificial" variables x'_i , $1 \leq i \leq m$:

$$\begin{array}{rcl} x'_1 & +a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = b_1 \\ x'_2 & +a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = b_2 \\ & \vdots & \\ & x'_m + a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & = b_m \end{array}$$

$$x'_i \geq 0 \ \forall i, \ x_j \geq 0 \ \forall j.$$

Minimize $\sum_{i=1}^m x'_i$, starting with the basic feasible solution in which $x'_i = b_i$ for $i = 1, 2, \dots, m$.

LEMMA 5. (Lemma 20 from [32]) *This new problem has optimal value zero if and only if the original problem is feasible.*

Now we have two options. If the optimal value resulting from Phase I is strictly positive, it is infeasible and STOP! If not, $x^*_1, x^*_2, \dots, x^*_n$ is the sequence of values of the *original* variables

returned by the algorithm, $x^* = \begin{bmatrix} x^*_1 \\ x^*_2 \\ \vdots \\ x^*_n \end{bmatrix}$ is a feasible point in our problem.

x^* is a vertex of our original problem if the Phase I linear program was solved by the Simplex Algorithm. If Phase I was not solved by the Simplex Method, we need to find basic feasible solutions v of no greater cost than that of x^* and a basis corresponding to v .

(Phase II) Solve

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Starting with the basis obtained in Phase I.

In summary, Phase I is often noted as the process of finding an initial feasible solution, Phase II is the process of going from a feasible solution to the optimal solution.

For further information on the Simplex Method refer to [32].

1.3.3. A Short Overview of Duality. In this section we want to review the basics of linear optimization, often used as a subroutine in the optimization problems (integer programming). We are also going to discuss two algebraic-geometric connections in the solution of linear programs.

The goals of these lectures are

- (1) Review the basics of optimization problems (integer programming)
- (2) Discuss two algebraic-geometric connections in the solution of linear programs
- (3) Introduce students to a large class of open problems.

A linear program, LP, is a problem of the form

$$(1.9) \quad \max / \min \{c^T x \mid Ax = b, Bx \geq d\}.$$

If it is easy to see that, by adding slack variables, one can consider (standard form)

$$(1.10) \quad (P) \quad \min \{c^T x \mid Ax = b, x \geq 0\}$$

where A is an $m \times n$ rational matrix, and b is a rational vector.

One of the key properties of LPS is a rich duality theory. LPS come in pairs, and there is an associated dual problem:

$$(1.11) \quad (D) \quad \max \{b^T y \mid A^T y \leq c\} \Rightarrow \max \{b^T y \mid A^T y + s = c, s \geq 0\}$$

THEOREM 10. (Weak Duality) For each feasible solution y of (D) and $x \in (P)$ we have $b^T y \leq c^T x$. In particular,

$$\begin{cases} (P) \text{ is unbounded} \Rightarrow (D) \text{ is infeasible} \\ (D) \text{ is unbounded} \Rightarrow (P) \text{ is infeasible} \end{cases}$$

THEOREM 11. (Strong Duality) For the pairs of linear programs (P),(D), only one of the following possibilities occurs:

- $(P), (D)$ both infeasible
- (P) unbounded, (D) infeasible
- (P) infeasible, (D) unbounded
- $(P), (D)$ both feasible

The optimal solution $\mathbf{x}^*, \mathbf{y}^*$ must satisfy $\Leftrightarrow \mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*$.

THEOREM 12. (Complementary Slackness) $\mathbf{x}, \mathbf{y}$ are optimal $\Leftrightarrow \mathbf{x}^T \mathbf{s} = 0$.

This theorem gives a strong combinatorial flavor to linear programs. The proof of Theorem 12 is Farka's Alternative theorem and we present it in the exercises.

COROLLARY 4. The set of optimal solutions of an LP pair (P, D) is precisely the set of solutions of the polynomial system of equalities/inequalities.

$\begin{array}{ll} A\mathbf{x} = \mathbf{b} & A^T \mathbf{y} - \mathbf{s} = \mathbf{c} \\ & \mathbf{x}^T \mathbf{s} = 0 \\ \mathbf{x} \geq 0 & \mathbf{s} \geq 0 \end{array}$

Most algorithms to solve the problem are obtained by fixing some of the 5 conditions and gradually adjusting the rest until satisfied (then we have reached optimality!).

1.3.4. Pivot Methods. Among all feasible solutions of an LP, we have:

DEFINITION 7. A feasible solution $\mathbf{x}$ of (P) is basic if there exists an m -subset of $\{1, 2, \dots, n\}$ such that A_B is a nonsingular matrix and $\mathbf{x}_j = 0$ for all $j \notin B$. We call such a solution a *basic feasible solution*, a term that is usually abbreviated by BFS. We say the index set B is a *basis*.

- LEMMA 6.**
- (1) If an optimal solution exists, then there exists a basic feasible solution (BFS).
 - (2) The set of feasible solutions $\{\mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ is an $(n - m)$ -dimensional polyhedron and its extreme points (vertices) correspond to the basic feasible solutions (BFS).

1.3.4.1. Simplex Pivot Methods. A simplex (primal) method starts with a primal basic feasible solution (BFS). If it is neither optimal nor the problem is unbounded, then find a new adjacent BFS such that

- (1) A dual infeasible variable enters the basis (and the primal leaves).
- (2) The new basis differs by exactly one element from the old basis.
- (3) Primal feasibility is preserved throughout.
- (4) The primal objective function is monotonically decreasing at each basis exchange.

1.3.5. Efficiency of the Simplex Method. The creation of the Simplex Algorithm sparked much curiosity and many mathematicians began to study it. Warren M. Hirsch was one of them. Hirsch was so motivated by the analysis of the Simplex Method, that he constructed a conjecture, that if true, would provide a lower bound on the number of steps needed by the simplex method.

The *combinatorial diameter* is described as the number of steps it takes to go from one vertex to another. A step is the action of traveling along an edge.

CONJECTURE 1. Hirsch Conjecture(1947) The (combinatorial) diameter of a polytope of dimension d with n facets cannot be greater than $n - d$.

Unfortunately, after 53 years, Francisco Santos proved that in fact the conjecture is false. He constructed a 43-dimensional polytope counterexample with 86 facets and a diameter at least of 44. Although Santos' counterexample is theoretically correct, it is computationally unfeasible. Thus, we will extensively analyze Santos' counterexample in Chapter 3 of this thesis, to see if there is a way to create a computationally feasible counterexample.

Although a counterexample was found to the Hirsch conjecture, this does not impact the Simplex method, since it would not rule out the possibility of a larger but still linear or polynomial number of steps.

Estimating the Number of Lattice Points Inside Polyhedra

2.1. Estimation and Approximate Counting

Since exact counting is difficult there is a rich and exciting theory of estimation and approximation, which is what we will focus on in half of this thesis.

Now we present various algorithms applicable to arbitrary polyhedra (much work has also been done for special families, such as contingency tables, see for instance [3, 4, 7, 18, 25, 47] and the many references there). Unfortunately we cannot avoid omitting part of this interesting literature. We recommend the surveys [28, 35, 34, 56] and the references therein for more information. One important point in the theory of approximate counting was that in 1986 Jerrum, Valiant and Vazirani showed that the existence of a good approximation (i.e. a fully polynomial randomized approximation scheme) is equivalent to almost uniform sampling from the set of lattice points [36]. Thus with the close connection to sampling the effort has often been on finding reasonable sampling procedures.

One approach is *direct Monte-Carlo sampling* or *dart-throwing techniques*. The basic framework is that we wish to estimate the size $|S|$ of the lattice point set S of a given polyhedron. To do this, we enclose S in an easy set S' which approximates S and can be sampled uniformly (think of S' a nice box around a nasty looking S). The process is then to sample from S' to get an estimate $\hat{p}$ of the proportion $|S|/|S'|$ of sampled points which lie in S . Estimate $|S|$ by $\hat{p}|S'|$. Of course, the key to using this approach is to find a set S' such that $S \subset S'$, $|S'|$ is easy to determine and can be easily sampled. Finally, it is desirable that $|S'|/|S|$ is polynomially bounded. There are several clever variations of this simple idea. One noteworthy example is the combination of dart-throwing with dynamic programming which was proposed by Dyer in [27]. Another example is the use of randomized rounding to improve the dart-throwing technique [38].

Sequential importance sampling uses the idea of divide and conquer to sequentially build up the proposed distribution by sampling variable entries one by one. Roughly speaking, the value of a particular solution for $Ax = b$, $x \geq 0$ can be done entry by entry, where x_i is constrained by the choices made for $x_1, \dots, x_{i-1}$. The sampling is easy to do if geometrically the possible solutions for x_i form an interval of values without gaps. In the paper [13] many real examples show that sequential importance sampling is applicable and useful, and can be used in cases where a Markov basis cannot be easily computed (e.g., multiway contingency tables). This method had great success on the problem of estimating the number of labeled graphs with a prescribed degree sequence (see [10]).

Of course there are other approaches that do not quite fit the above schemes. For instance, the integral representation of a counting problem can also be used to calculate asymptotic approximations. See [25, 40]. More lately, a strong connection between convex optimization and integer

feasibility and counting has been observed by several authors [33, 45, 46, 5]. Most recently Barvinok and Hartigan [5] presented probabilistic theorems for the estimation of the number of lattice points inside a polyhedron which are based in convex programming tools. To conclude this section, we take a closer look at this very recent development.

2.1.1. Estimation through Convex Optimization. We now briefly recount the main results presented in [5] that we used here to compute estimations on the number of lattice points inside a polyhedron by solving specially constructed convex optimization problems on polytopes. Unlike other techniques the Barvinok Hartigan idea uses no prior knowledge about the input polyhedron. In what follows, a *polytope* $P \subset \mathbb{R}^n$ is the set of solutions to the system $Ax = b, x \geq 0$ where $A = [a_1, \dots, a_n]$ is the matrix with d -dimensional vectors $a_1, \dots, a_n$ and x is thought of as a column vector $x = [\xi_1, \dots, \xi_n]^T$. For simplicity we assume that vectors $a_1, \dots, a_n$ span $\mathbb{R}^d$ and that in fact the vectors $a_1, \dots, a_n$ and b are integral, that is, $a_1, \dots, a_n; b \in \mathbb{Z}^d$. We also assume that P has a non-empty interior, that is, contains a point $x = (\xi_1, \dots, \xi_n)$, where the defining inequalities are strict (it is well-known this can be checked in fast both in theory and in practice). We wish to estimate the number $|P \cap \mathbb{Z}^n|$ of integer points in P . We also consider the problem of counting the set of all vectors in P with the coordinates 0 and 1.

Recall that a discrete random variable x has geometric distribution if $\Pr\{x = k\} = pq^k$ for $k = 0, 1, \dots$ for positive numbers p and q such that $p + q = 1$. For the expectation and variance of x we have $x = \frac{q}{p}$ and $x = \frac{q}{p^2}$ respectively. Conversely, if $x = \zeta$ for some $\zeta > 0$ then $p = \frac{1}{1+\zeta}$, $q = \frac{\zeta}{1+\zeta}$ and $x = \zeta^2 + \zeta$.

THEOREM 13 (Theorem 2.1 in [5]). *Let $P \subset \mathbb{R}^n$ be the intersection of an affine subspace in $\mathbb{R}^n$ and the non-negative orthant $\mathbb{R}_+^n$. Suppose that P is bounded and has a non-empty interior, that is contains a point $y = (\eta_1, \dots, \eta_n)$ where $\eta_j > 0$ for $j = 1, \dots, n$.*

Then the strictly concave function

$$g(x) = \sum_{j=1}^n \left((\xi_j + 1) \ln(\xi_j + 1) - \xi_j \ln \xi_j \right)$$

attains its maximum value on P at a unique point $z = (\zeta_1, \dots, \zeta_n)$ such that $\zeta_j > 0$ for $j = 1, \dots, n$.

Suppose now that x_j are independent geometric random variables with expectations ζ_j for $j = 1, \dots, n$. Let $X = (x_1, \dots, x_n)$. Then the probability mass function of X is constant on $P \cap \mathbb{Z}^n$ and equal to $e^{-g(z)}$ at every $x \in P \cap \mathbb{Z}^n$. In particular,

$$e^{g(z)} \Pr\{X \in P\} = |P \cap \mathbb{Z}^n|.$$

Similarly, Barvinok and Hartigan proposed a clever probability mass distribution which is constant on the 0-1 points. Let p and q be positive numbers such that $p + q = 1$. We recall that a discrete random variable x has Bernoulli distribution if $\Pr\{x = 0\} = p$ and $\Pr\{x = 1\} = q$. We have $x = q$ and $x = qp$. Conversely, if $x = \zeta$ for some $0 < \zeta < 1$ then $p = 1 - \zeta$, $q = \zeta$ and $x = \zeta - \zeta^2$. Their second main result is as follows.

THEOREM 14 (Theorem 2.4 in [5]). *Let $P \subset \mathbb{R}^n$ be the intersection of an affine subspace in $\mathbb{R}^n$ and the unit cube $\{0 \leq \xi_j \leq 1 : j = 1, \dots, n\}$. Suppose that P has a non-empty interior, that is, contains a point $y = (\eta_1, \dots, \eta_n)$ where $0 < \eta_j < 1$ for $j = 1, \dots, n$. Then the strictly concave*

function

$$h(x) = \sum_{j=1}^n \left(\xi_j \ln \frac{1}{\xi_j} + (1 - \xi_j) \ln \frac{1}{1 - \xi_j} \right)$$

attains its maximum value on P at a unique point $z = (\zeta_1, \dots, \zeta_n)$ such that $0 < \zeta_j < 1$ for $j = 1, \dots, n$.

Suppose now that x_j are independent Bernoulli random variables with expectations ζ_j for $j = 1, \dots, n$. Let $X = (x_1, \dots, x_n)$. Then the probability mass function of X is constant on $P \cap \{0, 1\}^n$ and equal to $e^{-h(z)}$ for every $x \in P \cap \{0, 1\}^n$. In particular,

$$e^{h(z)} \Pr\{X \in P\} = |P \cap \{0, 1\}^n|.$$

Now we discuss the important practical consequences of the above results. First of all, clearly the two theorems above suggest two simple sampling algorithms

- (1) Solve the optimization problem
 - $\gamma(P) = \max \sum_{j=1}^n g(x_j)$
 - subject to $x \in P$.
- (2) Let $z = (\zeta_1, \dots, \zeta_n)$ be the unique optimal point and γ the optimum value. Sample a random point $X = (x_1, \dots, x_n)$ by sampling x_j independently at random from the geometric distribution with expectation ζ_j , that is with parameter $p_j = (1 + \zeta_j)^{-1}$. Check whether $X \in P$. Repeat many times and compute the proportion α of times that the point X lands in P (i.e., $0 \leq \alpha \leq 1$).
- (3) Estimate $|P \cap \mathbb{Z}^n| \approx \alpha e^\gamma$.

Similarly, by using the entropy function h of Theorem 14, instead of the concave function g of Theorem 13, we can get a sampling algorithm to estimate 0-1 points inside P :

- (1) Solve the optimization problem
 - $\gamma(P) = \max \sum_{j=1}^n h(x_j)$
 - subject to $x \in P$.
- (2) Let $z = (\zeta_1, \dots, \zeta_n)$ be the unique optimal point and γ the optimum value. Sample a random binary point $X = (x_1, \dots, x_n)$ by sampling x_j independently at random from the Bernoulli distribution where x_i is equal to 1 with probability ζ_j .
- (3) Check whether $X \in P$. Repeat many times and compute the proportion α of times that the point X lands in P (i.e., $0 \leq \alpha \leq 1$). Estimate $|P \cap \{0, 1\}^n| \approx \alpha e^\gamma$.

More important, Barvinok and Hartigan developed a heuristic approximation to the above numbers that does not require sampling. They proved theorems stating sufficient conditions for the formulas to actually give a good approximation and showed, in particular, that for k -way contingency tables the formulas are asymptotically exact if $k \geq 4$. Later in [6] they computed the correction factor in the case of $k = 2$. We describe now the *Gaussian approximation method* of Barvinok and Hartigan, beginning with general integer programs. For a polytope P , defined by a system $Ax = b, x \geq 0$, we find the point $z = (\zeta_1, \dots, \zeta_n)$ maximizing

$$g(x) = \sum_{j=1}^n \left((\xi_j + 1) \ln (\xi_j + 1) - \xi_j \ln \xi_j \right)$$

on P . Let $A = (\alpha_{ij})$. We compute the $d \times d$ matrix $Q = (q_{ij})$ by

$$q_{ij} = \sum_{k=1}^n \alpha_{ik} \alpha_{jk} (\zeta_k^2 + \zeta_k).$$

We assume that A is an integer $n \times d$ matrix of rank $d < n$. Let $\Lambda = A(\mathbb{Z}^n)$ be image of the standard lattice, $\Lambda \subset \mathbb{Z}^d$. The proposed approximation the number of integer points in P is given by

$$|P \cap \mathbb{Z}^n| \approx \frac{\det \Lambda}{(2\pi)^{d/2} (\det Q)^{1/2}}.$$

Similarly, for estimating 0-1 points inside P , we find the point $z = (\zeta_1, \dots, \zeta_n)$ maximizing

$$h(x) = \sum_{j=1}^n \left(\xi_j \ln \frac{1}{\xi_j} + (1 - \xi_j) \ln \frac{1}{1 - \xi_j} \right)$$

on the truncated polytope defined by the equations $Ax = b$ and inequalities $0 \leq x \leq 1$ (that is, $0 \leq \xi_j \leq 1$ for $j = 1, \dots, n$). We then compute the $d \times d$ matrix $Q = (q_{ij})$ by

$$q_{ij} = \sum_{k=1}^n \alpha_{ik} \alpha_{jk} (\zeta_k - \zeta_k^2).$$

This way the desired approximation to the number of 0-1 vectors in P is given by

$$|P \cap \{0, 1\}^n| \approx \frac{\det \Lambda}{(2\pi)^{d/2} (\det Q)^{1/2}},$$

where $\Lambda = A(\mathbb{Z}^n)$.

It is well-known that all the above mentioned optimization problems can be solved quite efficiently, both in theory and in practice, by interior point methods, see [9, 49]. This is an additional attraction of these ideas. To conclude we report on the practical experimental performance of the two estimation algorithms derived from the theory in [5]. Our test sets included Knapsacks (general, some randomly generated and others artificially constructed), multiway transportation problems, market split problems. In all instances, the goal was to measure the accuracy of the estimation. We observed that, in a good number of cases, the Barvinok-Hartigan techniques perform rather well. Most important, the methods used here require no prior knowledge of the polyhedron (e.g. integral feasibility) and work for arbitrary general rational polyhedra. We divided our experiments into three families: (1) knapsack simplices, (2) two and three-way contingency tables, (3) market split problems. Each category is reported in the Appendix, Section 2.2. Note that categories (1) and (2) are polyhedra with lattice points with “large” integer entries, while category (3) is testing examples of binary polyhedra. In all cases the problems are given in the canonical form $Ax = b, x \geq 0$ and we checked for accuracy of the estimation using the software **LatTE** and **Azove**.

All computations were done on a 2 GHz Pentium PC running Red Hat Linux. For the solution of the necessary convex optimization problems we used the software **LOQO** a software package for solving general (smooth) nonlinear optimization problems. **LOQO** implements an infeasible-primal-dual path-following method. Even though **LOQO** can handle nonconvex problems in general, it performs much better with convex problems (see [8, 54, 55] and references therein). This was indeed the case in our situation and in absolutely all the instances that we tested the number of

iterations was never above twenty. Thus we can say that finding the necessary estimation parameters can be done easily and efficiently.

The simulation of the geometric and the Bernoulli distributions used in the two sampling algorithms was done using the statistics package of MAPLE version 12. Each experiment consisted on finding the parameters γ and the probability distribution numbers for each instance. Then we sampled three different times with (1000), (5000), and (10000) samples and the same test parameters read from LOQO we report on the average. As described in the algorithm, the sampling parameters are the exponential exponent γ and the parameters for geometric and Bernoulli random variables which come in the form of a point of the polytope.

The evidence we have presented in the Appendix supports the usefulness and accuracy of the Barvinok-Hartigan Gaussian estimation. The easy-to-use Gaussian estimation is widely applicable and fairly accurate in most instances we tried. In 40 percent of tests done using the gaussian estimation the relative error of the gaussian estimation was 5% or less. Only in the most pathological cases (e.g. market split, Aardal knapsacks) one can see a prediction worse than twice the correct value. See Figure 1 for details. We noted also that the algorithms seem to scale well as the number of variables grows. The sampling algorithm is nevertheless not very good. It performed best for knapsack problems but when two or more constraints are in play it is clearly not as reliable for estimating the number of lattice points in polyhedra. Using aggregation of constraints did not help with the problem.

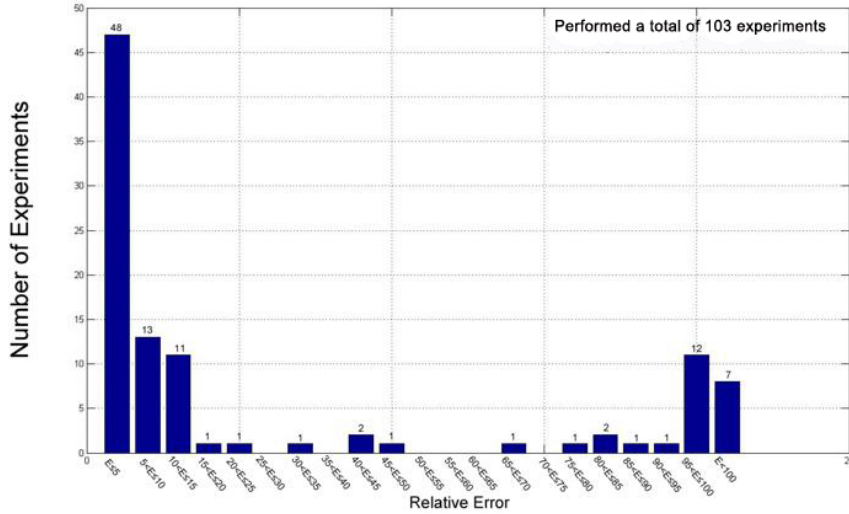


FIGURE 1. Error estimation over tests

Of course, as expected from the theoretical hardness of the problem, one can indeed find rather nasty problems for which the number of lattice points grows arbitrarily large and yet the estimation is consistently poor. Nevertheless, the experiments actually suggests that the convex optimization problems solved for finding the sampling parameters could in principle be used as a fairly good measure of how “well-behaved” is an integer linear program will turn out to be.

We conclude by saying that although other algorithms are available to carry out the same kind of estimation, we are not aware of any other algorithm that works in such a generality without prior knowledge of the integer feasibility of the polyhedron in question. For instance, Importance Sampling method rely on the ability of generating random lattice points according to a pivot or completion procedures. This is unavailable in most polyhedra. Thus the Barvinok-Hartigan process has a distinctive advantage of not requiring prior knowledge about the input polyhedron.

2.2. Details on Experiments

Here we present details for experiments with the Gaussian estimation of Barvinok and Hartigan. As described before, we needed the maximum objective values and the unique optimal solutions of the strictly concave optimization problems from Theorems 13 and 14 in order to calculate the Gaussian estimations. These values were obtained using LOQO and the rest of the calculation was implemented inside MAPLE version 12. There is no need for any sophisticated computation, one can easily calculate the covariance matrices and the lattice index using a Smith normal form calculation. Overall the evaluation step takes a negligible amount time in all instances, so we do not record it any time of computation.

2.2.1. General knapsacks. The first set of problems consist of knapsack-constrained simplices. The majority of the instances were randomly generated with coefficients between 1 and 20, but we also used a few famous hard knapsack problems from [1] and a few instances with coefficients in arithmetic progressions. The presence of these two kinds of tests we hope is useful on investigating how suitable are these techniques for feasibility detection and to test the sensibility of the estimate to the variation of right-hand-side vector. Tables 1 and 2 present the data used here and the results. On the line below an instance $ax = b$ we report the true number of lattice points and the relative error of the estimation after 5000 samples.

In Table 2 we present the results of estimation using the maximum entropy Gaussian estimate: Although there are a few difficult cases, the majority of instances have relative error less than one.

The Aardal style knapsack's (examples 13, 14 in the tables) [1] show a dramatically poor behavior which seems to be correlated to the value objective function value γ of the convex function we maximize. By Theorem 13 this number is a close indicator of success to generate a lattice point during sampling. For example, problem number #7 with right hand side 22382774 has no solutions, but if we increase it by one to 22382775 as shown in #7, there are 218842 integer solutions. In both cases the values for γ and probability parameters were identical ($\gamma = 47.58769$). Through sampling we expect to require $e^{47.6} = (4.7) \times 10^{20}$ samples before finding a lattice point.

Nevertheless, for the Aardal examples, in addition of the standard convex maximization of Theorem 13 we also used the weighted version and we were surprised to see that in both instances we found an actual integral feasible solution already using the convex optimization procedure. The weights were generated at random, so this raises the issue of understanding whether for good weights one can uncover integer solutions systematically.

2.2.2. Contingency Tables and Network Flows. We continue our investigations with the class of *multiway contingency tables*. A d -table of size $(n_1, \dots, n_d)$ is an array of non-negative integers $v = (v_{i_1, \dots, i_d})$, $1 \leq i_j \leq n_j$. For $0 \leq m < d$, an m -marginal of v is any of the $\binom{d}{m}$ possible m -tables obtained by summing the entries over all but m indices.

Problem	$ax = b$, sampling error, and number of solutions
#1	a=[2, 11, 18, 4, 17, 19, 6, 9, 2, 10, 16, 4, 18, 1, 15, 6, 17, 2, 8, 10, 7, 19, 7, 10, 14] b=5000 Error=0.0309433 numsols=8569641458133826414663483924452506094742800
#2	a=[5, 10, 10, 2, 8, 20, 15, 2, 9, 9, 7, 4, 12, 13, 19] b=34 Error=0.01329 numsols=2056
#3	a=[10, 15, 9, 12, 11, 20, 8, 9, 17, 18, 11, 20, 13, 8, 17] b=19 Error=0 numsols=6
#4	a=[15, 13, 6, 2, 12, 4, 6, 5, 17, 8, 5, 2, 18, 20, 11] b=500 Error=0.06840749 numsols=242818430132799
#5	a=[19, 14, 18, 15, 8, 10, 14, 12, 9, 13, 16, 1, 6, 13, 14] b=500 Error=0.290757 numsols=3489295417911
#6	a=[7, 7, 3, 14, 15, 15, 19, 12, 19, 8, 3, 17, 17, 3, 5] b= 50 Error=0.015823 numsols=8438
#7	a=[20601 , 40429 ,40429 ,45415 ,53725 ,61919 ,64470 ,69340 ,78539 ,95043] b=22382775 Error=1 numsols=218842
#8	a=[1,2,3,4,5,6,7,8,9,10,11,12] b=255 Error=0.131238 numsols=71660385050
#9	a=[1,2,3,4,5,6,7,8,9,10,11,12] b=24 Error=0.019237 numsols=1380
#10	a=[9, 1, 19, 4, 6, 3, 4, 10, 8, 2] b=50 Error=0.0159488 numsols=47761
#11	a=[16, 2, 11, 15, 11, 3, 5, 14, 7, 2] b=500 Error=0.037474 numsols=65552595947
#12	a=[8, 10, 9, 17, 2, 9, 3, 2, 5, 20] b=25 Error=0.00589 numsols=267
#13	a=[12137 ,24269 ,36405 ,36407 ,48545 ,60683] b=58925135 Error=1 numsols=2
#14	a=[12223,12224,36674,61119,85569] b= 89643482 Error=1 numsols=1
#15	a=[1,2,3,4,5,6] b=25 Error=0.03970 numsols=612
#16	a=[9, 10,17,5,2] b=73 Error=0.00208 numsols=209
#17	a=[9, 11, 14, 5,12] b=26 Error=0.015148 numsols=3
#18	a=[5, 3, 1, 4, 2] b=15 Error=0.02186 numsols=84
#19	a=[5, 13, 2, 8, 3] b=17 Error=0.02991 numsols=12
#20	a=[8,12,11] b=50 Error=0.1535 numsols=1

TABLE 1. Knapsack tests using sampling algorithm, information on relative error (best possible among three sampling tests)

Contingency tables appear naturally in statistics and operations research under various names such as *multi-way contingency tables*, or *tabular data*. Table counting has several applications in statistical analysis, in particular for independence testing, and has been the focus of much research (see [25] and the extensive list of references therein). Given a specified collection of marginals for d -tables of size $(n_1, \dots, n_d)$ (possibly together with specified lower and upper bounds on some of the table entries) the associated *multi-index transportation polytope* is the set of all non-negative *real valued* arrays satisfying these marginals and entry bounds. The counting problem can be formulated as that of counting the number of integer points in the associated multi-index transportation polytope.

First we treat *two-way contingency tables*. The polytope defined by a two-way contingency table is called the *transportation polytope*. We present the results in Table 2.2.2. This family of polytopes has been studied in their lattice points by several authors [3, 4, 7, 18, 47] and thus are good for testing accuracy of the estimation. We used several examples of 4×4 transportation polytopes, as presented in the table below. In all cases the relative error was very similar to the performance with simplices, but in this case we are simply using the Gaussian estimator proposed

Problem	$ax = b$, Gaussian estimate relative error, and number of solutions
#1	$a=[2, 11, 18, 4, 17, 19, 6, 9, 2, 10, 16, 4, 18, 1, 15, 6, 17, 2, 8, 10, 7, 19, 7, 10, 14]$ $b=5000$ Error= 0.00334 numsols=8569641458133826414663483924452506094742800
#2	$a=[5, 10, 10, 2, 8, 20, 15, 2, 9, 9, 7, 4, 12, 13, 19]$ $b=34$ Error= 0.00585 numsols=2056
#3	$a=[10, 15, 9, 12, 11, 20, 8, 9, 17, 18, 11, 20, 13, 8, 17]$ $b=19$ Error= 0.97855 numsols=6
#4	$a=[15, 13, 6, 2, 12, 4, 6, 5, 17, 8, 5, 2, 18, 20, 11]$ $b=500$ Error= 0.79371 numsols=242818430132799
#5	$a=[19, 14, 18, 15, 8, 10, 14, 12, 9, 13, 16, 1, 6, 13, 14]$ $b=500$ Error= 0.00618 numsols=3489295417911
#6	$a=[7, 7, 3, 14, 15, 15, 19, 12, 19, 8, 3, 17, 17, 3, 5]$ $b=50$ Error= 0.96326 numsols=8438
#7	$a=[20601, 40429, 40429, 45415, 53725, 61919, 64470, 69340, 78539, 95043]$ $b=22382775$ Error= 118163207.8 numsols=218842
#8	$a=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$ $b=255$ Error= 0.89587 numsols=71660385050
#9	$a=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$ $b=24$ Error= 0.02503 numsols=1380
#10	$a=[9, 1, 19, 4, 6, 3, 4, 10, 8, 2]$ $b=50$ Error= 0.01428 numsols=47761
#11	$a=[16, 2, 11, 15, 11, 3, 5, 14, 7, 2]$ $b=500$ Error= 3.16010 numsols=65552595947
#12	$a=[8, 10, 9, 17, 2, 9, 3, 2, 5, 20]$ $b=25$ Error= 0.04816 numsols=267
#13	$a=[12137, 24269, 36405, 36407, 48545, 60683]$ $b=58925135$ Error= 2634013736 numsols=2
#14	$a=[12223, 12224, 36674, 61119, 85569]$ $b=89643482$ Error= 95905325.28 numsols=1
#15	$a=[1, 2, 3, 4, 5, 6]$ $b=25$ Error=0.02599 numsols=612
#16	$a=[9, 10, 17, 5, 2]$ $b=73$ Error= 0.03213 numsols=209
#17	$a=[9, 11, 14, 5, 12]$ $b=26$ Error= 0.81445 numsols=3
#18	$a=[5, 3, 1, 4, 2]$ $b=15$ Error=0.02923 numsols=84
#19	$a=[5, 13, 2, 8, 3]$ $b=17$ Error= 0.40546 numsols=12
#20	$a=[8, 12, 11]$ $b=50$ Error= 1.0890 numsols=1

TABLE 2. Knapsack tests for Gaussian estimation, we report the relative error.

by Barvinok and Hartigan [5]. The results are good. It should be remarked that sampling did not do well at all for these instances. In fact, in all problems with multiple constraints (non-knapsack) we had a bad performance of sampling, thus we aggregated the constraints in a few instances, but that did not improve the behavior.

Margins	exact # of lattice points	Error on estimation
[1,1,1,1], [1,1,1,1]	24	0.36277
[300,300,300,300], [300,300,300,300]	20269699596926337751	0.12179
[220, 215, 93, 64], [108, 286, 71, 127]	1225914276768514	0.06062
[109, 127, 69, 109], [119, 86, 108, 101]	993810896945891	0.117701
[72, 67, 47, 96], [70, 70, 51, 91]	25387360604030	0.080141
[179909, 258827, 224919, 61909], [190019, 90636, 276208, 168701]	13571026063401838164668296635065899923152079	0.03177
[229623, 259723, 132135, 310952], [279858, 170568, 297181, 184826]	646911395459296645200004000804003243371154862	0.07002
[249961, 232006, 150459, 200438], [222515, 130701, 278288, 201360]	319720249690111437887229255487847845310463475	0.08213
[140648, 296472, 130724, 309173], [240223, 223149, 218763, 194882]	322773560821008856417270275950599107061263625	0.04441
[65205, 189726, 233525, 170004], [137007, 87762, 274082, 159609]	6977523720740024241056075121611021139576919	0.01276
[251746, 282451, 184389, 194442], [146933, 239421, 267665, 259009]	861316343280649049593236132155039190682027614	0.08974
[138498, 166344, 187928, 186942], [228834, 138788, 189477, 122613]	63313191414342827754566531364533378588986467	0.08647
[20812723, 17301709, 21133745, 27679151], [28343568, 18410455, 19751834, 20421471]	665711555567792389878908993624629379187969880179721169068827951	0.99997
[15663004, 19519372, 14722354, 22325971], [17617837, 25267522, 20146447, 9198895]	63292704423941655080293971395348848807454253204720526472462015	0.99987
[13070380, 18156451, 13365203, 20567424], [12268303, 20733257, 17743591, 14414307]	43075357146173570492117291685601604830544643769252831337342557	0.99925

Testing for 4×4 transportation polytopes.

Another closely related set of instances are those coming from flows on networks, which are the lattice points of *flow polytopes*. Again, testing accuracy of the estimation we looked at a simple network on the complete 4-vertex and the complete 5-vertex tournaments (directed complete graphs). Consider the directed complete graph K_n . We assign a number to each node of the graph. Then, we orient the arcs from the node of smaller index to the node of bigger index. Let N be the node set of the complete graph K_n , let w_i be a weight assigned to node i for $i = 1, 2, \dots, l$, and let A be the arc set of K_l . Then, we have the following constraints, with as many variables as arcs:

$$\sum_{(j,i) \text{ arc enters } i} x_{ji} - \sum_{(i,j) \text{ arc has tail } i} x_{ij} = w_i,$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in A.$$

These equalities and inequalities define a polytope and this polytope is a special case of network flow polytope. Some tests for the complete graphs K_4 and K_5 , with different weight vectors, are shown in Table ??.

Next we tested more complicated situations. We consider 3-way transportation polytopes where the constraints are given by 1-margins. In the first half of Table 5 one can see the behavior is like that of the 2-way transportation polytopes in Table 2.2.2. The second half has three 3-way

Margins	exact # of lattice points	Error on estimation
$[2, 3, 3, 3, 4],$ $[2, 3, 3, 3, 4]$	645	1.707307219
$[3, 3, 3, 3, 3],$ $[3, 3, 3, 3, 3]$	2040	1.505308160
$[3, 2, 3, 2, 2],$ $[2, 2, 2, 3, 3]$	1486	1.507041276
$[2, 2, 4, 1, 1],$ $[1, 2, 1, 3, 3]$	99	2.32612347
$[3, 2, 2, 3, 4],$ $[3, 3, 4, 1, 3]$	273	1.930376112
$[2, 4, 2, 2, 2],$ $[2, 4, 1, 3, 2]$	244	1.925455520
$[3, 3, 3, 2, 2],$ $[2, 2, 3, 4, 2]$	828	1.623584155
$[2, 3, 2, 1, 3],$ $[2, 3, 3, 2, 1]$	506	1.737362376
$[2, 2, 3, 2, 2],$ $[2, 2, 2, 2, 3]$	1656	1.505611878
$[1, 1, 2, 2, 3],$ $[2, 2, 1, 1, 3]$	281	1.930244570

TABLE 3. Future Experiments

Weights on nodes	Number of Lattice points	Estimation	Error
$[-6, -8, 5, 9];$	223;	203.7272028	0.08642509955
$[-9, -11, 12, 8];$	330	327.5118692	0.007539790303
$[-1000, -1, 1000, 1]$	3002	2853.211281	0.04956319753
$[-4383, 886, 2777, 720]$	785528058	$0.7862404052 \cdot 10^9$	0.0009068386454
$[-4907, -2218, 3812, 3313]$	20673947895	$0.1849005970 \cdot 10^{11}$	0.1056347927
$[-2569, -3820, 1108, 5281]$	14100406254	$0.1243766827 \cdot 10^{11}$	0.1179212818
$[-3842, -3945, 6744, 1043]$	1906669380	$0.2056244088 \cdot 10^{10}$	0.07844816179
$[-47896, -30744, 46242, 32398]$	19470466783680	$0.1778955624 \cdot 10^{14}$	0.08633129132
$[-54915, -97874, 64165, 88624]$	106036300535520	$0.8585026143 \cdot 10^{14}$	0.1903691374
$[-69295, -62008, 28678, 102625]$	179777378508547	$0.1349184241 \cdot 10^{15}$	0.2495250224
$[-3125352, -6257694, 926385, 8456661]$	34441480172695101274	$0.5258709961 \cdot 10^{18}$	0.9847314633

TABLE 4. Gaussian estimation for four node network problem

transportation polytopes given by sets of 2-margins. Due to the difference of error the examples indicate that the method of estimation seems to have its best performance when the number of variables is much larger than the number of constraints. This is particularly evident in the last two examples of Table 5. The first considers a 2-margin $2 \times 3 \times 3$ transportation polytope taken from [29]. Second, the $3 \times 3 \times 3$ analogue of the Birkhoff polytope has clear inaccuracy. Again, as we

Margins	# of lattice points	Error
[38,26,87], [69,11,71], [77,74]	2626216761994	0.02956
[381,264,871], [690,123,703], [672,844]	969328784998192447450409	0.02201
[179909, 258827, 224919], [190019, 276208, 197428], [331827, 331828]	10996570128188103700571192439719329377965299537177538734365	0.10601
[31,22,87], [50,13,77], [42,87,11]	8846838772161591	0.00185
[531,992,787], [750,913,577], [742,587,911]	75262943725025193225827940796161419321644384	0.11337
[11531,9922,13787], [13750,9913,11577], [9742,13587,11911]	131537460708108801553237287957587623890623217109985417250150842074021	0.12606
[1153100, 992200, 1378700], [1375000, 991300, 1157700], [974200, 1358700, 1191100]	1304703422295241015594871677227500622969143545937924621879639043865503921583490166934483708040820516427400011	0.12628
$\begin{bmatrix} 10002 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix},$ $\begin{bmatrix} 1000003 & 3 & 3 \\ 3 & 3 & 3 \end{bmatrix},$ $\begin{bmatrix} 1000003 & 3 \\ 3 & 3 \\ 3 & 3 \end{bmatrix}$	33	0.10437
$\begin{bmatrix} 282 & 199 & 212 \\ 21 & 14 & 9 \\ 1 & 2 & 2 \end{bmatrix},$ $\begin{bmatrix} 107 & 80 & 169 \\ 197 & 135 & 54 \end{bmatrix},$ $\begin{bmatrix} 329 & 364 \\ 23 & 21 \\ 4 & 1 \end{bmatrix}$	441	1.10058
(3 copies of) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	12	12.3843

TABLE 5. Testing for 3-way transportation polytopes. The top half deals with 1-margins and the second half with 2-margins.

observed before, when the computation of involves large entries the convex optimization problem becomes much harder to solve.

In the paper [16] the authors introduced a class of small but rather difficult problems for the purpose of detecting feasibility. Roughly speaking these are problems of the form $Ax = b, x \in \{0, 1\}$

where A is $m \times 10(m-1)$ matrix and the entries of A are integers between 0 and 99. The i -th entry of b is the sum of the corresponding row of A , divided by 2 and rounded to next integer value. We investigated the performance of estimation in these kind of problems. We generated all instances using the generator available at <http://did.mat.uni-bayreuth.de/~alfred/marketsplit.html>. We only investigated feasible instances with $m = 2, m = 3$ and $m = 4$. A very high percentage of instances produced are in fact infeasible.

Unfortunately, when we tried the sampling technique in the six feasible instances we found most of the the time sampling never found a single feasible solution. The second test was made using the Gaussian estimation and it is presented in Table 2.2.2. We list the predicted number of solutions and the true number of solutions, if any. The top part of the table consist of infeasible problems and there the prediction is consistent with that fact.

Constraints (of the form $L_i x = m_i$)	predicted v.s.exact solutions
$L_1 := [73, 93, 88, 63, 65, 27, 25, 6, 31, 70]; m_1 := 270;$ $L_2 := [17, 27, 32, 35, 36, 63, 64, 37, 91, 93]; m_2 := 247$	0.0271/0
$L_1 := [30, 29, 94, 77, 46, 10, 63, 40, 38, 37]; m_1 := 232;$ $L_2 := [16, 92, 10, 40, 29, 16, 51, 11, 31, 8]; m_2 := 152;$	0.0435/0
$L_1 := [10, 2, 88, 39, 33, 42, 48, 67, 25, 74, 86, 34, 76, 41, 20, 32, 42, 78, 26, 63]; m_1 := 463;$ $L_2 := [15, 94, 80, 1, 21, 99, 54, 48, 10, 46, 71, 20, 0, 60, 11, 33, 2, 59, 52, 79]; m_2 := 427;$ $L_3 := [85, 39, 14, 61, 80, 86, 46, 23, 16, 24, 86, 31, 19, 18, 85, 40, 69, 39, 40, 79]; m_3 := 490;$	0.0954/0
$L_1 := [66, 43, 98, 18, 69, 4, 40, 3, 72, 41, 20, 52, 42, 34, 60, 70, 27, 43, 64, 88]; m_1 := 477;$ $L_2 := [98, 17, 44, 94, 68, 84, 77, 70, 43, 77, 3, 61, 72, 53, 79, 41, 9, 71, 97, 81]; m_2 := 619;$ $L_3 := [12, 69, 33, 6, 55, 45, 29, 82, 88, 93, 70, 39, 62, 67, 85, 31, 51, 14, 1, 46]; m_3 := 489;$	0.0784/0
$L_1 := [29, 63, 85, 80, 64, 87, 22, 31, 5, 23, 96, 8, 14, 93, 23, 74, 78, 72, 0, 30]; m_1 := 488;$ $L_2 := [15, 44, 21, 83, 13, 49, 40, 13, 33, 46, 42, 62, 10, 80, 94, 26, 19, 68, 10, 24]; m_2 := 396;$ $L_3 := [43, 6, 84, 58, 51, 7, 84, 29, 79, 36, 11, 47, 33, 32, 30, 46, 33, 23, 11, 67]; m_3 := 405;$	0.1049/2221
$L_1 := [39, 65, 40, 15, 2, 35, 74, 87, 33, 46, 51, 21, 86, 80, 70, 60, 31, 30, 1, 8]; m_1 := 437;$ $L_2 := [12, 23, 87, 46, 49, 16, 14, 88, 91, 39, 8, 31, 4, 0, 46, 58, 36, 73, 45, 21]; m_2 := 393;$ $L_3 := [19, 49, 94, 58, 29, 17, 70, 61, 47, 71, 21, 59, 46, 8, 58, 95, 76, 72, 36, 68]; m_3 := 527;$	0.1363/748
$L_1 := [69, 0, 89, 59, 32, 73, 56, 26, 49, 42]; m_1 := 247;$ $L_2 := [42, 69, 59, 0, 57, 31, 84, 95, 61, 77]; m_2 := 287;$	0.02752/10
$L_1 := [46, 12, 3, 91, 76, 7, 39, 27, 95, 55]; m_1 := 225;$ $L_2 := [74, 85, 49, 39, 80, 22, 16, 25, 84, 11]; m_2 := 242;$	0.0334/8

Testing gaussian estimation on feasible market-split problems. The problems on the top are infeasible and those in the block below have only one integer solution. Thus, with one obvious exception, the estimated number of solutions is of comparable magnitude

Finally we considered ten instances of binary knapsack problems in Table 6. The estimation we present there was done using the Gaussian heuristic. This time the error is again smaller.

Constraints (of the form $Lx = m$)	# 0-1 solutions	Error
$L := [116, 192, 120, 401, 129]; m := 236;$	1	0.75480
$L := [11, 17, 33, 13, 3, 5, 6, 7, 4]; m := 33$	9	0.11174
$L := [16, 92, 10, 40, 29, 16, 51, 11, 31, 8]; m := 152;$	4	0.65327
$L := [46, 12, 3, 91, 76, 7, 39, 27, 95, 55]; m := 225;$	7	0.33358
$L := [9, 17, 16, 8, 6, 15, 14, 7, 8, 11]; m := 30;$	11	0.13587
$L := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]; m := 12;$	13	0.07061
$L := [11, 4, 13, 7, 3, 5, 10, 5, 6, 11]; m := 33;$	29	0.84788
$L := [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]; m := 19;$	6	0.04771
$L := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]; m := 15$	27	0.02667
$L := [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39]; m := 39$	41	0.05350
$L := [1, 3, 5, 7, 9, 11, \dots, 2i - 1, \dots, 47, 49]; m := 49$	93	0.03782
$L := [1, 2, 3, 4, \dots, i, \dots, 20]; m := 20;$	64	0.04844
$L := [1, 2, 3, 4, \dots, 30]; m := 30;$	296	0.04371
$L := [1, 2, 3, 4, \dots, 40]; m := 40;$	1113	0.04075
$L := [1, 2, 3, \dots, 50]; m := 50;$	3658	0.03619

TABLE 6. Gaussian estimation for binary knapsack problems

Constraints (of the form $Lx = m$)	solutions	Estimation
$L := [73, 103, 63, 109, 39, 100, 69, 34, 74, 101, 10, 98, 45, 22, 57, 14, 51, 85, 58, 118]; m := 89673;$	1468949513434886540505656822902989085261376;	0.1475084011 10 ⁴
$L := [122, 76, 47, 51, 121, 71, 37, 7, 85, 55, 6, 55, 25, 117, 39, 80, 19, 30, 84, 65]; m := 67973;$	291998636368571029765997170002536213655789	0.2932182580 10 ⁴
$L := [73, 46, 33, 45, 90, 79, 122, 44, 71, 88, 56, 68, 87, 33, 85, 75, 65, 19, 76, 89]; m := 62801;$	210682728869292564550256389445683491931	0.2115628234 10 ⁴
$L := [16, 15, 4, 90, 72, 8, 98, 102, 20, 98, 78, 20, 97, 26, 98, 78, 89, 77, 110, 69]; m := 68651;$	505129574819324212119149212297499287329320	0.5072394922 10 ⁴
$L := [78, 87, 69, 41, 38, 8, 75, 100, 49, 70, 22, 50, 50, 90, 63, 7, 88, 76, 23, 82]; m := 87820;$	19114162680134853637255438963128879326507291	0.1919398216 10 ⁴
$L := [48, 14, 51, 119, 67, 53, 116, 117, 112, 90, 44, 5, 22, 106, 35, 76, 78, 12, 115, 32]; m := 81461;$	2088958399570307731772462328567117058898353	0.2097682995 10 ⁴
$L := [68, 26, 93, 69, 107, 28, 43, 58, 13, 7, 48, 8, 64, 8, 60, 32, 105, 93, 46, 88]; m := 72058;$	16231732893860432484451040978238319647115898	0.1629952302 10 ⁴
$L := [56, 33, 99, 88, 116, 49, 70, 75, 66, 120, 104, 50, 20, 28, 93, 43, 72, 67, 55, 9]; m := 83871;;$	318613009580650552853932195841594408740892	0.3199437341 10 ⁴
$L := [100, 17, 6, 22, 92, 39, 122, 81, 107, 17, 100, 116, 35, 112, 94, 10, 40, 41, 6, 59]; m := 91895;$	520615049050701460701616736156390152604118125	0.5227889229 10 ⁴
$L := [52, 115, 56, 97, 24, 116, 4, 18, 26, 29, 38, 52, 9, 15, 51, 87, 12, 7, 96, 51]; m := 62626;$	54631823154896764877169682762360073091979	0.5486000068 10 ⁴
$L := [106, 45, 66, 55, 51, 120, 72, 60, 19, 39]; m := 40277;$	2431245141720187417	0.2451589736 10 ⁴
$L := [121, 51, 11, 42, 52, 48, 37, 99, 30, 70]; m := 39352;$	12144168080075516583	67.1007832
$L := [109, 66, 119, 32, 69, 101, 94, 52, 65, 89]; m := 50938;$	1263944963381215356	0.1274521533 10 ⁴
$L := [85, 106, 112, 106, 93, 76, 46, 119, 48, 94]; m := 46912;;$	176693490567784734	0.1781721072 10 ⁴
$L := [8, 68, 45, 33, 99, 83, 6, 7, 97, 30]; m := 32401;$	142772636388597989047	0.1439673055 10 ⁴
$L := [85, 25, 61, 96, 70, 1, 8, 69, 101, 23]; m := 42140;$	1094785736235471942376;	0.1103939496 10 ⁴
$L := [44, 5, 22, 106, 35, 76, 78, 12, 115, 32]; m := 48503;$	914264831725680577913	0.9219139398 10 ⁴
$L := [101, 114, 69, 47, 29, 61, 5, 117, 101, 87]; m := 47564;$	10832482612266038937	1092312509 10 ⁴
$L := [51, 2, 115, 88, 32, 48, 64, 55, 97, 109]; m := 42535;$	22816866155069770379	0.2300779969 10 ⁴
$L := [114, 84, 99, 73, 112, 108, 7, 50, 48, 108]; m := 63843;$	33823580063347460823	0.3410656787 10 ⁴
$L := [105, 106, 91, 77, 103]; m := 16803;$	437695	445051.278
$L := [103, 6, 74, 75, 74]; m := 18079;$	18188725	0.1849544905 10 ⁴
$L := [53, 116, 117, 112, 90]; m := 29493;;$	4493397	456893609
$L := [33, 31, 13, 22, 29]; m := 8835;$	30796702	313137098
$L := [120, 2, 114, 99, 96]; m := 20825;$	31107229	319300938
$L := [39, 60, 96, 33, 2]; m := 13496;$	96451455	980689260
$L := [25, 102, 90, 42, 94]; m := 19693;$	7150015	7287718.73
$L := [18, 100, 98, 82, 14]; m := 6744;$	932486	474101.298
$L := [97, 58, 5, 58, 21]; m := 11748;$	24120328	245246013
$L := [80, 57, 107, 89, 47]; m := 22755;$	5658373	575336847

TABLE 7. Gaussian estimation for Random knapsack problems

Analysis of the Recent Counterexample of the Hirsch Conjecture

3.1. Hirsch Conjecture

The original motivation for the Hirsch conjecture was the Simplex Algorithm for linear programming. The most commonly used method for linear programming is the *Simplex Method* which was devised by Dantzig in 1947. Geometrically speaking, the Simplex Method first finds an arbitrary vertex in the feasible polyhedron and then uses local rules to move from one vertex to the adjacent vertex in such a way that the $\xi(x)$ of the given linear functional ξ increases at every step. When there is no such *pivot step* that can increase the functional, the convexity implies that we have achieved the maximum possible value of it. The complexity of the Simplex Method depends on local pivot rules. The Simplex Method is discussed further in the introduction. In 1957 the Hirsch Conjecture was proposed in a letter to Dantzig from Hirsch. The main purpose of the Hirsch Conjecture was to add robustness to the Simplex Method by providing a lower bound to the number of steps needed in the Simplex Method. However, it has become one of the most fundamental open question in combinatorial geometry. It states that the graph of a d -dimensional polytope with n -facets cannot have a diameter greater than $n - d$. Meaning we can go from any vertex of the polytope to any other vertex using at most $n - d$ edges.

CONJECTURE 2. (*Hirsch Conjecture*) Let $n > d \geq 2$. Let P be a d -dimensional polytope with n facets, and let $G(P)$ be its graph. then $\text{diam}(G(P)) \leq n - d$.

The dimension of a polytope is the dimension of its affine hull, $\text{aff}(P)$. A d -dimensional polytope is called a d -polytope. *Facets* are the faces of dimension $d - 1$ of P , the number n of them is the minimum number of half-spaces needed to produce P as their intersection. The number $\text{diam}(G(P)) \in \mathbb{N}$ is the *diameter* of the graph of P . In other words, we can travel from any vertex of P to any other vertex of P by traveling on at most $n - d$ edges. [51] For the polytope P , we denote by $G(P)$ its graph or 1-skeleton, consisting of the vertices and edges of P . The distance between two vertices in a graph is the minimum number of edges needed to go from one to the other, and the diameter of a graph is the maximum distance between its vertices.

The Hirsch Conjecture would provide a lower bound to the number of steps needed for the simplex method. The Simplex Method is bounded under any pivot rule by the diameter of the polyhedron P .

A d -polytope is called *simple* if every vertex is the intersection of exactly d -facets. Equivalently, a d -polytope is simple if every vertex in the graph $G(P)$ has degree exactly d . Any polytope P given by its facet description, can be perturbed to a simple one P' by a generic and small change in the coefficients of its defining inequalities. This will make non-simple vertices “explode” and

become clusters of new vertices, all of which will be simple. This process can not decrease the diameter of the graph, since we can recover the graph of P from that of P' by collapsing certain edges. Thus, as we analyze the Hirsch Conjecture we only need to consider the simple polytopes. [51]

PROPOSITION 1. [51] *The diameter of P' is an upper bound on the diameter of P .*

LEMMA 7. [51] *The diameter of any polytope P is bounded above by the diameter of some simple polytope P' with the same dimension and number of facets.*

The graphs of the simple polytopes are better behaved than graphs of other polytopes. The main reason is because if u and v are vertices joined by an edge in a simple polytope then there is a single facet containing u and not v , and a single facet containing v and not u . Meaning that, as we travel on the edge of the polytope we leave a facet as we enter another. Every polytope P has a polar polytope P' whose vertices correspond to the facets of P and vice-versa. More generally, every $(d-i)$ -face of P' corresponds to a face of P of dimension $(i-1)$, and the incidence relations are reversed.

The polars of simple polytopes are called simplicial, and their defining property is that every facet is a $(d-1)$ -simplex. As an example, the d -dimensional cross polytope is the polar of the d -cube. Since cubes are simple polytopes, cross polytopes are simplicial. The polar of a simplex is a simplex, and simplices are the only polytopes of dimension greater than two which are at the same time simple and simplicial. Since all faces of a simplex are themselves simplices, all faces of a simplicial polytope are simplices. From this viewpoint, one can forget the geometry of P^* and look only at the combinatorics of the simplicial complex formed by its faces, the boundary of P^* . Topologically, this simplicial complex is a sphere of dimension $d-1$. For simplicial polytopes, we can state the Hirsch conjecture as asking how many ridges do we need to cross in order to walk between two arbitrary facets, if we are only allowed to move from one facet to another via a ridge. This suggests defining the dual graph $G^\Delta(P)$ of a polytope: The undirected graph has as nodes the facets of P and two nodes are connected by an edge if and only if their corresponding facets intersect in a ridge of P . In summary, $G^\Delta(P) = G(P^*)$.

DEFINITION 8. [52] Let $H(n, d)$ denote the maximum diameter of the graphs of the d -polytopes with n facets. Let $H_u(n, d)$ denote the maximum diameter of the graphs of d -polyhedra with n facets.

DEFINITION 9. [52] A d -dimensional polytope is Hirsch-sharp if it meets the Hirsch Conjecture with equality; that is if its diameter is exactly $n-d$.

Here are a couple of ways to construct Hirsch-sharp polytopes.

- (1) A Hirsch-sharp polytope exist if one of the following three conditions holds:
 - (a) $n \leq 2d$
 - (b) $n \leq 3d-3$
 - (c) $d \geq 7$
- (2) A Hirsch-sharp polytope does not exist for $d \leq 3$ if $n \geq 2d$ or for $(n, d) \in \{(10, 4), (11, 4), (12, 4)\}$.
- (3) They are unknown, but may exist in all other cases: that is, if $d \in \{4, 5, 6\}$ and $n > 3d-3$, except for the three pairs (n, d) mentioned above.

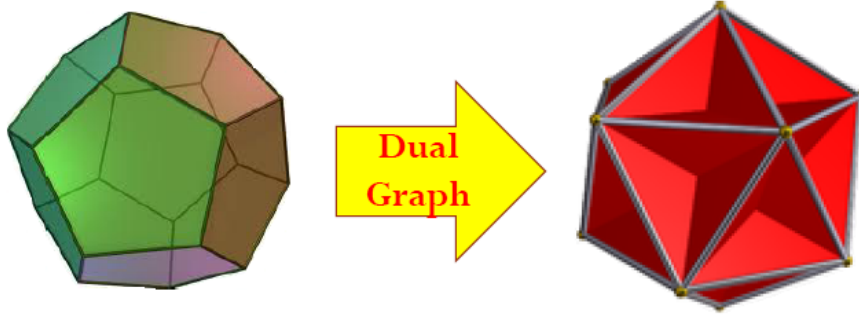


FIGURE 1. Example of the dodecahedron and its dual graph

Hirsch-sharp polytopes with $n \leq 2d$ are simple to find. For this reason they are considered “trivial” Hirsch- sharp polytopes. To some extent we only know of one non-trivial Hirsch- sharp polytope, the four dimensional polytope found by Klee and Walkup [?]. The 4-dimensional polytope by Klee and Walkup is called Q_4 with nine facets whose diameter is *five*. This polytope is at the basis of the construction of every other Hirsch-sharp polytope. It also plays a key part in the construction of disproving the unbounded and monotone version of the Hirsch-conjecture.

3.2. Santos' Counterexample

After 53-years, the open conjecture comes to a close. Francisco Santos, a Mathematician from the Universidad de Cantabria in Spain, found a counterexample in the 43rd dimension with 86 facets and at least 44 diameter.

LEMMA 8. (Lemma 1.1. in [51]) Let $H(n; d)$ denote the maximum possible diameter of the graph of a d -polytope with n -facets. Then: $\forall k \in \mathbb{N}; \max_{d \in \mathbb{N}} \{H(d+k, d)\} = H(2k, k)$

As a corollary, they got the following d-step Theorem:

THEOREM 15. [39]. The following statements are equivalent for a positive integer m :

- (1) Every d -polytope with $d+m$ facets has diameter at most m (The Hirsch Conjecture).
- (2) Every m -polytope with $2m$ facets has diameter at most m . (The d -step Conjecture)

There are two well-known reductions one usually makes when studying the Hirsch conjecture. One is the d -step theorem, the maximum diameter of a d -polytope with $d+m$ facets, for fixed m and varying d is achieved precisely when $m = d$. The other is that, for every m and d , the maximum is always achieved at a simple polytope. That is, a d -polytope in which every vertex belongs to exactly d facets. Simple polytopes are especially relevant for the Simplex Method, they are obtained when the system of inequalities is sufficiently generic. The first ingredient in our construction is the observation that if we start with a polytope which is at the same time non-simple and has $m > d$, then we can get a simple polytope with $m = d$ and not only maintain its diameter, but actually

increase it. Although the theorem can be stated more generally, the version we need has to do with the following class of polytopes:

DEFINITION 10. [51] A d -spindle is a d -polytope P having two distinguished vertices u and v such that every facet of P contains exactly one of them. The length of a spindle is the graph distance between u and v .

THEOREM 16. [39] (*Generalized d -step Theorem*). If P is a spindle of dimension d , with n facets and length l , then there is another spindle P' of dimension $n - d$, with $2n - 2d$ facets and with length at least $l + n - 2d$. In particular, if $l > d$ then P' violates the d -step conjecture, and also the Hirsch conjecture.

It is important to stress that in this statement the number of facets n of P is arbitrary, but still the maximum number of steps from u to v allowed in order for the Hirsch conjecture to survive is d . This gives much more freedom and eventually led to the construction of a counterexample. The second ingredient in the proof is the explicit construction of a spindle of dimension five and length six.

THEOREM 17. [52] *Theorem 1.5. There is a 5-dimensional spindle (with 48 facets and 322 vertices) of length six.*

COROLLARY 5. Corollary 1.6. There is a polytope of dimension 43 with 86 facets and diameter at least 44.

DEFINITION 11. [52] A prismatoid is a polytope Q with two (parallel) facets Q_+ and Q_- containing all vertices.

DEFINITION 12. [52] The width of a prismatoid is the dual-graph distance from Q_+ to Q_- .

THEOREM 18. [52] *Let Q be a prismatoid of dimension d , with $n > 2d$ vertices and width δ . Then there is another prismatoid Q_- of dimension $d + 1$, with $n + 1$ vertices and width $\delta + 1$.*

COROLLARY 6. [52] In particular, if a prismatoid Q has width $> d$ then there is another prismatoid Q_- (of dimension $n - d$, with $2n - 2d$ facets, and width $\geq \delta + n - 2d > n - d$) that violates (the dual of) the Hirsch conjecture.

So, to disprove the Hirsch Conjecture we need to find a prismatoid of dimension d and width larger than d . Its number of vertices and facets is irrelevant!

3.3. The Search for a Computationally Smaller Counterexample

Although Santos found a counterexample to the Hirsch conjecture, his counterexample is computationally unfeasible. For this reason, we began a search for a computationally smaller counterexample. We attempted to improve on his results by three distinct methods. Our original approach was similar to brute force. We used a software program named POLYMAKE to create different polytopes to then Cayley-embed them to create prismatoids.

3.3.1. Constructing Prismatoids. POLYMAKE is a software program designed for the algorithmic treatments of polytopes and polyhedra. One can download the software at the this address: <http://polymake.org/doku.php/download/start>

DEFINITION 13. Cayley-embedding is a method of construction in which 2 $n - \text{dimensional}$ polytopes, P and Q containing the origin in their interior, are embedded in $(n + 1)$ -space parallel and unit distant apart from one another and then the resulting polytope is simply the convex hull of these 2 embedded polytopes

In other words, we are joining two polytopes to create a prismatoid. Figure 2 is a simple example of joining two pentagons (1-dimensional polytope) to create a two- dimensional prismatoid.

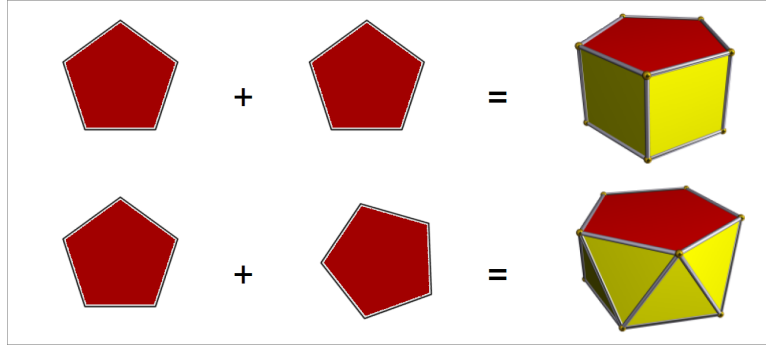


FIGURE 2. Example of a pentagon Cayley embedded with another pentagon.

The construction of these prismatoids was important in finding a counterexample to the Hirsch conjecture. It was not just about embedding any two polytopes; but it was a group effort to investigate what would be the best way to create different types of polytopes to embed together. The following are a few of the distinct selection of polytopes we used to generate the prismatoid.

- (1) d -cube- The vertices of the d -cube, the product of the d segments, are the 2^d points with ± 1 coordinates. Its facets are given by the $2d$ inequalities $-1 \leq x_i \leq 1$. Its graph has diameter d : the number of the steps needed to go from a vertex to another equals the number of coordinates in which the two vertices differ.
- (2) Random sphere- Produce a d -dimensional polytope with n random vertices on the unit sphere. Floating-point coordinates are used. Uniform distribution. The default precision is set to 6 digits.
- (3) d -Simplex- The convex hull of $d + 1$ points in $\mathbb{R}^d$ that do not lie on a common hyperplane is a d -dimensional simplex. It has $d + 1$ facets. Its graph is complete, so its diameter is 1.
- (4) Cross polytope - This is the convex hull of the d standard basis vectors and their negatives, which generalizes the 3-dimensional octahedron. It has 2^d facets, one in each orthant of $\mathbb{R}^d$. Its graph is almost complete; the only edges missing from it are those between opposite vertices.

Besides the basic constructions that were used through POLYMAKE, we also constructed many of the polytopes by using trigonometry and some implementation in MATLAB. This allowed us to embed polytopes with other polytopes which have been shifted by a trigonometric identity. Once we created sufficiently different polytopes, we begin to Cayley-embed them. Once we Cayley-embedded, we now have prismatoids. We used POLYMAKE to construct the Dual graphs of the prismatoid. As stated before, the Dual graph would allow us to work with only attempting to find

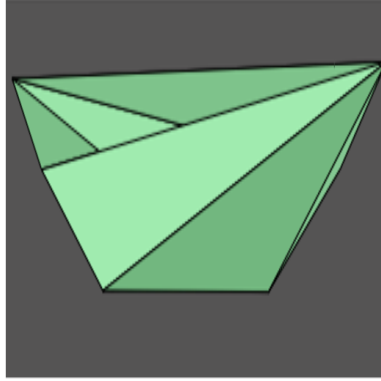


FIGURE 3. 3-dimensional polytope created with POLYMAKE using random generator with 7 points in dimension 3

the distance from the bottom facet to the top facet. With the information about the dual graph we would then use MAPLE to compute the distance between these two facets, in hopes of obtaining a width larger than the dimension of the polytope.

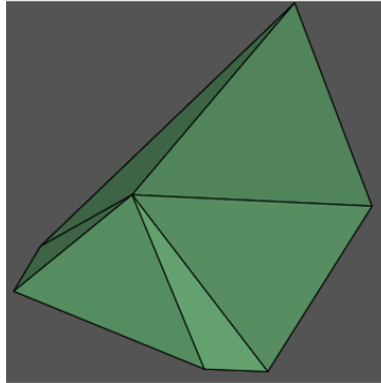


FIGURE 4. Three dimensional polytope created with POLYMAKE by creating a matrix in MATLAB with seven points

We recorded the data that we obtained and analyzed the different polytopes that were equal to the dimension. Although we attempted in many dimensions to obtain a smaller counterexample, we were unable to do so. Fortunately, we were able to find many polytopes whose diameters were equal to their dimension.

In the lab of five students, we were able to obtain the results demonstrated on the chart, where every column represents the amount of vertices and the rows demonstrates the dimension. The cells that are filled in represent the prisms which have a diameter equal to the dimension in which it lives. The initials represent who found the polytope. Unfortunately, this led to the conclusion that if we find one specific manner in which to create a polytope that satisfies this property then this

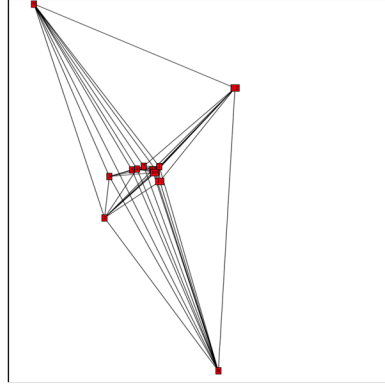


FIGURE 5. four dimensional prismatoid created by Cayley embedding the other two polytopes

# of vertices	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
dim 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
dim 4	-	-	-	-	X IR	X JG	X JG	X JG	X JG	X IR	X JG	X IR	X JG	X WX	X WX	X JG
dim 5	-	-	-	-	-	-	X JG	X IR	X IR	X IR	X WX	X WX	X JG	X WX	X JG	X JG
dim 6	-	-	-	-	-	-	-	-	X KL	-	X KL	X KL	X RC	X IR	X WX	X RC
dim 7	-	-	-	-	-	-	-	-	-	X KL	-	-	X WX	X IR	-	-
dim 8	-	-	-	-	-	-	-	-	-	-	-	-	X KL	X RC	X RC	X RC
dim 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
dim 10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

# of vertices	20	21	22	23	23	24	25	26	27	28	29	30	31	32	33	34
dim 4	X JG	X JG	X IR	X JG	X JG	X JG	X KL	X JG	X JG	X JG	-	-	-	-	-	-
dim 5	X JG	X JG	X IR	-	X WX	-	-	X WX	-	-	-	-	-	-	-	-
dim 6	X KL	X IR	X KL	X KL	-	X KL	X KL	X KL	X KL	-	-	-	-	-	-	-
dim 7	-	-	X IR	-	-	-	-	-	-	-	-	-	-	X IR	-	-
dim 8	X RC	X RC	X RC	X RC	-	X RC	X RC	X RC	X RC	X RC	X RC	X RC	X RC	X RC	X RC	X IR
dim 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
dim 10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

# of vertices	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
dim 4	-	-	-	-	X JG	-	-	-	-	-	-	-	-	-	-	-
dim 5	-	-	-	-	-	-	-	-	-	-	-	-	X IR	-	-	-
dim 6	-	-	-	X KL	-	X KL	-	X WX	-	-	-	-	X KL	-	X KL	X KL
dim 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
dim 8	X RC	X RC	X IR	X RC	X RC	X RC	X JG	X RC	X RC	X RC	X RC	X RC	X RC	X RC	X RC	X RC
dim 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
dim 10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

FIGURE 6. Results of the Polytope construction

same construction could easily be manipulated to satisfy the property in either another dimension, or with another set of vertices or both. Thus this type of construction was not very concrete, because there was much repetition and no improvement on obtaining the counterexample of the Hirsch Conjecture.

3.3.2. Geodesic Mapping. After noting that this may not be the best method of finding a counterexample, we switched gears and began to analyze Santos' counterexample through the geodesic maps. Before we continue further discussion, let us define a few key terms:

PROPOSITION 2. *If a graph G is three-connected, then the degree of the vertices is at least three.*

DEFINITION 14. A graph is three connected if for every two vertices v, w ($v \neq w$) there are three paths from v to w that are disjoint.

DEFINITION 15. A graph is planar if there is a way to draw it on a flat sheet.

THEOREM 19. (*Steinitz Theorem*) *A simple three-connected planar graph is always the graph of a three dimensional polytope.*

So we would like to find a 4-prismatoid of width > 4 . This is true if and only if the pair of (geodesic, polytopal) maps in S^2 , do not let you go from a blue vertex to a red vertex in two steps.

Replicating the following basic structure we can get a non-Hirsch periodic pair of maps in the plane:

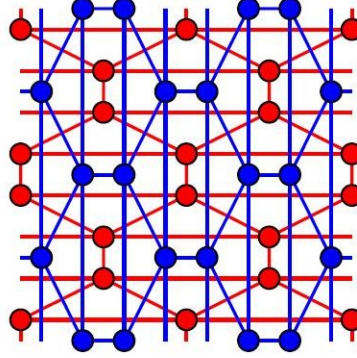


FIGURE 7. Santos' basic structure of the geodesic maps in S^2

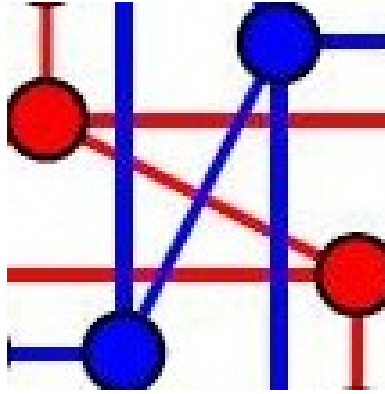


FIGURE 8. "Kiwi" extraction from Santos' basic structure

We attempted to replicate the "Kiwi" structure (Figure8) and modify it in order to see if we could find a pattern which would fail the d -step property. A way of drawing the geodesic maps on the sphere was to draw them on the two dimensional square, where everything on the edge is the north pole of the sphere. To demonstrate that the d -step fails in this case, we need to show that

every path between the red and blue vertices has at least two purple vertices. Separately, the blue and red graphs are three dimensional polytopes.

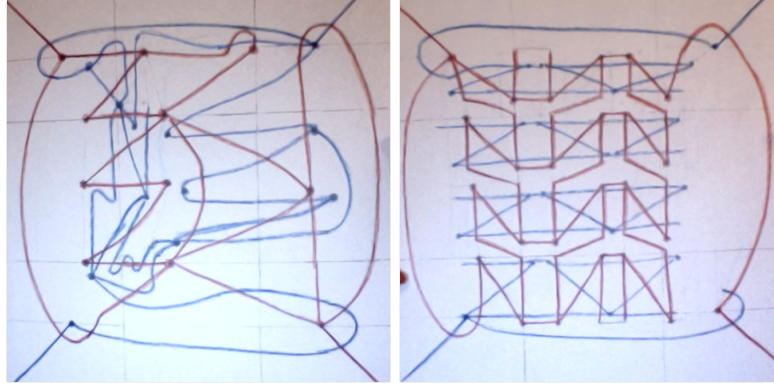


FIGURE 9. Attempt to create a "Kiwi" graph but this fails to be a polytope

In our example, we see that the figure on the left fails to be a polytope because it fails Steinitz Theorem. The figure on the right is an attempt to create a pattern of the "Kiwi". Unfortunately, we soon observe that the blue nodes can not be connected with another blue vertex by creating two purple vertices. After many attempts to create different patterns that would fail the $d - step$ property, we realized that no such counterexample could be possible in the fourth dimension, which was later proved by Santos.

Even though our progress was stunted in dimension 4 by an unexpected proof of the impossibility of a counter-example, given by Professor Santos, we were able to collect valuable data on the structure of Professor Santos' counterexample and on prisms in general. While we were unable to construct a more elementary counterexample. Professor Santos was able to construct a less complex counter (still unrealizable) which gives hope towards achieving a manageable counter in the future.

Bibliography

1. Aardal, K., Lenstra, A.K., and Lenstra, H.W. Jr.: Hard equality constrained integer knapsacks. In: W.J. Cook and A.S. Schulz (eds.), *Integer Programming and Combinatorial Optimization: 9th International IPCO Conference*, Lecture Notes in Computer Science vol. 2337, Springer-Verlag, 350–366, 2002.
2. Balcioglu, A. and Wood R.: Enumerating Near-Min s-t Cuts. In: D.L. Woodruff (editor) *Network Interdiction and Stochastic Integer Programming*, vol. 22, series operations research/computer science interfaces, Kluwer, Boston 2002.
3. Baldoni-Silva, W., De Loera, J., and Vergne, M.: Counting integral flows on Networks, *Foundations of Computational Mathematics*, Vol. 4, 277–314, 2004.
4. Barvinok A.I, *Asymptotic estimates for the number of contingency tables, integer flows, and volumes of transportation polytopes*, *International Mathematics Research Notices* 2009 (2009), 348385.
5. Barvinok A. I. and Hartigan J. *Maximum entropy Gaussian approximation for the number of integer points and volumes of polytopes* manuscript 2009, available at <http://www.math.lsa.umich.edu/~barvinok/papers.html> and at the Arxiv <http://arxiv.org/abs/0903.5223>
6. Barvinok A. I. and Hartigan J. *An asymptotic formula for the number of non-negative integer matrices with prescribed row and column sums* manuscript 2009, available at <http://www.math.lsa.umich.edu/~barvinok/papers.html> and at the Arxiv <http://arxiv.org/abs/0910.2477>
7. Beck M. and Pixton D., *The Ehrhart polynomial of the Birkhoff polytope*,. *Discrete Comput. Geom.* 30, no. 4 (2003), 623–637.
8. Benson H., Shanno D.F. and Vanderbei R.J. *A Comparative Study of Large Scale Nonlinear Optimization Algorithms* In *High Performance Algorithms and Software for Nonlinear Optimization*, (G. Di Pillo and A. Murli, editors), 94–126, Kluwer Academic Publishers, 2003.
9. Ben-Tal A. and Nemirovski, A. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, MPS-SIAM series on Optimization, 2001.
10. J. Blitzstein and P. Diaconis A sequential importance sampling algorithm for generating random graphs with prescribed degrees, manuscript (2009).
11. Y.Chen, I. Dinwoodie, A. Dobra, and M. Huber, *Lattice Points, Contingency Tables, and Sampling*. *Contemporary Mathematics*, Vol. 374, 65–78. American Mathematical Society, (2005).
12. Chen, Y., Diaconis, P., Holmes, S., and Liu, J. S. *Sequential Monte Carlo methods for statistical analysis of tables*. *Journal of the American Statistical Association*, 100, (2005) 109–120.
13. Chen, Y., Dinwoodie, I. H., and Sullivan, S. *Sequential importance sampling for multiway tables*. *The Annals of Statistics*, 34, (2006) 523–545.
14. Cox, L.H. *Contingency tables of network type: Models, Markov basis and applications* *Statistica Sinica* 17, (2007), 1371–1393
15. Clauss, P. and Loechner, V.: Parametric Analysis of Polyhedral Iteration Spaces, *Journal of VLSI Signal Processing*, Vol. 19, 179–194, 1998.
16. Cornuejols G and Dawande, M *A class of hard small 0-1 programs*. In R. E. Bixby, E.A. Boyd, and R.Z. Rios-Mercado, editors, *Integer Programming and Combinatorial Optimization. 6th International IPCO Conference*, Houston, Texas, June 1998. Springer Lecture Notes in Computer Science 1412, Heidelberg, 1998.
17. Danna E., Fenelon M., Gu Z., and Wunderling R. *Generating multiple solutions for mixed integer programming problems* In M. Fischetti and D.P. Williamson, eds., *Integer Programming and Combinatorial Optimization*, 12th International IPCO conference, Ithaca New York, June 2007. Springer Lecture Notes in Computer Science 4513, 280–294.

18. De Loera, J. and Sturmfels, B. *Algebraic unimodular counting*, Math. Programming Ser. B., Vol. 96, No. 2, (2003), 183–203.
19. De Loera, J. A., Haws, D., Hemmecke, R., Huggins, P., Tauzer, J. and Yoshida, R.: A User's Guide for **LattE**, software package **LattE** and manual are available at <http://www.math.ucdavis.edu/~latte/> 2003.
20. De Loera, J. A. and Hemmecke, R. and Tauzer, J. and Yoshida, R.: Effective lattice point counting in rational convex polytopes, *J. Symbolic Comp.*, vol. 38, 1273–1302, 2004.
21. De Loera, J.A, Hemmecke R., Köppe M., and Weismantel R. Integer polynomial optimization in fixed dimension. *Mathematics of Operations Research*. (2006), vol 31, No.1, 147–153.
22. De Loera, J.A. *ACTUALLY DOING IT: POLYHEDRAL COMPUTATION AND ITS APPLICATIONS* Lecture Notes for Dept. of Mathematics, University of California-Davis. <http://www.math.ucdavis.edu/~deloera/TEACHING/MATH114/justdoit.pdf>
23. De Loera, J.A, Hemmecke R. and Köppe M. Pareto optima of multicriteria integer programs, *INFORMS journal of Computing*. Vol. 21, No. 1, (2009), 39–48.
24. De Loera, J. A. and Onn, S. *Markov bases of three-way tables are arbitrarily complicated* *J. Symbolic Computation*, 41:173–181, 2006.
25. Diaconis, P. and Gangolli, A.: Rectangular arrays with fixed margins, In: *Discrete probability and algorithms* (Minneapolis, MN, 1993), 15–41, IMA Vol. Math. Appl., 72, Springer, New York, 1995.
26. Diaconis, P. and Sturmfels, B *Algebraic Algorithms for Sampling from Conditional Distributions*. *Annals of Statistics*, 26(1): 363–397, 1998.
27. Dyer, M. *Approximate counting by dynamic programming* Proceedings of the 35th Annual ACM Symposium on the Theory of Computing (STOC 03), ACM Press, pp. 693–699, 2003.
28. Dyer, M., Mount, D., Kannan, R. and Mount, J.: Sampling contingency tables, *Random Structures and Algorithms*, Vol. 10, 487–506, 1997.
29. Fienberg, S.E. and Makov, U.E. and Meyer, M.M. and Steele, R.J.: Computing the exact conditional distribution for a multi-way contingency table conditional on its marginal totals. In: *Data Analysis From Statistical Foundations*, 145–166, Nova Science Publishers, A. K. Md. E. Saleh, Huntington, NY, 2001.
30. Ghosh, S., Martonosi, M., Malik, S.: Cache miss equations: a compiler framework for analyzing and tuning memory behavior, *ACM transactions on programming languages and systems*, 21, 703–746, 1999.
31. Grötschel, M., Lovász, L., and Schrijver, A. *Geometric Algorithms and Combinatorial Optimization*, second edition. Algorithms and Combinatorics, 2, Springer-Verlag, Berlin, 1993.
32. Karloof, Howard *Linear Programming*
33. Jarre, F. *Relating max-cut problems and binary linear feasibility problems* available at http://www.optimization-online.org/DB_FILE/2009/02/2237.pdf
34. Jerrum, M.R. *Counting, sampling and integrating: algorithms and complexity*, Birkhäuser, Basel, 2003.
35. Jerrum, M.R. and Sinclair, A.: The Markov Chain Monte Carlo Method: An approach to approximate Counting and integration, in: Dorit Hochbaum (Ed.), *Approximation Algorithms*, PWS Publishing Company, Boston, 1997.
36. Jerrum M.R., Valiant L.G., and Vazirani V.V.. *Random generation of combinatorial structures from a uniform distribution*, *Theoretical Computer Science*, 43:169188, 1986.
37. Jerrum M., Sinclair A., and Vigoda E., *A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries*. *Journal of the ACM*, 51(4):671–697, 2004.
38. Kannan, R. and Vempala, S. *Sampling lattice points* in proceedings of the twenty-ninth annual ACM symposium on Theory of computing El Paso, Texas USA, 1997, 696–700 .
39. V. Klee and D. W. Walkup. The d-step conjecture for polyhedra of dimension $d \leq 6$. *Acta Math.*, 133:5378, 1967.
40. McKay B.D. and Wormald N.C., *Asymptotic enumeration by degree sequence of graphs of high degree*, *European J. Combin.*, 11, 565–580, 1990.
41. Micciancio, D. and Goldwasser, S. *Complexity of lattice problems. A cryptographic perspective*, Kluwer International Series in Engineering and Computer Science, 671. Kluwer, Boston, MA, 2002.
42. Loebl M., Zdeborova L., *The 3D Dimer and Ising Problems Revisited*, *European J. Combinatorics* No. 29 Vol.3, 2008.
43. Loechner, V., Meister, B., and Clauss, P.: Precise data locality optimization of nested loops, *J. Supercomput.*, Vol. 21, 37–76, 2002.

44. Mahadev N.V.R. and Peled U. *Threshold graphs and related topics*. Annals of Discrete Mathematics, Vol 56., North-Holland Publishing Co. Amsterdam.
45. Mangasarian O.L. *Knapsack Feasibility as an Absolute Value Equation Solvable by Successive Linear Programming*, Optimization Letters 3(2), 161–170, 2009.
46. Mangasarian O.L. and Recht, B. *Probability of unique Integer solution to a system of linear equations* Data Mining Institute, Univ. of Wisconsin, Technical Report 09-01, September 2009.
47. Mount, J. *Fast unimodular counting*, Combinatorics, Probability, and Computing 9 (2000) 277–285.
48. Morris, B. and Sinclair A. *Random walks on truncated cubes and sampling 0-1 knapsack solutions*. SIAM journal on computing, 34 (2004), pp. 195-226
49. Renegar, J. *A mathematical view of interior-point methods in convex optimization* MPS-SIAM series in Optimization, 2001
50. Sankaranarayanan S, Ivancic F, and Gupta A. *Program analysis using symbolic ranges*, in Proceedings of the Static Analysis Symposium 2007, vol. 4634 of Lecture Notes in Computer Science, Springer, 366–383.
51. Santos, F. *A counter-example to the Hirsch conjecture* in Bernoulli Conference on Discrete and Computational Geometry Lausanne, September 2, 2010 <http://personales.unican.es/santosf/Hirsch>
52. Santos, F. *Width of low-dimensional prisms* in Conference on Geometric Graph Theory Lausanne, September 28, 2010. <http://personales.unican.es/santosf/Hirsch>
53. Soprunov, I and Soprunova, J. *Toric surface codes and Minkowski length of polygons*, SIAM J. Discrete Math. 23 (2008/09), no. 1, 384–400.
54. Vanderbei, R. J. LOQO user’s manual version 3.10 Optimization Methods and Software, Vol. 11, Issue 1 - 4 (1999) , 485–514.
55. Vanderbei R.J and Shanno D.F. *An Interior-Point Algorithm for Nonconvex Nonlinear Programming*. Computational Optimization and Applications, 13: (1999) 231–252, 1999.
56. Welsh, D.J.A.: Approximate counting in Surveys in Combinatorics, (edited by R. A. Bailey), Cambridge Univ. Press, Cambridge, 1997.