# On the Circuit Diameters of Polyhedra

by

**Timothy James Yusun**

M.Sc., Simon Fraser University, 2011
B.Sc., Ateneo de Manila University, 2008

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
Department of Mathematics
Faculty of Science


**SIMON FRASER UNIVERSITY**
**Fall 2017**

# Approval

**Name:** **Timothy James Yusun**

**Degree:** **Doctor of Philosophy (Mathematics)**

**Title:** ***On the Circuit Diameters of Polyhedra***

**Examining Committee:** **Chair:** Abraham Punnen
Professor

**Tamon Stephen**
Senior Supervisor
Associate Professor

**Zhaosong Lu**
Supervisor
Professor

**Marni Mishna**
Internal Examiner
Associate Professor

**David Bremner**
External Examiner
Professor
Faculty of Computer Science
University of New Brunswick

**Date Defended:** December 4, 2017

# Abstract

In this thesis we develop a framework to study the circuit diameters of polyhedra. The circuit diameter is a generalization of the combinatorial (edge) diameter, where walks are permitted to enter the interior of the polyhedron as long as steps are parallel to its circuit directions. Because the circuit diameter is dependent on the specific realization of the polyhedron, many of the techniques used in the edge case do not transfer easily. We reformulate circuit analogues of the Hirsch conjecture, the $d$-step conjecture, and the non-revisiting conjecture, and recover some of the edge case relationships in the circuit case. To do this we adapt the notion of simplicity to work with circuit diameter, and so we define $\mathcal{C}$-simplicity and wedge-simplicity.

Then, we prove the circuit 4-step conjecture, including for unbounded polyhedra, by showing that the original counterexample $U_4$ to the combinatorial analogue satisfies the Hirsch bound in the circuit case, independent of its realization. This was the first known counterexample to Hirsch, and several families of counterexamples are constructed from $U_4$. In particular, the unbounded Hirsch conjecture could still hold in the circuit case.

We also use computational methods to study $Q_4$, the bounded counterpart to $U_4$, and give two realizations with different circuit diameters. It remains open whether $Q_4$ is circuit Hirsch-sharp; however, we are able to lower the distance bound for at least one direction between the two far vertices of $Q_4$. Finally, we present some auxiliary results involving representations of polyhedra and circuit calculations.

**Keywords:** discrete geometry; polytope diameters; circuit diameter; linear optimization

# Acknowledgements

This thesis is made possible through the support of a Postgraduate Scholarship (D) from the Natural Sciences and Engineering Research Council of Canada (NSERC). Thanks also to the Department of Mathematics at Simon Fraser University for their support during my postgraduate studies.

In the process of researching, writing, defending, and editing this thesis there were several individuals who have in their own ways contributed to the final product.

I am grateful for my supervisor Dr. Tamon Stephen for his unwavering support and constant encouragement. All our weekly meetings (both in person and online) helped push this study to its current state, and our eight years together from when I first started my MSc program have shaped me into who I am today.

Much of this work was a collaborative effort with Dr. Steffen Borgwardt at the University of Colorado Denver, and I would like to express my sincere appreciation for all our fruitful discussions that brought these results to life. To my committee members who read and critiqued my thesis – Dr. Zhaosong Lu, Dr. David Bremner, and Dr. Marni Mishna – thank you for taking time to examine my work and providing constructive feedback. I have also had the pleasure of working with the other faculty members at Simon Fraser University Surrey – Dr. Natalia Kouzniak, Dr. Abraham Punnen, and Dr. Randall Pyke – in academic and pedagogical roles; I have grown significantly as a mathematician and an educator thanks to all these individuals. Special mention as well to the other members of the SFU Surrey community for their friendship and camaraderie – Melissa, Pooja, and the many other graduate students and postdoctoral fellows I've gotten to know over my time in the department.

Thank you to my friends who have been sources of emotional support, and who have actually seen me stress out over the actual writing process – Marvin and Shai in Vancouver; Tim, Ginj, and Bernie in Toronto. I am also grateful for family and friends in Vancouver and in the Philippines (some of whom I only see once a year), for their steadfast encouragement. Finally, to Poch who has stayed by my side for the past 101 months, I hope I inspire you as much as you inspire me.

Toronto, ON | December 17, 2017

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

The classical simplex method in linear programming solves a linear optimization problem over a convex polyhedron $P = \{\mathbf{x} : A\mathbf{x} \leq \mathbf{b}\}$ by taking steps along its edges starting at a vertex and moving towards an optimal vertex. As such, the *combinatorial diameter* of $P$, which is the largest distance between two vertices on its vertex-edge graph, offers some insight into the performance of this algorithm – for example, a superpolynomial lower bound on the diameter would imply that there is no polynomial-time pivot rule for simplex. Recent years have seen encouraging progress in this regard, with both upper bounds and lower bounds (for generalizations) being discovered (e.g. Todd [Tod14], Sukegawa and Kitahara [SK15], Sukegawa [Suk17]; Borgwardt et al. [BLF17], Eisenbrand et al. [EHRR10], Labbé et al. [LMS17]). Much of the work in this field was spurred by the following claim, put forth in 1957 by W. Hirsch to G. Dantzig via written communication:

**Conjecture 1.1** (Hirsch, 1957)**.** *The diameter of a $d$-dimensional polyhedron with $f$ facets is bounded above by $f - d$.*

Surprisingly it was proven in (Klee and Walkup [KW67]) that the Hirsch conjecture is equivalent to each of the following claims:

**Conjecture 1.2** ($d$-step)**.** *The diameter of a $d$-dimensional polyhedron with $2d$ facets is bounded above by $d$.*

**Conjecture 1.3** (nonrevisiting)**.** *Between any two vertices in a polyhedron there is a walk that does not enter any facet it has already left.*

However we now know these claims to not be true in general – in the same paper Klee and Walkup give an unbounded counterexample in 4 dimensions that has 8 facets and diameter 5; Santos constructed in [San12] the first bounded counterexample, this time in 43 dimensions, having 86 facets and diameter at least 44 (it was further improved to a 20-dimensional counterexample in Matschke et al. [MSW15]).

Standard approaches to obtaining bounds on diameters include looking at generalizations of polyhedra and studying variants of the combinatorial diameter. In this thesis we consider the diameter variant called the *circuit diameter*, introduced in (Borgwardt et al. [BFH15]), where we allow vertex-vertex walks to enter the interior of the polyhedron, using the *circuit directions* of $P$.

**Definition 1.4** (Circuit direction)**.** Given a polyhedron

$$P = \{\mathbf{x} \in \mathbb{R}^d : A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2\},$$

with $A^i \in \mathbb{Q}^{m_i \times d}$ and $\mathbf{b}^i \in \mathbb{Q}^{m_i}$ for $i = 1, 2$, the set of *circuit directions*, or *circuits*, $\mathcal{C}(A^1, A^2)$ of $A^1$ and $A^2$ are those vectors $\mathbf{g} \in \ker(A^1)\backslash\{\mathbf{0}\}$ for which $A^2\mathbf{g}$ is support-minimal in $\{A^2\mathbf{x} : \mathbf{x} \in \ker(A^1)\backslash\{\mathbf{0}\}\}$

Here, $\ker(A^1) = \{\mathbf{x} \in \mathbb{R}^d : A^1\mathbf{x} = \mathbf{0}\}$. When $f_1 = 0$ we assume $\ker(A^1) = \mathbb{R}^d$.

We refer to the circuits of a polyhedron $P$ by $\mathcal{C}(P)$. These were first introduced by Rockafellar as *elementary vectors* [Roc69]. Note that since $\mathbf{g} \in \mathcal{C}(P) \Rightarrow \lambda\mathbf{g} \in \mathcal{C}(P)$, without loss of generality we can normalize the members of $\mathcal{C}(P)$ to have coprime integer components. Also, observe that the definition does not depend on the vectors $\mathbf{b}^1, \mathbf{b}^2$. In fact, $\mathcal{C}(P)$ is exactly the set of *potential edge directions* of $P$ for varying right-hand sides. In particular, $\mathcal{C}(P)$ contains the actual edge directions of $P$.

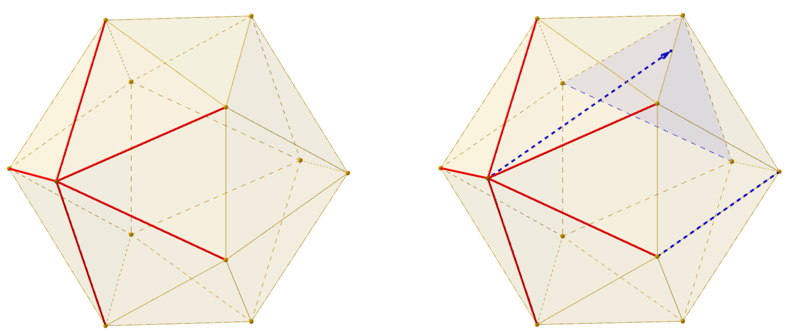

Figure 1.1: (L) Incident edge directions at a vertex; (R) A step along a circuit direction that passes through the interior

Each step in a *circuit walk* in a polyhedron is parallel to some circuit in $\mathcal{C}(P)$, and is *feasible* and *maximal* (i.e. the circuit direction is traversed as far as possible such that feasibility is preserved). Circuit distance and circuit diameter are then defined similarly as in the combinatorial scenario.

**Definition 1.5** (Circuit distance)**.** Given two points $\mathbf{x}, \mathbf{y} \in P$, the *circuit distance* from $\mathbf{x}$ to $\mathbf{y}$, denoted $\mathrm{dist}_{\mathcal{C}}(\mathbf{x}, \mathbf{y})$, is the smallest number of steps in a circuit walk from $\mathbf{x}$ to $\mathbf{y}$.

**Definition 1.6** (Circuit diameter)**.** Given a polyhedron $P$, the *circuit diameter* of $P$, denoted $\Delta_{\mathcal{C}}(P)$, is the length of the longest shortest circuit walk connecting any two of its vertices. That is,

$$\Delta_{\mathcal{C}}(P) = \max_{\mathbf{x},\mathbf{y} \in V(P)} \text{dist}_{\mathcal{C}}(\mathbf{x}, \mathbf{y}).$$

It is appealing to consider this diameter variant because the circuit diameter bounds the combinatorial diameter from below; this is because the set of edge directions of a polyhedron is a subset of its circuit directions. Analysis is not straightforward, however, since the circuit diameter is dependent on the realization of the polyhedron.

In this thesis we reformulate Conjectures 1.1, 1.2, and 1.3 using circuits; whether or not the Hirsch bound holds for circuit diameter is still open. Let $\Delta_{\mathcal{C}}(f, d)$ be the largest circuit diameter achieved among $d$-dimensional polyhedra with $f$ facets.

**Conjecture 1.7** (Circuit Hirsch, Borgwardt et al. [BFH15])**.** *$\Delta_{\mathcal{C}}(f, d) \leq f - d$.*

We formulate the *circuit $d$-step conjecture*, a conjecture on nonrevisiting circuit walks, and one on Dantzig figures, which are $d$-dimensional polyhedra with $2d$ facets and with two distinguished vertices $\mathbf{u}$ and $\mathbf{v}$ such that exactly $d$ facets are incident to each of $\mathbf{u}$ and $\mathbf{v}$.

To adapt Klee-Walkup constructions to the circuit framework we modified key polyhedral definitions. For example, $d$-dimensional polyhedra whose vertices are each contained in exactly $d$ facets are called *simple* polyhedra. These polyhedra satisfy the nice property that each step in an edge walk leaves and then enters exactly one facet. Moreover, any non-simple polyhedra can be made simple by a small perturbation of its facets. For circuit walks, since they are allowed to pass through the interior of the polyhedron, more than one facet may be left at each step. So we define the concept of $\mathcal{C}$-simplicity (or *circuit-simplicity*) to apply to polyhedra where circuit walks satisfy the desired property of only entering one new facet at each step. It turns out that via a series of perturbations one can also make any non-$\mathcal{C}$-simple polyhedron into a $\mathcal{C}$-simple one.

Moreover, we study the *wedging* operation that was applied in both (Klee and Walkup [KW67]) and (Santos [San12]) to build new polyhedra from old ones, and how the circuits and circuit diameters are affected by this construction. This operation will be formally defined in Definition 2.47.

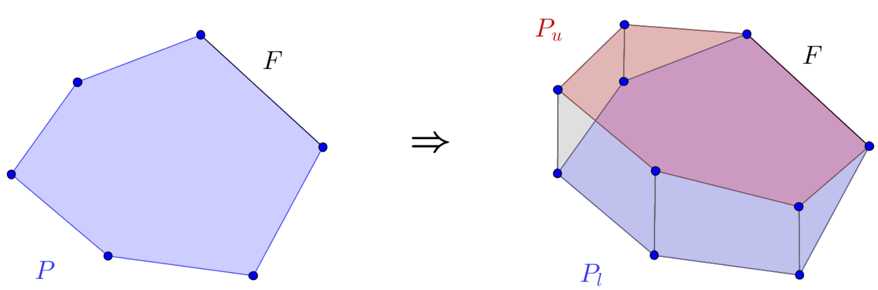


Figure 1.2: The wedge on $P$ over facet $F$.

When taking the wedge on a polyhedron over one of its faces, one obtains a new polyhedron whose facets and vertices relate back to those of the original polyhedron in a particular way. This operation also preserves simplicity; that is, the wedge on a simple polyhedron is also simple. Since this transfer does not hold for wedges on $\mathcal{C}$-simple polyhedra, for proving the main theorem it is necessary to define a stronger version of $\mathcal{C}$-simplicity where wedges can be taken so that the result is still $\mathcal{C}$-simple. Using these techniques we prove the following relationships among these circuit conjectures:

**Theorem 1.8.** *Consider the following statements:*

*(1) Let* $\mathbf{u}, \mathbf{v}$ *be two vertices of a* $k$*-wedge-simple polyhedron* $P$ *for* $k \geq f$*. Then there is a non-revisiting circuit walk from* $\mathbf{u}$ *to* $\mathbf{v}$*.*

*(2)* $\Delta_{\mathcal{C}}(f, d) \leq f - d$ *for all* $f \geq d$

*(3)* $\Delta_{\mathcal{C}}(2d, d) \leq d$ *for all* $d$

*(4) For all* $d$*-dimensional Dantzig figures* $(P, \mathbf{u}, \mathbf{v})$*, the circuit distance from* $\mathbf{u}$ *and* $\mathbf{v}$ *is at most* $d$*.*

*Then (1)* $\Rightarrow$ *(2)* $\Rightarrow$ *(3)* $\Rightarrow$ *(4).*

Another main result of this thesis is to prove that the circuit diameter of the Klee-Walkup counterexample satisfies the Hirsch bound, independent of realization. In Section 3.4 we use this to prove the circuit 4-step conjecture for unbounded polyhedra (i.e. 4-polyhedra with 8 facets have circuit diameter at most 4, regardless of the realization), in contrast to the combinatorial case.

**Theorem 1.9.** *The maximum circuit diameter of a 4-dimensional polyhedron with 8 facets is 4, independent of realization.*

An alternative proof of this is also given, that builds upon the results of Santos et al. [SST12]. As for the Santos counterexample (and others of the same construction) it is a challenging open problem to determine whether they satisfy circuit Hirsch, or if there is some realization that does not.

The remainder of this thesis is structured as follows: Chapter 2 discusses preliminaries and a brief survey of results on polyhedral diameters in the edge and circuit frameworks; Chapter 3 contains our main results (the adaptation of a number of polyhedral concepts into the circuit diameter framework; formulating and proving relationships among a number of circuit diameter-related theorems; a more concrete result in the resolution of the circuit 4-step conjecture); Chapter 4 discusses some of the computations we did to aid in the analysis, the challenges we encountered along the way, and some auxiliary results; Chapter 5 concludes the thesis. A preliminary version of some of the results is available as an ArXiv preprint (Borgwardt, Stephen, Yusun [BSY16]), but note the discussion of Theorem 1.8 contains an error. The figures in this paper were created using Geogebra [geo].

# Chapter 2

# Background and Related Results

In this chapter we briefly go over notation and relevant definitions, and summarize results about the diameter of polyhedra (in both the combinatorial and the circuit settings).

## 2.1 Preliminaries

We follow Bertsimas and Tsitsiklis ([BT97]) and Grünbaum ([Grü03]) for the relevant definitions and properties.

**Definition 2.1.** A *convex polyhedron*, or simply *polyhedron*, is a set of the form

$$P = \{\mathbf{x} \in \mathbb{R}^d : A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2\},$$

where $A^i \in \mathbb{Q}^{m_i \times d}$ and $\mathbf{b}^i \in \mathbb{Q}^{m_i}$, for $i = 1, 2$.

That is, $P$ is the finite intersection of the hyperplanes described by the equalities $A^1\mathbf{x} = \mathbf{b}^1$, and the half-spaces of the form $\mathbf{a}^j\mathbf{x} \geq b_j$, where the $\mathbf{a}^j$ vectors are the rows of $A^2$, and $b_j$ is the $j$th component of $\mathbf{b}^2$.

Now let us make the following important definitions:

**Definition 2.2** (Affine and convex combination)**.** Given points $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k \in \mathbb{R}^d$ and scalars $\lambda_1, \lambda_2, \ldots, \lambda_k \in \mathbb{R}$ such that $\sum_i \lambda_1 = 1$, the point $\sum_{i=1}^k \lambda_i\mathbf{x}^i$ is said to be an *affine combination* of the points $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k$.

If we restrict the scalars $\lambda_i$ to be nonnegative, then the point $\sum_{i=1}^k \lambda_i\mathbf{x}^i$ is called a *convex combination* of the points $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k$.

**Definition 2.3** (Affine and convex hull)**.** The *affine hull* $\mathrm{aff}(X)$ (*convex hull* $\mathrm{conv}(X)$) of a set $X$ of points is the set of all finite affine (convex) combinations of points in $X$.

**Definition 2.4** (Affine and convex set)**.** A set $S \subseteq \mathbb{R}^d$ is said to be *affine* if for all pairs $\mathbf{x}^1, \mathbf{x}^2 \in S$, the point $(1 - \lambda)\mathbf{x} + \lambda\mathbf{y}$ is also contained in $S$, for any $\lambda \in \mathbb{R}$.

The set $S \subseteq \mathbb{R}^d$ is called *convex* if for all pairs $\mathbf{x}^1, \mathbf{x}^2 \in S$, the point $(1-\lambda)\mathbf{x} + \lambda\mathbf{y}$ is also contained in $S$, for any $\lambda \in (0, 1)$.

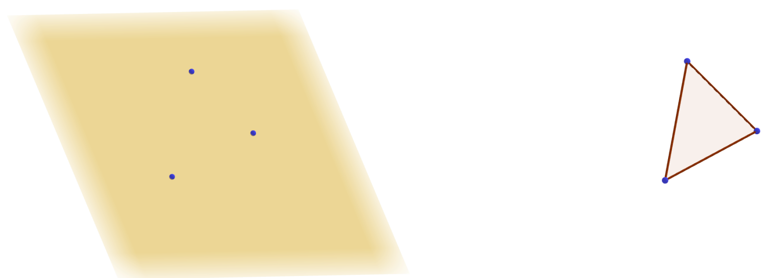

Figure 2.1: The affine hull (L) and convex hull (R) of three points in space.

Because we do not restrict the scalars $\lambda_i$ to be nonnegative when taking affine combinations, the affine hull of a set $X$ of points is just the collection of all *lines* that pass through any two distinct points in $X$; in contrast to this, the convex hull contains all *line segments* formed by taking two points $X$ as endpoints.

Geometrically, a set $S$ is convex if given any two points in $S$, the segment connecting the two points is also contained in $S$. These sets are 'easy' to optimize over, as we will see in the next section. Moreover, the intersection of two (or more) convex sets is also convex. Since half-spaces are convex, polyhedra are convex as well.

Instead of using inequalities to describe polyhedral sets, we can also use points:

**Definition 2.5** (Extreme point)**.** A point $\mathbf{x} \in \mathbb{R}^d$ is said to be an *extreme point* of a polyhedron $P$ if it cannot be written as a convex combination of points $\mathbf{y}, \mathbf{z} \in P$, that is, if $\mathbf{x} = \lambda\mathbf{y} + (1-\lambda)\mathbf{z}$ with $\lambda \in (0, 1)$ and $\mathbf{y}, \mathbf{z} \in P$, then $\mathbf{x} = \mathbf{y} = \mathbf{z}$.

**Definition 2.6** (Vertex)**.** A point $\mathbf{x} \in \mathbb{R}^d$ is said to be a *vertex* of the polyhedron $P$ if there exists a linear functional $\mathbf{c} \in \mathbb{R}^d$ such that $\mathbf{c}^T\mathbf{x} < \mathbf{c}^T\mathbf{y}$ for all $\mathbf{y} \in P$, $\mathbf{y} \neq \mathbf{x}$.

**Definition 2.7** (Basic solution)**.** A point $\mathbf{x} \in \mathbb{R}^d$ is said to be a *basic solution* of $P = \{\mathbf{x} \in \mathbb{R}^d : A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2\}$ if

(a) All equality constraints are active at $\mathbf{x}$ (so $A^1\mathbf{x} = \mathbf{b}^1$);

(b) Out of the constraints active at $\mathbf{x}$, there are $d$ of them that are linearly independent.

If $\mathbf{x} \in P$ as well, then it is called a *basic feasible solution* of $P$.

What is interesting about these definitions is that they are equivalent:

**Proposition 2.8.** *If $P = \{\mathbf{x} : \mathbb{R}^d : A\mathbf{x} \geq \mathbf{b}\}$ is a nonempty polyhedron and $\mathbf{x} \in P$, then the following are equivalent:*

*(a) $\mathbf{x}$ is an extreme point;*

*(b)* $\mathbf{x}$ *is a vertex;*

*(c)* $\mathbf{x}$ *is a basic feasible solution.*

Hence in what follows we use the above three terms interchangably, and denote by $V(P)$ the set of such points in $P$. Note that it is possible for $V(P)$ to be empty even if $P$ is nonempty; for instance, consider the polyhedron $\{(x_1, x_2) \in \mathbb{R}^2 : x_1 \geq 0\}$. Since $d = 2$ but only one linear inequality is given, this polyhedron does not have any basic feasible solutions, and so $V(P) = \emptyset$. We call polyhedra with at least one vertex *pointed*; in this thesis we consider pointed polyhedra unless otherwise noted.

We mention an easy characterization of pointedness:

**Proposition 2.9.** *A polyhedron is pointed if and only if it does not contain a line.*

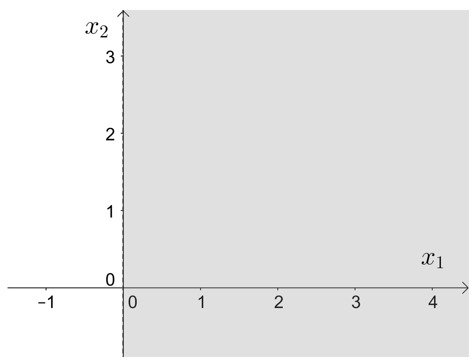


Figure 2.2: A polyhedron with no vertices.

If $P$ is bounded, then it cannot contain a line, and we have the following:

**Proposition 2.10.** *Suppose $P$ is a nonempty polytope. Then* $P = \mathrm{conv}(V(P))$.

This means that any point in $P$ can be written as a convex combination of its vertices.

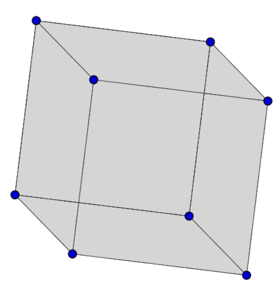

Figure 2.3: The 3-cube is the convex hull of its eight vertices.

For unbounded polyhedra, the vertices do not give a complete description. To see why, we define *rays* and *directions*:

**Definition 2.11** (Ray)**.** A ray is a set of the form $\{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} = \mathbf{x}^0 + \lambda\mathbf{w}, \lambda \geq 0\}$, where $\mathbf{w} \neq \mathbf{0}$. Here $\mathbf{w}$ is called the *direction of the ray*.

**Definition 2.12** (Recession direction)**.** Given a polyhedron $P$, a nonzero vector $\mathbf{w}$ is called a *direction of recession*, or simply *direction* of $P$ if for each $\mathbf{x}^0 \in P$, the ray $\{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} = \mathbf{x}^0 + \lambda\mathbf{w}, \lambda \geq 0\}$ is contained in $P$.

If a polyhedron is bounded, it does not have any directions of recession; this implies that we can prove that a polyhedron is unbounded by showing that it contains a ray.

**Definition 2.13** (Extreme direction)**.** Given a polyhedron $P$, a nonzero vector $\mathbf{w}$ is called an *extreme direction* of $P$ if it cannot be expressed as a positive combination of two distinct directions of $P$; that is, it cannot be written in the form $\mathbf{w} = \lambda_1\mathbf{w}^1 + \lambda_2\mathbf{w}^2$ for $\mathbf{w}^1, \mathbf{w}^2$ are directions of $P$ such that $\mathbf{w}^1 \neq k\mathbf{w}^2$ for any $k \neq 0$, and $\lambda_1, \lambda_2 > 0$.

**Definition 2.14** (Extreme ray)**.** Given a polyhedron $P$, a ray $\{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} = \mathbf{x}^0 + \lambda\mathbf{w}, \lambda \geq 0\}$ is called an *extreme ray* of $P$ if $\mathbf{w}$ is an extreme direction of $P$.

Note that if $\mathbf{w}$ is a direction of $P$, then so is $\lambda\mathbf{w}$ for any $\lambda > 0$. Hence when speaking of the set of extreme directions of a polyhedron, we implicitly assume that we are taking a representative from each equivalence class.

We can also define extreme directions using the *recession cone* of a polyhedron:

**Definition 2.15** (Cone)**.** A set $C \subset \mathbb{R}^d$ is a *cone* if $\lambda\mathbf{x} \in C$ for all $\lambda \geq 0$ and all $\mathbf{x} \in C$.

**Definition 2.16** (Polyhedral cone)**.** A *polyhedral cone* is a set of the form $C = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \geq \mathbf{0}\}$.

**Definition 2.17** (Recession cone)**.** Given a polyhedron $P = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \geq \mathbf{b}\}$, the *recession cone* of $P$ is the polyhedral cone $C = \{\mathbf{z} \in \mathbb{R}^d : A\mathbf{z} \geq \mathbf{0}\}$.

Note that in general, cones may not be convex. However, the polyhedral cone and the recession cone are examples of convex cones. The recession cone of a polyhedron is exactly the set of its directions of recession; so for polytopes the recession cone is the set $\{\mathbf{0}\}$. The extreme directions of a polyhedron can thus be seen as the extreme directions of its recession cone. Algebraically, the extreme directions of $C$ are those elements $\mathbf{w} \in C$ such that there are $d-1$ linearly independent inequalities among $A\mathbf{z} \geq 0$ that are met at $\mathbf{w}$ with equality. Now we can state the following fundamental result:

**Theorem 2.18** (Resolution Theorem (see, e.g. Bertsimas and Tsitsiklis [BT97]))**.** *Let*

$$P = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \geq \mathbf{b}\}$$

*be a nonempty polyhedron with at least one extreme point. Let* $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^k$ *be the extreme points, and let* $\mathbf{w}^1, \mathbf{w}^2 \ldots, \mathbf{w}^r$ *be the extreme directions of* $P$*. Let*

$$Q = \left\{ \sum_{i=1}^{k} \lambda_i \mathbf{x}^i + \sum_{j=1}^{r} \theta_j \mathbf{w}^j : \lambda_i \geq 0, \theta_j \geq 0, \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

*Then* $Q = P$*.*

The summation $\sum_{j=1}^{r} \theta_j \mathbf{w}^j$, $\theta_j \geq 0$, $j = 1, 2, \ldots, r$ is called a *conic combination* of the points $\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^r$. Hence this theorem says that a point in a polyhedron can be expressed as the sum of a convex combination of its vertices and a conic combination of its extreme rays. This has various implications in the study of linear optimization problems, which we will discuss in the next section. For now we continue discussing the geometry and structure of polyhedra.

**Definition 2.19** (Supporting hyperplane)**.** Given a polyhderon $P = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \geq \mathbf{b}\}$, a hyperplane $H = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{a}^T\mathbf{x} = b\}$ is called a *supporting hyperplane* of $P$ if

(a) $P$ is contained in either one of the two half-spaces $H^{\geq} = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} \geq b\}$ or $H^{\leq} = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} \leq b\}$; and

(b) $P \cap H \neq \emptyset$.

The vector $\mathbf{a}$ is called the *outward normal* of this supporting hyperplane.

**Definition 2.20** (Dimension)**.** The *dimension* of an affine space is the largest number of affinely independent vectors it contains, minus 1.

In particular, the dimension of the empty set is $-1$, the dimension of a singleton set is 0, the dimension of a line is 1, and the dimension of a hyperplane in $\mathbb{R}^d$ is $d-1$. Now we can define the *faces* of polyhedra.

**Definition 2.21** (Face)**.** Let $P = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \geq \mathbf{b}\}$. Then the set $F$ is said to be a *face* of $P$ if:

1. $F = \emptyset$;
2. $F = P$; or
3. $F = P \cap H$ for some supporting hyperplane $H$ of $P$.

The *dimension* of $F$ is the dimension of $\operatorname{aff}(F)$. We say that $F$ is a *k-face* of $P$ if it is a face of dimension $k$.

**Definition 2.22** (Facet)**.** Suppose that $P$ is a polyhedron of dimension $d$. Then:

1. The faces of $P$ of dimension $d-1$ are called its *facets*; and

2. The faces of $P$ of dimension 1 are called its *edges*.

We denote the set of edges of a polyhedron $P$ by $E(P)$. Also, observe that the 0-dimensional faces of $P$ are its vertices $V(P)$. In Figure 2.4 the above definitions are illustrated – $H_1 \cap P$ is a facet, $H_2 \cap P$ is an edge, and $H_3 \cap P$ is a vertex. The $d$-cube in general has $2d$ facets and $2^d$ vertices (so, the 3-cube has 6 facets and 8 vertices).

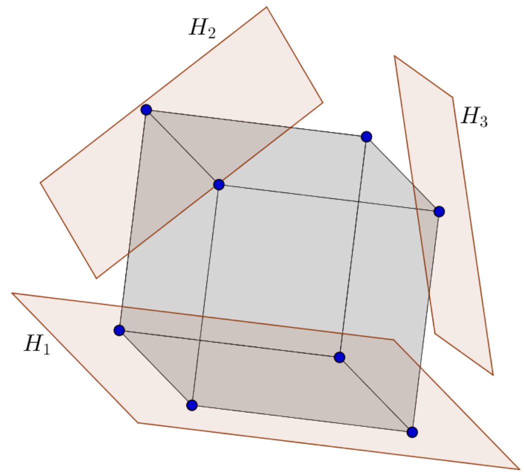


Figure 2.4: Supporting hyperplanes for a facet, an edge, and a vertex of the 3-cube.

The facets of a polyhedron play a special role in its description, in the same way as vertices and rays do.

**Definition 2.23** (Facet-defining)**.** An inequality $\mathbf{a}^T\mathbf{x} \geq b$ is said to be *facet-defining* for the polyhedron $P$ if the set $P \cap \{\mathbf{x} \in \mathbb{R}^d : \mathbf{a}^T\mathbf{x} \geq b\}$ is a facet of $P$. The hyperplane $\mathbf{a}^T\mathbf{x} = b$ is called a *defining hyperplane* for $P$.

**Definition 2.24** (Irredundant)**.** The representation of a polyhedron $P$ in the form $P = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \geq \mathbf{b}\}$ is said to be *irredundant* or *minimal* if removing any inequality from $A\mathbf{x} \geq \mathbf{b}$ results in a different polyhedron.

**Proposition 2.25** (see, e.g. Grünbaum [Grü03])**.** *Suppose that $P \subseteq \mathbb{R}^d$ is a d-dimensional poyhedron. Then there is a unique irredundant description of $P$ given by the set of its facet-defining inequalities.*

The faces of a polyhedron also satisfy the following properties:

**Proposition 2.26.** *Let $P$ be a polyhedron, and let $F_1$ and $F_2$ be faces of $P$. Then $F_1 \cap F_2$ is a face of $P$.*

**Proposition 2.27.** *Let $P$ be a polyhedron with vertex set $V(P)$, and let $F$ be a face of $P$. Then $F$ is a polyhedron with vertex set $V(P) \cap \mathrm{aff}(F)$.*

**Proposition 2.28.** *Let $P$ be a polyhedron, let $F_1$ be a face of $P$, and let $F_2$ be a face of $F_1$. Then $F_2$ is a face of $P$.*

We can also define a $k$-face $F$ of $P$ by the number of linearly independent facets active at $F$. For a general polyhedron $P \subseteq \mathbb{R}^d$, vertices are the intersections of $d$ linearly independent facets, edges are the intersections of $d-1$ of them, and so on. Furthermore, the combinatorial structure of $P$ is determined by its *face lattice*:

**Definition 2.29** (Face lattice)**.** Given a polyhedron $P$, its *face lattice* $\mathcal{F}(P)$ is the set of its faces, partially ordered by inclusion.

If two polyhedra have the same face lattice, they are combinatorially identical:

**Definition 2.30** (Combinatorial equivalence)**.** Two polyhedra $P_1$ and $P_2$ are said to be *combinatorially equivalent*, or *isomorphic*, or *of the same combinatorial type*, if $\mathcal{F}(P_1)$ is isomorphic to $\mathcal{F}(P_2)$. That is, there is an inclusion-preserving bijection between the faces of $P_1$ and the faces of $P_2$.

This is an important definition as many properties of a polytope can be inferred from its combinatorial type. In fact, when it is clear we will talk about polyhedra in terms of the entire equivalence class (so, *the* 3*-cube* will mean *polytopes that are isomorphic to the standard* 3*-cube in Figure 2.3.*)

We give names to the classes of polyhedra whose vertices are contained in the minimum number of facets (or, whose facets contain the minimum number of vertices).

**Definition 2.31** (Simple)**.** A $d$-polyhedron is *simple* if each vertex is incident to exactly $d$ facets.

**Definition 2.32** (Simplicial)**.** A $d$-polyhedron is *simplicial* if each facet is incident to exactly $d$ vertices.

We will also use *simple* to refer to vertices and *simplicial* for facets where the corresponding conditions are true.

Note that the roles of facets and vertices in polyhedra are similar, in that either can be used to fully describe a polyhedron. The *dual* of a polyhedron can be defined as:

**Definition 2.33** (Dual polyhedron)**.** Given polyhedra $P, P^* \subseteq \mathbb{R}^d$, we say that $P^*$ is *dual to* $P$ if between $\mathcal{F}(P)$ and $\mathcal{F}(P^*)$ there is an inclusion-reversing bijection $\phi$. That is, vertices of $P$ are mapped to facets of $P^*$, edges of $P$ to $(d-2)$-faces of $P^*$, and so on, such that if $F_1 \subset F_2$ in $P$, then $\phi(F_2) \subset \phi(F_1)$ in $P^*$.

For example, the dual of the $d$-cube (which has $2d$ facets and $2^d$ vertices) is called the *cross-polytope* (which has $2^d$ facets and $2d$ vertices). In 3 dimensions, this is the octahedron. In general, the cross-polytope is the convex hull of the elementary basis vectors in $\mathbb{R}^d$ together with their negatives.

Note that cubes are simple polytopes, while cross-polytopes are simplicial. This holds in general - the dual of a simple polytope is always simplicial.

**Definition 2.34** (Graph of a polytope)**.** Given a polytope $P$, its *graph* is defined as the graph $G_P = (V, E)$ where $V = V(P)$ and $E = E(P)$.

For an unbounded polyhedron, its graph is defined in the same way – as the graph induced by its vertices. As such, information about unbounded rays is not captured in the graph. Sometimes we will append rays to a graph when it suits the discussion (see Figure 2.7).

**Definition 2.35** (Distance)**.** Given a polyhedron $P$ and two vertices $\mathbf{x}, \mathbf{y} \in V(P)$, the *distance* between $\mathbf{x}$ and $\mathbf{y}$, denoted by $\text{dist}_{\mathcal{E}}(\mathbf{x}, \mathbf{y})$, is the length of the shortest path from $\mathbf{x}$ to $\mathbf{y}$ on the graph of $P$.

**Definition 2.36** (Diameter)**.** Given a polyhedron $P$, its *diameter*, denoted by $\Delta_{\mathcal{E}}(P)$, is the maximum value of $\text{dist}_{\mathcal{E}}(\mathbf{x}, \mathbf{y})$ attained over all pairs $\mathbf{x}, \mathbf{y} \in V(P)$.

Finally, we define the product operation applied on two polyhedra; this will be an essential part of the later chapters.

**Definition 2.37** (Product)**.** Given polyhedra $P \subset \mathbb{R}^{d_1}$ and $Q \subset \mathbb{R}^{d_2}$, the *product* of $P$ and $Q$, denoted by $P \times Q$, is the polyhedron

$$P \times Q = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d_1+d_2} : \mathbf{x} \in P, \mathbf{y} \in Q\}.$$

If $F$ is a $k$-dimensional face of $P \times Q$, then it can be written as $F = F_1 \times F_2$, where $F_1$ and $F_2$ are $k_1$- and $k_2$-dimensional faes of $P$ and $Q$, respectively, and such that $k_1 + k_2 = k$.

The last property implies in particular that the vertices (0-dimensional faces) of $P \times Q$ are of the form $(\mathbf{x}_P, \mathbf{x}_Q)$, where $\mathbf{x}_P \in V(P)$ and $\mathbf{x}_Q \in V(Q)$. Moreover, an edge of $P \times Q$ is a product of a vertex of one factor and an edge of the other, while each facet of $P \times Q$ is a product of one factor and a facet of the other.

Edge walks in $P \times Q$ have a specific structure; a step taken from a vertex $(\mathbf{x}_p, \mathbf{x}_Q)$ along an incident edge can only affect one of the two elements $\mathbf{x}_P$, $\mathbf{x}_Q$. Hence edge walks in the product can be decomposed into two edge walks, one in each factor. This implies:

**Proposition 2.38.** $\Delta_{\mathcal{E}}(P \times Q) = \Delta_{\mathcal{E}}(P) + \Delta_{\mathcal{E}}(Q)$.

**Example 2.39.** The 3-cube $C_3$ has diameter 3, since starting from any vertex one can reach any other vertex in at most 3 edge steps. In general, the $d$-cube $C_d$ has diameter $d$. This can be seen by recursively-defining the cube as $C_{d+1} = C_d \times [0, 1]$, and using Proposition 2.38.

## 2.2 Linear Programming and the Simplex Method

The diameter of a polyhedron has a deep connection with the classical simplex method for linear programming, an augmentation algorithm that moves along the edges of a polyhedron towards an optimal solution. Specifically, the combinatorial diameter gives a lower bound on the number of augmentations needed in the best-case performance of simplex. The linear programming problem is defined as follows:

**Definition 2.40** (Linear program)**.** A *linear program* (LP), or a *linear programming problem*, is an optimization problem of the form $\min\{\mathbf{c}^T\mathbf{x} : A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2\}$ (or max).

It is an optimization problem that seeks the best value attained by a linear function $\mathbf{c}^T\mathbf{x}$ on a convex polyhedron $P$.

Linear programming is extremely useful, and is widely seen in a large number of real-world applications. It is the heart of mathematical programming, which is (broadly speaking) the field of mathematics concerned with finding the best solution out of a set of alternatives. The history of linear programming is rich and goes back to World War II and even before then – see (Dantzig [Dan63]) for a thorough discussion of the history of the field. Some examples of areas that linear programming has appeared in include the oil industry (Garvin et al. [GCJS57]), cancer research (Mangasarian et al. [MSW95]), portfolio optimization (Mansini et al. [MOS14]), and employee scheduling (Hanssmann and Hess [HH60]).

The linear function $\mathbf{c}^T\mathbf{x}$ that is being minimized (or maximized) is called the *objective function* of the linear program, and the set of linear equations and inequalities $\{A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2\}$ is called its *feasible region.* Moreover, the individual rows in the relations $A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2$ are called the *constraints* of the linear program. Let us call a problem of the form

$$\begin{aligned} \min\ & \mathbf{c}^T\mathbf{x} \\ \text{such that } & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{2.1}$$

(and its feasible region) to be in *standard form.* Intuitively, this can be seen as a weighting problem where we assign nonnegative multiplers $x_i$ to the columns $A_i$ of $A$ such that the weighted sum is exactly $\mathbf{b}$, and the weighting is best possible with respect to some linear function $\mathbf{c}^T\mathbf{x}$. Any polyhedron with a vertex, and hence any linear program on it, can be written in this standard form.

Note that we can always rewrite the feasible region of a linear program to look like Definition 2.1, or Equation (2.1), or the inequality form $A\mathbf{x} \geq \mathbf{b}$. When the specific formulation is not relevant, we write linear programs as

$$\min\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in P\} \tag{2.2}$$

for convex polyhedron $P \subset \mathbb{R}^d$. Note that while we are guaranteed the existence of a basic feasible solution in formulation (2.1), polyhedra in general may not be pointed, so formulation (2.2) may not have a vertex.

We have the following important properties of linear programs:

**Definition 2.41** (Feasibility, optimality, unboundedness)**.** Given the linear program (2.2),

(a) If $P$ is nonempty, then the linear program is said to be *feasible*; if $P$ is empty, then the linear program is *infeasible*.

(b) If there is a point $\mathbf{x} \in P$ such that $\mathbf{c}^T\mathbf{x} \leq \mathbf{c}^T\mathbf{y}$ for all $\mathbf{y} \in P$, then the linear program is said to have an *optimal solution* $\mathbf{x}$.

(c) If, given any point $\mathbf{x} \in P$, there is another point $\mathbf{y} \in P$ such that $\mathbf{c}^T\mathbf{y} < \mathbf{c}^T\mathbf{x}$, we say the linear program is *unbounded*.

Here are some examples to illustrate the above possibilities (the red arrows show the objective function $\mathbf{c}^T\mathbf{x} = c_1x_1 + c_2x_2$).

$$\begin{aligned} \min\ & x_1 + x_2 \\ \text{s.t.}\ & x_1 \leq 0 \\ & x_1 \geq 1 \end{aligned} \qquad \begin{aligned} \min\ & x_1 + x_2 \\ \text{s.t.}\ & -x_1 + x_2 \geq -1 \\ & 3x_1 + x_2 \geq -1 \\ & -x_1 - 2x_2 \geq -3 \end{aligned} \qquad \begin{aligned} \min\ & x_1 + x_2 \\ \text{s.t.}\ & -x_1 + x_2 \geq -1 \\ & x_1 - 3x_2 \geq -1 \\ & -x_1 - 2x_2 \geq -3 \end{aligned}$$

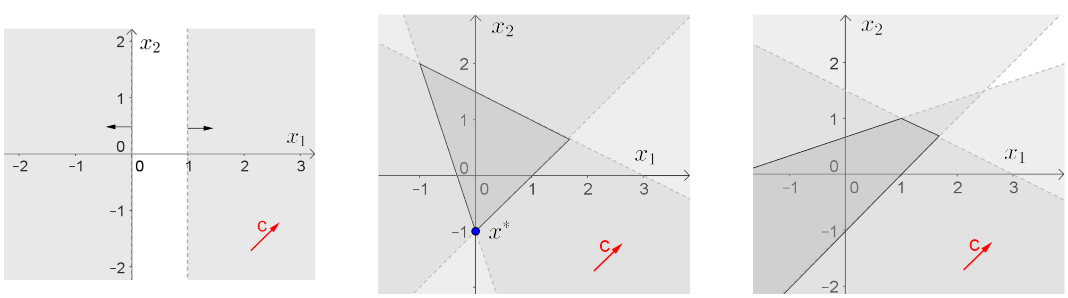

Figure 2.5: (L-R) An infeasible LP, an LP with an optimal solution, and an LP with optimal cost $-\infty$.

The following two propositions show the importance of the vertices and extreme rays of a polyhedron in linear programming:

**Proposition 2.42.** *If the linear program* (2.2) *has an optimal solution, where $P$ is a pointed polyhedron, then there is a vertex $\mathbf{x} \in V(P)$ where optimality is achieved.*

**Proposition 2.43.** *If the linear program* (2.2) *is unbounded, then there is an extreme direction* $\mathbf{w}$ *of* $P$ *such that* $\mathbf{c}^T\mathbf{w} < 0$.

Note that we use the term *unbounded* here to refer to the fact that the optimal cost is $-\infty$, and not to describe the feasible region. This is because a linear program with an unbounded feasible set may still have a finite optimal solution (consider the last example above but with the objective function $x_1 + x_2$ instead).

The **simplex method** for solving linear programs takes advantage of Proposition 2.42 by traversing the vertices of the feasible region until the optimal solution is found. What makes the simplex method so powerful is the fact that it systematizes the search for an optimal extreme point by using the geometric structure of the feasible region. Indeed, any linear program can be solved by enumerating all vertices of the feasible polyhedral set, but clearly this does not scale to larger problems. Instead, simplex performs the following steps:

(1) Look for an initial feasible extreme point $\mathbf{x}^0$. If one cannot be found, the LP is infeasible.

(2) Check if there is an edge incident to $\mathbf{x}^k$ along which the objective function is decreasing.

   (a) If there is such an edge, there are two possibilities: either it leads to another vertex $\mathbf{x}^{k+1}$ of the polyhedron, after which step (2) is repeated; or, it doesn't end, in which case the LP is unbounded.

   (b) If there is no such improving (decreasing) edge from $\mathbf{x}^k$, then $\mathbf{x}^k$ is the optimal vertex.

A key fact about the simplex method is the fact that when no improving edge directions are found at the current vertex $\mathbf{x}^*$ (a *local* condition), it can be concluded that $\mathbf{x}^*$ is the optimal solution to the linear program (a *global* implication). This follows from the convexity of the feasible region and linearity of the objective function.

### 2.2.1 The Simplex Method and the Geometry of Polyhedra

The simplex method uses the standard form LP in Equation (2.1):

$$\begin{aligned} \min\ & \mathbf{c}^T\mathbf{x} \\ \text{such that } & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Suppose that $A$ is $m \times d$ and $\mathbf{b}$, $\mathbf{c}$ are $d \times 1$. Then the extreme points of the polyhedron $P = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ are exactly those points $\mathbf{x}$ satisfying the equality conditions $A\mathbf{x} = \mathbf{b}$, and in addition $d - m$ of the nonnegativity constraints are met with equality at $\mathbf{x}$ (we say that these constraints are *active* at $\mathbf{x}$). This means that $d - m$ variables among the $x_i$'s are zero; let $N$ be the set of indices of these $d - m$ variables, and let $B$ be the set of indices corresponding to the remaining $m$ variables. Then we can rewrite the system

$$A\mathbf{x} = \mathbf{b}$$

as

$$A_B\mathbf{x}_B + A_N\mathbf{x}_N = \mathbf{b} \tag{2.3}$$

where $A_B$ is $m \times m$, $A_N$ is $(d - m) \times m$, and each contains the columns of $A$ corresponding to indices in $B$ and $N$, respectively. The variables $\{x_i : i \in B\}$ are called *basic variables*, while those in $\{x_i : i \in N\}$ are *nonbasic variables*; we call $B$ a *basis* for this basic feasible solution. Now since $x_i = 0$ for $i \in N$, we have $x_B = A_B^{-1}\mathbf{b}$ as the values for the basic variables.

Moving from a vertex to an adjacent vertex entails choosing an *entering variable* $x_l$ to enter the basis $B$, such that the objective function is improved when $x_l$ is increased. Increasing $x_l$ affects the other basic variables – at some point a variable that was in the basis may drop to zero, in which case we encounter another vertex. Or, we might discover that we can increase the value of $x_l$ indefinitely without violating any of the other conditions. When this happens, the LP is unbounded.

This method of picking an entering variable and a leaving variable is called a *pivot rule*; choosing an appropriate one to use in an implementation of the simplex method is crucial.

Note that basic variables are not restricted to be nonzero; there can be multiple bases that represent the same point $\mathbf{x}$. This phenomenon is called *degeneracy*, which may result in pivot steps where the objective function does not improve (or worse, *cycling*, when the simplex implementation loops through the same sequence of basic feasible solution). It is possible to design a pivot rule that prevents cycling. Bland's rule (Bland [Bla77]) was the first to do so; it picks the entering variable lexicographically from among the improving nonbasic variables, and breaks ties among candidate leaving variables lexicographically as well.

Regardless of the pivot rule used, the geometric interpretation of the simplex method is the same – moving from vertex to adjacent vertex of a polyhedron along its edges. Hence, the simplex method traces a walk on the graph of the feasible polyhedron from the starting vertex $\mathbf{x}^0$ to the optimal vertex $\mathbf{x}^*$. This walk can be no shorter than $\mathrm{dist}_{\mathcal{E}}(\mathbf{x}^0, \mathbf{x}^*)$, and so in the best-case performance of the simplex method, the quantity $\max \mathrm{dist}_{\mathcal{E}}(\mathbf{x}, \mathbf{y})$ is a lower bound on the number of steps to find the optimum (where $\mathbf{x}, \mathbf{y}$ are vertices of $P$). This is exactly $\Delta_{\mathcal{E}}(P)$.

The connection between polyhedral diameters and the performance of simplex leads to the question: *What is the largest possible combinatorial diameter of a $d$-dimensional polyhedron with a given number $f$ of facets?*

Dimension and number of facets are selected as parameters as they correspond to the numbers of variables and inequality constraints in the linear program. If there is a family of polyhedra with combinatorial diameter exponential in $d$ and $f$, then a pivot rule cannot exist for the simplex method that is polynomial-time in these inputs.

We introduce notation for the combinatorial diameter of the class of $d$-polyhedra with $f$ facets.

- Let $\Delta_{\mathcal{E}}(f,d)$ denote the maximum combinatorial diameter attained by $d$-dimensional polyhedra with $f$ facets;
- Let $\Delta^u_{\mathcal{E}}(f,d)$ and $\Delta^b_{\mathcal{E}}(f,d)$ denote the maximum combinatorial diameter attained by unbounded and bounded $d$-polyhedra with $f$ facets, respectively.
- In general, we refer to a $d$-dimensional polyhedron with $f$ facets as an $(f,d)$-polyhedron.

In the late 1950s Hirsch conjectured the bound $\Delta_{\mathcal{E}}(f,d) \leq f-d$ (by written communication to Dantzig, see [Dan63]). This quantity has since become a benchmark for talking about diameters of classes of polyhedra.

In the next section we outline some of the main results about $\Delta_{\mathcal{E}}(f,d)$, including various lower and upper bounds, and prove the equivalence of the Hirsch conjecture with other related statements.

## 2.3 Combinatorial Diameter: Survey of Results

Throughout the years both positive and negative results have been proven about the quantity $\Delta_{\mathcal{E}}(f,d)$ and the Hirsch conjecture. Techniques developed in the 60's still prove relevant to this day; we begin with a powerful reduction in the study of diameters.

**Proposition 2.44** ([KW67]). *For any $f > d \geq 1$, $\Delta^b_{\mathcal{E}}(f,d)$ is realized as the distance between two vertices $\mathbf{x},\mathbf{y}$ of a simple $d$-polytope with $f$ facets. For any $f > d \geq 2$, $\Delta^u_{\mathcal{E}}(f,d)$ is realized as the distance between two vertices $\mathbf{x},\mathbf{y}$ of a simple $d$-polyhedron with $f$ facets such that $\mathbf{x}$ and $\mathbf{y}$ are incident to unbounded edges of the polyhedron. In both cases, if $f \geq 2d$, $\mathbf{x}$ and $\mathbf{y}$ may be chosen so they do not share a facet.*

This implies that the quantity $\Delta_{\mathcal{E}}(f,d)$ is achieved at a simple polyhedron. The proof of Proposition 2.44 proceeds by showing that any non-simple $d$-polyhedron with $f$ facets can be transformed to a simple $d$-polyhdron with the same number of facets and such that the diameter does not decrease. This is done by *perturbing* the facets by a small amount – this means that facets are translated a small amount (by changing the right-hand side) so that a vertex that has more than $d$ facets incident to it is transformed into multiple vertices, each with $d$ facets. Figure 2.6 illustrates this operation on a pyramid with a non-simple vertex having $5 > 3$ facets incident to it; the two polyhedra on the right show two different ways of perturbing the facets to produce a simple polyhedra. These polyhedra have diameters bounded below by the original diameter, since we only introduced new vertices and edges.

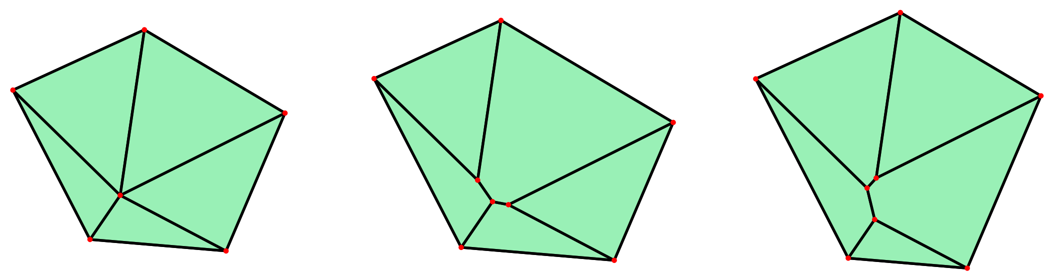

Figure 2.6: Two ways of perturbing the facets of a non-simple polytope (a pentagonal pyramid) to simplicity.

Edge walks on simple polyhedra are easier to work with than in the general case; this is because if two vertices $\mathbf{x}, \mathbf{y}$ of a simple polyhedron are adjacent, then exactly one facet is removed and one added when going from $\mathbf{x}$ to $\mathbf{y}$.

Most results in this area focus on the bounded case, or $\Delta^b_{\mathcal{E}}(f, d)$. This is because Klee and Walkup already give an unbounded 4-dimensional counterexample to Hirsch, that has 8 facets and diameter 5 (Klee and Walkup [KW67]). Let us call this polyhedron $U_4$; it has the 'graph' shown in Figure 2.7.[1]

Observe that $U_4$ is simple, since each vertex is contained in exactly 4 facets. Hence an edge walk from `V1234` to `V5678` has length at least 4 (one for each facet in the destination). It is clear from the graph, however, that $\mathrm{dist}_{\mathcal{E}}(\texttt{V1234}, \texttt{V5678}) = 5$. This implies that $\Delta^u_{\mathcal{E}}(8, 4) \geq 5$; Klee and Walkup also showed that $\Delta^u_{\mathcal{E}}(8, 4) = 5$. The polyhedron $U_4$ can also be used to construct infinitely many non-Hirsch polyhedra whose *excess* over the Hirsch bound tends to infinity (we omit the proof).

**Proposition 2.45** (Klee and Walkup [KW67])**.** $\Delta_{\mathcal{E}}(f, d) \geq f - d + \min\left(\lfloor \frac{d}{4} \rfloor, \lfloor \frac{f-d}{4} \rfloor\right)$.

The value of $\Delta^b_{\mathcal{E}}(f, d)$ is known for the following cases; this list is given in the comprehensive survey (Kim and Santos [KS10]).

**Theorem 2.46.**

- $\Delta^b_{\mathcal{E}}(f, 2) = \lfloor \frac{f}{2} \rfloor$ *(trivial.)*
- $\Delta^b_{\mathcal{E}}(f, 3) = \lfloor \frac{2f}{3} \rfloor - 1$ *(Klee [Kle64])*
- $\Delta^b_{\mathcal{E}}(8, 4) = 4$ *(Klee [Kle64])*
- $\Delta^b_{\mathcal{E}}(9, 4) = \Delta^b_{\mathcal{E}}(10, 5) = 5$ *(Klee and Walkup [KW67])*
- $\Delta^b_{\mathcal{E}}(10, 4) = 5$, $\Delta^b_{\mathcal{E}}(11, 5) = 6$ *(Goodey [Goo72])*

[1] Note that this is technically not a graph as we are using additional segments to represent unbounded rays.

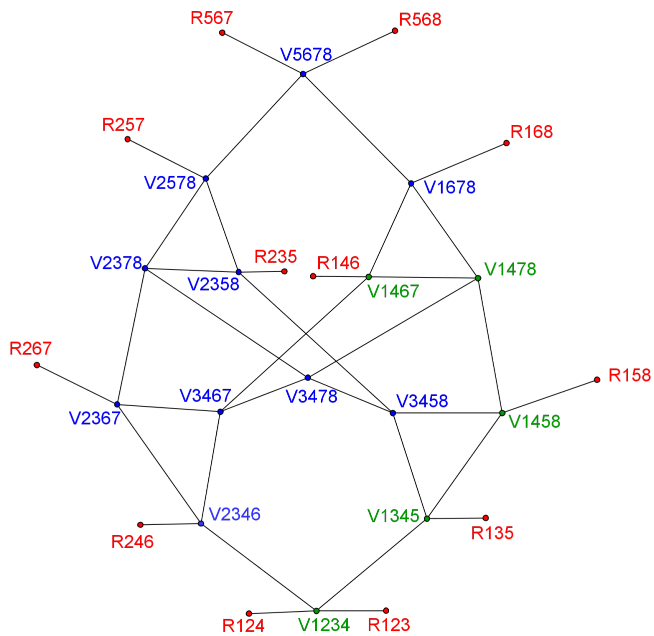


Figure 2.7: The graph of $U_4$, the Klee-Walkup counterexample to unbounded Hirsch. Each vertex and ray is labeled by its containing facets.

- $\Delta^b_{\mathcal{E}}(11, 4) = \Delta^b_{\mathcal{E}}(12, 6) = 6$ *(Bremner and Schewe [BS11])*
- $\Delta^b_{\mathcal{E}}(12, 4) = \Delta^b_{\mathcal{E}}(12, 5) = 7$ *(Bremner et al. [BDHS13])*

Observe that these pairs $(f, d)$ all satisfy $f \geq 2d$. This is because for a fixed $f - d$, the maximum value attained by $\Delta_{\mathcal{E}}(f, d)$ is when $f = 2d$. To see why this is the case, we first define the *wedge* operation for polyhedra.

**Definition 2.47** (Wedge)**.** Let $P$ be a $d$-dimensional polyhedron and let $F$ be a facet of $P$. Let $L = [0, \infty)$. Take the product $P \times L$ and let $H^{\leq} \subset \mathbb{R}^{d+1}$ be a closed halfspace containing $P \times \{0\}$, such that the defining hyperplane $H$ of $H^{\leq}$ intersects $P \times \{0\}$ exactly in $F$. Then the $(d+1)$-dimensional polyhedron $P' = H^{\leq} \cap (P \times L)$ is called the *wedge on $P$ over $F$*.

The wedge operation can also be viewed as taking the product $P \times [0, 1]$ and then identifying the two faces $F \times \{0\}$ and $F \times \{1\}$ together. Figure 2.8 depicts an example. Note that we only consider the wedging operation when it is done over a facet of $P$. The operation can be extended to faces of smaller dimension, but we don't use this here. Also, note that there is a distinction between 'the' wedge on $P$ over $F$ (the combinatorial class) and 'a' wedge on $P$ over $F$ (one realization in this combinatorial class; a different realization arises when choosing a different $H^{\leq}$). In what follows, which one we refer to will be clear from the context.

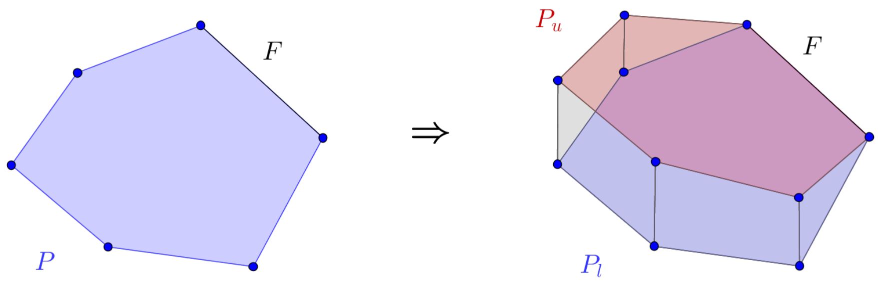


Figure 2.8: The wedge $P'$ on the hexagon $P$ over facet $F$. Bases are $P_l$ and $P_u$.

By construction, $P'$ has $f+1$ facets. The *lower base* $P_l = P \times \{0\}$ of $P'$ and the *upper base* $P_u = H \cap (P \times L)$ of $P'$ are facets of $P'$, and both are isomorphic to the original polyhedron $P$. The remaining $f-1$ facets of $P'$ are contained in spaces of the form $G \times L$, where $G \neq F$ is a facet of $P$; we call them the *sides* of the wedge. The lower base $P_l$ lies in the affine subspace $\mathbb{R}^d \times \{0\}$ while the upper base $P_u$ lies in the affine subspace $H$ of dimension $d$. We use $\phi$ to denote the projection to the upper base that is parallel to $L$, and $\phi^{-1}$ to denote the projection to the lower base, also moving parallel to $L$.

Wedges are a basic building block for results on combinatorial diameters, due to some nice properties:

**Proposition 2.48** (Properties of wedges)**.** *Let $P \subseteq \mathbb{R}^d$ be a polyhedron and $F$ one of its facets. Let $P' \subseteq \mathbb{R}^{d+1}$ be the wedge on $P$ over $F$. Then:*

*(i) If $P$ is simple, $P'$ is simple.*

*(ii)* $\Delta_{\mathcal{E}}(P) \leq \Delta_{\mathcal{E}}(P') \leq \Delta_{\mathcal{E}}(P)+1$.

*Proof.* (i) If $P$ is simple, then each of its vertices is incident to exactly $d$ facets. If $\mathbf{x} \in V(P)$ and $\mathbf{x} \notin F$, then it corresponds to two vertices $\mathbf{x}^l$ and $\mathbf{x}^u$ in the lower and upper bases of $P'$, respectively. The vertex $\mathbf{x}^l \in V(P')$ is incident to $d+1$ facets in $P'$: $d$ sides corresponding to the $d$ original facets it was incident to in $P$, and the lower base of $P'$. Similarly, $\mathbf{x}^u$ is incident to $d+1$ facets in $P'$: its original $d$ facets, and the upper base of $P'$.

If $\mathbf{x} \in F$, then in $P$ it was incident to $d$ facets, including $F$. In $P'$, it is incident to [the sides formed by] $d-1$ of the original facets, and also both lower and upper bases.

(ii) Any two vertices in $P$ can be connected by a path of length at most $\Delta_{\mathcal{E}}(P)$. Since any vertex $\mathbf{x} \in P_l$ has a neighbouring vertex $\phi(\mathbf{x}) \in P_u$, to connect any two vertices in $P'$ there is a path of length at most $\Delta_{\mathcal{E}}(P)+1$. So, $\Delta_{\mathcal{E}}(P') \leq \Delta_{\mathcal{E}}(P)+1$.

On the other hand, any edge walk in $P'$ transfers to a possibly shorter edge walk in $P$, by projecting to $\mathbb{R}^d \times \{0\}$. So, $\Delta_{\mathcal{E}}(P) \leq \Delta(P')$. $\square$

We can now prove the following:

**Proposition 2.49** (Klee and Walkup [KW67]). *$\Delta_{\mathcal{E}}(f,d) \leq \Delta_{\mathcal{E}}(f+1,d+1)$ for all $f \geq d$, with equality if $f \leq 2d$.*

*Proof.* Suppose $P$ is a $d$-polyhedron with $f$ facets and diameter $\Delta_{\mathcal{E}}(f,d)$. The wedge on $P$ over one of its facets is a $(d+1)$-polyhedron with $f+1$ facets, and whose diameter is no smaller than $\Delta_{\mathcal{E}}(P)$. This gives the inequality $\Delta_{\mathcal{E}}(f,d) \leq \Delta_{\mathcal{E}}(f+1,d+1)$.

For the second claim, suppose that $f \leq 2d$, and let $Q$ be a $(d+1)$-polyhedron with $(f+1)$ facets and diameter $\Delta_{\mathcal{E}}(f+1,d+1)$. Assume that $\mathbf{x},\mathbf{y} \in V(Q)$ such that $\mathrm{dist}_{\mathcal{E}}(\mathbf{x},\mathbf{y}) = \Delta_{\mathcal{E}}(f+1,d+1)$. Since $Q$ has $f+1 \leq 2d+1$ facets, and both $\mathbf{x},\mathbf{y}$ must be incident to at least $d+1$ facets, there must be a facet $K$ of $Q$ that contains both $\mathbf{x}$ and $\mathbf{y}$. The facet $K$ is also a $d$-polyhedron, and each of its facets is the intersection of $K$ with a facet of $Q$ (other than $K$) – hence $K$ has at most $f$ facets, and thus has diameter $\Delta_{\mathcal{E}}(K) \leq \Delta_{\mathcal{E}}(f,d)$. But $\mathrm{dist}_{\mathcal{E}}(\mathbf{x},\mathbf{y})$ in $K$ is no smaller than $\Delta_{\mathcal{E}}(Q) = \Delta_{\mathcal{E}}(f+1,d+1)$. This implies $\Delta_{\mathcal{E}}(K) \geq \Delta_{\mathcal{E}}(f+1,d+1)$, completing the proof of $\Delta_{\mathcal{E}}(f+1,d+1) \leq \Delta_{\mathcal{E}}(f,d)$. □

A consequence of this series of relations is that the Hirsch conjecture and the $d$-step conjecture are equivalent.

**Theorem 2.50.** *The following two statements are equivalent:*

*(1) (Hirsch) $\Delta_{\mathcal{E}}(f,d) \leq f-d$ for all $f,d$.*

*(2) (d-step) $\Delta_{\mathcal{E}}(2d,d) \leq d$ for all $d$.*

*Proof.* (1) $\Rightarrow$ (2): $d$-step is just a special case of Hirsch, for $f = 2d$.

(2) $\Rightarrow$ (1): By Proposition 2.49, if $f \leq 2d$, then $\Delta_{\mathcal{E}}(f,d) = \Delta_{\mathcal{E}}(2d,d)$. On the other hand, if $f > 2d$, then $\Delta_{\mathcal{E}}(f,d) \leq \Delta_{\mathcal{E}}(f+1,d+1) \leq \cdots \leq \Delta_{\mathcal{E}}(f+(f-2d),d+(f-2d))$, which is by (2) at most $f-d$. ($\Delta_{\mathcal{E}}(2(f-d),f-d) \leq f-d$) □

Note that proving $d$-step for a particular $d$ does not also prove Hirsch in that dimension; the relevant parameter in the equivalence is $f-d$.

Klee and Walkup proved a stronger result involving four statements. One of these talks about Dantzig figures:

**Definition 2.51** (Dantzig figure). Let $P$ be a $d$-dimensional polyhedron with $2d$ facets, of which exactly $d$ are incident to a vertex $\mathbf{u}$ and the other $d$ are incident to a vertex $\mathbf{v}$. Then the tuple $(P,\mathbf{u},\mathbf{v})$ is a *Dantzig figure*.

Essentially, Dantzig figures are the intersection of two $d$-dimensional cones of $d$ facets. In fact, it will suffice to consider the distances of $\mathbf{u}$ and $\mathbf{v}$ in such a Dantzig figure, as Klee and Walkup's result implies:

**Theorem 2.52** (Klee and Walkup [KW67])**.** *The following statements are equivalent:*

*(A) Let* $\mathbf{u}$*,* $\mathbf{v}$ *be two vertices of a simple polytope* $P$*. Then there is an edge walk between* $\mathbf{u}$ *to* $\mathbf{v}$ *that does not visit a facet more than once.*

*(B)* $\Delta^b_{\mathcal{E}}(f, d) \leq f - d$ *for all* $f \geq d$*.*

*(C)* $\Delta^b_{\mathcal{E}}(2d, d) \leq d$ *for all* $d$*.*

*(D) For all* $d$*-dimensional Dantzig figures* $(P, \mathbf{u}, \mathbf{v})$*,* $\mathrm{dist}_{\mathcal{E}}(\mathbf{u}, \mathbf{v}) = d$*.*

*Proof.* Theorem 2.50 takes care of the equivalence $\mathbf{B} \iff \mathbf{C}$. To prove the rest of the equivalences we show $\mathbf{A} \Rightarrow \mathbf{B}$, $\mathbf{C} \Rightarrow \mathbf{D} \Rightarrow \mathbf{A}$.

That $\mathbf{A} \Rightarrow \mathbf{B}$ follows immediately from the fact that in a simple $d$-dimensional polytope with $f$ facets, any non-revisiting walk starting at a vertex will necessarily take $f - d$ steps at most to get to any other vertex ($f$ facets in all, less $d$ incident at the starting vertex, gives $f - d$ facets left to visit).

Assuming $\mathbf{C}$ holds, if $P$ is an $d$-dimensional Dantzig figure, then $\Delta_{\mathcal{E}}(P) \leq d$. So there is a walk of length at most $d$ connecting the two distinguished vertices $\mathbf{x}$ and $\mathbf{y}$. However *at least* $d$ steps are needed in such a walk, so we have the exact length $d$. Hence $\mathbf{C} \Rightarrow \mathbf{D}$.

To show the last implication $\mathbf{D} \Rightarrow \mathbf{A}$, assume that the claim in $\mathbf{D}$ holds, and take a simple $d$-dimensional polytope $P$ with $f$ facets. Let $\mathbf{x}$ and $\mathbf{y}$ be any two vertices in $P$ – we will show that there is a non-revisiting path connecting them.

First, set $\mathbf{y}^0 = \mathbf{y}$, and let $F_0$ the smallest face of $P$ that is incident to both $\mathbf{x}$ and $\mathbf{y}$. Then $F_0$ is a simple polytope in $d'$ dimensions such that there are $d'$ facets incident at $\mathbf{x}$, $d'$ *other* facets incident at $\mathbf{y}$, and some $k$ facets that are not incident to either of $\mathbf{x}$ or $\mathbf{y}^0$. Thus $F_0$ has a total of $2d' + k$ facets.

If $k = 0$ then $F_0$ is a Dantzig figure in $d'$ dimensions, and by $\mathbf{D}$, the distance between $\mathbf{x}$ and $\mathbf{y}$ is exactly $d'$, and any walk connecting them must necessarily be non-revisiting.

If $k > 0$ then we will once again employ the wedging operation. Let $G$ be any facet of $F_0$ that does not contain either $\mathbf{x}$ or $\mathbf{y}$, and let $F_1$ be the wedge of $F_0$ over $G$. Then consider the resulting polytope $F_1$ with distinguished vertices $\mathbf{x}$ as before, and $\mathbf{y}^1$ the image of $\mathbf{y}^0$ in its upper base. The polytope $F_1$ is $d' + 1$ dimensional with $2d' + k + 1$ facets. Further, there are now $d' + 1$ facets incident to $\mathbf{x}$ and another $d' + 1$ facets incident to $\mathbf{y}^1$, leaving $k - 1$ facets not incident to either. Repeating this process we arrive at a simple $(d' + k)$-dimensional polytope $F_k$ with distinguished vertices $\mathbf{x}$, which is incident to $d' + k$ facets, and $\mathbf{y}^k$, incident to another $d' + k$ facets. Now $F_k$ is a Dantzig figure in dimension $d' + k$, so by $\mathbf{D}$, the shortest walk between $\mathbf{x}$ and $\mathbf{y}^k$ has length exactly $d' + k$, and so has to be non-revisiting (since we add one new facet at each step). By projecting this non-revisiting path down through all the wedges $F_k, F_{k-1}, \ldots, F_0$, we get a non-revisiting path between $\mathbf{x}$ and $\mathbf{y}^0 = \mathbf{y}$ in $F_0$, hence in $P$. Thus $\mathbf{D} \Rightarrow \mathbf{A}$ is proved, and so is the theorem. □

Klee and Walkup also give some relations involving $\Delta_{\mathcal{E}}(f, d)$, obtained from several constructions.

**Proposition 2.53.** *(a)* $\Delta^u_{\mathcal{E}}(f, d) < \Delta^u_{\mathcal{E}}(f+1, d)$

*(b)* $\Delta^b_{\mathcal{E}}(f, d) \leq \Delta^b_{\mathcal{E}}(f+1, d)$

*(c)* $\Delta^u_{\mathcal{E}}(f, d) < \Delta^u_{\mathcal{E}}(f+2, d+1)$ *and* $\Delta^b_{\mathcal{E}}(f, d) < \Delta^b_{\mathcal{E}}(f+2, d+1)$

*(d)* $\Delta^u_{\mathcal{E}}(f, d) \leq \Delta^u_{\mathcal{E}}(f+1, d+1)$ *and* $\Delta^b_{\mathcal{E}}(f, d) \leq \Delta^b_{\mathcal{E}}(f+1, d+1)$

*(e) If* $f \leq 2d$, $\Delta^u_{\mathcal{E}}(f, d) = \Delta^u_{\mathcal{E}}(f+1, d+1)$ *and* $\Delta^b_{\mathcal{E}}(f, d) = \Delta^b_{\mathcal{E}}(f+1, d+1)$

*(f)* $\Delta^u_{\mathcal{E}}(2d+k, d+k) = \Delta^u_{\mathcal{E}}(2d, d)$ *and* $\Delta^b_{\mathcal{E}}(2d+k, d+k) = \Delta^b_{\mathcal{E}}(2d, d)$ *for* $k > 0$*.*

*Proof.* (a) and (b) are obtained by truncating a vertex (in the unbounded case, a vertex incident to an unbounded edge); (c) from $P \times [0, 1]$; (d), (e), and (f) from the proof of Proposition 2.49. □

The strict inequalities in (a) and (c) also explain why a table of known values for $\Delta^u_{\mathcal{E}}(f, d)$ in the same vein as Theorem 2.46 is not as interesting. Because the Klee-Walkup counterexample implies $\Delta^u_{\mathcal{E}}(8, 4) = 5$, by (a) and (c) we have $\Delta^u_{\mathcal{E}}(f, d) > f - d$ for $f \geq 2d$, $d \geq 4$. That is, for each such pair $(f, d)$, there exists a non-Hirsch unbounded polyhedron.

It is also possible to prove relationships between $\Delta^b_{\mathcal{E}}(f, d)$ and $\Delta^u_{\mathcal{E}}(f, d)$. For instance, given a bounded polytope $P$, projecting one of its facets to infinity produces an unbounded polyhedron whose diameter is no smaller than that of $P$ – this shows $\Delta^b_{\mathcal{E}}(f, d) \geq \Delta^u_{\mathcal{E}}(f-1, d)$. A consequence of this is the fact that given a bounded polytope whose diameter meets the Hirsch bound exactly – which we call *Hirsch-sharp* – one can produce a non-Hirsch polyhedron by projecting a vertex to infinity. So, performing this operation on $U_4$ (Fig. 2.7) yields a Hirsch-sharp 4-polytope with 9 facets, which we will denote as $Q_4$ (following Kim and Santos [KS10]) and will talk about in more detail in Chapter 4.

Here are the best known upper bounds proven over the years for general $f$ and $d$:

1. $\Delta^b_{\mathcal{E}}(f, d) \leq 2^{d-3} f$ (Larman [Lar70])
2. $\Delta^b_{\mathcal{E}}(f, d) \leq f^{\log_2(d+2)}$ (Kalai and Kleitman [KK92])
3. $\Delta^b_{\mathcal{E}}(f, d) \leq (f - d)^{\log_2 d}$ (Todd [Tod14])
4. $\Delta^b_{\mathcal{E}}(f, d) \leq (f - d)^{\log_2(d)-3+O(1/d)}$ (Sukegawa [Suk17])

Many classes of polytopes have been proven to satisfy the Hirsch bound, including $\{0, 1\}$-polytopes (Naddef [Nad89]), and the broad class of network flow and transportation polytopes (Borgwardt et al. [BLF17]).

Moreover, dual transportation polytopes that are defined on complete bipartite graphs are also Hirsch (Balinski [Bal84]). Given a bipartite graph $G$ with vertex sets $V_1 = \{0, 1, \ldots, M-1\}$ and $V_2 = \{M, M+1, \ldots, M+N-1\}$, and edges $E$ having one endpoint in each of $V_1$ and $V_2$, a dual transportation polyhedron associated to $G$ and given by a vector $\mathbf{c} \in \mathbb{R}^{|E|}$ is

$$P_{G,\mathbf{c}} = \left\{ \mathbf{x} \in \mathbb{R}^{M+N} : -x_u + x_v \leq c_{uv},\ \forall u \in V_1, v \in V_2,\ \text{and } uv \in E, x_0 = 0 \right\}.$$

**Theorem 2.54** (Balinski [Bal84])**.** *The combinatorial diameter of* $P_{G,\mathbf{c}}$ *is bounded above by* $(M-1)(N-1) = MN - (M+N-1)$.

The class of dual network flow polyhedra is a superclass of dual transportation polyhedra, where the restriction that the underlying graph be bipartite is removed. For this class a quadratic upper bound has been proven:

**Theorem 2.55** (Borgwardt et al. [BFH16])**.** *The combinatorial diameter of the dual network flow polyhedron* $P_{G,\mathbf{c}}$ *is bounded above by* $\min\{(|V|-1)|E|, \frac{1}{6}|V|^3\}$.

On the other hand, the first bounded counterexample violating the Hirsch bound was only discovered in 2010 by Santos:

**Theorem 2.56** (Santos [San12])**.** *There is a 43-dimensional polytope with 86 facets with combinatorial diameter at least 44.*

A modified version of the wedging operation, called the *perturbed wedge*, was vital in this discovery. This is performed on a special type of polyhedron called *spindles*, which generalize Dantzig figures:

**Definition 2.57** (Spindle)**.** A *spindle* $P(\mathbf{u}, \mathbf{v})$ is a polyhedron $P$ with two distinguished vertices $\mathbf{u}$ and $\mathbf{v}$ such that each facet of $P$ is incident to exactly one of $\mathbf{u}$ and $\mathbf{v}$. The *length* of the spindle $P$ is defined to be $\text{dist}_{\mathcal{E}}(\mathbf{u}, \mathbf{v})$.

**Lemma 2.58** (Strong $d$-step Lemma, Santos [San12])**.** *For every* $d$*-spindle* $P$ *with* $f > 2d$ *facets and length* $\lambda$ *there is a* $(d+1)$*-spindle with one more facet and length* $\lambda + 1$.

In a later paper Matschke, Santos, and Weibel proved the existence of a smaller non-Hirsch polytope:

**Theorem 2.59** (Matschke et al. [MSW15])**.** *There is a 20-dimensional polytope with 40 facets (and 36442 vertices) with combinatorial diameter 21.*

The methods used to construct this counterexample is the same as in the original – starting with a low-dimensional spindle, the strong $d$-step lemma is applied until the number of facets is twice the dimension. The crux of this result is the construction of the initial 5-dimensional spindle with 25 facets and length 6 (to which strong $d$-step is applied 15 times).

Interestingly, Santos, Stephen, and Thomas proved that one cannot use a 4-dimensional spindle in this way to obtain non-Hirsch polytopes.

**Theorem 2.60** (Santos et al. [SST12])**.** *4-dimensional spindles have length at most 4.*

Some other approaches to studying the combinatorial diameter involves looking at abstractions of polyhedra, e.g. simplicial complexes (Provan and Billera [PB80], Klee and Kleinschmidt [KK87], de Loera and Klee [DLK12]) and connected layer families (Eisenbrand et al. [EHRR10]). While these avenues of analysis seem promising, still the polynomial Hirsch conjecture remains unresolved:

**Conjecture 2.61** (Polynomial Hirsch)**.** *The diameter of a $d$-dimensional polyhedron with $f$ facets is bounded above by a polynomial in $d$ and $f$.*

## 2.4 Circuits and the Circuit Diameter

For these definitions and properties we refer to Borgwardt et al. [BFH15] and Borgwardt et al. [BSY16]. The circuits of a polyhedron are defined as follows:

**Definition 2.62** (Circuit direction)**.** Given a polyhedron

$$P = \{\mathbf{x} \in \mathbb{R}^d : A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2\},$$

with $A^i \in \mathbb{Q}^{m_i \times d}$ and $\mathbf{b}^i \in \mathbb{Q}^{m_i}$ for $i = 1, 2$, the set of *circuit directions*, or *circuits*, $\mathcal{C}(A^1, A^2)$ of $A^1$ and $A^2$ are those vectors $\mathbf{g} \in \ker(A^1)\backslash\{\mathbf{0}\}$ for which $A^2\mathbf{g}$ is support-minimal in $\{A^2\mathbf{x} : \mathbf{x} \in \ker(A^1)\backslash\{\mathbf{0}\}\}$, where $\mathbf{g}$ is normalized to coprime integer components.

Here, $\ker(A^1) = \{\mathbf{x} \in \mathbb{R}^d : A^1\mathbf{x} = \mathbf{0}\}$.

Sometimes, we will define polyhedra as a system of linear inequalities $\{\mathbf{x} : A\mathbf{x} \geq \mathbf{b}\}$ – if there are no equality constraints, then it is not necessary to check the first kernel condition (that is, we look for support-minimal vectors over $\mathbb{R}^d$ itself).

We refer to the circuits of a polyhedron $P$ by $\mathcal{C}(P)$. These were first introduced as *elementary vectors* in (Rockafellar [Roc69]). Note that the definition does not depend on the vectors $\mathbf{b}^1, \mathbf{b}^2$. In fact, $\mathcal{C}(P)$ is exactly the set of *potential edge directions* of $P$ for varying right-hand sides. Here, a vector $\mathbf{g}$ is said to be a potential edge direction of the polyhedron $P$ if:

- It is an actual edge direction of the polyhedron, that is, it arises as the direction vector with an adjacent pair of vertices at its head and tail; or
- It does not arise as such in $P$, but it does in some other polyhedron obtained by changing the right-hand-side values of the constraints of $P$.

This means that $\mathcal{C}(P)$ contains the actual edge directions of $P$.

**Proposition 2.63.** *$\mathcal{C}(A^1, A^2)$ is the set of all edge directions of*

$$P = \{\mathbf{x} \in \mathbb{R}^d : A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2\}$$

*for varying right-hand sides $\mathbf{b}^1, \mathbf{b}^2$.*

A proof of this statement for $P = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ can be found in (Onn et al. [ORT05]).

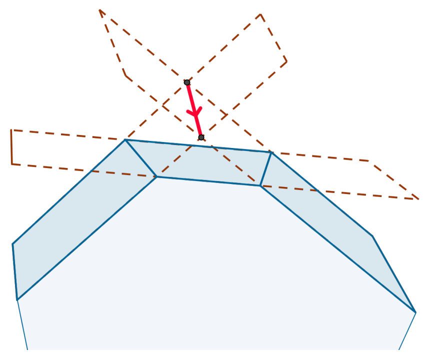

Figure 2.9: The circuits of a polyhedron are its potential edge directions for varying right-hand sides. The vector in red is a circuit direction here because it arises as an edge direction in the polyhedron obtained when the middle facet is translated upwards.

We define a circuit analogue to edge walks on polyhedra:

**Definition 2.64** (Circuit walk)**.** Given a polyhedron $P$, a sequence $\mathbf{y}^{(0)}, \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(k)}$ is a *circuit walk of length $k$* if for all $i = 0, 1, \ldots, k-1$ we have

(i) $\mathbf{y}^{(i)} \in P$,

(ii) $\mathbf{y}^{(i+1)} - \mathbf{y}^{(i)} = \alpha_i \mathbf{g}^{(i)}$ for some $\mathbf{g}^{(i)} \in \mathcal{C}(P)$ and $\alpha_i > 0$, and

(iii) $\mathbf{y}^{(i)} + \alpha_i \mathbf{g}^{(i)} \notin P$ for all $\alpha > \alpha_i$.

Informally, a circuit walk takes steps of maximal length along circuit of $P$. Note that the points are not restricted to be vertices of $P$ (not even the initial and final point). The circuit analogue for distance can also be defined for any two points of the polyhedron, but the circuit diameter only looks at circuit walks from vertex to vertex.

**Definition 2.65** (Circuit distance)**.** Given two points $\mathbf{x}, \mathbf{y} \in P$, the *circuit distance* from $\mathbf{x}$ to $\mathbf{y}$, denoted $\text{dist}_{\mathcal{C}}(\mathbf{x}, \mathbf{y})$, is the smallest number of steps in a circuit walk from $\mathbf{x}$ to $\mathbf{y}$.

**Definition 2.66** (Circuit diameter)**.** Given a polyhedron $P$, the *circuit diameter* of $P$, denoted $\Delta_{\mathcal{C}}(P)$, is the length of the longest shortest circuit walk connecting any two of its vertices. That is,

$$\Delta_{\mathcal{C}}(P) = \max_{\mathbf{x},\mathbf{y} \in V(P)} \text{dist}_{\mathcal{C}}(\mathbf{x}, \mathbf{y}).$$

Note that in Definition 2.65 the order in which $\mathbf{x}$ and $\mathbf{y}$ are written matters – this is because unlike combinatorial distance, the circuit distance is not symmetric. Figure 2.10 illustrates an example (originally from Borgwardt et al. [BFH15]) where $\text{dist}_{\mathcal{C}}(\mathbf{v}^{(1)}, \mathbf{v}^{(4)}) = 2$ while $\text{dist}_{\mathcal{C}}(\mathbf{v}^{(4)}, \mathbf{v}^{(1)}) = 3$ (since at least two more circuit steps are needed to go to $\mathbf{v}^{(1)}$ from any point reachable from $\mathbf{v}^{(4)}$ after the initial step).

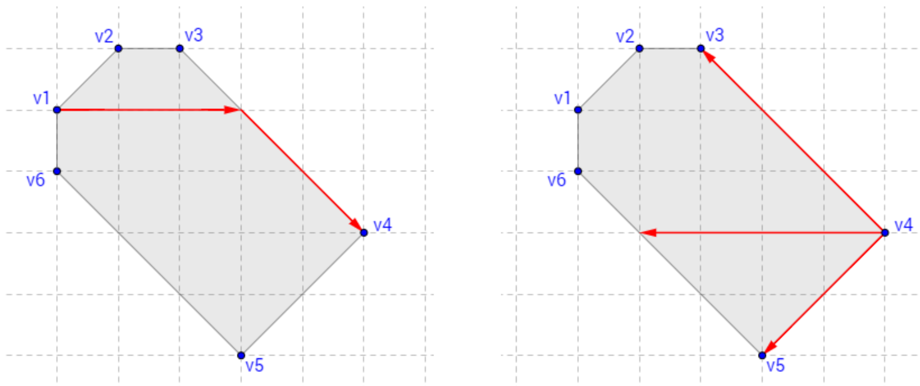


Figure 2.10: Circuit distance is not symmetric.

Furthermore, unlike the combinatorial diameter, which is invariant in a fixed combinatorial class, the circuit diameter of a polyhedron depends on both its representation and its realization in $\mathbb{R}^d$. We would like to work with polyhedra that are given by their unique minimal representations, rather than allow for arbitrary systems of inequalities, where redundant inequalities may give rise to additional circuit directions and reduce the circuit diameter. Figure 2.11 illustrates two combinatorially equivalent polytopes (6-gons) that have different circuit diameters – the one on the left has circuit diameter 3 (since $\text{dist}_{\mathcal{C}}(\mathbf{v}^{(4)}, \mathbf{v}^{(1)}) = 3$) while the one on the right, being regular, has circuit diameter 2.

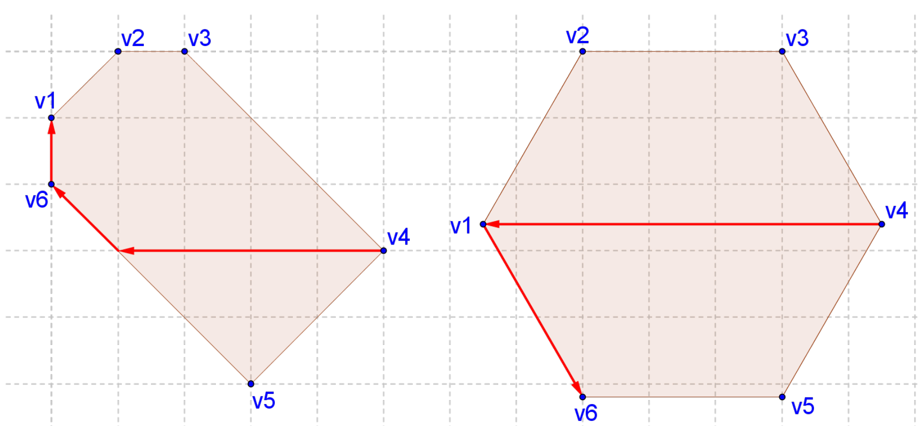


Figure 2.11: Circuit diameter is realization-dependent.

This dependence on the geometry of the polyhedron makes proving statements about its circuit diameter more complicated, as one would have to take into account all possible realizations. However it is still appealing to do so, because of its relationship with the combinatorial diameter.

**Proposition 2.67.** *For any polyhedron $P$, $\Delta_{\mathcal{E}}(P) \geq \Delta_{\mathcal{C}}(P)$.*

*Proof.* This follows immediately from the fact that the edge directions of a polyhedron form a subset of its circuit directions. □

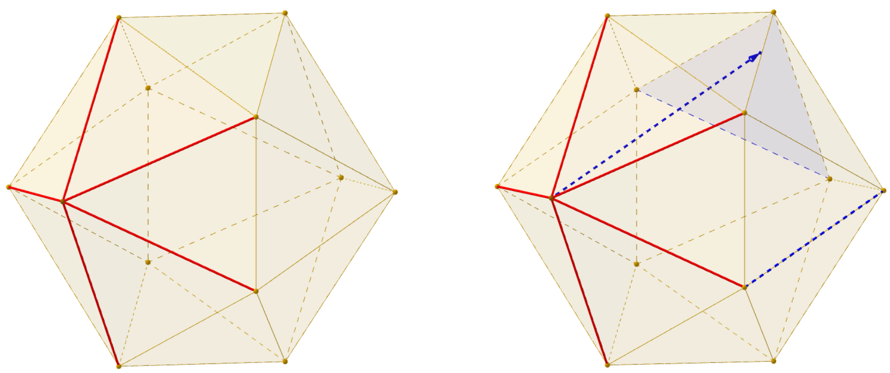

Figure 2.12: Circuits allow for shortcuts through the interior.

Denote by $\Delta_{\mathcal{C}}(f, d)$ the maximum circuit diameter attained by $d$-dimensional polyhedra with $f$ facets. (Also use $\Delta_{\mathcal{C}}^{b}(f, d)$ and $\Delta_{\mathcal{C}}^{u}(f, d)$ for the bounded and unbounded cases.) Then we have:

**Corollary 2.68.** *$\Delta_{\mathcal{E}}(f, d) \geq \Delta_{\mathcal{C}}(f, d)$ for any $f \geq d$.*

In fact, if we drop the maximality condition from the circuit walk definition (Condition (iii) in Definition 2.64), we obtain a third type of walk called *feasible circuit walks*. Using $\Delta_f$ for the diameter arising from these walks, in (Borgwardt et al. [BLF16]) it was proven that $\Delta_f(f, d) \leq f - d$ for all $f \geq d$; that is, the conjecture of Hirsch holds for this type of diameter. Furthermore, $\Delta_f$ is a lower bound on $\Delta_{\mathcal{C}}$, giving the following:

**Proposition 2.69** (Borgwardt et al. [BLF16])**.** *$\Delta_{\mathcal{E}}(f, d) \geq \Delta_{\mathcal{C}}(f, d) \geq \Delta_f(f, d)$ for any $f \geq d$.*

It is interesting to note here that while Hirsch fails for the combinatorial diameter $\Delta_{\mathcal{E}}$, it holds for $\Delta_f$. Hence the circuit diameter $\Delta_{\mathcal{C}}$ marks a point along the Hirsch/non-Hirsch spectrum where it is yet unclear if the conjecture holds.

**Conjecture 2.70** (Circuit Hirsch, Borgwardt et al. [BFH15])**.** *$\Delta_{\mathcal{C}}(f, d) \leq f - d$ for all $f \geq d$.*

If this conjecture were true, then conceptually there is a significant difference between walking along edges and walking along circuits. If it were false, then there is a significant difference between taking steps of maximal length and just staying feasible. This gives credence to the study of the circuit diameter, as an exploration of the reasons that the Hirsch bound is violated by the combinatorial diameter.

## 2.5 Circuit Diameter: Summary of Previous Results

Borgwardt et al. [BFH15] proved an upper bound for the circuit diameter of dual transportation polyhedra that is smaller than the bound for combinatorial diameter in Theorem 2.54. For general dual network flow polyhedra, the best known bound is quadratic.

**Theorem 2.71** (Borgwardt et al. [BFH15])**.** *The circuit diameter of a dual transportation polyhedron $P_{G,\mathbf{c}}$ is bounded above by $M + N - 2$.*

**Theorem 2.72** (Borgwardt et al. [BFH16])**.** *The circuit diameter of a dual network flow polyhedron $P_{G,\mathbf{c}}$ is bounded above by $\frac{|V|(|V|-1)}{2}$.*

Perhaps more interesting is the fact that the set of circuit directions of a polyhedron form a universal *test set* for linear programming. This means that a certificate of optimality at a point is given by the nonexistence of a feasible improving circuit direction from that point. In other words, when each feasible circuit emanating from a point only makes worse the objective value, that point is optimal. Note that the simplex method takes advantage of the fact that incident edge directions at each point also form a test set [for that point].

This has some implications in the development of augmentation algorithms for linear programming. In particular, under certain pivot rules, a simplex-like algorithm to solve linear programs using circuits directions will find the optimum in a polynomial number of augmentations:

**Theorem 2.73** (de Loera et al. [DLHL15])**.** *Let $A \in \mathbb{Z}^{n\times d}$, $\mathbf{b} \in \mathbb{Z}^n$, and $\mathbf{c} \in \mathbb{Z}^d$, and consider the linear program*

$$\min\{\mathbf{c}^T\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \le \mathbf{x} \le \mathbf{u}, \mathbf{x} \in \mathbb{R}^d\}.$$

*Let $\mathbf{x}_0$ be an initial feasible solution, let $\mathbf{x}_{\min}$ be an optimal solution, let $\gamma$ be the maximum non-zero entry (in absolute value) in any feasible solution, and let $\delta$ denote the least common multiple of all subdeterminants of $A$. Then the number of augmentations to reach an optimal solution from $\mathbf{x}_0$ is bounded as follows:*

*(a) The number of discrete deepest-descent augmentations needed to reach an optimal solution from $\mathbf{x}_0$ is bounded by $2d\log(\delta\mathbf{c}^T(\mathbf{x}_0 - \mathbf{x}_{\min}))$.*

*(b) The number of discrete Dantzig-descent augmentations needed to reach an optimal solution from* $\mathbf{x}_0$ *is no more than* $2d^2\delta\gamma\log(\delta\mathbf{c}^T(\mathbf{x}_0 - \mathbf{x}_{\min}))$*.*

*(c) Any discrete steepest-descent direction (which by definition belongs to* $\mathcal{C}(A)$*) is an overall steepest-descent direction (which could be any applicable direction from* $\mathbb{R}^d$*). Moreover, the number of discrete steepest-descent augmentations to each an optimal solution of the given linear program is bounded by* $\mathcal{C}(A)$*.*

Here, *discrete-descent*, *Dantzig-descent*, and *steepest-descent* augmentations refer to specific pivot rules for choosing a circuit to augment along. A corollary of their result is the following bound for the circuit diameter of polyhedra with totally unimodular coefficient matrices:

**Corollary 2.74.** *For a* $n \times d$ *totally-unimodular matrix* $A$*, the circuit diameter of the polyhedron*

$$P := \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\}$$

*is bounded above by* $2d(n+1)(d-n)$*.*

Relations involving the circuit diameter in the vein of Proposition 2.53 are scarce. Again, this is because $\Delta_{\mathcal{C}}(P)$ depends on the geometry of the polyhedron. On the other hand, we prove here that one property *does* transfer:

**Theorem 2.75.** $\Delta_{\mathcal{C}}(f,d) \leq \Delta_{\mathcal{C}}(f+2,d+1)$

Recall from the proof of Proposition 2.53 that the construction $P \times [0,1]$ is used to prove the corresponding statement for combinatorial diameter. To prove it for circuit diameter, we first need to characterize the circuits of a product of two polyhedra:

**Proposition 2.76.** *Let* $P = \{\mathbf{x} \in \mathbb{R}^{d_1} : A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \leq \mathbf{b}^2\}$ *and* $Q = \{\mathbf{x} \in \mathbb{R}^{d_2} : C^1\mathbf{x} = \mathbf{w}^1, C^2\mathbf{x} \leq \mathbf{w}^2\}$ *be pointed polyhedra. Then* $\mathcal{C}(P \times Q) = (\mathcal{C}(P) \times \{\mathbf{0}\}) \cup (\{\mathbf{0}\} \times \mathcal{C}(Q))$*.*

*Proof.* By definition, $\mathcal{C}(P)$ consists of vectors $\mathbf{g}$ that are in $\ker(A^1) = \{\mathbf{g} : A^1\mathbf{g} = \mathbf{0}\}$, and for which $(A^2\mathbf{g})$ is support-minimal in $K_P = \{A^2\mathbf{g} : \mathbf{g} \in \ker(A^1), \mathbf{g} \neq \mathbf{0}\}$. Similarly, $\mathcal{C}(Q)$ are those vectors $\mathbf{h} \in \ker(C^1)$ and such that $(C^2\mathbf{h})$ is support-minimal in $K_Q = \{C^2\mathbf{h} : \mathbf{h} \in \ker(C^1), \mathbf{h} \neq \mathbf{0}\}$.

We would like to characterize the set of circuits $\mathcal{C}(P \times Q)$. Recall that

$$P \times Q = \left\{(\mathbf{x},\mathbf{y}) \in \mathbb{R}^{d_1+d_2} : A^1\mathbf{x} = \mathbf{b}^1, C^1\mathbf{y} = \mathbf{w}^1, A^2\mathbf{x} \leq \mathbf{b}^2, C^2\mathbf{y} \leq \mathbf{w}^2\right\}.$$

For brevity denote by $M_i$ the matrix $\begin{pmatrix} A^i & \mathbf{0} \\ \mathbf{0} & C^i \end{pmatrix}$. Then $\mathcal{C}(P \times Q)$ is composed of vectors $(\mathbf{g},\mathbf{h})^T$ such that $A^1\mathbf{g} = \mathbf{0}$, $C^1\mathbf{h} = \mathbf{0}$, and

$$\begin{pmatrix} A^2 & \mathbf{0} \\ \mathbf{0} & C^2 \end{pmatrix}\begin{pmatrix} \mathbf{g} \\ \mathbf{h} \end{pmatrix} = M\begin{pmatrix} \mathbf{g} \\ \mathbf{h} \end{pmatrix} = \begin{pmatrix} A^2\mathbf{g} \\ C^2\mathbf{h} \end{pmatrix}$$

is support-minimal in the set

$$K_{PQ} = \left\{ M \begin{pmatrix} \mathbf{g} \\ \mathbf{h} \end{pmatrix} : \mathbf{g} \in \ker(A^1), \mathbf{h} \in \ker(A^2), \mathbf{g}, \mathbf{h} \neq \mathbf{0} \right\}.$$

First, we claim that

$$\begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix} \in \mathcal{C}(P \times Q) \iff \mathbf{g} \in \mathcal{C}(P).$$

This follows from the fact that $A^2\mathbf{g}$ is support-minimal over nonzero $\mathbf{g} \in \ker(A^1)$ if and only if

$$\begin{pmatrix} A^2\mathbf{g} \\ \mathbf{0} \end{pmatrix}$$

is support-minimal over nonzero

$$\begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix} \in \ker(M).$$

The pointedness assumption is crucial here. By Proposition 2.9, pointed polyhedra cannot contain a line. This means that there is no nonzero vector $\mathbf{g} \in P$ such that $A^2\mathbf{g} = \mathbf{0}$, that is, $A^2\mathbf{g}$ has nonempty support as long as $\mathbf{g} \neq \mathbf{0}$. Similarly, $C^2\mathbf{h} \neq \mathbf{0}$ for nonzero $\mathbf{h}$.

Hence, given a fixed nonzero $\mathbf{g} \in \ker(A^1)$ for which $A^2\mathbf{g}$ is support-minimal, the support of

$$\begin{pmatrix} A^2\mathbf{g} \\ \mathbf{0} \end{pmatrix}$$

cannot contain the support of

$$\begin{pmatrix} A^2\mathbf{g} \\ C^2\mathbf{h} \end{pmatrix}$$

for any nonzero $\mathbf{h} \in \ker(C^1)$. So,

$$M \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix}$$

is support-minimal over $K_{PQ}$, while also being in $\ker(M)$, implying

$$\begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix} \in \mathcal{C}(P \times Q).$$

Thus $\mathcal{C}(P) \times \{\mathbf{0}\} \subseteq \mathcal{C}(P \times Q)$.

The argument for $\{\mathbf{0}\} \times \mathcal{C}(Q) \subseteq \mathcal{C}(P \times Q)$ is similar. To show that $\mathcal{C}(P \times Q)$ has no other circuits, we only need to prove that if

$$\begin{pmatrix} \mathbf{g} \\ \mathbf{h} \end{pmatrix} \in \ker(M_1)$$

where both $\mathbf{g}$ and $\mathbf{h}$ are nonzero, then

$$\begin{pmatrix} \mathbf{g} \\ \mathbf{h} \end{pmatrix}$$

cannot be a circuit of $P \times Q$. But this is immediate – the support of

$$M_2 \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix}$$

has a strictly smaller support than

$$M_2 \begin{pmatrix} \mathbf{g} \\ \mathbf{h} \end{pmatrix}$$

again because pointedness dictates $C^2\mathbf{h} \neq \mathbf{0}$ for nonzero $\mathbf{h}$. The result follows. □

A consequence of this is a result on the circuit diameter of the product $P \times Q$, the circuit counterpart of Proposition 2.38.

**Proposition 2.77.** $\Delta_{\mathcal{C}}(P \times Q) = \Delta_{\mathcal{C}}(P) + \Delta_{\mathcal{C}}(Q)$.

*Proof.* The previous result implies that when moving along circuit directions in the product $P \times Q$, one's position in one of the factors does not change. Hence, any circuit walk in $P \times Q$ can be decomposed into two circuit walks, one in $P$ and one in $Q$, giving the bound. □

Proposition 2.75 then follows from using the same construction $P \times [0, 1]$ as in Proposition 2.53. It is important to note here that since circuit diameter is dependent on realization, the previous results consider specific realizations for $P$ and $Q$, and hence $P \times Q$ (as opposed to the entire combinatorial class, in the combinatorial diameter setting).

# Chapter 3

# The Circuit Diameter Conjectures

In Theorem 2.52 we gave Klee and Walkup's proof of the equivalences of the Hirsch, non-revisiting, and the $d$-step conjectures. An important first step was to show first that it is enough to work with simple polyhedra, and that any non-simple polyhedron could be transformed into a simple one via a perturbation of facets that does not decrease its diameter. This is not easily transferred to the circuit framework, so in Section 3.1 we introduce a refined notion of simplicity that is used in subsequent proofs.

The advantage of working with a simple polyhedron is that each step along an edge leaves exactly one facet and enters exactly one other facet. In circuit walks, steps along non-edge circuits will leave more than one facet at a time; however, we can still require that circuit steps do not enter more than one facet at a time.

One of the main challenges in this regard is that in the circuit context, different geometric realizations of polyhedra with the same combinatorial structure may have different circuit diameters (see Figure 2.11). Thus the refined notion of simplicity should depend on the geometry of the problem, including the circuits. This is what we explore in Section 3.1.

The rest of the chapter is structured as follows: in Section 3.2 we study how the set of circuit directions is affected by the wedging operation (Definition 2.47) and define wedge-simplicity; in Section 3.3 we reformulate the combinatorial diameter conjectures in the circuit setting, and expound on their interrelationships; in Section 3.4 we show that the circuit $d$-step conjecture is true for $d = 4$ (i.e. $\Delta_{\mathcal{C}}(8, 4) \leq 4$), in contrast to the combinatorial case. We are able to prove this in two ways – showing that the original Klee-Walkup counterexample to unbounded Hirsch remains Hirsch for circuit diameter; and by extending a result on 4-prismatoids (Santos et al. [SST12]).

## 3.1 $\mathcal{C}$-simplicity

In the following, let $P = \{\mathbf{x} \in \mathbb{R}^d : \; A\mathbf{x} \geq \mathbf{b}\}$ be a $d$-dimensional polyhedron.

For $\mathbf{x}^{(i)} \in P$, let $H^{(i)}$ denote the set of facets of $P$ that are incident to $\mathbf{x}^{(i)}$. First, let us introduce some terminology for circuit walks in which we enter only one new facet in each step.

**Definition 3.1** (Simple walks)**.** Let $P$ be a polyhedron. A circuit walk $\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(k)}$ in $P$ is *simple* if $|H^{(i+1)} \backslash H^{(i)}| = 1$ for $i = 1, 2, \ldots, k$, where $H^{(i)}$ denotes the set of facets incident to $\mathbf{x}^{(i)}$. Walks that violate this condition are called *non-simple*.

We are particularly interested in polyhedra for which it suffices to only consider simple circuit walks. As the combinatorial diameter is an upper bound on the circuit diameter of a polyhedron, for the study of Conjecture 2.70, it suffices to consider circuit walks of length at most $\Delta_{\mathcal{E}}(f, d)$, which is bounded above for example by $f^{\log d+2}$ (Klee and Kleinschmidt [KK92]). This leads to the following definition.

**Definition 3.2** ($\mathcal{C}$-simple)**.** Let $P$ be a $d$-polyhedron with $f$ facets. We say $P$ is $\mathcal{C}$-simple if all circuit walks of length at most $\Delta_{\mathcal{E}}(f, d)$ and that start from a vertex of $P$ are simple.

Let $M$ be a finite set of points in $P$ that includes the set of vertices. If all circuit walks starting at any point in $M$ and of length at most $\Delta_{\mathcal{E}}(f, d) + d$ are simple, then we say $P$ is *$\mathcal{C}$-simple with respect to $M$*.

Note that $\mathcal{C}$-simplicity is a strictly stronger condition than simplicity of a polyhedron, as edge walks are a special type of circuit walks. The goal for this section is to prove that it suffices to consider $\mathcal{C}$-simple polyhedra for the study of Conjecture 2.70, which leads to the following variant of the circuit Hirsch conjecture.

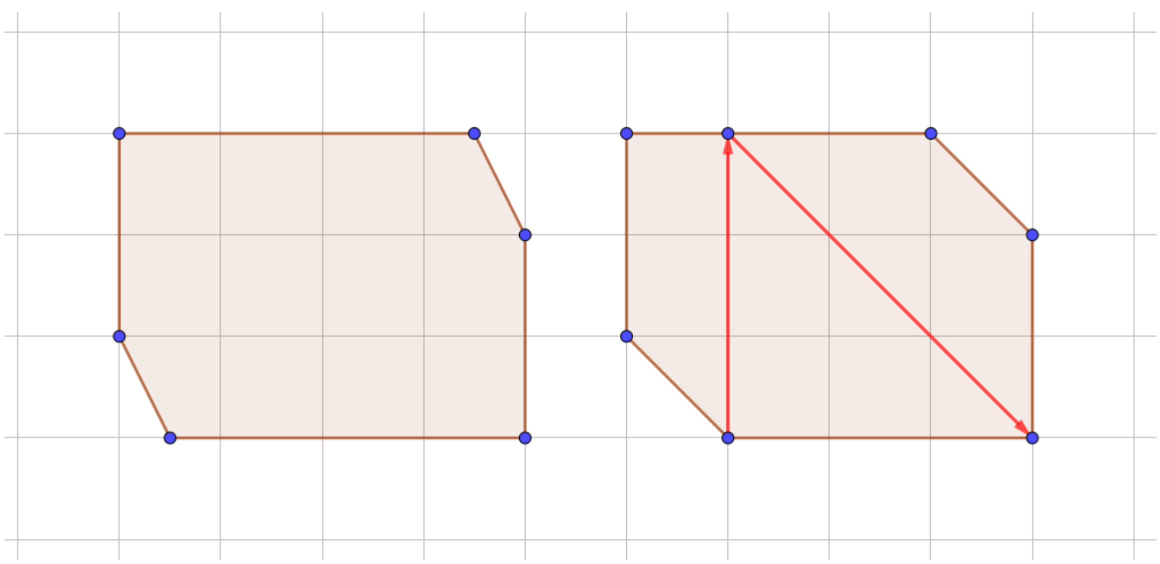

Figure 3.1: The polytope on the left is $\mathcal{C}$-simple – this can be verified by checking that all circuit walks of length at most $\Delta^b_{\mathcal{E}}(6, 2) = 3$ are simple. The polytope on the right is not $\mathcal{C}$-simple because the circuit walk shown enters two facets at the second step.

**Conjecture 3.3** ($\mathcal{C}$-Simplicity)**.** *For any $\mathcal{C}$-simple $d$-dimensional polyhedron with $f$ facets the circuit diameter is bounded above by $f - d$.*

We prove the equivalence of Conjecture 3.3 and Conjecture 2.70 by showing that for fixed $f$ and $d$, $\Delta_{\mathcal{C}}(f, d)$ can be realized by a $\mathcal{C}$-simple polyhedron. We do so by describing a perturbation of a polyhedron $P$ such that the perturbed polyhedron $P'$ is $\mathcal{C}$-simple and has at least the same circuit diameter as $P$.

The perturbations we consider are to the right hand sides of the defining equations, and thus do not change the set of circuits, which depends only on $A$. That is, the right-hand side $\mathbf{b}$ is changed to $\mathbf{b} \to \mathbf{b}' = \mathbf{b} + \mathbf{p}$ for some vector $\mathbf{p}$ with $\|\mathbf{p}\| \leq \delta$ for a sufficiently small $\delta$. We call such a perturbation a *$\delta$-perturbation.* The perturbed polyhedron is

$$P' = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \geq \mathbf{b}'\}.$$

Note that for a $\delta$-perturbation, if $\delta$ is small enough each facet remains a facet and the dimension does not change.

The challenge here lies in the fact that the circuit diameter of a polyhedron depends on its realization, and not only on its combinatorial structure (see Figure 2.11, e.g.). Hence the effect of a perturbation, in theory, might reduce the diameter. We have to check that there is a $\delta$-perturbation for which this is not the case.

**Lemma 3.4.** *Let $P$ be a polyhedron. Then there is a $\mathcal{C}$-simple polyhedron $P'$ in the same dimension and with the same number of facets with $\Delta_{\mathcal{C}}(P) \leq \Delta_{\mathcal{C}}(P')$.*

*Proof.* Let $P$ be a $d$-dimensional polyhedron with $f$ facets. It suffices to prove that a $\delta$ can be determined such that applying a $\delta$-perturbation to $P$ produces a $\mathcal{C}$-simple polyhedron $P'$ with at least the same circuit diameter as $P$.

First, recall that $P$ and any polyhedron produced by a perturbation of $P$ share the same finite set of circuits. Further, observe that it suffices to only consider circuit walks of length at most $\Delta_{\mathcal{E}}(f, d)$ to validate $\mathcal{C}$-simplicity. There are a finite number of points $\mathbf{x} \in P'$ that may appear in such a walk. Hence the condition $|H^{(i+1)} \backslash H^{(i)}| = 1$ only has to be satisfied for a finite set of pairs $(\mathbf{x}^{(i)}, \mathbf{x}^{(i+1)})$. This implies that (for fixed $\delta$) the set of right-hand sides $\mathbf{b}' = \mathbf{b} + \mathbf{p}$ with $\|\mathbf{p}\| \leq \delta$ that do not give a $\mathcal{C}$-simple polyhedron $P'$ is of measure 0. In turn, for any given $\delta$ there are infinitely many perturbations that yield a $\mathcal{C}$-simple polyhedron $P'$.

It remains to see that there is such a perturbation for which the circuit diameter of $P'$ is at least the circuit diameter of $P$. Let $Y_P$ denote the set of all points on circuit walks in $P$ of length at most $\Delta_{\mathcal{E}}(f, d)$. A simple but important observation is that the points in $Y_P$ are at least a certain fixed distance from each other. We say that a pair of points $\mathbf{x}^*, \mathbf{y}^*$ are *close* if

$$\|\mathbf{x}^* - \mathbf{y}^*\| \leq \min_{\mathbf{x}, \mathbf{y} \in Y_P} \tfrac{1}{2}\|\mathbf{x} - \mathbf{y}\|,$$

that is, they are less than half the minimum distance between pairs of points in $Y_P$. Consider then a perturbation for which $\delta$ is small enough that basic solutions in $P'$ remain close to basic solutions in $P$. Take $\mathbf{x} \in P$ and $\mathbf{y} \in P'$. Let $I(\mathbf{x})$ denote the cone of feasible directions of $\mathbf{x}$ with respect to $P$ and $I'(\mathbf{y})$ denote the cone of feasible directions of $\mathbf{y}$ with respect to $P'$. Then we have the following:

(P1) There is a one-to-many correspondence between vertices $\mathbf{x}$ of $P$ and vertices $\mathbf{y}$ of $P'$ where each $\mathbf{y}$ is close to a unique $\mathbf{x}$ and at least one $\mathbf{y}$ is close to a given $\mathbf{x}$ (and possibly many are). In particular, each vertex $\mathbf{y}$ of $P'$ is associated to precisely one close vertex $\mathbf{x}$ in $P$.

(P2) Let $\mathbf{x} \in Y_P$, $\mathbf{y} \in P'$ be close and let $\mathbf{g}$ be a circuit in $I(\mathbf{x}) \cap I'(\mathbf{y})$. Then a step along $\mathbf{g}$ from $\mathbf{y} \in P'$ gives a $\mathbf{y}' \in P'$ that is close to precisely one $\mathbf{x}' \in Y_P$, which is derived from a step along $\mathbf{g}$ from $\mathbf{x} \in P$. (It may also happen that $\mathbf{g}$ is an unbounded direction of $P$, in which case it is also an unbounded direction of $P'$. This is because $P'$ is obtained from $P$ via small translations of facets, and so their recession cones are identical.)

(P3) Let $\mathbf{x} \in Y_P$, $\mathbf{y} \in P'$ be close. Then a step along $\mathbf{g} \in I'(\mathbf{y}) \backslash I(\mathbf{x})$ from $\mathbf{y}$ will give a $\mathbf{y}' \in P'$ that is also close to precisely $\mathbf{x}$.

Let us consider a circuit walk $\mathbf{y}^{(0)}, \dots, \mathbf{y}^{(k')} \in P'$ for $k' \leq \Delta_{\mathcal{E}}(f, d)$. Informally, the above properties tell us that it starts close to a vertex of $P$ (P1) and stays close to points in $Y_P$ in each step (P2, P3). More precisely, each $\mathbf{y}^{(i)}$ is close to precisely one $\mathbf{x}^{(i)} \in Y_P$. If (P2) is valid for $\mathbf{x}^{(i)}, \mathbf{y}^{(i)}$, then $\mathbf{x}^{(i+1)} \neq \mathbf{x}^{(i)}$. Else if (P3) holds, then $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)}$. This implies that each $\mathbf{y}^{(0)}, \dots, \mathbf{y}^{(k')}$ corresponds to a circuit walk $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(k)}$ in $P$ with $k \leq k'$.

Let now $\Delta_{\mathcal{C}}(P) = k$ and let $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(k)}$ be a walk in $P$ realizing the diameter. Further, let $\mathbf{y}^{(0)}$ be a vertex of $P'$ close to $\mathbf{x}^{(0)}$ and $\mathbf{y}^{(k')}$ be a vertex of $P'$ close to $\mathbf{x}^{(k)}$. If $\mathrm{dist}(\mathbf{y}^{(0)}, \mathbf{y}^{(k')}) < k$, then there is a circuit walk $\mathbf{y}^{(0)}, \dots, \mathbf{y}^{(k')}$ and $i, i' \leq k$ such that $\mathbf{y}^{(i')}$ is close to $\mathbf{x}^{(i)}$ for $i' < i$. By the above, we then know that the walk $\mathbf{y}^{(0)}, \dots, \mathbf{y}^{(i')}$ corresponds to a walk $\mathbf{x}^{(0)}, \mathbf{x}'^{(1)}, \dots, \mathbf{x}'^{(i')} = \mathbf{x}^{(i)}$ of length $i' < i$. This implies $\mathrm{dist}(\mathbf{x}^{(0)}, \mathbf{x}^{(k)}) < k$, a contradiction. Thus $\mathrm{dist}(\mathbf{y}^{(0)}, \mathbf{y}^{(k')}) \geq k$, which proves the claim.

Note that the parameter $\delta$ can always be made smaller to ensure properties (P1-P3) hold.

□

Hence we have proven the following:

**Corollary 3.5.** *For any $f > d > 1$, $\Delta_{\mathcal{C}}(f, d)$ is attained by a $\mathcal{C}$-simple $d$-dimensional polyhedron with $f$ facets.*

## 3.2 Wedge-simplicity

The wedge operation also played an important role in the derivation of combinatorial diameter bounds, as well as in the proof of Theorem 2.52. Let us recall Definition 2.47 and the accompanying discussion in Chapter 2:

**Definition 2.47** (Wedge). Let $P$ be a $d$-dimensional polyhedron and let $F$ be a facet of $P$. A *wedge* on $P$ over $F$ is a $(d+1)$-dimensional polyhedron $P' = H^{\leq} \cap (P \times L)$, where $P \times L$ denotes the product of $P$ with $L = [0, \infty)$ and $H^{\leq} \subset \mathbb{R}^{d+1}$ is a closed halfspace with $P \times \{0\} \subset H^{\leq}$ that is defined by a hyperplane $H$ that intersects the interior of $P \times L$ and satisfies $H \cap (P \times \{0\}) = F \times \{0\}$.

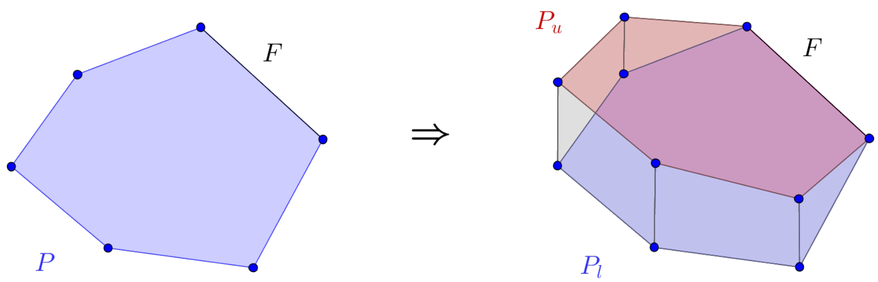


Figure 3.2: The wedge $P'$ on the hexagon $P$ over facet $F$. Bases are $P_l$ and $P_u$.

We again note that we only consider wedges over facets in this thesis, although a more general definition exists where wedges can be taken over lower-dimensional faces. For reference, we include Figure 2.8 here (renumbered as Figure 3.2). The set of circuits of the wedge $P'$, $\mathcal{C}(P')$, contains precisely the normalized vectors in the linear subspaces coming from the intersection of any subset of $d$ linearly independent facets. Recall that the facets of a wedge are: the lower base $P_l$; the upper base $P_u$; and its sides, each defined by supporting hyperplanes of the form $G \times L$, where $G$ is a supporting hyperplane containing a facet of $P$ (other than the facet we wedge over). The following lemma shows that $\mathcal{C}(P')$ can be characterized using $\mathcal{C}(P)$.

**Lemma 3.6** (Circuits of a wedge)**.** *Let $P \subseteq \mathbb{R}^d$ be a $d$-dimensional polyhedron with set of circuits $\mathcal{C}(P)$ and let $F$ be one of its facets. Then if $P'$ is a wedge on $P$ over $F$, the set of circuits $\mathcal{C}(P')$ is comprised of vectors of the form*

*(i)* $(0, 0, \ldots, 0, \pm 1)^T \in \mathbb{R}^{d+1}$

*(ii)* $(\pm \mathbf{g}, 0)^T \in \mathbb{R}^{d+1}$*, where* $\mathbf{g} \in \mathcal{C}(P) \subseteq \mathbb{R}^d$

*(iii)* $\phi\left((\pm \mathbf{g}, 0)^T\right) \in \mathbb{R}^{d+1}$*, where* $\mathbf{g} \in \mathcal{C}(P) \subseteq \mathbb{R}^d$*, and* $\phi(\mathbf{x})$ *is the projection of the vector* $\mathbf{x}$ *to the upper base, moving parallel to* $L$*.*

*Proof.* Each circuit direction of $P'$ is defined by a selection of $d$ facets with linearly independent outer normals.

(i) First, consider the intersection of $d$ sides of $P'$ and recall that they correspond to $d$ facets $G_1, \ldots, G_d$ of $P$. Their intersection in $\mathbb{R}^d$, $G_1 \cap G_2 \cap \cdots \cap G_d$, is a single point $\mathbf{u} \in \mathbb{R}^d$ which may either be in $P$ (in which case it is a vertex), or not in $P$. Hence the intersection of the corresponding sides in $\mathbb{R}^{d+1}$ is

$$(G_1 \times L) \cap \cdots \cap (G_d \times L) = (G_1 \cap \cdots \cap G_d) \times L = \mathbf{u} \times L,$$

which corresponds to the circuit direction $(0, 0, \ldots, 0, \pm 1)^T \in \mathbb{R}^{d+1}$.

(ii) Next, let the lower base $P_l$ be one of the $d$ facets in the intersection. Then the other $d-1$ facets again correspond to facets $G_1, \ldots, G_{d-1}$ of $P$: For the sides, these are the same facets as above; if $P_u$ is one of the facets then the corrsponding facet in $P$ is $F$ (i.e. $G_i = F$ for some $i$). For this, the intersection of these $d$ facets is in one-to-one correspondence with the intersection

$$(G_1 \times L) \cap \cdots \cap (G_{d-1} \times L) \cap P_l = (G_1 \cap \cdots \cap G_{d-1}),$$

which gives a pair of circuits $\pm\mathbf{g} \in \mathcal{C}(P) \subseteq \mathbb{R}^d$. We obtain circuits $(\pm\mathbf{g}, 0)^T \in \mathbb{R}^{d+1}$.

(iii) It remains to consider the intersection of the upper base $P_u$ with $d-1$ sides. (Note that the case with both $P_u$ and $P_l$ in the intersection is already covered above.) These sides correspond to facets $G_1, \ldots, G_{d-1}$ in $P$.

In the wedge $P'$, we get the intersection of facets

$$(G_1 \times L) \cap \cdots (G_{d-1} \times L) \cap P_u = ((G_1 \cap \cdots \cap G_{d-1}) \times L) \cap P_u = \phi(G_1) \cap \cdots \phi(G_{d-1}).$$

Hence their intersection in the upper base $P_u$ is the image of the circuit $\mathbf{g} \in \mathcal{C}(P)$ corresponding to $G_1 \cap \cdots \cap G_{d-1}$ in $P \subseteq \mathbb{R}^d$. That is, we get circuits $\phi\left((\pm\mathbf{g}, 0)^T\right) \in \mathbb{R}^{d+1}$.

□

Wedges are also helpful in the analysis of circuit diameters. However, for circuit walks in wedges, the situation is more involved. Recall from Proposition 2.48 that wedges $P'$ on $P$ over a facet $F$ satisfy $\Delta_{\mathcal{C}}(P) \leq \Delta_{\mathcal{C}}(P') \leq \Delta_{\mathcal{C}}(P) + 1$, because there is an easy correspondence between edge walks in $P$ and edge walks in $P'$. In contrast, circuit walks in $P'$ do not necessarily transfer to circuit walks in $P$. This is because it is possible to hit the interior of $P_u$, as depicted in Figure 3.4: Let a walk begin at a vertex $\mathbf{u} \in P_l \backslash F$. Then take a step along a circuit parallel to $P_u$, which will give a $\mathbf{y}$ in a side of the wedge. Then continue from $\mathbf{y}$ along a circuit parallel to $P_l$ to $\mathbf{y}'$, which may lie in the interior of $P_u$. Projecting this walk down to $P_l$ gives a circuit step in $P$ that does not use maximal step length and is thus not a circuit walk. The point $\mathbf{y}'$ may also be incident to both $P_u$ and

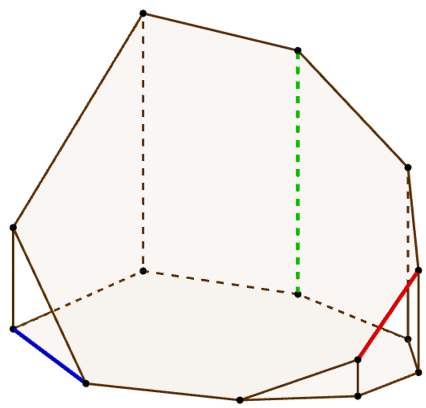

Figure 3.3: The circuits of a wedge – circuits in the lower base (blue), circuits in the upper base (red), and circuits in the $(d+1)$st coordinate direction (green)

a side of the wedge, or lead to another point where this happens, in which case $P'$ is not $\mathcal{C}$-simple – and this is possible even if $P$ itself is $\mathcal{C}$-simple. See Figure 3.5.

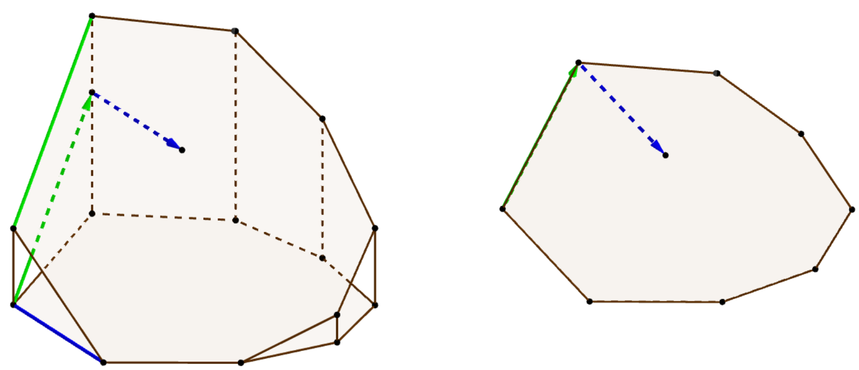

Figure 3.4: A circuit walk in the wedge $P'$ that does not project to a circuit walk in $P$.

Thus the corresponding circuit formulations of Proposition 2.48 do not hold in general. In particular, the wedge operation may reduce the circuit diameter by creating 'shortcuts' between vertices of $P_l$ by walks that take an intermediary step into the interior of $P_u$. In fact, it is possible to construct a polyhedron $P$ and a wedge $P'$ with $\Delta_{\mathcal{C}}(P) > \Delta_{\mathcal{C}}(P')$ (based on the construction for Lemma 15 in Borgwardt et al. [BLF16]).

As we will see, we will not need this property for our arguments. On the other hand, we will require that polyhedra constructed by wedging are $\mathcal{C}$-simple polyhedra (compare Proposition 2.48 (i)). The wedge operation may create a $P'$ that is not $\mathcal{C}$-simple, even if the underlying polyhedron $P$ is $\mathcal{C}$-simple (see Figure 3.5). In the following, we explain how to deal with this problem. First, we require some new terminology that strengthens the $\mathcal{C}$-simple property.

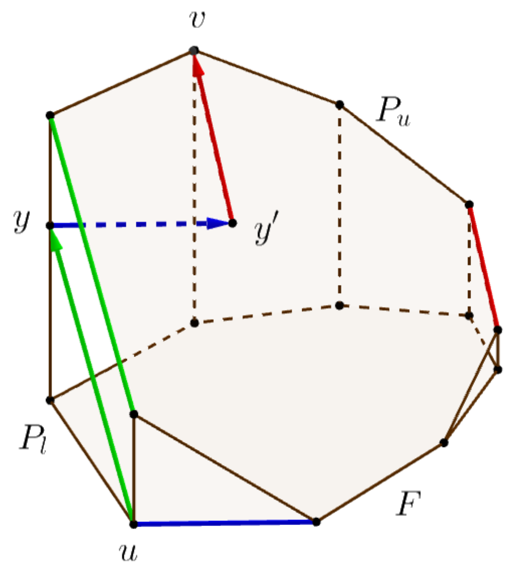


Figure 3.5: Polyhedron $P = P_l$ is $\mathcal{C}$-simple but the wedge $P'$ is not: the walk $\mathbf{u}, \mathbf{y}, \mathbf{y}', \mathbf{v}$ is a non-simple circuit walk in $P'$ because two new facets are seen in the step from $\mathbf{y}'$ to $\mathbf{v}$.

**Definition 3.7** (Wedge-simple)**.** Let $P$ be a $\mathcal{C}$-simple polyhedron. $P$ is a *[1-]wedge-simple polyhedron with respect to a facet* $F$ if there is a wedge $P'$ on $P$ over $F$ that is $\mathcal{C}$-simple. $P$ is *wedge-simple* if for all facets $F$ there is a wedge $P'$ on $P$ over $F$ that is $\mathcal{C}$-simple. We can then recursively define $P$ to be *k-wedge-simple* for $k \geq 2$, if for all facets $F$ there is a wedge $P'$ on $P$ over $F$ that is $(k-1)$-wedge-simple.

With this terminology, we are ready to prove that a $\mathcal{C}$-simple polyhedron can be perturbed to obtain a wedge-simple polyhedron.

**Lemma 3.8.** *Let $P$ be a $\mathcal{C}$-simple polyhedron. Then there is a wedge-simple polyhedron $P^*$ in the same dimension and with the same number of facets with $\Delta_{\mathcal{C}}(P) \leq \Delta_{\mathcal{C}}(P^*)$.*

*Proof.* Let $P' \subseteq \mathbb{R}^{d+1}$ be a wedge on $P \subseteq \mathbb{R}^d$ over facet $F$, and consider a perturbation of the facets of $P'$. Assume without loss of generality that we keep the lower base $P_l$ fixed when perturbing $P'$ – that is, the facet $x_{d+1} \geq 0$ is not translated. When one of the sides of $P'$ is translated, this corresponds to a translation of the corresponding facet in $P$ before wedging. When the upper base $P_u$ is translated, this corresponds to translating the facet $F$ over which we performed the wedging operation. Thus, it is possible to guarantee some properties of $P'$ by perturbing $P$ before wedging.

By Lemma 3.4, if $P'$ is not $\mathcal{C}$-simple we can perturb it to get a $\mathcal{C}$-simple polyhedron with at least the same circuit diameter. Translating the resulting polyhedron so that the lower base remains at $x_{d+1} = 0$, we conclude using the arguments in the previous paragraph that this is equivalent to wedging over a perturbed $P$. Thus it is always possible to perturb an original polyhedron such that the wedge over a given facet $F$ is $\mathcal{C}$-simple.

So now let $P$ be perturbed so that the wedge $P'$ over facet $F$ is $\mathcal{C}$-simple. Further, let $F' \neq F$ be a different facet of $P$ and consider a wedge $P''$ over facet $F'$. If $P''$ is not $\mathcal{C}$-simple, again a perturbation of $P''$ will make it $\mathcal{C}$-simple. This corresponds to another perturbation of $P$, so we need to make sure that $\mathcal{C}$-simplicity of $P'$ is not lost when ensuring it for $P''$.

But note that from the proof of Lemma 3.4, as long as we are performing *$\delta$-perturbations*, that is, perturbations that are sufficiently small (where the $\delta$ depends on the minimum distance between any two points in $Y_P$), we get/retain $\mathcal{C}$-simplicity. Since we only need to consider wedges on $P$ over its facets, of which there is a finite number, we can ensure that $P$ and each wedge constructed stay $\mathcal{C}$-simple by scaling the parameter $\delta$ by some small factor each time. This proves the claim.

□

Additionally, Lemma 3.8 transfers to $k$-wedge-simplicity.

**Corollary 3.9.** *Let $P$ be a $\mathcal{C}$-simple polyhedron and let $k \in \mathbb{N}$ be given. Then there is a $k$-wedge-simple polyhedron $P^*$ in the same dimension and with the same number of facets with $\Delta_{\mathcal{C}}(P) \leq \Delta_{\mathcal{C}}(P^*)$.*

*Proof.* Let $P = P_0, P_1, \ldots, P_k$ be a sequence of wedges $P_{i+1}$ on $P_i$ for $0 \leq i < k$ and suppose $P_k$ is not $\mathcal{C}$-simple. This can be amended by a slight perturbation of $P_k$ (Lemma 3.4), which translates to a perturbation of $P_{k-1}$, which in turn translates to a perturbation of $P_{k-2}$ and so on up to a perturbation of $P = P_0$. Repeated application of the proof of Lemma 3.8 thus gives the claim. □

The above statements sum up to the equivalence of the following conjecture to the original formulation in Conjecture 2.70.

**Conjecture 3.10** (Wedge-Simplicity)**.** *For any $k$-wedge-simple $d$-dimensional polyhedron with $f$ facets and any $k \in \mathbb{N}$ the circuit diameter is bounded above by $f - d$.*

In light of the many applications of wedging over polyhedra in the studies of the combinatorial diameter, Conjecture 3.10 may be useful in its own right. It tells us that we may restrict our studies to $\mathcal{C}$-simple polyhedra for which an arbitrarily large number of wedging operations still gives a $\mathcal{C}$-simple polyhedron.

We conclude this section by explaining how $k$-wedge-simplicity transfers from a polyhedron $P$ to its faces.

**Lemma 3.11.** *Let $P$ be a wedge-simple $d$-dimensional polyhedron and $F$ be any $d'$-face of $P$ for $1 < d' < d$. Then $F$ is also wedge-simple.*

*Proof.* Let $F$ be a $d'$-dimensional face of the wedge-simple polyhedron $P$. In order to prove that $F$ is wedge-simple, we need to show that there is a $\mathcal{C}$-simple wedge on $F$ over each of its facets. To this end let $G$ be a facet of $F$. We know that $G = F \cap \tilde{G}$, where $\tilde{G}$ is a facet of $P$. By wedge-simplicity of $P$, there is a wedge $W$ on $P$ over $\tilde{G}$ that is $\mathcal{C}$-simple. Recalling Definition 2.47, let $H^{\leq}$ be the closed halfspace intersected with $P \times L = P \times [0, \infty)$ to produce the wedge $W$, and let $H$ be the defining hyperplane for $H^{\leq}$ in $\mathbb{R}^{d+1}$. In particular this means that $H \cap (P \times \{0\}) = \tilde{G} \times \{0\}$; that is, the hyperplane intersects the lower base $P_l$ in the facet $\tilde{G}$. Hence we have:

$$\begin{aligned} H \cap (F \times \{0\}) &= H \cap ((P \times \{0\}) \cap (F \times \{0\})) \\ &= (H \cap (P \times \{0\}) \cap (F \times \{0\}) \\ &= (\tilde{G} \times \{0\}) \cap (F \times \{0\}) \\ &= G \times \{0\} \end{aligned}$$

So we can use the same halfspace $H^{\leq}$ intersected with $F \times L$ to produce a wedge $W_F$ on $F$ over $G$ that is contained in $W$.

Further $W_F$ is a face of $W$. To see this, let $H_F \subset \mathbb{R}^d$ be a supporting hyperplane of $P$ that intersects $P$ exactly in $F$: $H_F \cap P = F$. Then we have the following implications:

$$\begin{aligned} H_F \cap P &= F \\ \Rightarrow (H_F \times L) \cap (P \times L) &= (F \times L) \\ \Rightarrow (H_F \times L) \cap (P \times L) \cap H^{\leq} &= (F \times L) \cap H^{\leq} \\ \Rightarrow (H_F \times L) \cap W &= W_F \\ \Rightarrow (H_F \times \mathbb{R}) \cap W &= W_F \quad \text{(since } W, W_F \subseteq \mathbb{R}^d \times L\text{)} \end{aligned}$$

Clearly $W \subseteq P \times L$ is contained in $H_F^{\leq} \times \mathbb{R}$, since $P$ is contained in $H_F^{\leq}$. This means that $W_F$ is a face of $W$, with supporting hyperplane $H_F \times \mathbb{R}$. Circuit walks in $W_F$ are therefore also circuit walks in $W$, and so if $W$ is $\mathcal{C}$-simple, then $W_F$ must be $\mathcal{C}$-simple as well. Since the facet $G$ of $F$ was selected arbitrarily at the beginning, the proof works for any facet; this proves that $F$ is wedge-simple. □

**Lemma 3.12.** *Let $P$ be a $k$-wedge-simple $d$-dimensional polyhedron and $F$ any $d'$-face of $P$ for $1 < d' < d$. Then $F$ is also $k$-wedge-simple.*

*Proof.* Following the proof of Lemma 3.11, let $F$ be a $d'$-dimensional face of the $k$-wedge-simple polyhedron $P$, and let $G$ be a facet of $F$. Call $\tilde{G}$ the facet of $P$ such that $G = F \cap \tilde{G}$, $W$ the wedge on $P$ over $\tilde{G}$, and $W'$ the wedge on $F$ over $G$, constructed using the same closed halfspace as $W$. Then we saw above that $W'$ is a $(d'+1)$-face of $W$.

This means that we can proceed with this proof using induction on $k$. The base case, $k = 2$, is handled immediately by Lemma 3.11: If $P$ is 2-wedge-simple, then $W$ is wedge-simple, and thus so is $W'$, implying $F$ is 2-wedge-simple by the arbitrary choice of $G$.

The same argument can be used to prove the inductive step: if it is true for $(k-1)$, then using the same constructions, if $P$ is $k$-wedge-simple, then $W$ is $(k-1)$-wedge-simple, and thus so is $W'$. Since $G$ was chosen arbitrarily at the beginning, this implies that $F$ itself is $k$-wedge-simple. □

## 3.3 The Conjectures

In this section we discuss how several variants of Conjecture 2.70 are related – these are circuit analogues of variants of the combiantorial Hirsch conjecture. We begin with a discussion and formal statements for these conjectures.

### 3.3.1 Non-revisiting circuit walks

One of the most useful variants in the studies of the Hirsch conjecture for the combinatorial diameter is the *non-revisiting conjecture*. A walk is *non-revisiting* if no facet is left during the walk then entered again at a later step; the non-revisiting conjecture was that any two vertices are connected by such a walk. In particular this means that if two vertices $\mathbf{u}, \mathbf{v}$ lie in the same face of a polyhedron, there would be a walk connecting the two vertices that stays in this face and adheres to the given bound.

In an edge walk in a simple polyhedron, at each step exactly one facet is left and exactly one other facet is entered. The non-revisiting conjecture requires that the entered facet is new, i.e. it has not been left before. For circuit walks the situation is a bit more complicated. Circuit steps that are not along edges will leave multiple facets simultaneously, and, in a non-$\mathcal{C}$-simple polyhedron, may enter multiple facets as well. To transfer the concept of a non-revisiting walk, we consider the facets that are entered during a walk.

Recall that the connection of the non-revisiting conjecture for edge walks and the original Hirsch conjecture, in fact, comes from the 'positive' interpretation of the above: One enters exactly one new facet in each step. As one starts at a vertex, to which $d$ facets are incident, entering a new facet in each step immediately gives the bound $f - d$. We use this interpretation to give a viable formulation for circuit walks.

**Conjecture 3.13** (Non-revisiting)**.** *For any polyhedron $P$ and two vertices $\mathbf{u}, \mathbf{v} \in P$, there is a circuit walk from $\mathbf{u}$ to $\mathbf{v}$ that enters a new facet in each step, that is, each step produces an active facet that was inactive at all previous steps.*

In a non-revisiting circuit walk, each circuit step enters at least one new facet, and may leave any number of old facets. So generally, it is possible to enter 'old' facets in a step as long as one also enters a new facet. However, for $\mathcal{C}$-simple polyhedra, only exactly one new facet is entered in each step. Then the above formulation is equivalent to asking for a circuit walk from $\mathbf{u}$ to $\mathbf{v}$ that does not enter a facet it left before.

It is easy to see that Conjecture 3.13 implies Conjecture 2.70 with the same arguments as before: One begins at a vertex, to which (at least) $d$ facets are incident, and enters a new facet in each step, which gives a bound of $f - d$.

### 3.3.2 Any start

In contrast to edge walks, which only walk between vertices of a polyhedron, one may consider circuit walks that begin at a feasible point in a polyhedron that is not a vertex: indeed many steps in circuit walks do not begin at vertices. To study partial circuit walks, we present a variant of the conjectures that deals with the generalization of the starting point. Recall that a set of facets is linearly independent if the corresponding outer normals are linearly independent.

**Conjecture 3.14** (Any start)**.** *For any $d$-dimensional polyhedron $P$ with $f$ facets and any finite set $M$ of points in $P$, the length of a circuit walk from any point $\mathbf{u} \in M$ to any vertex in $P$ is bounded above by $f - d'$, where $d'$ is the number of linearly independent facets active at $\mathbf{u}$.*

Note the number of active facets for any vertex $\mathbf{u}$ of a $d$-dimensional polyhedron is at least $d$. The above conjecture gives rise to a generalized notion of circuit walk that may start anywhere in the polyhedron, not only at a vertex.

Let us briefly look at an example to see why the above formulation is plausible: Consider a simplex in $\mathbb{R}^d$ and recall it has $d+1$ vertices and $d+1$ facets, which implies $f - d = 1$. It has combinatorial diameter 1, which transfers to circuit diameter 1. However, the number of steps to a vertex can be much larger for non-vertices. For example, let a walk begin at a feasible point $\mathbf{u}$ in the strict interior of a facet $F$. Then $d' = 1$ and $f - d' = d$. If the simplex is $\mathcal{C}$-simple with respect to a set $M$ containing the starting point $\mathbf{u}$, walking to the unique vertex not incident to $F$ requires exactly $d$ steps. Figure 3.6 depicts an example in dimension 3.

Consider a non-revisiting walk (as in Conjecture 3.13) starting at a non-vertex. Picking up a new facet in each step that did not appear before would transfer to a bound of $f - d'$, as only $d' \leq d$ facets are active in the beginning. It is clear that Conjecture 3.14 is at least as strong as Conjecture 2.70, as it encompasses the corresponding statement for $M$ as the set of all vertices and because $d' \geq d$ for all vertices. But Conjecture 3.13 also implies Conjecture 3.14:

**Lemma 3.15.** *Let $P$ be a $d$-dimensional polyhedron with $f$ facets, let $\mathbf{u}$ be a feasible point in $P$ that is incident to $d'$ linearly independent facets of $P$, and let $\mathbf{v}$ be a vertex of $P$. Suppose further that the non-revisiting conjecture (Conjecture 3.13) is true. Then there is a circuit walk from $\mathbf{u}$ to $\mathbf{v}$ that enters a new facet in each step.*

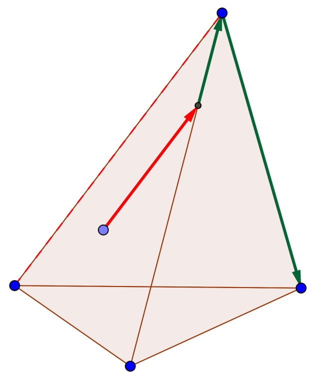

Figure 3.6: A simplex in $\mathbb{R}^3$. The circuit distance of a point on the boundary to any vertex is at most three.

*Proof.* Our strategy for the proof is as follows: We construct a polyhedron $P' \subset P$ such that $\mathbf{u}, \mathbf{v}$ are vertices of $P'$. Then we show that a non-revisiting circuit walk from $\mathbf{u}$ to $\mathbf{v}$ in $P'$ transfers to a non-revisiting circuit walk in $P$.

Let $F_1^v, \dots, F_d^v$ be $d$ linearly independent facets incident to $\mathbf{v}$ in $P$ with outer normals $\mathbf{a}_1^v, \dots, \mathbf{a}_d^v$. Let further $F_1^u, \dots, F_{d'}^u$ be $d'$ linearly independent facets incident to $\mathbf{u}$ in $P$ and let $\mathbf{a}_1^u, \dots, \mathbf{a}_{d'}^u$ be the corresponding outer normals. For $d' \geq d$, there is nothing to prove. For $d' < d$, there are $d - d'$ outer normals among $\mathbf{a}_1^v, \dots, \mathbf{a}_d^v$, without loss of generality $\mathbf{a}_1^v, \dots, \mathbf{a}_{d-d'}^v$, such that the set $\{\mathbf{a}_1^u, \dots, \mathbf{a}_{d'}^u, \mathbf{a}_1^v, \dots, \mathbf{a}_{d-d'}^v\}$ is linearly independent.

Let now $F_{d'+i}^{\geq} = \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{a}_i^v)^T \mathbf{x} \geq (\mathbf{a}_i^v)^T \mathbf{u}\}$ for $i \leq d - d'$. Note that by definition of the $\mathbf{a}_i$, $(\mathbf{a}_i^v)^T \mathbf{v} \geq (\mathbf{a}_i^v)^T \mathbf{u}$ for all $i \leq d - d'$. Thus $P' = P \cap \bigcap_{i=1}^{d-d'} F_i^{\geq}$ contains both $\mathbf{u}$ and $\mathbf{v}$, and both are vertices of $P'$. Informally $P'$ is the intersection of $P$ with a cone (starting at $\mathbf{u}$) of facets that are parallel to facets incident to $\mathbf{v}$.

By assumption of Conjecture 3.13 there is a walk from vertex $\mathbf{u}$ to vertex $\mathbf{v}$ in $P'$ that is non-revisiting. As $P$ and $P'$ only differ in facets incident to the starting point $\mathbf{u}$ of the walk, this implies that in each step of such a walk there is a facet from the original polyhedron $P$ that bounds the step length. Thus the corresponding walk is also a circuit walk in $P$ and it is non-revisiting in $P$. $\square$

### 3.3.3 Dantzig figures and the circuit $d$-step conjecture

Next, we consider the connection of Conjecture 2.70 to the so-called $d$-step conjecture and Dantzig figures. It is well known that for the maximal combinatorial diameter of $d$-dimensional polyhedra, it suffices to consider polyhedra with $2d$ facets. The maximal value of $f - d$ is realized in a polyhedron with $2d$ facets. This leads to the circuit equivalent of the $d$-step conjecture.

**Conjecture 3.16** ($d$-step)**.** *For any $d$-dimensional polyhedron with $2d$ facets the circuit diameter is bounded above by $d$.*

With $f = 2d$, clearly $f - d = d$. Conjecture 3.16 treats a special class of polyhedra, and thus is a specialization of Conjecture 2.70. In fact, one may even restrict the studies to Dantzig figures (Definition 2.51).

**Conjecture 3.17** (Dantzig figure)**.** *For any $d$-dimensional Dantzig figure and vertices $\mathbf{u}, \mathbf{v}$ not sharing a facet, the circuit distance from $\mathbf{u}$ to $\mathbf{v}$ is bounded above by $d$.*

Note that for a $\mathcal{C}$-simple Dantzig figure, the circuit distance from $\mathbf{u}$ to $\mathbf{v}$ is at least $d$. If it is equal to $d$, then a corresponding walk is non-revisiting.

### 3.3.4 Equivalence of the Conjectures

We now prove a sequence of implications relating Conjectures 2.70, 3.13, 3.16, and 3.17. The proof methods are inspired by the corresponding proof for the combinatorial diameter in (Yemelichev et al. [YKK84]) and (Klee and Walkup [KW67]), but we have to pay significantly more attention to technical detail. Moreover we do not recover the last implication (4) ⇒ (1); we explain why later in this section.

**Theorem 1.8.** *Consider the following statements:*

*(1) Let $\mathbf{u}, \mathbf{v}$ be two vertices of a $k$-wedge-simple polyhedron $P$ for $k \geq f$. Then there is a non-revisiting circuit walk from $\mathbf{u}$ to $\mathbf{v}$.*

*(2) $\Delta_{\mathcal{C}}(f, d) \leq f - d$ for all $f \geq d$*

*(3) $\Delta_{\mathcal{C}}(2d, d) \leq d$ for all $d$*

*(4) For all $d$-dimensional Dantzig figures $(P, \mathbf{u}, \mathbf{v})$, the circuit distance from $\mathbf{u}$ to $\mathbf{v}$ is at most $d$.*

*Then (1) ⇒ (2) ⇒ (3) ⇒ (4).*

*Proof.* (1) ⇒ (2): First, recall that $\Delta_{\mathcal{C}}(f, d)$ is realized by a $k$-wedge-simple polyhedron. In a non-revisiting circuit walk from vertex $\mathbf{u}$ to $\mathbf{v}$, a new facet is entered in each step. As $P$ is $k$-wedge-simple, it in particular is $\mathcal{C}$-simple so this is exactly one new facet per step. Since $\mathbf{u}$ is incident to exactly $d$ facets, this walk has at most $f - d$ steps.

(2) ⇒ (3): $\Delta_{\mathcal{C}}(2d, d) \leq d$ is the special case of $\Delta_{\mathcal{C}}(f, d) \leq f - d$ for $f = 2d$.

(3) ⇒ (4): The $d$-dimensional Dantzig figure $(P, \mathbf{u}, \mathbf{v})$ in particular has $2d$ facets. □

We note here that this result mirrors Klee and Walkup's proof of the equivalences in Theorem 2.52, except for the last implication (4) ⇒ (1). The combinatorial version of this implication is proven by wedging on a starting polyhedron $F_0$, wedging over the result, and repeating this process until a Dantzig figure $F_k$ is obtained. Then the edge walk in the Dantzig figure is non-revisiting, and it projects down to a non-revisiting edge walk in $F_0$.

The idea of $k$-wedge-simplicity is introduced because it is necessary to ensure $\mathcal{C}$-simplicity of the final Dantzig figure $P'$ produced; this and the circuit diameter bound from the assumption (4) gives a non-revisiting circuit walk in the Dantzig figure. However this might not project down to a non-revisiting circuit walk in the intermediate wedge constructions, or the starting polyhedron itself – see Figure 3.7.

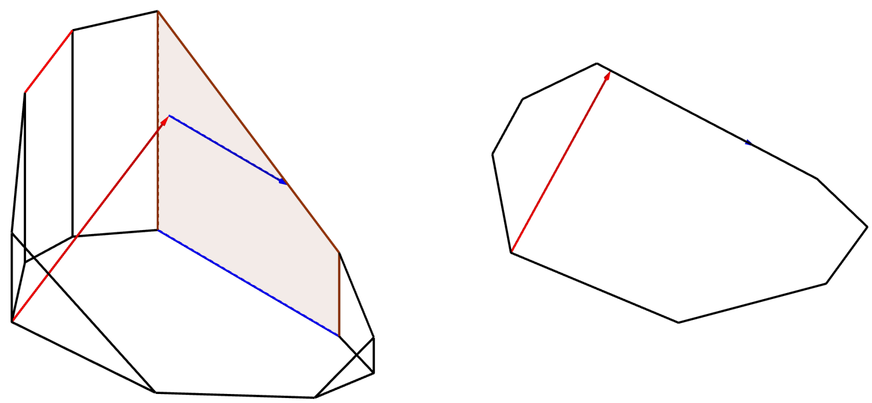

Figure 3.7: The circuit walk in the wedge $P'$ on the left is non-revisiting. However it does not project to a circuit walk in $P$.

We can still guarantee existence of a non-revisiting circuit walk from $\mathbf{x}$ to $\mathbf{y}^k$ in the final Dantzig figure $F_k$, however. By $k$-wedge-simplicity of the starting polyhedron $F_0$, we can wedge on $F_0$ such that $F_1$ is $(k-1)$-wedge-simple. In general there is a choice of wedge such that $F_i$ is $(k-i)$-wedge-simple, and hence the Dantzig figure $F_k$ will be $\mathcal{C}$-simple, with the property that $\mathrm{dist}_{\mathcal{C}}(\mathbf{x}, \mathbf{y}^k) = d' + k$. So, any circuit walk of length $d' + k$ from $\mathbf{x}$ to $\mathbf{y}^k$ has to be non-revisiting – the number of facets incident to $\mathbf{y}^k$ and the number of steps in the walk are equal.

Where problems may arise are in the intermediate wedges $F_i$ for $i < k$. Note that this method could still work – if there were some way to control these non-revisiting walks such that at the circuit step when the upper base is entered, the circuit direction traversed was vertical (that is, it left [only] the lower base and entered the upper base). Such a circuit walk in $F_i$ would project down to a circuit walk in $F_{i-1}$ with one fewer step, and so the same argument as in the previous paragraph will guarantee that this walk is non-revisiting (using strong induction on $i$ from $k$ down to 0).

### 3.3.5 A connection of unbounded and bounded circuit diameters

In the proof of Lemma 3.15, we added extra facets incident to a boundary point $\mathbf{u}$ of $P$ to obtain a polyhedron $P' \subset P$ in which $\mathbf{u}$ is a vertex. The added facets were parallel to existing facets, such that $\mathcal{C}(P) = \mathcal{C}(P')$. By performing a similar construction it is possible to transform an unbounded polyhedron $P$ to a bounded polytope $P'$ with $\mathcal{C}(P) = \mathcal{C}(P')$ and without cutting off any vertices. We will do so by adding facets through $\mathbf{u}$ that are parallel to facets through a particular vertex $\mathbf{v}$. In doing so, we derive a relationship between the circuit diameters of bounded polytopes and unbounded polyhedra.

In the following, for a facet with outer normal $\mathbf{a}_i$, we call a facet with outer normal $-\mathbf{a}_i$ an *opposite facet*. We begin by examining how many opposite facets have to be added to a single vertex of an unbounded polyhedron $P$ to obtain a bounded polytope $P'$. Here we say a facet *blocks* an edge direction incident to vertex $\mathbf{v}$ if the half-line starting at $\mathbf{v}$ in edge direction intersects the facet. Essentially, an edge direction only is unbounded if there is no blocking facet.

**Lemma 3.18.** *Let $P$ be a $d$-dimensional unbounded polyhedron with $f$ facets, and let $\mathbf{u}, \mathbf{v}$ be two vertices of $P$ that do not share a facet. Then there is a bounded polyhedron $P' \subset P$, where $P'$ is constructed by adding at most $d-1$ facets incident to $\mathbf{u}$ that are opposite facets for facets incident to $\mathbf{v}$. By this construction, $\mathbf{u}, \mathbf{v} \in P'$ are vertices and $\mathcal{C}(P) = \mathcal{C}(P')$.*

*Proof.* By construction, $P'$ still contains both $\mathbf{u}$ and $\mathbf{v}$, as all facets that are added are opposite facets of facets that are incident to $\mathbf{v}$. In particular, the facets are parallel to existing ones, so $\mathcal{C}(P) = \mathcal{C}(P')$. Thus it suffices to prove that at most $d-1$ facets are necessary to create a bounded polytope.

Consider the recession cone of $P$, or its cone of unbounded directions (Definition 2.17). In particular, it is contained in the inner cone of $\mathbf{v}$. Thus it suffices to block the edge directions of the inner cone in $P'$ by the addition of extra facets. A simple way to do so would be to add opposite facets incident to $\mathbf{u}$ to *all* facets incident to $\mathbf{v}$. In fact, this would give a bounded box that contains $P'$.

However, not all of these facets are necessary. Let $S$ be a simple cone coming from the selection of exactly $d$ linearly independent facets of the $d^* \geq d$ facets incident to $\mathbf{v}$. Then $S$ contains the inner cone and it suffices to block the corresponding $d$ edge directions of $S$.

The graph of $P$ is connected and thus there is an edge incident to $\mathbf{v}$ that leads to a neighbouring vertex. It is possible to choose the facets $F_1, \ldots, F_d$ for $S$ as a superset of those $d-1$ facets that define such an edge. This implies that we only have to block $d-1$ edge directions to validate the claim. In fact, if we add to $\mathbf{u}$ exactly the opposite facets to these $d-1$ facets then we obtain the desired result.

$\square$

Note that the number of facets that are necessary for the construction may be lower than $d-1$ if the unbounded cone is of lower dimension than $d$ or if multiple edges lead from $\mathbf{v}$ to neighbors in the graph of $P$. Lemma 3.18 is our key ingredient to connect the maximal circuit diameter $\Delta^u_{\mathcal{C}}(f,d)$ of an unbounded $d$-dimensional polyhedron with $f$ facets and the maximal circuit diameter $\Delta^b_{\mathcal{C}}(f,d)$ of a bounded $d$-dimensional polytope with $f$ facets.

**Theorem 3.19.** *If all $\mathcal{C}$-simple bounded $(f', d')$-polytopes with $f' \leq f + d - 1$ and $d' \leq d$ satisfy the non-revisiting conjecture (Conjecture 3.13), then $\Delta^u_{\mathcal{C}}(f, d) \leq f - 1$.*

*Proof.* Let $P$ be an unbounded $d$-dimensional polyhedron with $f$ facets and let $\mathbf{u}, \mathbf{v}$ be two of its vertices. We assume $\mathbf{u}$ and $\mathbf{v}$ do not share a facet; otherwise consider the minimal face that contains $\mathbf{u}$ and $\mathbf{v}$ in place of $P$. Further, let $P'$ be constructed as in the proof of Lemma 3.18, where new facets are added to $\mathbf{u}$. Then $P'$ has at most $f + (d - 1)$ facets. Now perturb polytope $P'$ to obtain the $\mathcal{C}$-simple polytope $P''$. Let $\mathbf{u}'$ be a vertex of $P''$ that corresponds to $\mathbf{u}$ in the non-perturbed $P'$, and to which all (at most $d-1$) extra facets are incident. (The existence of such a $\mathbf{u}'$ in $P''$ can be guaranteed by first fixing a set of $d$ facets that contains all extra facets and relaxing all other facets slightly. Any subsequent perturbation then keeps the single point of intersection $\mathbf{u}'$ of these facets feasible, i.e. $\mathbf{u}'$ is a vertex of $P''$.) Let further $\mathbf{v}'$ be a vertex of $P''$ that corresponds to $\mathbf{v}$ in $P'$.

Now consider a non-revisiting circuit walk from $\mathbf{u}'$ to $\mathbf{v}'$ in $P''$. As it is non-revisiting, none of the extra facets is the only blocking facet in any step – in other words the step length is always bounded by one of the original facets. This means that the circuit walk transfers to a circuit walk from $\mathbf{u}$ to $\mathbf{v}$ in $P$, with the same number of steps. The circuit walk thus has length at most $f - 1$, as none of the initial $d$ facets is revisited.

□

Note that the constructed walk for the unbounded polyhedron $P$ may be revisiting. An interesting special case arises for Dantzig figures. For this case, Theorem 3.19 can be refined.

**Corollary 3.20.** *If all $d$-dimensional bounded spindles $P(\mathbf{u}, \mathbf{v})$ with $2d - 1$ facets incident to $\mathbf{u}$ and $d$ facets incident to $\mathbf{v}$ have a non-revisiting circuit walk from $\mathbf{u}$ to $\mathbf{v}$ of length at most $d$, then Conjecture 3.17 is true in dimension $d$, even for unbounded polyhedra.*

Note the non-revisiting condition in the above corollary is needed to transfer a circuit walk in the bounded polytope to be a circuit walk in the unbounded polyhedron. In dimension 4, all spindles have length at most 4 (Santos et al. [SST12]). Showing that such a walk can be realized in a non-revisiting manner gives the circuit 4-step conjecture for bounded and unbounded polyhedra. In the following section, we give two proofs of the circuit 4-step conjecture: first by a careful analysis of the Klee-Walkup polyhedron $U_4$ and second via Corollary 3.20 by showing that 4-spindle walks can in fact be made non-revisiting.

## 3.4 The Circuit 4-step Conjecture

### 3.4.1 Proof via the Klee-Walkup polyhedron

The first unbounded counterexample to the Hirsch conjecture was given by Klee and Walkup in [KW67], where they constructed a 4-dimensional polyhedron with 8 facets and combinatorial diameter 5. Stephen and Yusun in [SY15] prove that this polyhedron satisfies the Hirsch bound in the circuit diameter setting. We detail the proof here, and then consider the more general 4-step conjecture afterwards.

Denote by $U_4$ the polyhedron defined by the system of linear inequalities $\{\mathbf{x} \in \mathbb{R}^4 : A\mathbf{x} \geq \mathbf{b}\}$, where

$$A = \begin{pmatrix} -6 & -3 & 0 & 1 \\ -3 & -6 & 1 & 0 \\ -35 & -45 & 6 & 3 \\ -45 & -35 & 3 & 6 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} -1 \\ -1 \\ -8 \\ -8 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Its graph is shown in Figure 3.8: here the vertices are indexed by the four facets containing each one, while the points labelled with `R`'s represent extreme rays. It is clear from the graph that vertices `V5678` and `V1234` are at graph distance five apart.

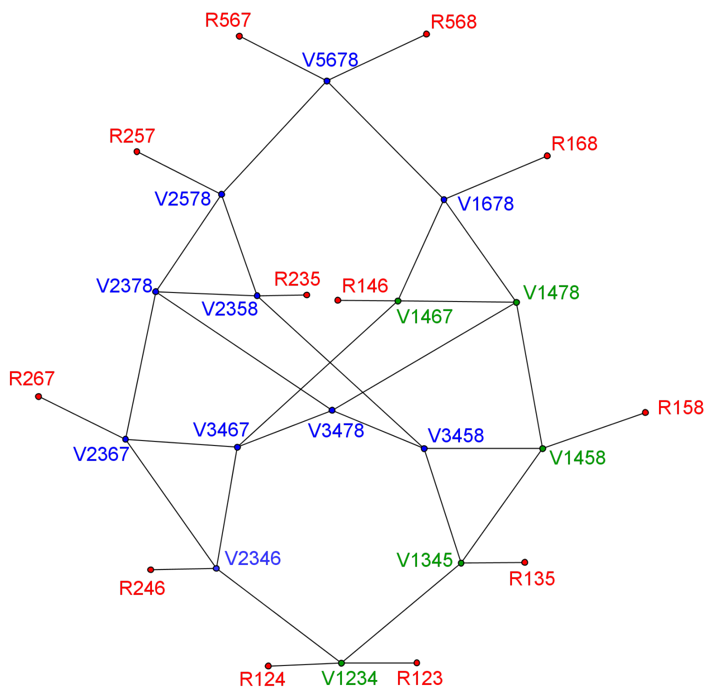


Figure 3.8: The graph of $U_4$.

**Theorem 3.21.** *The circuit diameter of the Klee-Walkup polyhedron $U_4$ is at most 4, independent of realization.*

*Proof.* First we demonstrate the existence of a circuit walk of length 4 from `V5678` to `V1234`. Observe that we can take two edge steps as follows: `V5678` $\rightarrow$ `V1678` $\rightarrow$ `V1478`. Vertices `V1478` and `V1234` are both contained in the 2-face determined by facets `1` and `4`, so we can complete the walk on this face. Note that this 2-face is an unbounded polyhedron on six facets. Figure 3.9 is a topological illustration of this face, showing the order of the vertices and rays.

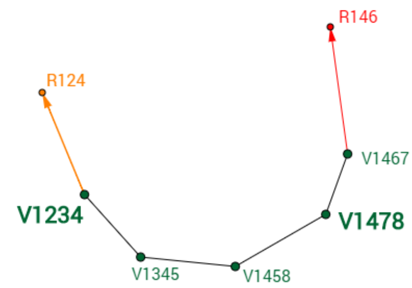


Figure 3.9: The 2-face determined by facets `1` and `4`.

Now consider a vector $\mathbf{g}$ corresponding to the edge direction from `V1458` to `V1345` – this is the blue vector in Figure 3.10. Note that this is always a circuit direction in any realization of $U_4$ since it corresponds to an actual edge of the polyhedron.

To see that $\mathbf{g}$ is a feasible direction at `V1478`, consider vector $\mathbf{h}$ in the edge direction from `V1478` to `V1458`, and vector $\mathbf{r}$ in the direction of ray `R124`. Observe that $\mathbf{g}$ and $-\mathbf{h}$ are the two incident edge directions at `V1458`, and so $\mathbf{r}$ must be a strict conic combination of $\mathbf{g}$ and $-\mathbf{h}$, i.e. $\mathbf{r} = \alpha_1(\mathbf{g}) + \alpha_2(-\mathbf{h})$ for $\alpha_1, \alpha_2 > 0$. By rearranging terms we see that $\mathbf{g}$ is a strict conic combination of $\mathbf{h}$ and $\mathbf{r}$: $\mathbf{g} = (\alpha_2/\alpha_1)\mathbf{h} + (1/\alpha_1)\mathbf{r}$, with $\alpha_2/\alpha_1, 1/\alpha_1 > 0$. Feasibility of $\mathbf{r}$ and $\mathbf{h}$ at `V1478` implies that $\mathbf{g}$ is a feasible direction at `V1478`.

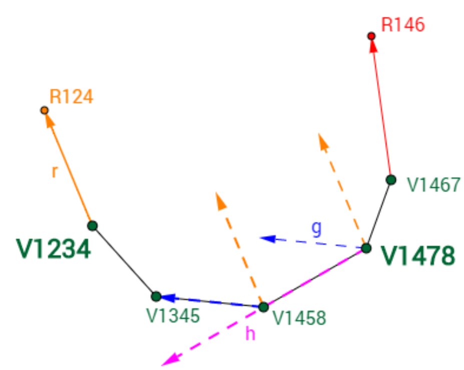


Figure 3.10: Feasibility of the circuit direction $\mathbf{g}$.

Now starting at `V1478` traverse $\mathbf{g}$ as far as feasibility allows. This direction is bounded since we eventually exit the polyhedron when following $\mathbf{g}$ from `V1458`. We will hit the 2-face at a point along the boundary, and at one of the following positions:

- exactly at `V1234`,
- on the edge connecting `V1234` and `V1345`, or
- on the ray `R124` emanating from `V1234`.

Hitting exactly `V1234` gives a circuit walk of length 3 from `V5678`, while the other two cases give circuit walks of length 4 since we only need one step to `V1234`. These two situations are illustrated in Figure 3.11.

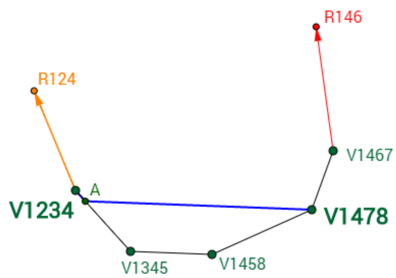


Figure 3.11: Getting from `V1478` to `V1234` in at most 2 steps.

The argument is the same for the reverse direction (`V1234` to `V5678`). We can construct a similar walk by first traversing edges `V1234` $\to$ `V2346` $\to$ `V3467`, and then taking a maximal step in the circuit direction arising from the edge connecting `V1467` and `V1678`. Here we stay in the 2-face determined by facets `6` and `7`. We can then arrive at `V5678` in at most two steps from `V3467`. $\square$

One consequence of Theorem 3.21 is the general circuit 4-step conjecture, however we will need the following fact to prove it:

**Lemma 3.22.** *Up to isomorphism, $U_4$ is the only non-Hirsch simple (8,4)-polyhedron.*

*Proof.* The simple bounded (8,4)-polytopes are enumerated in Grünbaum and Sreedharan [GS67]. A simple unbounded (8,4)-polyhedron can be truncated with an additional facet that cuts off the vertex at infinity, this produces a simple (9,4)-polytope. Observe that the diameter of this polytope will remain at least 5, as any route between `V1234` and `V5678` through the new facet will have to add the new ninth facet along with facets 5, 6, 7 and 8. Klee and Kleinschmidt in [KK87] mention that there is a unique simple polytope with $d = 4$, $f = 9$ and diameter 5, following directly from the complete enumeration of all polytopal simplicial 3-spheres, completed by Altshuler et al. [ABS80]. Thus this must be exactly that polytope, which we denote by $Q_4$. The result follows, as any non-Hirsch 4-polyhedron with 8 facets comes from $Q_4$ by projecting to infinity the ninth facet that does not contain either of the two vertices at distance 5. $\square$

Polytopes such as $Q_4$ that satisfy $\Delta_{\mathcal{E}}(P) = f - d$ are known as *Hirsch-sharp* polytopes. Now we can prove the circuit 4-step conjecture:

**Theorem 1.9 (Circuit 4-step).** $\Delta_{\mathcal{C}}(8, 4) = 4$.

*Proof.* By Lemma 3.4, it suffices to consider $\mathcal{C}$-simple polyhedra – let $P$ be a 4-dimensional polyhedron with 8 facets. If $P$ is bounded then it has combinatorial diameter 4 (Santos et al. [SST12]), so suppose $P$ is unbounded. By Lemma 3.22, $P$ is combinatorially equivalent to $U_4$, and by Theorem 3.21 it has circuit diameter 4. $\square$

### 3.4.2 Proof via facial paths in 4-prismatoids

Here we present a second proof of Theorem 1.9. Recall that a *spindle* is a polytope with two distinguished vertices $\mathbf{u}$ and $\mathbf{v}$ such that each facet is incident to exactly one of $\mathbf{u}$ and $\mathbf{v}$. Polar to this, a *prismatoid* is a polytope with two disjoint facets $Q^+$ and $Q^-$ (called its *bases*) that together contain all the vertices of the polytope. So edge walks between the two distinguished vertices in a spindle correspond to facial paths between $Q^-$ and $Q^+$, that is, a path that moves from facet to facet via $(d-2)$-dimensional faces.

The *length* of a spindle is the graph distance between the two vertices $\mathbf{u}$ and $\mathbf{v}$, while the *width* of a prismatoid is the dual graph distance between the two bases, that is, the length of the shortest facial path between $Q^-$ and $Q^+$.

These constructions were essential in finding counterexamples to the combinatorial Hirsch conjecture; the starting point of Santos's non-Hirsch construction was a 5-dimensional prismatoid (see Santos [San12], and Matschke et al. [MSW15] for an improvement). On the other hand, Santos et al. ([SST12]) proved that 4-dimensional prismatoids have width at most 4, implying that $d = 5$ is the lowest possible dimension one could start this construction.

We strengthen this result here by showing that at each step of the facial path from $Q^+$ to $Q^-$, a new vertex of $Q^-$ is seen (where *new* here means it is new for that particular step, and not necessarily for the whole walk). Back in the spindle, this is an edge walk from $\mathbf{u}$ to $\mathbf{v}$ such that at each step in the walk, a new (with respect to that step) facet of $\mathbf{v}$ is entered. Note that this is different from the usual nonrevisiting property for edge walks, which requires entered facets to be new with respect to the entire walk up to that point.

**Lemma 3.23.** *In a 4-prismatoid with parallel faces $Q^+$ and $Q^-$, there exists a facial path from $Q^+$ to $Q^-$ such that at each step at least one new vertex of $Q^-$ is encountered.*

*Proof.* Suppose a 4-prismatoid $Q$ is given, with bases $Q^+$ and $Q^-$. We know that $Q$ has width at most 4 (Santos et al. [SST12]). If $Q$ has width 2 then there is a facet of $Q$ that is adjacent to both bases. The claim then follows as the number of vertices of $Q^-$ incident to each facet in the facial path is strictly increasing.

If $Q$ has width 3, suppose the facial path of length 3 is $Q^+ \to F \to G \to Q^-$. Then $F$ must have at either 1 or 2 vertices from $Q^-$ – any more and it would itself be adjacent to $Q^-$, and there would be a shorter path between the bases. Also, $G$ must have at least 3 vertices in common with $Q^-$ to be adjacent to it. Hence the number of vertices of $Q^-$ incident to each facet in this facial path is also strictly increasing.

Suppose now that $Q$ has width 4. Santos et al. ([SST12]) prove that there is a facial path of length 4 between the bases, say $Q^- \to F_1 \to F_2 \to F_3 \to Q^+$, such that $F_2$ is a tetrahedron with $|V(Q^-) \cap V(F_2)| = 2$ and $|V(Q^+) \cap V(F_2)| = 2$. That is, $F_2$ has two vertices on $Q^-$ and two vertices on $Q^+$. Otherwise, if $F_2$ were incident to more than two vertices of say, $Q^+$, then it would be adjacent to $Q^+$ and we would have a shorter facial path between the bases, contradicting the assumption that $Q$ has width 4. Denote by $\mathbf{x}$ and $\mathbf{y}$ ($\mathbf{z}$ and $\mathbf{w}$) the two vertices of $Q^+$ ($Q^-$) on $F_2$.

Let us now consider each step of the path. The first step $Q^+ \to F_1$ and the last step $F_3 \to Q^-$ clearly satisfy the condition we require. Moreover, going from $F_2$ to $F_3$, the number of vertices on $Q^-$ increases from 2 to at least 3 – a strict increase as well.

As for the step from $F_1$ to $F_2$, the crucial observation is that the triangle of $F_2$ that is incident to $F_1$ contains two vertices of $Q^+$ ($\mathbf{x}$ and $\mathbf{y}$) and one of $Q^-$ (assume without loss of generality that it is $\mathbf{z}$). This means that $F_1$ cannot contain $\mathbf{w}$ as well, or else $F_1$ would contain $F_2$ entirely. Therefore $\mathbf{w}$ is the new vertex of $Q^-$ seen when moving from $F_1$ to $F_2$.

□

In Figure 3.12 we show an example of a facial path in a 4-prismatoid; this figure forms part of a *Schlegel diagram*, which is a way of visualizing a 4-dimensional polytope by projecting it through a point beyond one of its facets, which results in a polytopal subdivision in 3-space.

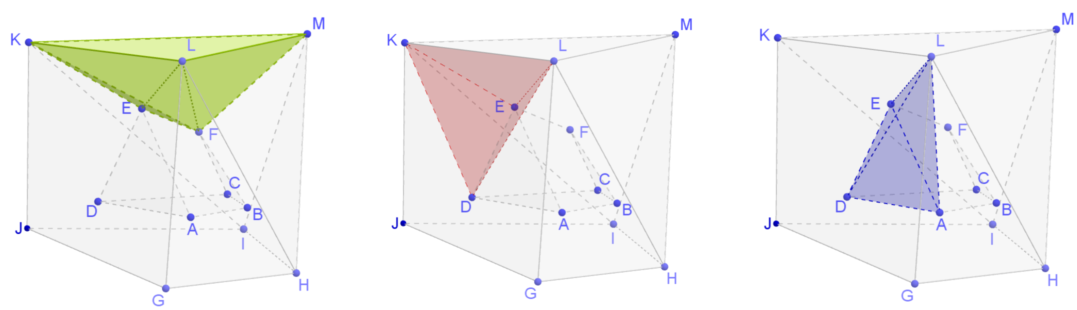


Figure 3.12: Intermediate steps in a facial path of length 4 between the two bases of a prismatoid, visualized as a partial Schlegel diagram.

The two bases are the polytopes with vertex sets $V(Q^+) = \{G, H, I, J, K, L, M\}$ and $V(Q^-) = \{A, B, C, D, E, F\}$. The facial path shown walks from the outer facet $Q^+$ to the inner facet $Q^-$ as follows:

- Step 1: $Q^+$ to the facet $F_1$ with vertices $V(F_1) = \{K, L, M, E, F\}$ via the 2-face $\triangle KLM$.
- Step 2: $F_1$ to the tetrahedron $F_2$ with vertices $V(F_2) = \{K, L, E, D\}$ via $\triangle KLE$.
- Step 3: $F_2$ to the facet $F_3$ with vertices $V(F_3) = \{L, E, D, A\}$ via $\triangle LED$.
- Step 4: $F_3$ to $Q^-$ via $\triangle EDA$.

Figure 3.13 shows the entire facial path; observe that this is a *revisiting* path since the vertex $F$ is left at the second step and then seen again in the last step. However this path still satisfies the condition we need, that at least one new vertex of $Q^-$ is seen *at each step*: $E$ and $F$ for step 1, $D$ for step 2, $A$ for step 3, and $B$, $C$, and $F$ for step 4 (we still list $F$ here because it is a vertex of $Q^-$ but not of $F_3$, although it has already appeared in the walk before).

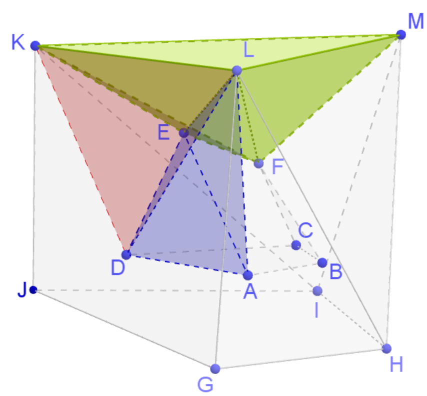


Figure 3.13: A revisiting facial path in a 4-prismatoid.

The polar result for spindles is that there is a length 4 edge walk from $\mathbf{u}$ to $\mathbf{v}$ such that a new facet of $\mathbf{v}$ is entered at each step. This implies the next result:

**Corollary 3.24.** *Let $P(\mathbf{u}, \mathbf{v}) = P(\mathbf{u}) \cap P(\mathbf{v}) \subset \mathbb{R}^4$ be a bounded spindle coming from the intersection of two cones $P(\mathbf{u}), P(\mathbf{v})$ at $\mathbf{u}$, respectively $\mathbf{v}$. Then there is an edge walk from $\mathbf{u}$ to $\mathbf{v}$ such that in each step, a new facet of $P(\mathbf{v})$ becomes active.*

**Lemma 3.25.** *Let $P(\mathbf{u}, \mathbf{v}) = P(\mathbf{u}) \cap P(\mathbf{v}) \subset \mathbb{R}^4$ be an unbounded spindle coming from the intersection of two cones $P(\mathbf{u}), P(\mathbf{v})$ at $\mathbf{u}$, respectively $\mathbf{v}$. Let further $P(\mathbf{v})$ be simple. Then there is a circuit walk of length at most 4 from $\mathbf{u}$ to $\mathbf{v}$.*

*Proof.* Let $\mathbf{a}_1, \dots, \mathbf{a}_4$ be the outer normals of facets $F_1, \dots, F_4$ incident to $\mathbf{v}$. Let further $Q_i = \{\mathbf{x} \in \mathbb{R}^d : (-\mathbf{a}_i)^T\mathbf{x} \leq (-\mathbf{a}_i)^T\mathbf{u}\}$, informally an opposite halfspace of the one created by $F_i$, but now moved to be incident to $\mathbf{u}_i$. Set $P'(\mathbf{u}) = P(\mathbf{u}) \cap Q$ with $Q = \bigcap_{i=1}^{4} Q_i$ and

$P'(\mathbf{u},\mathbf{v}) = P'(\mathbf{u}) \cap P(\mathbf{v})$. Clearly, $P'(\mathbf{u},\mathbf{v}) \subset P(\mathbf{u},\mathbf{v})$ is a bounded spindle with simple cone $P(\mathbf{v})$ so that we may apply Corollary 3.24.

Thus there is a circuit walk from $\mathbf{u}$ to $\mathbf{v}$ in $P'(\mathbf{u},\mathbf{v})$ of length at most 4 such that in each step at least one of the facets of $P(\mathbf{v})$ becomes active. This means that the 'extra' facets introduced as $Q$ are never the only facets to bound the step length. Combining this fact with $P'(\mathbf{u},\mathbf{v}) \subset P(\mathbf{u},\mathbf{v})$, we see that the circuit walk from $\mathbf{u}$ to $\mathbf{v}$ in $P'(\mathbf{u},\mathbf{v})$ of length at most $f$ is a circuit walk in $P(\mathbf{u},\mathbf{v})$, as well. This proves the claim. □

The bounded case $\Delta_{\mathcal{C}}(8,4) \leq 4$ already follows from the combinatorial diameter bound $\Delta_{\mathcal{E}}(8,4) \leq 4$; hence Lemma 3.25 takes care of the only other possible bad case – when the $(8,4)$-polyhedron is unbounded. So, given an unbounded $(8,4)$-polyhedron, we take two of its vertices $\mathbf{u}$ and $\mathbf{v}$, which we can assume to have no facets in common (otherwise we reduce to the 3- or fewer-dimensional case.)

An application of Lemma 3.25 gives Theorem 1.9.

# Chapter 4

# Computational Methods

In this chapter we detail some of the computational methods used in our exploration of circuit directions and circuit diameters, as well as a number of auxiliary results that do not fit in Chapter 3. We begin with a discussion of the Hirsch-sharp polytope $Q_4$ obtained from $U_4$ by truncating the vertex at infinity. We prove an analogue to Theorem 3.21 – that a circuit distance bound of 4 holds for the far pair of vertices of $Q_4$, for at least one direction. Then, we give two representations of $Q_4$ with different circuit diameters – for this we mainly used MATLAB and polymake, though we also wrote code in python that would carry out limited polyhedral computations (since polymake does not run on all systems).

We discuss differences between the inequality form $A\mathbf{x} \geq \mathbf{b}$ and the standard form $A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ presentation for polyhedra, and their implications on circuit computations. We prove that an inequality-form polyhedron with nonnegativity constraints can be rewritten in standard form (using slack variables) while retaining the same circuits. On the other hand, adding new inequalities to a polyhedron will always result in more circuits that originally existed – e.g. recall the classic linear programming exercise of writing unconstrained variables $x$ as $x^+ - x^-$.

Finally, we end with a short proof that the circuit directions of a certain polyhedron arise as the vertices of another, hinting at another possible method to compute circuits efficiently.

## 4.1 The Klee-Walkup Polytope

By applying a projective transformation to $U_4$ we can obtain a bounded 4-dimensional polytope with $f = 9$ and diameter 5. As in (Kim and Santos [KS10]), we call this polytope $Q_4$. Since its diameter is exactly $5 = f - d = 9 - 4$, $Q_4$ is *Hirsch-sharp* – this is the smallest Hirsch-sharp polytope outside of cubes and tetrahedra, and is the basis for the constructions used to produce more Hirsch-sharp polytopes (Holt and Klee [HK98], Fritsche and Holt [FH99]). With this in mind, we investigate the circuit diameter of $Q_4$.

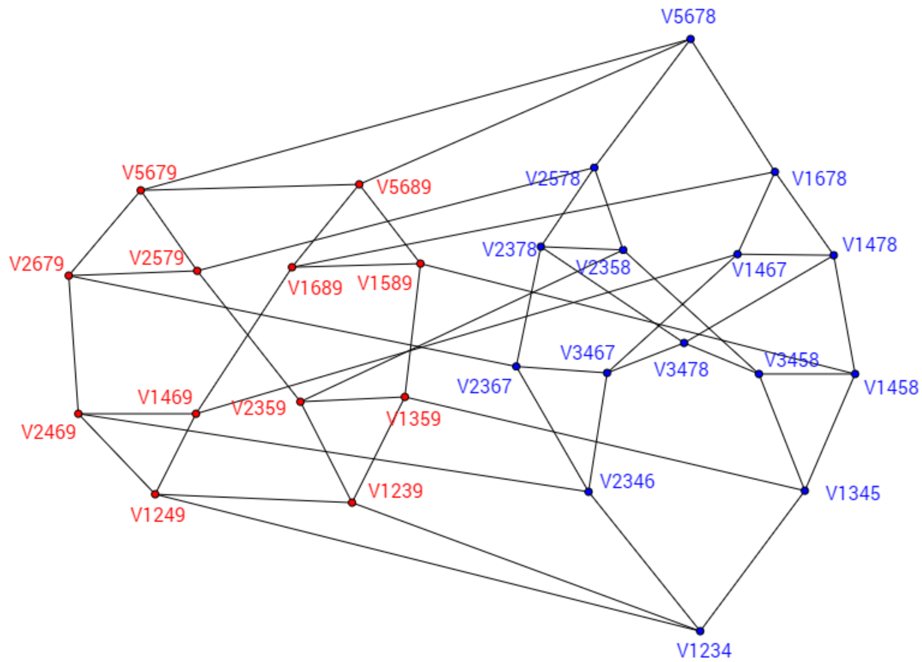


Figure 4.1: The skeleton of $Q_4$.

The graph of $Q_4$ is shown in Figure 4.1. Here the red vertices, those containing the number `9`, are the ones obtained by adding the ninth facet to $U_4$. Note that we follow the same vertex labeling as in Figure 2.7. It is easy to see that $Q_4$ has graph diameter 5: starting at `V5678` and using only blue vertices, we stay in $U_4$, and `V1234` is at distance 5; if we move to a red vertex at any point, we will need at least 4 other steps to introduce facets `1`, `2`, `3`, and `4`. It is easy to see that all other pairs of vertices are connected by walks of length at most 4.

The arguments given in the proof of Theorem 3.21 cannot be applied to $Q_4$ because we lose unboundedness. That is, in the 2-face determined by facets `1` and `4` (which is now a bounded 7-gon), we cannot anymore be certain that we can get to `V1234` in two steps from `V1478` (or `V1467`, which is at distance 2 from `V5678`). Figure 4.2 illustrates an example where we need three steps to arrive at `V1234` coming from either `V1478` or `V1467`.

It is interesting to note, however, that the non-existence of a 2-walk from `V1478` to `V1234` in this face implies the existence of a 2-walk from `V1234` to `V1478`, and hence a circuit walk of length 4 from `V1234` to `V5678` (see Figure 4.3). As a consequence we are able to attain a walk of distance 4 between `V1234` and `V5678` *for at least one direction*, irrespective of representation.

**Theorem 4.1.** *In any realization of $Q_4$, if $\mathbf{u}$ and $\mathbf{v}$ are the two vertices such that $\operatorname{dist}_{\mathcal{E}}(\mathbf{u},\mathbf{v}) = 5$, then either $\operatorname{dist}_{\mathcal{C}}(\mathbf{u},\mathbf{v}) \leq 4$ or $\operatorname{dist}_{\mathcal{C}}(\mathbf{v},\mathbf{u}) \leq 4$.*

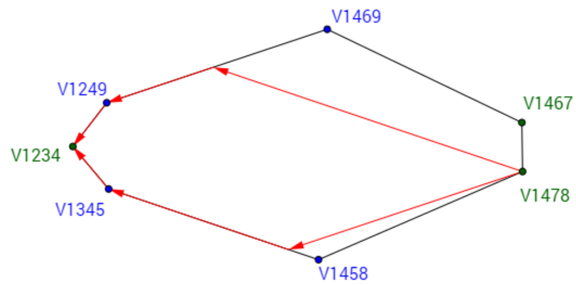


Figure 4.2: At least three steps are needed to go from `V1478` to `V1234`.

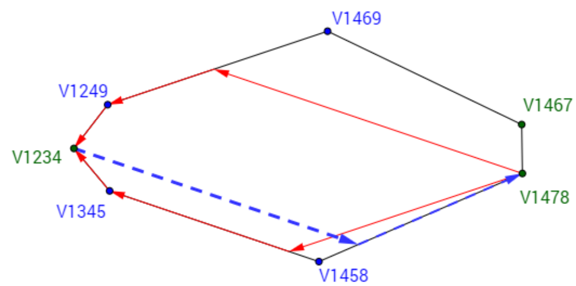


Figure 4.3: Starting at `V1478`, moving parallel to the edge `V1458-V1345` one hits the edge `V1469-V1249`. Traversing the same circuit in the opposite direction from `V1234`, we end up on the edge between `V1458` and `V1478`, and one more step is sufficient to arrive at `V1478`.

We speculate that no representation of $Q_4$ is circuit Hirsch-sharp, i.e. that $\Delta_{\mathcal{C}}(Q_4) < 5$ for all representations. Below we show one representation of $Q_4$ that has circuit diameter 3, and another that has circuit diameter 4.

### 4.1.1 A Representation of $Q_4$ with Circuit Diameter 3

We first consider the representation of $Q_4$, given by the facet description $\tilde{Q}_4 = \{\mathbf{x} \in \mathbb{R}^4 : A\mathbf{x} \geq \mathbf{b}\}$ from [KS09]. This representation has several symmetries in its coefficients:

$$A = \begin{pmatrix} 3 & -3 & -1 & -2 \\ -3 & 3 & -1 & -2 \\ -2 & 1 & -1 & -3 \\ 2 & -1 & -1 & -3 \\ -3 & -3 & 1 & -2 \\ 3 & 3 & 1 & -2 \\ 1 & 2 & 1 & -3 \\ -1 & -2 & 1 & -3 \\ 0 & 0 & 0 & 2 \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}.$$

Its $f$-vector is $(27, 54, 36, 9)$, and its graph diameter is 5 (refer to Figure 4.1 for its graph). Other than `V1234` and `V5678`, all other pairs of vertices are connected by paths of lengths at most 4.

By brute force enumeration, we are able to generate a circuit walk of length 2 from vertex `V1234` to vertex `V5678`, that uses the circuits $\mathbf{g}^0 = (-1, 0, -3, 0)^T$ and $\mathbf{g}^1 = (1, 0, -3, 0)^T$. This shows that the circuit diameter of $\tilde{Q}_4$ is less than 5.

Further, by an exhaustive search we find circuit walks of length 3 between every other pair of vertices. Some of these pairs admit a walk of length 2 while some pairs do not. Hence $\tilde{Q}_4$ has circuit diameter 3. See Section 4.2 for more details of this computation.

### 4.1.2 A Representation of $Q_4$ with Circuit Diameter 4

We are able to use the same methods to explore the circuit diameters of perturbed, less symmetric versions of $Q_4$, with the goal of finding representations with different diameters. For instance, consider the realization $\tilde{Q}_4' = \{x \in \mathbb{R}^4 : C\mathbf{x} \geq \mathbf{b}\}$, where

$$C = \begin{pmatrix} 3.2 & -3 & -1 & -2 \\ -3 & 3.2 & -1 & -2 \\ -2 & 1 & -1 & -3 \\ 2 & -1 & -1 & -3 \\ -3 & -3 & 1.05 & -2 \\ 3 & 3 & 1.05 & -2 \\ 1.05 & 2 & 1 & -3 \\ -1 & -2.05 & 1 & -3 \\ 0 & 0 & 0 & 2 \end{pmatrix} = A + \begin{pmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0.05 & 0 \\ 0.05 & 0 & 0 & 0 \\ 0 & -0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

and $A$ and $\mathbf{b}$ are as defined in the standard representation. (A quick check is done in polymake to verify the isomorphism and generate the vertices.) Vertices `V5678` and `V1234` of $\tilde{Q}_4'$, while still separated by a distance-5 path along the skeleton of the polytope, are now at circuit distance 4 apart. Therefore $\tilde{Q}_4'$ has circuit diameter exactly 4, as there are multiple pairs of vertices separated by circuit distance 4. This brings to light the fragility of the circuit diameter with respect to geometric realization.

## 4.2 Computational Details

We performed these circuit diameter computations using polymake [GJ00], MATLAB [mat14], and Maple [map14]. Given the representation $\{\mathbf{x} \in \mathbb{R}^4 : A\mathbf{x} \geq \mathbf{b}\}$ of $Q_4$, we first use polymake to compute its vertices, then we compute circuit directions in MATLAB. Since each $4 \times 4$ submatrix of $A$ is nonsingular, we are able to compute all vectors $\mathbf{g}$ such that $A\mathbf{g}$ is support minimal by solving the systems $A_{ijk}\mathbf{g} = 0$ for $1 \leq i < j < k \leq 9$, where $A_{ijk}$ is the $3 \times 4$ submatrix of $A$ consisting of rows $i$, $j$, and $k$. Each $A_{ijk}$ matrix is rank 3 and so the solution to $A_{ijk}\mathbf{g} = 0$ is a line passing through the origin. Normalizing to coprime integer components will give the required circuit direction $\mathbf{g}$. Any circuit step can then be computed using a function that, given a starting point $\mathbf{x}$ and a circuit direction $\mathbf{g}$ as inputs, finds $\alpha_g$ so that $A(\mathbf{x} + \alpha_g \mathbf{g}) \geq \mathbf{b}$ but $A(\mathbf{x} + \alpha \mathbf{g}) \not\geq \mathbf{b}$ for $\alpha > \alpha_g$.

We can then enumerate exhaustively all circuit walks of length 2 or 3 emanating from any point, by considering all possible triples of circuit directions. With $Q_4$, we have 84 circuit directions, and taking into account both the positive and negative of each one, we find that there are at most $168 \cdot (166)^2 = 4629408$ triples to check. The MATLAB code can perform this check in a few minutes. Afterwards, to check for circuit walks of length 4, we use the set of points output by the above enumeration procedure and compute circuit steps for each feasible circuit. Again, this does not take long to run.

We mention here that precision may be lost due to MATLAB's use of floating point arithmetic, and this may be compounded at each circuit step taken. To get around this potential issue we include a function that rounds to zero any value that lies within some interval $(-\varepsilon, \varepsilon)$, where $\varepsilon$ is the machine epsilon in MATLAB, that is, the relative accuracy of floating-point arithmetic.[1] So, checking two quantities for equality is done by taking their difference and checking if it falls in this $\varepsilon$-interval. Certainly, a value larger than $\varepsilon$ may be used for this check – in fact, given a fixed polyhedron and a fixed upper bound for circuit diameter (say $\Delta_{\mathcal{E}}(P)$), one could likely determine what $\varepsilon$ would be appropriate for that polyhedron. (This relates back to the discussion in the proof of Lemma 3.4.)

We perform checks (in Maple) using exact arithmetic for the generated walks. All these computations were performed on a laptop in a few minutes.

In this way we find that $\tilde{Q}_4$ has circuit diameter 3, and $\tilde{Q}_4'$ has circuit diameter 4.

## 4.3 Alternate Representations of Polyhedra

Most of the computations we do work with the inequality form $\{\mathbf{x} : A\mathbf{x} \geq \mathbf{b}\}$ for polyhedra. We claimed at the beginning of Section 3.1 that we can assume that polyhedra we consider are full-dimensional, and are represented using an irredundant description of facets; here we explore this in more detail.

First, we prove the following statement:

**Proposition 4.2.** *Let $A \in \mathbb{Q}^{m \times d}$, $\mathbf{b} \in \mathbb{Q}^{m \times 1}$, and let $P = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. Further, let $Q = \{(\mathbf{x}, \mathbf{s}) \in \mathbb{R}^{d+m} : A\mathbf{x} - \mathbf{s} = \mathbf{b}, \mathbf{x}, \mathbf{s} \geq \mathbf{0}\}$ be the polyhedron obtained by adding slack variables to $P$ to convert it into standard form. Then, $\mathbf{g} \in \mathcal{C}(P)$ if and only if $(\mathbf{g}, A\mathbf{g}) \in \mathcal{C}(Q)$.*

[1] The number $\varepsilon$ is the distance from 1.0 to the next larger double-precision number, i.e. $\varepsilon = 2^{-52}$, see `https://www.mathworks.com/help/matlab/ref/eps.html`

*Proof.* First observe that $P$ can be written as

$$P = \left\{ \mathbf{x} \in \mathbb{R}^d : \begin{pmatrix} A \\ I_d \end{pmatrix} \mathbf{x} \geq \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \right\},$$

and $Q$ as

$$Q = \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{s} \end{pmatrix} \in \mathbb{R}^{d+m} : \begin{pmatrix} A & -I_m \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{s} \end{pmatrix} = \mathbf{b}, \begin{pmatrix} \mathbf{x} \\ \mathbf{s} \end{pmatrix} \geq \mathbf{0} \right\}.$$

Given a $\mathbf{g} \in \mathcal{C}(P)$, by definition, $\begin{pmatrix} A \\ I_d \end{pmatrix} \mathbf{g} = \begin{pmatrix} A\mathbf{g} \\ \mathbf{g} \end{pmatrix}$ is support-minimal among nonzero vectors $\mathbf{g}$. For $(\mathbf{g}, A\mathbf{g})$ to be in $\mathcal{C}(Q)$, we need to show $\begin{pmatrix} A & -I_m \end{pmatrix} \begin{pmatrix} \mathbf{g} \\ A\mathbf{g} \end{pmatrix} = \mathbf{0}$ and $(\mathbf{g}, A\mathbf{g})$ support-minimal in $\ker\left(\begin{pmatrix} A & -I_m \end{pmatrix}\right) \setminus \{\mathbf{0}\}$. But this kernel is isomorphic to the set $\mathbb{R}^d \setminus \{\mathbf{0}\}$, since the first $d$ components determine the last $m$ components completely. Also, $\begin{pmatrix} A & -I_m \end{pmatrix} \begin{pmatrix} \mathbf{g} \\ A\mathbf{g} \end{pmatrix} = A\mathbf{g} - A\mathbf{g} = \mathbf{0}$. Hence $\mathbf{g} \in \mathcal{C}(P)$ if and only if $(\mathbf{g}, A\mathbf{g}) \in \mathcal{C}(Q)$. $\square$

What Proposition 4.2 shows is that adding slack variables to convert from an inequality form to standard form does not change the set of circuits, in the sense that there is a bijection between $\mathcal{C}(P)$ and $\mathcal{C}(Q)$. Observe, however, that Proposition 4.2 begins with an inequality description of $P$ including the nonnegativity constraints $\mathbf{x} \geq \mathbf{0}$. If these are absent, the proposition will hold, but $Q$ will not be in standard form since the $\mathbf{x}$ variables are not constrained to be nonnegative. That is, we can set

$$P' = \{\mathbf{x} : A\mathbf{x} \geq \mathbf{b}\}$$

and

$$Q' = \{(\mathbf{x}, \mathbf{s}) : A\mathbf{x} - \mathbf{s} = \mathbf{b}, \mathbf{s} \geq \mathbf{0}\},$$

and we get the same bijection between circuit sets $\mathcal{C}(P')$ and $\mathcal{C}(Q')$.

On the other hand, the standard preprocessing technique to introduce non-negativity constraints on the way to standard form will introduce new circuits. Recall that this is to split unconstrained variables $\mathbf{x}$ into $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$, hence obtaining the polyhedron

$$Q = \{(\mathbf{x}^+, \mathbf{x}^-) : A(\mathbf{x}^+ - \mathbf{x}^-) \geq \mathbf{b}, \mathbf{x}^+, \mathbf{x}^- \geq \mathbf{0}\}.$$

One might hypothesize that $(\mathbf{g}^+, \mathbf{g}^-) \in \mathcal{C}(Q)$ gives a corresponding circuit $\mathbf{g} = \mathbf{g}^+ - \mathbf{g}^- \in \mathcal{C}(P)$, but this is not the case, as we show in Example 4.3.

**Example 4.3.** Consider the realization of $U_4$ given by $P = \{\mathbf{x} \in \mathbb{R}^4 : A\mathbf{x} \geq \mathbf{b}\}$, where

$$A = \begin{pmatrix} 3 & -3 & -1 & -4 \\ -3 & 3 & -1 & -4 \\ -2 & 1 & -1 & -5 \\ 2 & -1 & -1 & -5 \\ -3 & -3 & 1 & -4 \\ 3 & 3 & 1 & -4 \\ 1 & 2 & 1 & -5 \\ -1 & -2 & 1 & -5 \end{pmatrix} \text{ and } \mathbf{b} = (-1,-1,-1,-1,-1,-1,-1,-1)^T.$$

Then, let $Q = \{(\mathbf{x}^+, \mathbf{x}^-) \in \mathbb{R}^8 : A\mathbf{x}^+ - A\mathbf{x}^- \geq \mathbf{b},\ \mathbf{x}^+, \mathbf{x}^- \geq \mathbf{0}\}$.

Using python [pyt], we generate the circuits of $Q$ and see that vector

$$(\mathbf{g}^+, \mathbf{g}^-) = (10, 0, 0, 0, 0, 0, 5, 3)^T$$

is a circuit of $Q$. This means $\begin{pmatrix} A & -A \\ I & \mathbf{0} \\ \mathbf{0} & I \end{pmatrix} \begin{pmatrix} \mathbf{g}^+ \\ \mathbf{g}^- \end{pmatrix} = \begin{pmatrix} A\mathbf{g}^+ - A\mathbf{g}^- \\ \mathbf{g}^+ \\ \mathbf{g}^- \end{pmatrix}$ is support-minimal for nonzero $(\mathbf{g}^+, \mathbf{g}^-)$. This evaluates to the vector

$$(47, -13, 0, 40, -23, 37, 20, 0, 10, 0, 0, 0, 0, 0, 5, 3)^T.$$

If we attempt to pull this back into a circuit for $P$, we compute $\mathbf{g} = \mathbf{g}^+ - \mathbf{g}^- = (10, 0, -5, -3)^T$, which is *not* in $\mathcal{C}(P)$. This is because we can find another circuit $\mathbf{g}' \in \mathcal{C}(P)$ such that the support of $(A\mathbf{g}')$ is smaller. In fact, the circuit

$$\mathbf{g}' = (41, 47, 50, -17)^T \in \mathcal{C}(P)$$

works; we get $A\mathbf{g}' = (0, 36, 0, 70, -146, 382, 270, 0)^T$, and $A\mathbf{g} = (47, -13, 0, 40, -23, 37, 20, 0)^T$, showing that $\mathbf{g} \notin \mathcal{C}(P)$.

The reason why $\mathbf{g}^+ - \mathbf{g}^-$ appears as a circuit of $Q$ is because in $Q$ we check support-minimality of $\begin{pmatrix} A(\mathbf{g}^+ - \mathbf{g}^-) \\ \mathbf{g}^+ \\ \mathbf{g}^- \end{pmatrix}$ – so the support of the vector itself $(\mathbf{g}^+, \mathbf{g}^-)$ is taken into consideration. Hence, even if $A\mathbf{g}'$ has smaller support than $A\mathbf{g} = A(\mathbf{g}^+ - \mathbf{g}^-)$, $\mathbf{g}$ has one more zero entry than $\mathbf{g}'$, and so $(\mathbf{g}^+, \mathbf{g}^-)$ becomes a circuit of $\mathcal{C}(Q)$.

The implication of Proposition 4.2 and Example 4.3 is that while we are able to freely convert from the inequality form to the standard form of a polyhedron without substantially affecting the circuit sets, this does not transfer if new facets involving the existing variables have to be added. This is relevant to the computation of circuits, since most available software is written to compute circuits of polyhedra in standard form only (e.g. `4ti2`[4ti]). So discrepancies arise when computing circuits based on their inequality form, which is what we did in the MATLAB and python code in Appendix A.

We note also that for polytopes that are not full-dimensional, the set of circuits depends on whether we use an equality or inequality representation. Writing a polyhedron

$$P = \{\mathbf{x} : A^1\mathbf{x} = \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2\}$$

in inequality form

$$Q = \{\mathbf{x} : A^1\mathbf{x} \geq \mathbf{b}^1, -A^1\mathbf{x} \geq \mathbf{b}^1, A^2\mathbf{x} \geq \mathbf{b}^2\}$$

produces different circuits; this is because of the addition of more inequalities, which affects the minimality-check of the supports. In particular, while support-minimality of $A^2\mathbf{g}$ is checked for both $\mathcal{C}(P)$ and $\mathcal{C}(Q)$, $\mathcal{C}(P)$ is also restricted to be a subset of the kernel of $A^1$. Hence in $\mathcal{C}(Q)$, if we can find a vector $\mathbf{g}$ *not* in $\ker(A^1)$ but for which $A^2\mathbf{g}$ has smaller support than any other vectors, then $\mathbf{g} \in \mathcal{C}(Q) \setminus \mathcal{C}(P)$. This is easy to accomplish – suppose $A^2 = I$. Then we can pick one of the elementary coordinate vectors, and this would likely be a circuit of $Q$.

For instance, consider the $d$-simplex

$$\Delta_d = \{\mathbf{x} \in \mathbb{R}^{d+1} : \mathbf{1}^T\mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}.$$

Its circuits are the set

$$\mathcal{C}(\Delta_d) = \{\pm(\mathbf{e}_i - \mathbf{e}_j) : 1 \leq i < j \leq d+1\}.$$

But, when represented as

$$\Delta'_d = \{\mathbf{x} \in \mathbb{R}^{d+1} : \mathbf{1}^T\mathbf{x} \geq 1, -\mathbf{1}^T\mathbf{x} \geq -1, \mathbf{x} \geq \mathbf{0}\},$$

the set of circuits becomes

$$\mathcal{C}(\Delta'_d) = \{\pm(\mathbf{e}_i - \mathbf{e}_j) : 1 \leq i < j \leq d+1\} \cup \{\pm\mathbf{e}_i : 1 \leq i \leq d+1\}.$$

This is surprising, as $P$ and $Q$ are equal. In fact, to any polyhedron in inequality form one could simply add a redundant inequality and produce additional circuits – this demonstrates the dependence of the circuit set on how the polyhedron is represented. For full-dimensional polyhedron in $\mathbb{R}^d$, there is a unique minimal facet representation. This presentation is assumed in Chapter 3, which presents a purely geometric picture.

## 4.4 From Circuits to Vertices

We end this chapter by showing a nice equivalence that might be of use for computing circuits. Consider the polyhedron defined by

$$P_1 = \left\{ \mathbf{x} \in \mathbb{R}^d : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \right\},$$

where $A \in \mathbb{Z}^{f \times d}$, $\mathbf{b} \in \mathbb{Z}^f$. The circuits of this polyhedron are exactly the set

$$\mathbb{C}(A) = \left\{ \mathbf{g} \in \mathbb{R}^d : A\mathbf{g} = \mathbf{0}, \mathbf{g} \text{ has inclusion-minimal support } \right\}.$$

Now form a second polyhedron

$$P_2 = \left\{ \begin{pmatrix} \mathbf{y}^+ \\ \mathbf{y}^- \end{pmatrix} \in \mathbb{R}^{2d} : A(\mathbf{y}^+ - \mathbf{y}^-) = \mathbf{0}, \|\mathbf{y}^+\|_1 + \|\mathbf{y}^-\|_1 = 1, \mathbf{y}^+, \mathbf{y}^- \geq \mathbf{0} \right\}.$$

Here, $\|\mathbf{y}\|_1$ is the 1-norm of the vector $\mathbf{y}$, or $\|\mathbf{y}\| = \sum_{i=1}^d |y_i|$.

**Theorem 4.4.** *If $\mathbf{g} \in \mathbb{R}^d$ is a circuit of $A$, normalized so that $\|\mathbf{g}\|_1 = 1$, then the vector $\begin{pmatrix} \mathbf{y}^+ \\ \mathbf{y}^- \end{pmatrix} \in \mathbb{R}^{2d}$ defined by $y_i^+ = \max(g_i, 0)$ and $y_i^- = -\min(g_i, 0)$ $(1 \leq i \leq d)$ is a vertex of $P_2$.*

Proof: By construction of $\mathbf{y} = \begin{pmatrix} \mathbf{y}^+ \\ \mathbf{y}^- \end{pmatrix}$, $\mathbf{y} \geq \mathbf{0}$ and $\|\mathbf{y}\|_1 = \|\mathbf{y}^+\|_1 + \|\mathbf{y}^-\|_1 = 1$. Also at most one of $y_i^+$ and $y_i^-$ is nonzero for each $i$, so $A(\mathbf{y}^+ - \mathbf{y}^-) = A\mathbf{g} = \mathbf{0}$. Hence $\mathbf{y} \in P_2$.

Since $\mathbf{g}$ is support-minimal in $\ker(A) = \{\mathbf{x} : A\mathbf{x} = \mathbf{0}\}$, this means $\mathbf{y} = \begin{pmatrix} \mathbf{y}^+ \\ \mathbf{y}^- \end{pmatrix}$ has inclusion-minimal support among points in $P_2$. To see this, assume the opposite; that there is some vector $\mathbf{y}' = \begin{pmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \end{pmatrix} \in P_2$ with a smaller support, where $\mathbf{y}^1, \mathbf{y}^2 \in \mathbb{R}^d$. This means $\text{supp}(\mathbf{y}^1) \subseteq \text{supp}(\mathbf{y}^+)$ and $\text{supp}(\mathbf{y}^2) \subseteq \text{supp}(\mathbf{y}^-)$ (with at least one of these containments strict), implying that for each $i$, at most one of $y_i^1$ and $y_i^2$ is nonzero. This further implies that the vector $\mathbf{x} = \mathbf{y}^1 - \mathbf{y}^2$ satisfies $\text{supp}(\mathbf{x}) \subsetneq \text{supp}(\mathbf{g})$, as well as $A\mathbf{x} = \mathbf{0}$, contradicting the fact that $\mathbf{g}$ is a circuit of $A$.

Now since $\mathbf{y}$ is inclusion-minimal in $P_2$, this directly implies it is a vertex of $P_2$. This is because we cannot set any more variables to zero, i.e. the set of active inequalities at $\mathbf{y}$ is maximal. (Otherwise, we could stay in the same affine space and travel along a feasible direction until another inequality becomes active.) □

As for the reverse implication, we investigate whether all vertices of $P_2$ arise from circuits of $P_1$. This is generally not the case, as we show in the following example.

**Example 4.5.** Consider

$$P_1 = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{pmatrix} 1 & 2 & 1 \end{pmatrix} \mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \right\}$$

and the corresponding

$$P_2 = \left\{ \begin{pmatrix} \mathbf{y}^+ \\ \mathbf{y}^- \end{pmatrix} \in \mathbb{R}^3 \times \mathbb{R}^3 : \begin{pmatrix} 1 & 2 & 1 & -1 & -2 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{y}^+ \\ \mathbf{y}^- \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix},\ \mathbf{y}^+, \mathbf{y}^- \geq \mathbf{0} \right\}.$$

Then the normalized circuits of $P_1$ are

$$\mathcal{C}\left(\begin{pmatrix} 1 & 2 & 1 \end{pmatrix}\right) = \left\{ \pm \begin{pmatrix} \frac{2}{3} \\ -\frac{1}{3} \\ 0 \end{pmatrix}, \pm \begin{pmatrix} 0 \\ -\frac{1}{3} \\ \frac{2}{3} \end{pmatrix}, \pm \begin{pmatrix} \frac{1}{2} \\ 0 \\ -\frac{1}{2} \end{pmatrix} \right\}.$$

Using polymake, the vertices of $P_2$ are found to be

$$V(P_2) = \left\{ \begin{pmatrix} \frac{2}{3} \\ 0 \\ 0 \\ 0 \\ \frac{1}{3} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{1}{3} \\ 0 \\ \frac{2}{3} \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{1}{3} \\ 0 \\ 0 \\ 0 \\ \frac{2}{3} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \frac{2}{3} \\ 0 \\ \frac{1}{3} \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{1}{2} \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{1}{2} \\ 0 \\ 0 \\ \frac{1}{2} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \\ 0 \\ 0 \\ \frac{1}{2} \end{pmatrix}, \right\}$$

Computing $\mathbf{y}^+ - \mathbf{y}^-$ for the first six vertices gives exactly the circuits of $P_1$, while doing the same for the last three give the zero vector, which is not a circuit of $P_1$.

This leads to the following claim:

**Theorem 4.6.** *The vertices of $P_2$ are*

$$V(P_2) = \left\{ \begin{pmatrix} \mathbf{y}^+ \\ \mathbf{y}^- \end{pmatrix} : y_i^+ = \max(g_i, 0), y_i^- = -\min(g_i, 0), 1 \leq i \leq d,\ \forall g \in \mathbb{C}(P_1) \right\}$$

$$\cup \left\{ \begin{pmatrix} \frac{1}{2} e_i \\ \frac{1}{2} e_i \end{pmatrix} : 1 \leq i \leq d \right\}$$

*Proof.* Let $R = \left\{ \begin{pmatrix} \mathbf{y}^+ \\ \mathbf{y}^- \end{pmatrix} : y_i^+ = \max(g_i, 0), y_i^- = -\min(g_i, 0), 1 \leq i \leq d,\ \forall g \in \mathbb{C}(P_1) \right\}$ and $S = \left\{ \begin{pmatrix} \frac{1}{2} e_i \\ \frac{1}{2} e_i \end{pmatrix} : 1 \leq i \leq d \right\}$.

The vertices of $P_2$ are exactly its nonzero inclusion-minimal vectors

$$\begin{pmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \end{pmatrix},$$

and there are two possibilities: at most one of $y_i^1$ and $y_i^2$ is nonzero for each $1 \leq i \leq d$, or there is some index $1 \leq i \leq d$ such that both $y_i^1$ and $y_i^2$ are nonzero. The first scenario corresponds exactly to the vectors in $R$, as we have shown previously. We now show that the second type correspond to exactly the vectors in $S$.

Observe that if $y_i^1 > y_i^2 > 0$, we can modify these components to get a new vector $\mathbf{y}' = \begin{pmatrix} \mathbf{y}'^1 \\ \mathbf{y}'^2 \end{pmatrix} \in P_2$ where $y_i'^1 = y_i^1 + y_i^2$ and $y_i'^2 = 0$. Furthermore, we have $\mathrm{supp}(\mathbf{y}') \subsetneq \mathrm{supp}(\mathbf{y})$, implying $\mathbf{y}$ is not a vertex. If $y_i^2 > y_i^1 > 0$ instead, then the same argument holds but setting $y_i'^1 = 0$ and $y_i'^2 = y_i^1 + y_i^2$.

Finally, assume $y_i^1 = y_i^2 > 0$. If there is at least one other nonzero component in $\mathbf{y}$, then the vector $\mathbf{y}'$ obtained from $\mathbf{y}$ by setting $y_i^1$ and $y_i^2$ to zero has smaller support, so $\mathbf{y}$ is not a vertex of $P_2$. However, if there are no other nonzero components in $\mathbf{y}$, then it is support-minimal among nonzero vectors in $P_2$, since doing the same operation will result in just the zero vector. The only vectors satisying this condition are those that have $y_i^1 = y_i^2 = \frac{1}{2}$ and all other entries zero – this is exactly the set $S$.

$\square$

# Chapter 5

# Conclusion

## 5.1 Summary

This thesis builds up a theoretical toolkit to study the circuit diameters of polyhedra. This involves transferring the various techniques known and used by discrete geometers to prove bounds on polyhedral diameters – including wedging (Lemma 3.6), taking products (Proposition 2.77), operations on spindles (Section 3.4.2; Corollary 3.24), and reformulations of the non-revisiting and $d$-step conjectures. We see, working on small examples, that it is already non-trivial to prove bounds on the circuit diameter, since the set of circuit directions of a polyhedron depends on how it is represented (as a set of algebraic inequalities and equations) and how it is realized (as a geometric figure in $\mathbb{R}^d$). This is apparent already in two dimensions, and more so in four.

There are two results in this thesis that stand out in importance – the reformulation and implications between the circuit conjectures (Theorem 1.8) and the two proofs of the circuit 4-step conjecture (Section 3.4). The first gives us tools, showing that the general conjecture can be approached using key special cases, while the second removes a key counterexample from the edge case, and allows us to reinstate major conjectures for the circuit case.

The relationships among the circuit versions of the Hirsch, non-revisiting, and $d$-step conjectures, plus an additional one involving Dantzig figures, are established using an approach patterned after Klee and Walkup's original proof in [KW67] under the combinatorial diameter framework. We modified the notion of simplicity for polyhedra using wedge-simplicity and $k$-wedge-simplicity (Section 3.2), in order to have appropriate tools for dealing with circuit walks in wedges; however the combinatorial result (Theorem 2.52) is not fully transferred to the circuit framework because non-revisiting circuit walks in wedges do not necessarily project down to circuit walks in the original polyhedron – see the discussion in Section 3.3.4.

We are also able to prove some auxiliary results including Theorem 3.19 (a connection between unbounded and bounded circuit diameters) and an interesting equivalence between circuits and vertices in Section 4.4.

As for the circuit 4-step conjecture, the key ingredient to one of the proofs is the fact that the circuit diameter of $U_4$, the original counterexample to unbounded Hirsch, satisfies the circuit Hirsch conjecture (Theorem 3.21). We prove this is true independent of the specific realization of $U_4$; this is especially relevant because $U_4$ is the starting point for constructions that produce many other non-Hirsch unbounded polyhedra.

We present here the current known values for $\Delta_{\mathcal{E}}^{b}(f,d)$ as given in Theorem 2.46, and what is known for the circuit diameter $\Delta_{\mathcal{C}}(f,d)$.

## 5.2 Looking Forward

Of main priority is to complete the proof of the equivalences in Theorem 1.8, or to prove a partial equivalence (say (2) $\Rightarrow$ (1)). It is possible that the circuit non-revisiting conjecture is strictly stronger than the other circuit conjectures we formulate – since circuit walks allow steps through the interior of the polyhedron (or the relative interior of its faces), the non-revisiting property represents a much stronger restriction to these walks as compared to the combinatorial case.

The resolution of the (combinatorial) polynomial Hirsch conjecture remains a fundamental objective, and studying the relaxation of edge walks to circuit walks provides a way of furthering our understanding. Borgwardt et al. in [BLF16] give a hierarchy of polyhedral diameters corresponding to which properties are relaxed among feasibility, maximality, or walking along edges – the circuit diameter we study in this thesis $\Delta_{\mathcal{C}}(P)$ is in the middle of this hierarchy, as we discuss in Chapter 2. That is, we have

$$\Delta_{\mathcal{E}}(P) \geq \Delta_{\mathcal{C}}(P) \geq \Delta_{f}(P),$$

where $\Delta_f(P)$ denotes the diameter obtained when circuit steps are not restricted to be maximal. Moreover, $\Delta_f(P)$ satisfies the Hirsch bound for any polyhedron; this means the Hirsch bound breaks down somewhere in between $\Delta_f(P)$ and $\Delta_{\mathcal{E}}(P)$, and where the circuit diameter $\Delta_{\mathcal{C}}(P)$ is situated may lend clues on the general conjecture.

Presently there is no clear way to resolve the circuit Hirsch conjectures; even small cases are challenging, such as the circuit 5-step conjecture. A possible approach for 5-step would be an enumeration of simple 5-polyhedra with 10 facets and showing that they are derived from $U_4$ (and the literature supports this - see Borwardt et al. [BDHS13], Holt and Klee [HK98], Kim and Santos [KS10]); another is to look at whether all Hirsch-sharp 5-polytopes with 11 facets are derived from $Q_4$, following methods of Firsching [Fir17] and Bremner et al. [BDHS13]. This seems out of reach at present – Firsching [Fir17] outlines the state-of-the-art in enumeration; Lutz [Lut08] and Fukuda et al. [FMM13] are examples of this type of approach. Alternatively, one could prove that 5-spindles with 9 facets on one vertex and 5 on the other satisfy the non-revisiting conjecture – circuit 5-step follows from this via Corollary 3.20.

It may also be worthwhile to search for counterexamples. Santos-type polytopes would be one place to start; note that known examples of non-Hirsch polyhedra are based on $U_4$ is the Santos constructions. Also, given the relationships proven in Theorem 1.8, it may now be possible to understand the general situation by studying spindles.

It will be interesting to also consider *circuit Hirsch-sharp* polyhedra, or polyhedra whose circuit diameter is exactly the Hirsch bound $f - d$. The combinatorial Hirsch-sharp polytopes were extensively studied (Fritzsche and Holt [FH99], Holt and Klee [HK98]) before the bounded Hirsch conjecture was disproved; these include *trivial* Hirsch-sharp polytopes (where $f \leq 2d$) like the $d$-cube and the $d$-simplex, and *non-trivial* Hirsch-sharp polytopes (where $f > 2d$), which include $Q_4$ and others obtained from $Q_4$ by performing operations. For circuit diameter, the $d$-simplex remains Hirsch-sharp independent of realization, since $f - d = 1$. A regular $d$-cube is also Hirsch-sharp, but it is not obvious whether this remains true for non-regular realizations. We conjecture that Hirsch-sharpness transfers:

**Conjecture 5.1.** *Any realization of the $d$-cube is circuit Hirsch-sharp.*

Similarly, it is open whether there exists a realization of $U_4$ that has circuit diameter less than 4.

Another avenue to be explored is the calculation of circuit diameters for important classes of polyhedra. Known results include bounds for the circuit diameters of dual transportation polyhedra (Borgwardt et al. [BFH15]), dual network flow polyhedra (Borgwardt et al. [BFH16]) and $2 \times n$ transportation polytopes (Borgwardt et al. [BLFM15]). More recently, Kafer et al. characterized the circuit diameters of polytopes that arise from combinatorial optimization problems – the matching polytope, the traveling salesman polytope, and the fractional stable set polytope, showing that for these classes the circuit diameter is much smaller than their combinatorial diameters (Kafer et al. [KPS17]).

Finally, de Loera et al. showed in [DLHL15] that a circuit augmentation algorithm for linear programming that uses certain pivot rules finds a solution close to the optimum in a polynomial number of augmentations. However this polynomial bound depends on the number of circuits of the coefficient matrix, and one also has to consider the implementation of the various pivot rules as well. A practical implementation of a circuit-following algorithm will be nice to have – in particular, there does not exist yet an implementation of these pivot rules, which likely will prove to be the bottleneck for this algorithm. The search for other interesting pivot rules might also be fruitful in this regard, as a more efficient method of circuit selection will impact the algorithm significantly.

# Bibliography


[4ti] *4ti2—a software package for algebraic, geometric and combinatorial problems on linear spaces*, 4ti2 team.

[ABS80] A. Altshuler, J. Bokowski, and L. Steinberg, *The classification of simplicial 3-spheres with nine vertices into polytopes and nonpolytopes*, Discrete Mathematics **31** (1980), no. 2, 115–124.

[Bal84] M. L. Balinski, *The Hirsch conjecture for dual transportation polyhedra*, Mathematics of Operations Research **9** (1984), no. 4, 629–633.

[BDHS13] David Bremner, Antoine Deza, William Hua, and Lars Schewe, *More bounds on the diameters of convex polytopes*, Optimization Methods and Software **28** (2013), no. 3, 442–450.

[BFH15] Steffen Borgwardt, Elisabeth Finhold, and Raymond Hemmecke, *On the circuit diameter of dual transportation polyhedra*, Siam Journal on Discrete Mathematics **29** (2015), no. 1, 113–121.

[BFH16] ———, *Quadratic diameter bounds for dual network flow polyhedra*, Mathematical Programming, Series A **159** (2016), no. 1-2, 237–251.

[Bla77] Robert G. Bland, *New finite pivoting rules for the simplex method*, Mathematics of Operations Research **2** (1977), no. 2, 103–107.

[BLF16] S. Borgwardt, J. A. De Loera, and E. Finhold, *Edges vs circuits: a hierarchy of diameters in polyhedra*, Advances in Geometry **16** (2016), no. 4, 511–530.

[BLF17] ———, *The diameters of network-flow polytopes satisfy the Hirsch conjecture*, Mathematical Programming, Series A (2017), in press.

[BLFM15] S. Borgwardt, J.A. De Loera, E. Finhold, and J. Miller, *The hierarchy of circuit diameters and transportation polytopes*, Discrete Applied Mathematics (2015), in press.

[BS11] David Bremner and Lars Schewe, *Edge-graph diameter bounds for convex polytopes with few facets*, Experimental Mathematics **20** (2011), no. 3, 229–237.

[BSY16] Steffen Borgwardt, Tamon Stephen, and Timothy Yusun, *On the circuit diameter conjecture*, eprint arXiv:1611.08039v1, November 2016.

[BT97] Dimitris Bertsimas and John N. Tsitsiklis, *Introduction to linear optimization*, 1st ed., Athena Scientific, Belmont, MA, 1997.

[Dan63] George B. Dantzig, *Linear programming and extensions*, Princeton Univ. Press, Princeton, NJ, 1963.

[DLHL15] Jesús A. De Loera, Raymond Hemmecke, and Jon Lee, *On augmentation algorithms for linear and integer-linear programming: From Edmonds–Karp to Bland and beyond*, SIAM Journal on Optimization **25** (2015), no. 4, 2494–2511.

[DLK12] Jesús A. De Loera and Steven Klee, *Transportation problems and simplicial polytopes that are not weakly vertex-decomposable*, Mathematics of Operations Research **37** (2012), no. 4, 670–674.

[EHRR10] F. Eisenbrand, N. Hähnle, A. Razborov, and T. Rothvoß, *Diameter of polyhedra: Limits of abstraction*, Mathematics of Operations Research **35** (2010), no. 4, 786–794.

[FH99] K. Fritzsche and F. B. Holt, *More polytopes meeting the conjectured Hirsch bound*, Discrete Mathematics **205 (1-3)** (1999), 77–84.

[Fir17] Moritz Firsching, *Realizability and inscribability for simplicial polytopes via nonlinear optimization*, Mathematical Programming **166** (2017), no. 1, 273–295.

[FMM13] Komei Fukuda, Hiroyuki Miyata, and Sonoko Moriyama, *Complete enumeration of small realizable oriented matroids*, Discrete & Computational Geometry **49** (2013), no. 2, 359–381.

[FP96] Komei Fukuda and Alain Prodon, *Double description method revisited*, Combinatorics and Computer Science: 8th Franco-Japanese and 4th Franco-Chinese Conference Brest, France, July 3–5, 1995 Selected Papers (Michel Deza, Reinhardt Euler, and Ioannis Manoussakis, eds.), Springer, Berlin, Heidelberg, 1996, pp. 91–111.

[GCJS57] W. W. Garvin, H. W. Crandall, J. B. John, and R. A. Spellman, *Applications of linear programming in the oil industry*, Management Science **3** (1957), no. 4, 407–430.

[geo] *GeoGebra 4.4*, International GeoGebra Institute, `http://www.geogebra.org`.

[GJ00] Ewgenij Gawrilow and Michael Joswig, *polymake: a framework for analyzing convex polytopes*, Polytopes — Combinatorics and Computation (Gil Kalai and Günter M. Ziegler, eds.), Birkhäuser, 2000, pp. 43–74.

[Goo72] P. R. Goodey, *Some upper bounds for the diameters of convex polytopes*, Israel Journal of Mathematics **11** (1972), no. 4, 380–385.

[Grü03] Branko Grünbaum, *Convex polytopes*, Springer, 2003.

[GS67] Branko Grünbaum and V.P. Sreedharan, *An enumeration of simplicial 4-polytopes with 8 vertices*, Journal of Combinatorial Theory **2** (1967), 437–465.

[HH60] Fred Hanssmann and Sidney W. Hess, *A linear programming approach to production and employment scheduling*, Management Science **MT-1** (1960), no. 1, 46–51.

[HK98] F. B. Holt and V. Klee, *Many polytopes meeting the conjectured Hirsch bound*, Discrete and Computational Geometry **20** (1998), 1–17.

[KK87] V. Klee and P. Kleinschmidt, *The d-step conjecture and its relatives*, Mathematics of Operations Research **12** (1987), no. 4, 718–755.

[KK92] G. Kalai and D. Kleitman, *A quasi-polynomial bound for the diameter of graphs of polyhedra*, Bulletin of the American Mathematical Society **26** (1992), no. 2, 315–316.

[Kle64] Victor Klee, *Diameters of polyhedral graphs*, Canadian Journal of Mathematics **16** (1964), 602–614.

[KPS17] Sean Kafer, Kanstantin Pashkovich, and Laura Sanità, *On the circuit diameter of some combinatorial polytopes*, eprint arXiv:1709.09642v1, 2017.

[KS09] Edward D. Kim and Francisco Santos, *Companion to "An update on the Hirsch conjecture"*, eprint arXiv:0912.4235, 2009.

[KS10] E. D. Kim and F. Santos, *An update on the Hirsch conjecture*, Jahresbericht der Deutschen Mathematiker-Vereinigung **112** (2010), no. 2, 73–98.

[KW67] V. Klee and D. W. Walkup, *The d-step conjecture for polyhedra of dimension* $d < 6$, Acta Mathematica **133** (1967), 53–78.

[Lar70] D. G. Larman, *Paths on polytopes*, Proceedings of the London Mathematical Society **s3-20** (1970), no. 1, 161–178.

[LMS17] Jean-Philippe Labbé, Thibault Manneville, and Francisco Santos, *Hirsch polytopes with exponentially long combinatorial segments*, Mathematical Programming **165** (2017), no. 2, 663–688.

[Lut08] Frank H. Lutz, *Combinatorial 3-manifolds with 10 vertices*, Beiträge zur Algebra und Geometrie **49** (2008), no. 1, 97–106.

[map14] *Maple 18*, Maplesoft, a Division of Waterloo Maple, Inc., Waterloo, ON, 2011-2014.

[mat14] *Matlab version 8.4.0 (r2014b)*, The Mathworks, Inc., Natick, MA, 2014.

[McK81] Brendan D. McKay, *Practical graph isomorphism*, Congressus Numerantium **30** (1981), 45–87.

[MOS14] Renata Mansini, Wlodzimierz Ogryczak, and M. Grazia Speranza, *Twenty years of linear programming based portfolio optimization*, European Journal of Operational Research **234** (2014), no. 2, 518 – 535, 60 years following Harry MarkowitzâĂŹs contribution to portfolio theory and operations research.

[MRTT53] T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall, *The double description method*, Contributions to the Theory of Games (AM-28) (Harold William Kuhn and Albert William Tucker, eds.), Annals of Mathematics Studies 2, vol. II, Princeton: Princeton University Press, 1953, pp. 51–74.

[MSW95] Olvi L. Mangasarian, W. Nick Street, and William H. Wolberg, *Breast cancer diagnosis and prognosis via linear programming*, Operations Research **43** (1995), no. 4, 570–577.

[MSW15] Benjamin Matschke, Francisco Santos, and Christophe Weibel, *The width of five-dimensional prismatoids*, Proceedings of the London Mathematical Society **110** (2015), no. 3, 647–672.

[Nad89] D. Naddef, *The Hirsch Conjecture is true for* $(0,1)$*-polytopes*, Mathematical Programming **45** (1989), no. 1, 109–110.

[ORT05] Shmuel Onn, Uriel G. Rothblum, and Yoav Tangir, *Edge-directions of standard polyhedra with applications to network flows*, Journal of Global Optimization **33** (2005), no. 1, 109–122.

[PB80] J. Scott Provan and Louis J. Billera, *Decompositions of simplicial complexes related to diameters of convex polyhedra*, Mathematics of Operations Research **5** (1980), no. 4, 576–594.

[pyt] *Python language reference, version 2.7*, Python Software Foundation, Available at http://python.org.

[Roc69] R. T. Rockafellar, *The elementary vectors of a subspace of* $R^N$, Combinatorial Mathematics and its Applications, University of North Carolina Press, Chapel Hill, NC, 1969, pp. 104–127.

[San12] Francisco Santos, *A counterexample to the Hirsch conjecture.*, Annals of Mathematics **176** (2012), no. 1, 383–412.

[SK15] Noriyoshi Sukegawa and Tomonari Kitahara, *A refinement of Todd's bound for the diameter of a polyhedron*, Oper. Res. Lett. **43** (2015), no. 5, 534–536.

[SST12] F. Santos, T. Stephen, and H. Thomas, *Embedding a pair of graphs in a surface, and the width of 4-dimensional prismatoids*, Discrete Computational Geometry **47** (2012), 569–576.

[Suk17] Noriyoshi Sukegawa, *Improving bounds on the diameter of a polyhedron in high dimensions*, Discrete Mathematics **340** (2017), no. 9, 2134 – 2142.

[SY15] Tamon Stephen and Timothy Yusun, *The circuit diameter of the Klee-Walkup polyhedron*, Electronic Notes in Discrete Mathematics, The Eight European Conference on Combinatorics, Graph Theory and Applications, EuroComb 2015 **49** (2015), 505–512.

[Tod14] Michael J. Todd, *An improved Kalai-Kleitman bound for the diameter of a polyhedron*, SIAM Journal on Discrete Mathematics **28** (2014), no. 4, 1944–1947.

[YKK84] V. A. Yemelichev, M. M. Kovalëv, and M. K. Kravtsov, *Polytopes, graphs and optimisation*, Cambridge University Press, Cambridge, 1984, Translated from the Russian by G. H. Lawden.

# Appendix A

# Code

## A.1 MATLAB Code for Computations

In the following, we assume that the polyhedron is given as $P = \{\mathbf{x} : A\mathbf{x} \geq \mathbf{b}\}$, where $A \in \mathbb{Q}^{n\times d}$ and $\mathbf{b} \in \mathbb{R}^{d\times 1}$,

The main functions we use are ones that compute the circuits of $P$, and a script that traverses circuit steps to maximality within the feasible region $A\mathbf{x} \geq \mathbf{b}$. We also have functions to perform operations on $P$ like compute its vertices, but for the most part we defer to `polymake` for these operations, so we omit those here and only include the following:

| **function** | **description** |
|---|---|
| `compute_circuits.m` | Computes the circuits of $P$ by checking ranks of $(d-1)\times d$ submatrices (only for $d = 3, 4, 5$). |
| `traverse.m` | Moves along a given direction from a given starting point while maintaining feasiblity (or detects infeasibility of the direction). |
| `adjust_for_rounding.m` | Rounds to zero any entries in a vector smaller than a fixed tolerance. |
| `compute_paths4.m` | Script that computes circuit walks in a polyhedron starting from a given point. |

```
function result = compute_circuits(A)

%% Given an m-by-d matrix A
%% reduces each (d-1)-by-d submatrix to row echelon form
%% and outputs the rightmost column (as long as rank of
%% the submatrix is equal to d-1)
%% Brute forces; only works for d = 3, 4, 5.

[m, d] = size(A);
all_ind = 1:m;
```

```
all_perms = nchoosek(all_ind,d-1);

result = zeros(size(all_perms,1),d);

if (d == 5)

    for i = 1:size(all_perms,1)

        if rank(A(all_perms(i,:),:)) == 4
            [reduced, basis] = rref(A(all_perms(i,:),:));

            if sum(basis) == 10     % 1 2 3 4
                temp = [-reduced(:,d) ; 1];
                [num den] = rat(temp);
                result(i,:) = sign(temp(find(temp,1)))*max(abs(den))*temp';

            elseif sum(basis) == 11 % 1 2 3 5
                temp = [-reduced(1:d-2,d-1) ; 1 ; 0];
                [num den] = rat(temp);
                result(i,:) = sign(temp(find(temp,1)))*max(abs(den))*temp';

            elseif sum(basis) == 12 % 1 2 4 5
                temp = [-reduced(1:d-3,d-2) ; 1 ; -reduced(d-2:d-1,d-2)];
                [num den] = rat(temp);
                result(i,:) = sign(temp(find(temp,1)))*max(abs(den))*temp';

            elseif sum(basis) == 13 % 1 3 4 5
                temp = [-reduced(1:d-4,d-3) ; 1 ; -reduced(d-3:d-1,d-3)];
                [num den] = rat(temp);
                result(i,:) = sign(temp(find(temp,1)))*max(abs(den))*temp';

            elseif sum(basis) == 14 % 2 3 4 5
                % First column all zeros
                temp = [1 ; 0 ; 0 ; 0 ; 0];
                result(i,:) = sign(temp(find(temp,1)))*temp';

            end


        end
    end
elseif (d == 4)

    for i = 1:size(all_perms,1)
        [reduced, basis] = rref(A(all_perms(i,:),:));

        if sum(basis) == 6 % basis [1 2 3]
            temp = [-reduced(:,d) ; 1];
            [num den] = rat(temp);
            result(i,:) = sign(temp(find(temp,1)))*lcm(sym(abs(den)))*temp';

        elseif sum(basis) == 7 % basis [1 2 4]
            temp = [-reduced(1:d-2,d-1) ; 1 ; 0];
```

```
            [num den] = rat(temp);
            result(i,:) = sign(temp(find(temp,1)))*lcm(sym(abs(den)))*temp';

        elseif sum(basis) == 8 % basis [1 3 4]
            temp = [-reduced(1:d-3,d-2) ; 1 ; -reduced(d-2:d-1,d-2)];
            [num den] = rat(temp);
            result(i,:) = sign(temp(find(temp,1)))*lcm(sym(abs(den)))*temp';

        elseif sum(basis) == 9 % basis [2 3 4]
            % Only possible if first column is all zeros
            temp = [1 ; 0 ; 0 ; 0];
            result(i,:) = sign(temp(find(temp,1)))*temp';

        end

    end

elseif (d == 3)

        for i = 1:size(all_perms,1)
        [reduced, basis] = rref(A(all_perms(i,:),:));

        if sum(basis) == 3 % basis [1 2]
            temp = [-reduced(:,3) ; 1];
            [num den] = rat(temp);
            result(i,:) = sign(temp(find(temp,1)))*max(abs(den))*temp';

        elseif sum(basis) == 4 % basis [1 3]
            temp = [-reduced(1,2) ; 1 ; 0];
            [num den] = rat(temp);
            result(i,:) = sign(temp(find(temp,1)))*max(abs(den))*temp';

        elseif sum(basis) == 5 % basis [2 3]
            % Only possible if first column is all zeros
            temp = [1 ; 0 ; 0];
            result(i,:) = sign(temp(find(temp,1)))*temp';

        end

    end

end

result = unique(result,'rows')



%%%%%%%%%%%%%%%%%%%%%%%

function [point feas] = traverse(in_point,dir,A,b)

%% Given a point in_point and
%% a direction dir,
```

```
%% computes first whether dir is feasible.
%% If it is, computes the last point in the polytope
%% that can be reached from in_point
%% when traveling in the direction dir
%% If dir is not feasible,
%% Then outputs feas = 0 and point = in_point

%% Assume polytope is A*x >= -1
%% Assume polytope is A*x >= b??

%% Check if dimensions match.
dim = size(A,2);
inequalities = size(A,1);

if size(in_point,1) == dim
else
    disp('Input point has different number of variables than polytope');
    point = in_point;
    feas = -1;
    return
end

%% Check feasibility. (Only check dir, not -dir)
% Feasible if for tight inequalities at in_point, A*dir is >= 0

tight = (adjust_for_rounding(A*in_point-b) == 0);
A_dpos = (adjust_for_rounding(A*dir) >= 0);

if all(A_dpos(tight)) == 1
   % Then feasible
   feas = 1;

   % Now compute coefficients and compute point

   % Only consider non-tight inequalities now
   %current = and(not(tight),adjust_for_rounding(A*dir) < 0);
   current = and(not(tight),(A*dir < 0));

   % Get smallest possible value for coefficient of dir
   alpha = min((b(current) - A(current,:)*in_point)./(A(current,:)*dir));
   if alpha > 0
       point = in_point + alpha*dir;
   else % Then unbounded
       point = in_point;
       feas = 0;
   end

else
    % Then not feasible
    feas = 0;
    point = in_point;
end
```

```
end


%%%%%%%%%%%%%%%%%%%%%%%

function result = adjust_for_rounding(vect)

%% Given a vector vect
%% Rounds any entry with absolute value <= eps
%% to exactly 0

result = vect;
ind = abs(vect)<=eps;
result(ind) = 0;

end


%%%%%%%%%%%%%%%%%%%%%%%



function [paths points] = compute_paths4(A,b,gravers,start)

%% For computing all graver paths starting from vertex 3
%% Of length 2 or 3.

%% First generate all possible combinations of
%% directions
total_g = size(gravers,1);
temp = nchoosek(1:total_g,4);
perms4 = perms(1:4);
dir_set = zeros(0,4,'int8');

% all 4 different first!
for i = 1:size(perms4,1)
   dir_set =  [dir_set; temp(:,perms4(i,:))];
end

clear temp perms4;

%temp1 = nchoosek(1:84,3);
%dir_set = [temp1; temp1(:,[2 3 1]); temp1(:,[3 1 2])];
%temp2 = nchoosek(1:84,2);
%dir_set = int8(sortrows([dir_set; temp2(:,[1 2 1]); temp2(:,[2 1 2])]));
%clear temp1 temp2;



% Get 292824 combinations
total = size(dir_set,1);
% total is 46308024 for only all 4 different
```

```
% Initialize output matrices:
% One for list of triples of directions (3 entries per row) and
% one for points generated (12 entries per row)
paths = zeros(0,4,'int8'); % Initialize as int8 for now
points = zeros(0,16);

%% Consider each combination then compute
%% node coordinates

for k = 45500001:46308024 % Change manually

    % Combination k, directions
    d1 = gravers(dir_set(k,1),:)';
    d2 = gravers(dir_set(k,2),:)';
    d3 = gravers(dir_set(k,3),:)';
    d4 = gravers(dir_set(k,4),:)';

    % First direction d1
    % Check if feasible direction.
    % Feasible if A*d1>=0 or A*d1<=0
    % For tight facets at start

    % Call traverse function to go to next point
    % Have to take into account both +d and -d for
    % directions...
    [point1_pos, feas1_pos] = traverse(start,d1,A,b);
    [point1_neg, feas1_neg] = traverse(start,-d1,A,b);

    % First branch: d1 feasible
    if feas1_pos == 1 % then feasible
        [point2_pos, feas2_pos] = traverse(point1_pos,d2,A,b);
        [point2_neg, feas2_neg] = traverse(point1_pos,-d2,A,b);

        % Same, branch off
        if feas2_pos == 1 % d2 feasible
            [point3_pos, feas3_pos] = traverse(point2_pos,d3,A,b);
            [point3_neg, feas3_neg] = traverse(point2_pos,-d3,A,b);


            if feas3_pos == 1 % d3 feasible

                [point4_pos, feas4_pos] = traverse(point3_pos,d4,A,b);
                [point4_neg, feas4_neg] = traverse(point3_pos,-d4,A,b);

                if feas4_pos == 1
                    paths = [paths; dir_set(k,:)];
                    points = [points; [point1_pos' point2_pos' point3_pos' point4_pos']];
                end

                if feas4_neg == 1
                    paths = [paths; dir_set(k,:).*int8([1 1 1 -1])];
                    points = [points; [point1_pos' point2_pos' point3_pos' point4_neg']];
                end
```

```
    end

    if feas3_neg == 1 % -d3 feasible

        [point4_pos, feas4_pos] = traverse(point3_neg,d4,A,b);
        [point4_neg, feas4_neg] = traverse(point3_neg,-d4,A,b);

        if feas4_pos == 1
            paths = [paths; dir_set(k,:).*int8([1 1 -1 1])];
            points = [points; [point1_pos' point2_pos' point3_neg' point4_pos']];
        end

        if feas4_neg == 1
            paths = [paths; dir_set(k,:).*int8([1 1 -1 -1])];
            points = [points; [point1_pos' point2_pos' point3_neg' point4_neg']];
        end

    end
end

if feas2_neg == 1 % -d2 feasible
    [point3_pos, feas3_pos] = traverse(point2_neg,d3,A,b);
    [point3_neg, feas3_neg] = traverse(point2_neg,-d3,A,b);


    if feas3_pos == 1 % d3 feasible

        [point4_pos, feas4_pos] = traverse(point3_pos,d4,A,b);
        [point4_neg, feas4_neg] = traverse(point3_pos,-d4,A,b);

        if feas4_pos == 1
            paths = [paths; dir_set(k,:).*int8([1 -1 1 1])];
            points = [points; [point1_pos' point2_neg' point3_pos' point4_pos']];
        end

        if feas4_neg == 1
            paths = [paths; dir_set(k,:).*int8([1 -1 1 -1])];
            points = [points; [point1_pos' point2_neg' point3_pos' point4_neg']];
        end

    end

    if feas3_neg == 1 % -d3 feasible

        [point4_pos, feas4_pos] = traverse(point3_neg,d4,A,b);
        [point4_neg, feas4_neg] = traverse(point3_neg,-d4,A,b);

        if feas4_pos == 1
            paths = [paths; dir_set(k,:).*int8([1 -1 -1 1])];
            points = [points; [point1_pos' point2_neg' point3_neg' point4_pos']];
        end
```

```
                if feas4_neg == 1
                    paths = [paths; dir_set(k,:).*int8([1 -1 -1 -1])];
                    points = [points; [point1_pos' point2_neg' point3_neg' point4_neg']];
                end

            end
        end

    elseif feas1_neg == 1 % Note since start is a vertex, d1 and -d1 can't be both feasible

        [point2_pos, feas2_pos] = traverse(point1_neg,d2,A,b);
        [point2_neg, feas2_neg] = traverse(point1_neg,-d2,A,b);

        % Same, branch off
        if feas2_pos == 1 % d2 feasible
            [point3_pos, feas3_pos] = traverse(point2_pos,d3,A,b);
            [point3_neg, feas3_neg] = traverse(point2_pos,-d3,A,b);

            if feas3_pos == 1 % d3 feasible

                [point4_pos, feas4_pos] = traverse(point3_pos,d4,A,b);
                [point4_neg, feas4_neg] = traverse(point3_pos,-d4,A,b);

                if feas4_pos == 1
                    paths = [paths; dir_set(k,:).*int8([-1 1 1 1])];
                    points = [points; [point1_neg' point2_pos' point3_pos' point4_pos']];
                end

                if feas4_neg == 1
                    paths = [paths; dir_set(k,:).*int8([-1 1 1 -1])];
                    points = [points; [point1_neg' point2_pos' point3_pos' point4_neg']];
                end

            end

            if feas3_neg == 1 % -d3 feasible

                [point4_pos, feas4_pos] = traverse(point3_neg,d4,A,b);
                [point4_neg, feas4_neg] = traverse(point3_neg,-d4,A,b);

                if feas4_pos == 1
                    paths = [paths; dir_set(k,:).*int8([-1 1 -1 1])];
                    points = [points; [point1_neg' point2_pos' point3_neg' point4_pos']];
                end

                if feas4_neg == 1
                    paths = [paths; dir_set(k,:).*int8([-1 1 -1 -1])];
                    points = [points; [point1_neg' point2_pos' point3_neg' point4_neg']];
                end

            end

        end
```

```
        if feas2_neg == 1 % -d2 feasible
            [point3_pos, feas3_pos] = traverse(point2_neg,d3,A,b);
            [point3_neg, feas3_neg] = traverse(point2_neg,-d3,A,b);


            if feas3_pos == 1 % d3 feasible

                [point4_pos, feas4_pos] = traverse(point3_pos,d4,A,b);
                [point4_neg, feas4_neg] = traverse(point3_pos,-d4,A,b);

                if feas4_pos == 1
                    paths = [paths; dir_set(k,:).*int8([1 -1 1 1])];
                    points = [points; [point1_pos' point2_neg' point3_pos' point4_pos']];
                end

                if feas4_neg == 1
                    paths = [paths; dir_set(k,:).*int8([1 -1 1 -1])];
                    points = [points; [point1_pos' point2_neg' point3_pos' point4_neg']];
                end

            end

            if feas3_neg == 1 % -d3 feasible

                [point4_pos, feas4_pos] = traverse(point3_neg,d4,A,b);
                [point4_neg, feas4_neg] = traverse(point3_neg,-d4,A,b);

                if feas4_pos == 1
                    paths = [paths; dir_set(k,:).*int8([-1 -1 -1 1])];
                    points = [points; [point1_neg' point2_neg' point3_neg' point4_pos']];
                end

                if feas4_neg == 1
                    paths = [paths; dir_set(k,:).*int8([-1 -1 -1 -1])];
                    points = [points; [point1_neg' point2_neg' point3_neg' point4_neg']];
                end

            end

        end

    end


end

%% Then, find if the target point is reached by checking output
%% array in the appropriate columns
```

## A.2 python Code for Polyhedral Computation

We use `python` mainly to aid in the polyhedral computations needed - e.g. computing vertices of a polyhedron, computing combinatorial diameter, etc. The original intent in using `python` was for it to serve as a `polymake` replacement, as the latter cannot be installed on Windows computers without resorting to third-party software like `cygwin`.[1] Towards the end of this project we discovered that `polymake` could be run on `docker`[2], a much easier set-up than installing `cygwin` or another virtual machine, so we went back to `polymake` for vertex computations, etc. The `python` code still remained useful, however, as we are able to do there most of what we did in `MATLAB`, like traversing circuit directions in a given feasible region. It is as of now incomplete still, and we hope to improve the code in the future.

What we did was to write a class `Poly` for polyhedra, taking as input a coefficient matrix $A$ and a right-hand-side vector $\mathbf{b}$. The `Poly` object generated contains as attributes its vertices, rays, facet-vertex incidences, circuits, and so on. Because of this, for moderately large coefficient matrices it already is not quick to implement – especially since our circuit computations are also brute force here as in `MATLAB`.

Some additional notes on packages used:

- We use `pycddlib`, a wrapper for Fukuda's implementation of the double description method for generating vertices of a polyhedron [MRTT53, FP96].[3]
- We use `pynauty`, which implements McKay's graph isomorphism algorithm [McK81]. This has to be installed manually though, and only worked when we ran `python` from inside `cygwin`.[4]
- We use `NetworkX` for incidence graph construction and computing combinatorial diameter; more importantly, we also use it for checking isomorphism, eliminating the need for the `pynauty` package.[5]

```
"""
Python 2.7

This file contains the functions needed to carry out computations on polyhedra
involving circuit directions.

To call the functions below, make sure it is in the same directory as the script
file and use the line

>> from circuitcomp import *

Inputs:
# mat = n x (d+1) matrix encoding Ax >= b as [-b | A]
# A = n x d coefficient matrix; b = n x 1 rhs vector
# these are extracted from mat
```

[1] http://www.cygwin.com/
[2] https://www.docker.com/
[3] https://pypi.python.org/pypi/pycddlib
[4] https://web.cs.dal.ca/~peter/software/pynauty/html/index.html
[5] https://networkx.github.io

```
Assumptions:
# Polyhedron is defined by a system of linear inequalities Ax >= b

"""

# PACKAGES REQUIRED
import sympy as sp
import itertools
import numpy as np
import cdd
import fractions
import networkx as nx
import copy


class Poly:
    """ Class of polyhedra """
    """ Assume Ax >= b, mat = [-b | A] """
    """ Matrices are in np.matrix """

    def __init__(self,coef):
        if coef.shape[1] == 0:
            print 'Error: empty input matrix'
        else:
            mat = copy.deepcopy(coef)
            self.A = mat[:,1:]
            self.b = -mat[:,0]
            self.n_facets = self.A.shape[0]
            self.dimension = int(self.A.shape[1])
            self.mat = mat
            self.vertices = get_vertices(mat)
            self.rays = get_rays(mat)
            self.n_vertices = self.vertices.shape[0]
            self.n_rays = self.rays.shape[0]
            self.bounded = (len(self.rays)==0)
            self.facets_thru_vertices = {j: set([i for i in range(self.n_facets)...
            ...if ((self.A*self.vertices[j].T-self.b)==0)[i]])...
            ...for j in range(self.n_vertices)}
            self.facets_thru_rays = {j: set([i for i in range(self.n_facets)...
            ...if ((self.A*self.rays[j].T)==0)[i]]) for j in range(self.n_rays)}
            self.circuits = (get_circuits(self.A))
            self.n_circuits = self.circuits.shape[0]
            self.facets_thru_circuits = {j: set([i for i in range(self.n_facets)...
            ...if ((self.A*self.circuits[j].T)==0)[i]]) for j in range(self.n_circuits)}
            self.v_adjacency = vertex_adjacency(self.facets_thru_vertices,self.dimension)
            self.diameter = compute_diameter(self.v_adjacency)
            self.FVgraph = compute_bipartite_facet_vertex(self.facets_thru_vertices,...
            ...self.n_vertices,self.n_facets)
            G = nx.Graph()
            G.add_nodes_from(self.v_adjacency.keys())
            for i in self.v_adjacency.keys():
                for j in self.v_adjacency[i]:
```

```
                G.add_edge(i,j)
            self.nxgraph = G ## create NetworkX Graph
            G1 = nx.Graph()
            G1.add_nodes_from(self.FVgraph.keys())
            for i in self.FVgraph.keys():
                for j in self.FVgraph[i]:
                    G1.add_edge(i,j)
            self.nxFVgraph = G1 ## create NetworkX Graph for FVgraph so we can...
            ...check isomorphism without resorting to pynauty

    def __str__(self):
        print 'Polyhedron Ax >= b where A is'
        print self.A
        print 'and b is'
        print self.b
        return ''

def get_vertices(mat):
    m = cdd.Matrix(mat.tolist(),number_type='fraction')
    m.rep_type = cdd.RepType.INEQUALITY
    poly = cdd.Polyhedron(m)
    vert = np.matrix(poly.get_generators())
    return np.compress(np.array(vert[:,0]>0).flatten(), vert, axis=0)[:,1:]

def get_rays(mat):
    m = cdd.Matrix(mat.tolist(),number_type='fraction')
    m.rep_type = cdd.RepType.INEQUALITY
    poly = cdd.Polyhedron(m)
    vert = np.matrix(poly.get_generators())
    return np.compress(np.array(vert[:,0]==0).flatten(), vert, axis=0)[:,1:]


def get_circuits(mat):
    """
    # Given coefficient matrix A of size n x d (in system Ax >= b)
    # compute its circuits, i.e. null spaces of each
    # full rank (d-1) x d submatrix.
    """
    # Initialize output array
    r, c = mat.shape
    result = sp.zeros(0,c)
    # Generate all combinations of 1..n choose (d-1)
    for comb in itertools.combinations(range(r),c-1):
        temp = sp.Matrix(mat[comb,:])
        M, b = temp.rref()
        if len(b) == c-1: # Do computation only if rank is correct
            ind = [i for i in range(c) if i not in b][0]
            circ = M[:,ind].row_insert(ind,sp.Matrix([-1])).T
            # Normalize to coprime integers!
            cand = circ*(sp.lcm([circ[j].q for j in range(c) if circ[j] != 0]))
            if [cand[j] for j in range(c) if cand[j] != 0][0] < 0:
                cand *= -1
            if tuple(cand) not in [tuple(row) for row in result.tolist()]:
```

```
                result = result.col_join(cand)
    # Now convert sympy result matrix to numpy
    npresult = np.zeros([result.shape[0],c],dtype=fractions.Fraction)
    for i in range(result.shape[0]):
        for j in range(c):
            npresult[i,j] = result[i,j] # Normalized to coprime integers already anyway
    return np.matrix(npresult,dtype=fractions.Fraction)
    ## Uses exact arithmetic for this computation


def vertex_adjacency(FV,dim):
    """
    Take Vertex-Facet adjacency dictionary and generate
    Vertex-Vertex adjacency list.
    ** dim is dimension of polyhedron
    """
    num_vert = len(FV)
    VV = dict([(i,set()) for i in range(num_vert)])
    for i in range(num_vert-1):
        for j in range(i+1,num_vert):
            if len(FV[i] & FV[j]) == (dim - 1):
                VV[i] = VV[i] | {j}
                VV[j] = VV[j] | {i}
    return VV

def bfs(graph,start):
    """
    Breadth-first search on graph (a dictionary) starting from
    node start
    Output is a dictionary {0: start; 1: graph[start]; etc} (levels)
    """
    notvisited = set(graph.keys())
    result = {0:{start}}
    #notvisited.remove(start)
    current = {start}
    currentlevel = 1
    while notvisited:
        new_nodes = set()
        for i in current:
            new_nodes = new_nodes | graph[i]
            notvisited.remove(i)
        result[currentlevel] = new_nodes & notvisited
        currentlevel += 1
        current = new_nodes & notvisited
    del result[currentlevel-1]
    return result

def compute_diameter(VV):
    """
    Use Vertex-Vertex adjacency dictionary to compute
    diameter of polyhedron
    """
    result = 0
```

```
    for i in VV.keys():
        if len(bfs(VV,i))-1 > result:
            result = len(bfs(VV,i))-1
    return result

def compute_bipartite_facet_vertex(FV,nv,nf):
    """
    Given facet-vertex adjacency list,
    generate bipartite graph of vertex/facet incidences
    for use with pynauty
    """
    total_nodes = nv + nf
    result = dict([(i,set()) for i in range(total_nodes)])
    for i in FV.keys():
        for j in FV[i]:
            new_j = int(j + nv)
            result[i] = result[i] | {new_j}
            result[new_j] = result[new_j] | {i}
    for i in result.keys():
        result[i] = list(result[i])
    return result

def check_connected(VV):
    """
    Check if the graph determined by the vertex-vertex adjacency
    matrix is connected. Do this by performing one BFS - result must hit all vertices
    """
    k = VV.keys()
    bfsearch = bfs(VV,k[0])
    visited = set()
    for i in bfsearch:
        visited = visited | bfsearch[i]
    if len(visited) == len(VV):
        return 1
    else:
        return 0

def isomorphic(P,Q):
    """
    #Inputs: two Poly objects P, Q
    # Requires NetworkX
    """
    return nx.is_isomorphic(P.nxFVgraph,Q.nxFVgraph)

def traverse(in_point,circ,A,b):
    """
    # Starting at in_point, traverse circ as far as possible
    # inside the polyhedron Ax >= b
    # A is n x d, b is n x 1,
    # in_point and circ are d x 1
    """
    # Check feasibility of circ at in_point
    facets, dim = A.shape
```

```
    tight = {i for i in range(facets) if (A*in_point - b)[i] == 0}
    not_tight = {i for i in range(facets) if i not in tight}
    feas = check_feas(in_point,circ,A,b)
    if feas == 2:
        new_point = in_point
    elif feas == 1:
        candidates = not_tight & {i for i in range(facets) if (A*circ)[i] < 0}
        diffs = b - A*in_point
        ratios = {(diffs[j]/((A*circ)[j])).item(0) for j in candidates}
        alpha = (min(ratios))
        new_point = in_point + alpha*circ
    else:
        new_point = in_point
    return (feas, new_point)

def check_feas(in_point,circ,A,b):
    """
    # Checks feasiblity of circuit circ at point in_point
    # in the polyhedron Ax >= b
    # returns 0 if infeasible, 1 if feasible, 2 if unbounded
    """
    facets, dim = A.shape
    tight = {i for i in range(facets) if (A*in_point - b)[i] == 0}
    #not_tight = {i for i in range(facets) if i not in tight}
    if all(A*circ >= 0):
        feas = 2
    elif all([((A*circ)[i] >= 0) for i in tight]):
        feas = 1
    else:
        feas = 0
    return feas

# Also redefine for taking as input a Poly object instead
def traverse_in_P(in_point,circ,P):
    """
    # Starting at in_point, traverse circ as far as possible
    # inside the polyhedron P: Ax >= b
    # in_point and circ are d x 1
    """
    return traverse(in_point,circ,P.A,P.b)

def check_feas_in_P(in_point,circ,P):
    """
    # Checks feasiblity of circuit circ at point in_point
    # in the polyhedron Ax >= b
    # returns 0 if infeasible, 1 if feasible, 2 if unbounded
    """
    return check_feas(in_point,circ,P.A,P.b)

def tight_facets(in_point,P):
    """
    # Outputs indices of facets of polyhedron P tight at point in_point
    """
```

```
    facets, dim = P.A.shape
    tight = {i for i in range(facets) if (P.A*in_point - P.b)[i] == 0}
    if {i for i in range(facets) if (P.A*in_point - P.b)[i] < 0}:
        print 'Infeasible point'
        return set([-1])
    else:
    #not_tight = {i for i in range(facets) if i not in tight}
        return tight

def row_in_matrix(r,mat):
    """
    # Checks if row r is contained in a matrix mat.
    # Outputs first occurrence it finds if yes,
    # and outputs -1 if it is not a member.
    """
    found = False
    index = -1
    for ind,row in enumerate(mat):
        if np.all(row == r):
            found = True
            index = ind
            break
    print("Found: ", found)
    print("Row: ", index)
    return index

## From https://stackoverflow.com/questions/1987694/print-the-full-numpy-array
## The following class allows for printing of full arrays
class fullprint:
    'context manager for printing full numpy arrays'

    def __init__(self, **kwargs):
        if 'threshold' not in kwargs:
            kwargs['threshold'] = np.nan
        self.opt = kwargs

    def __enter__(self):
        self._opt = np.get_printoptions()
        np.set_printoptions(**self.opt)

    def __exit__(self, type, value, traceback):
        np.set_printoptions(**self._opt)
```