Introduction to Polyhedral Computation

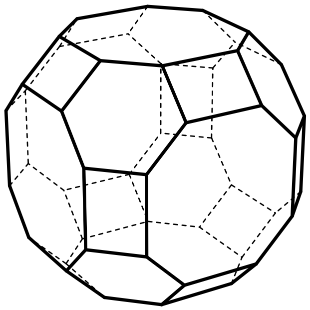
Komei Fukuda

ETH Zurich, Switzerland

fukuda@math.ethz.ch

February 21, 2012

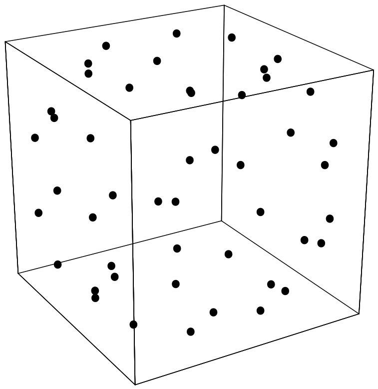
Convex Polyhedra



A convex polyhedron or simply polyhedron P in $\mathbb{R}^d$ is the set of solutions to a (finite) system of linear inequalities in d-variables:

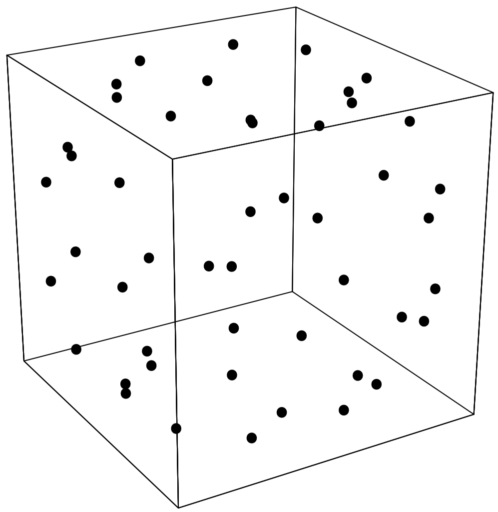
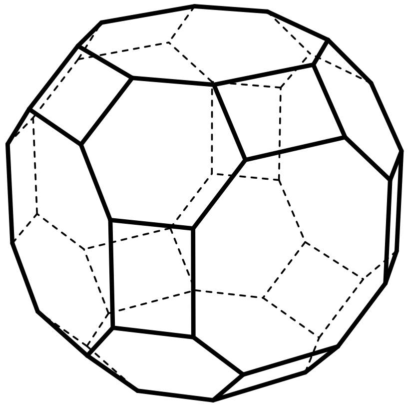
$$P = \{x \in \mathbb{R}^d : Ax \leq b\}$$

where $A \in \mathbb{R}^{m \times d}$ and $b \in \mathbb{R}^m$. A convex polytope is a bounded polyhedron.



A polyhedron is called H-polyhedron (resp. V-polyhedron) if it is given by an inequality system (resp. a set of generators).

Facet Listing (Representation Conversion)

Input A	Output $\lambda_{\mathrm{EXT}}(A)$
	
a set of $n(=48)$ points in $d(=3)$ space, a V-polytope	all $m(=26)$ inequalities, an H-polytope

- It is also known as the Convex Hull problem.
- The reverse problem Vertex Listing is equivalent by duality.
- For $d = 2, 3$, there is an optimal $O(n \log n)$ algorithm.

An Example

```
* filename: mit729-9.ine
* Ternary Alloy Ground State Analysis
* See, Ceder, G., Garbulski, G.D.,  Avis, D. and Fukuda, K.,
* "Ground states of a ternary lattice model with nearest
* and next-nearest neighbor interactions,"
* This polytope has 4862 vertices.
H-representation
begin
   729    9    integer
   12     2     0     0     0     0    -3     0     0
   36     5     1     0     0     0    -6    -3     0
    0     0     0     0     0     0    -1    -2    -1
    0     0     0     0     0     0    -1     0     1
    0     0     0     0     0     0    -1     2    -1
    0    -1     1     0     0     0     1    -1     0
   48    -4    12     0     0     0     3    -6    -9
    0    -2     2     0     0     0     1     0    -3
    0    -1     1     0     0     0     0     0    -3
    0    -1     1     0     0     0     0    -3    -3
    0    -1    -1     0     0     0     1     1     0
    0    -1    -1     0     0     0     0     3    -3
    0    -1    -1     0     0     0     0     0    -3
    0    -2    -2     0     0     0     1     0    -3
   24     2     0     0     0     0    -1     0     0
    .
    .
    .
  320    16    16    -1    -2    -1    -4    -8    -4
    0    -8     8     3     2    -5    -4   -32   -12
    0    -6     2     1     2    -3     1     2   -15
end
```

Polyhedral Computation: When Was It Born (to me)?

- Public releases of representation conversion (RC) codes:
  cdd (KF) v.0.23, and lrs (Avis) v.1.1 in 1993.
  qhull (Barber-Huhdanpaa) v2.b05 in 1994.
- Questions from users started to overwhelm my work in late 1997.
- Polyhedral Computation FAQ in November 26, 1997.
  (Latest version in 2004.)

What is Polyhedral Computation? Parallels in History

- Mathematical Programming $\Longleftarrow$ Major Progress in LP
- Polyhedral Computation $\Longleftarrow$ Major Progress in RC

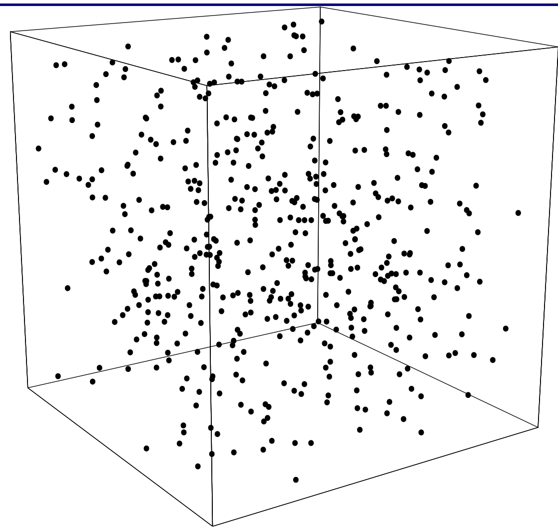
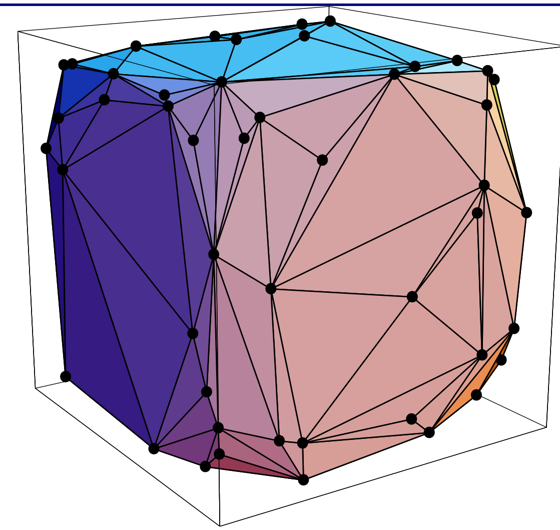
Fundamental Problems in Polyhedral Computation

- Representation Conversion (V-Polytope $\Longleftrightarrow$ H-Polytope)
- Redundancy Removal (for V- and H-Polytopes)
- Arrangement/Zonotope Construction
- Minkowski Addition of Polytopes
- Gröbner Walk and Gröbner Fan Construction
- Multiparametric LP/LCP
- Lattice Points in a Polytope, Polytope Projection, Triangulations, etc.

Ideal Algorithms

- Time-Efficient Algorithm (Polynomial-Time)
- Space-Efficient Algorithm (Compactness)

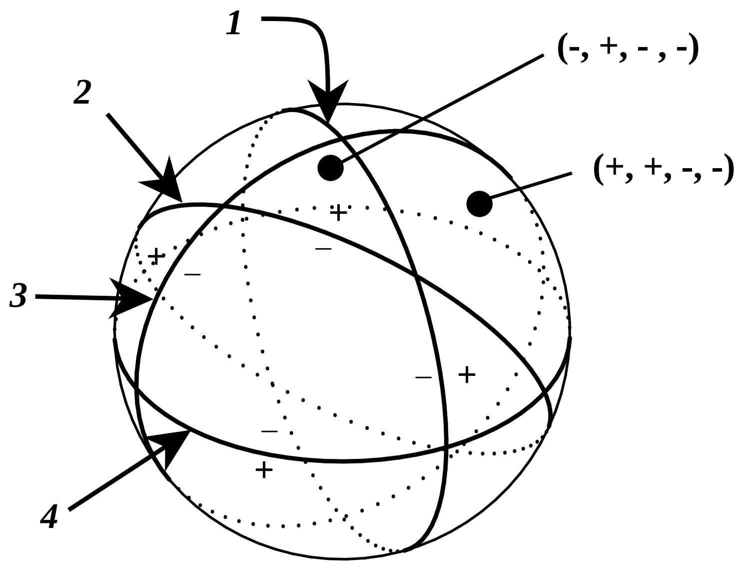
Redundancy Removal (for V-Polytopes)

Input A	**Output** $\lambda_{\mathrm{ESS}}(A)$
	
a set of $n(=500)$ points in $d(=3)$ space, a V-polytope	all $n'(=69)$ extreme points, a minimal V-representation

- In general, much easier than the representation conversion.
  (One can compute it for very large d, by solving many LPs.)

- Yet, in lower dimensions (2, 3), it is faster to use the repr. conversion.

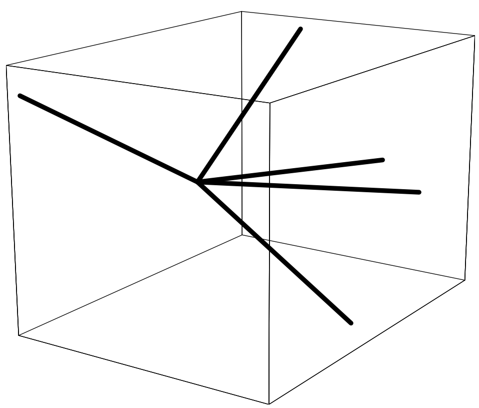
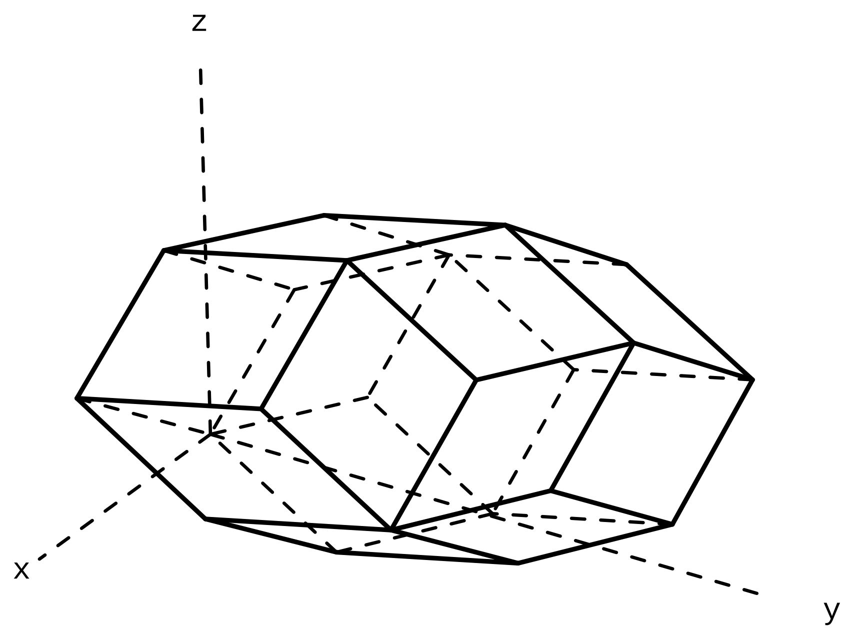
Arrangement Construction

<table>
<tr><th>Input V</th><th>Output $\lambda_{\text{CELL}}(V)$</th></tr>
<tr><td>

$k(=4)$ hyperlane normals
in dimension $d(=3)$</td>
<td>$(-,+,-,-), (+,+,-,-),$
$(-,-,-,-), (+,-,-,-),$
$(+,+,+,-), (-,-,+,-),$
$(-,+,+,-),$
and the negatives.

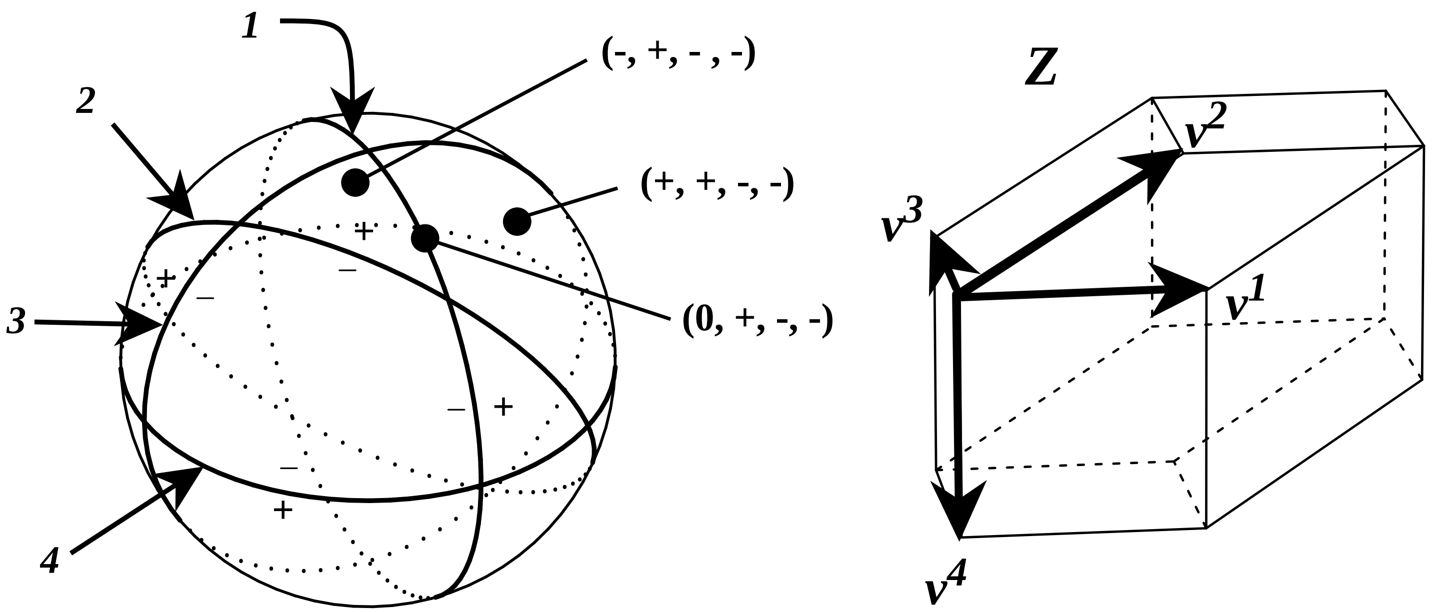
all 14 sign vectors
(the underlying oriented matroid)</td></tr>
</table>

Zonotope Construction

Input $I_1, I_2, \ldots, I_k$	**Output** $\lambda_{\text{ZO}}(I_1, I_2, \ldots, I_k)$
 $k(=5)$ line segments in dimension $d(=3)$	 all $n(=22)$ extreme points of $I_1 + I_2 + \cdots + I_k$

Minkowski-sum: $P = P_1 + P_2 := \{v_1 + v_2 : v_1 \in P_1 \text{ and } v_2 \in P_2\}$.

Duality of Arrangements and Zonotopes

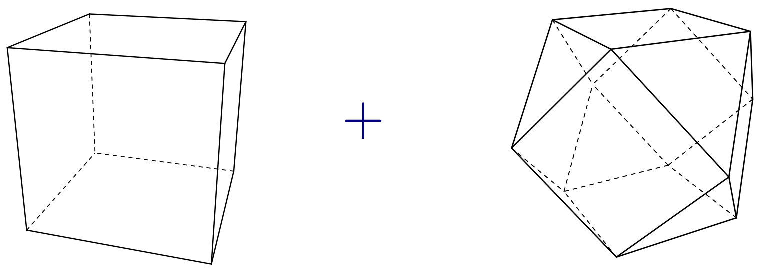
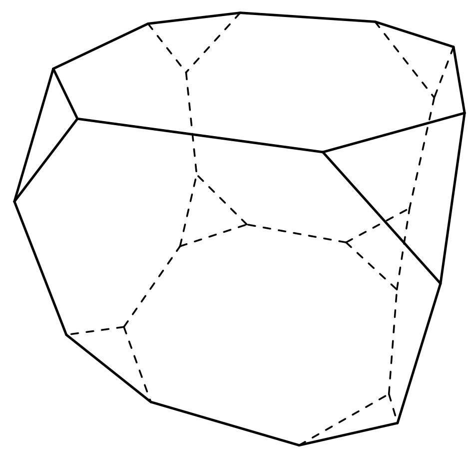


Cells　　　　Extreme points

$$X = (-, +, -, -) \iff z = v^2$$

$$X = (+, +, -, -) \iff z = v^1 + v^2$$

Minkowski Addition of V-Polytopes

Input $V_1, V_2, \ldots, V_k$	**Output** $\lambda_{\mathrm{MI}}(V_1, V_2, \ldots, V_k)$
	
$k(=2)$ V-polytopes in dimension $d(=3)$	all $n(=24)$ extreme points, a V-representation of the sum

Minkowski-sum: $P = P_1 + P_2 := \{v_1 + v_2 : v_1 \in P_1 \text{ and } v_2 \in P_2\}$.

Gröbner Basis Listing (Gröbner Fan Construction)

Input V	**Output** $\lambda_{\mathrm{GR}}(V)$
A polynomial ideal $I =$ $\langle b-a^2, c-a^3, d-a^6, b^3-d\rangle$ $\subset \mathbb{Q}[a,b,c,d]$ $n(=4)$ generating polynomials in $d(=4)$ variables	$\mathcal{G}_0 = \{b-a^2, c-a^3, d-a^6\}$, $\mathcal{G}_1 = \{c^2-d, ab-c, b^2-ac, a^2-b\}$, $\mathcal{G}_2 = \{c^2-d, a^3-c, b-a^2\}$, $\vdots$ $\mathcal{G}_{11} = \{a^6-d, b-a^2, c-a^3\}$. all $m(=12)$ reduced Gröbner bases

For example, b^3-d is redundant in the input, because

$$b^3-d = (b^2+ba^2+a^4)(b-a^2)+(-1)(d-a^6).$$

Ideal Algorithms?

There are no uniformly accepted complexity notions for LISTING problems for which the output size can be LARGE.

Nevertheless, one can extend the notion of polynomial algorithms naturally.

- An algorithm is polynomial if it runs in TIME polynomial in both the input size and the output size. (This is sometimes called "total polynomial" or "output sensitive".)
- An algorithm is compact if it runs in SPACE polynomial in the input size ONLY.

An ideal algorithm is a compact polynomial algorithm.

[Alternative goal: Worst-output-case optimal algorithms.]

Current Status of General Dimensional Polyhedral Computation

Problem	Algorithms	Eff.	Implementations
Representation conversion	IS (Motzkin'53,Grünbaum'63, etc)	!po, !co	cdd, cgal, qhull,...
	RS (Avis-KF '91)	po*, co	lrs
	PD (Bremner-KF-Marzetta '96)	po*, co*	pd (based on lrs)
Arr./Zonotope construction	IS (Edelsbrunner et al '86)	po, !co	
	RS (Avis-KF '92)	po, co	rs_tope(+cddlib)
Minkowski addition	IS (Gritzmann-Sturmfels'93)	!po, !co	
	RS (KF '02)	po, co	minksum(+cddlib)
Gröbner bases	RS (KF-Jensen-Thomas 04')	opo, oco	gfan(+cddlib)

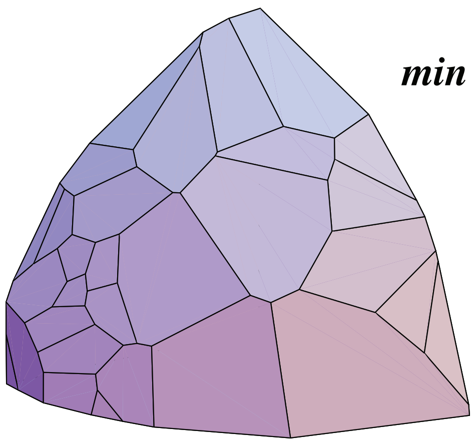
po=polynomial; co= compact; o= oracle; != not; (*)under non-degeneracy

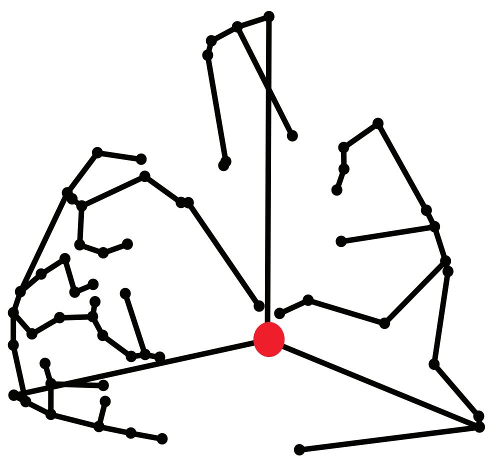
IS = Incremental Search; RS=Reverse S.; PD=Primal-Dual

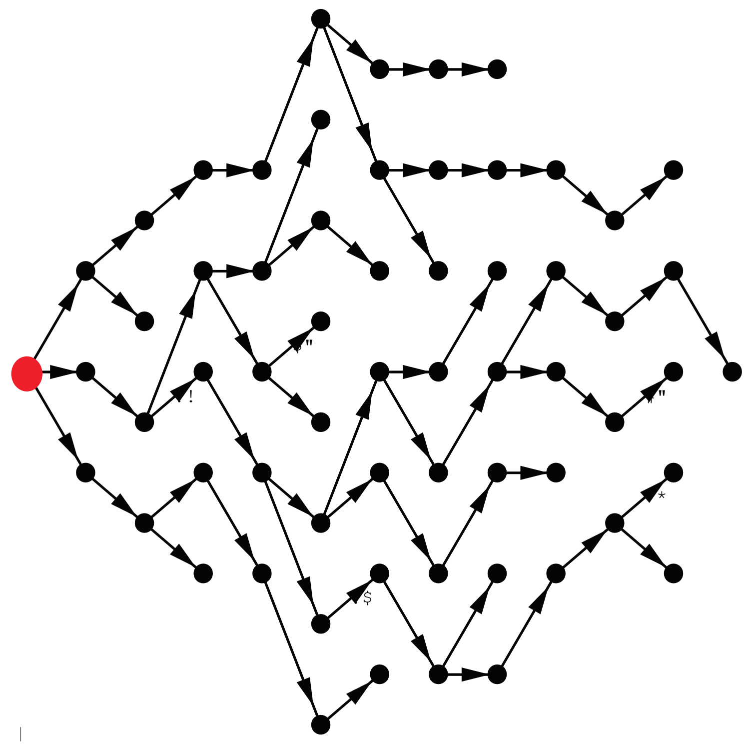
cdd(KF),cgal(many),gfan(Jensen),qhull(Barbar),lrs(Avis),minksum(Weibel),pd(Marzetta)

Reverse Search for Vertex Listing

Reverse the simplex method from the optimal vertex in all possible ways:







Complexity: $O(md\, f_0)$-time and $O(md)$-space under nondegeneracy.

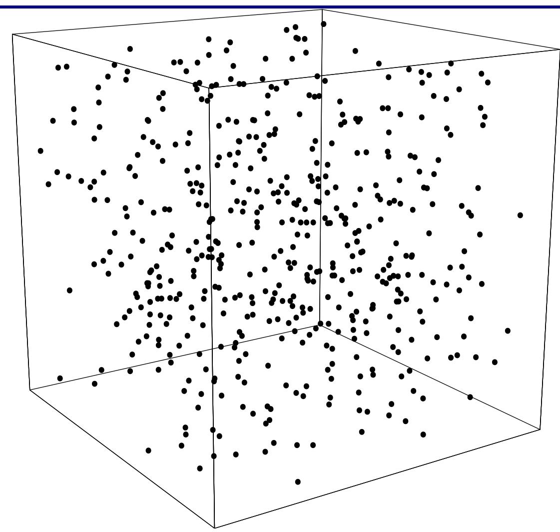
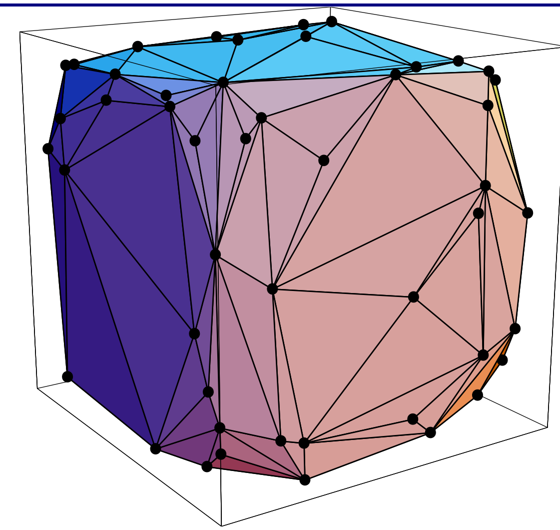
A Challenge in Polyhedral Representation Conversion

Polyhedral Verification Problem (Lovasz):

Given a rational H-polytope P and a rational V-polytope P', decide whether $P = P'$.

- PVP is clearly in coNP.
- Is PVP in coNPC?
  (A substantial progress was made by Khachiyan et al in 2005.)
- PVP is in P $\Longleftrightarrow$ the representation conversion admits an "incrementally" polynomial algorithm.
  (See the Polyhedral Computation FAQ [4] for the only-if part).

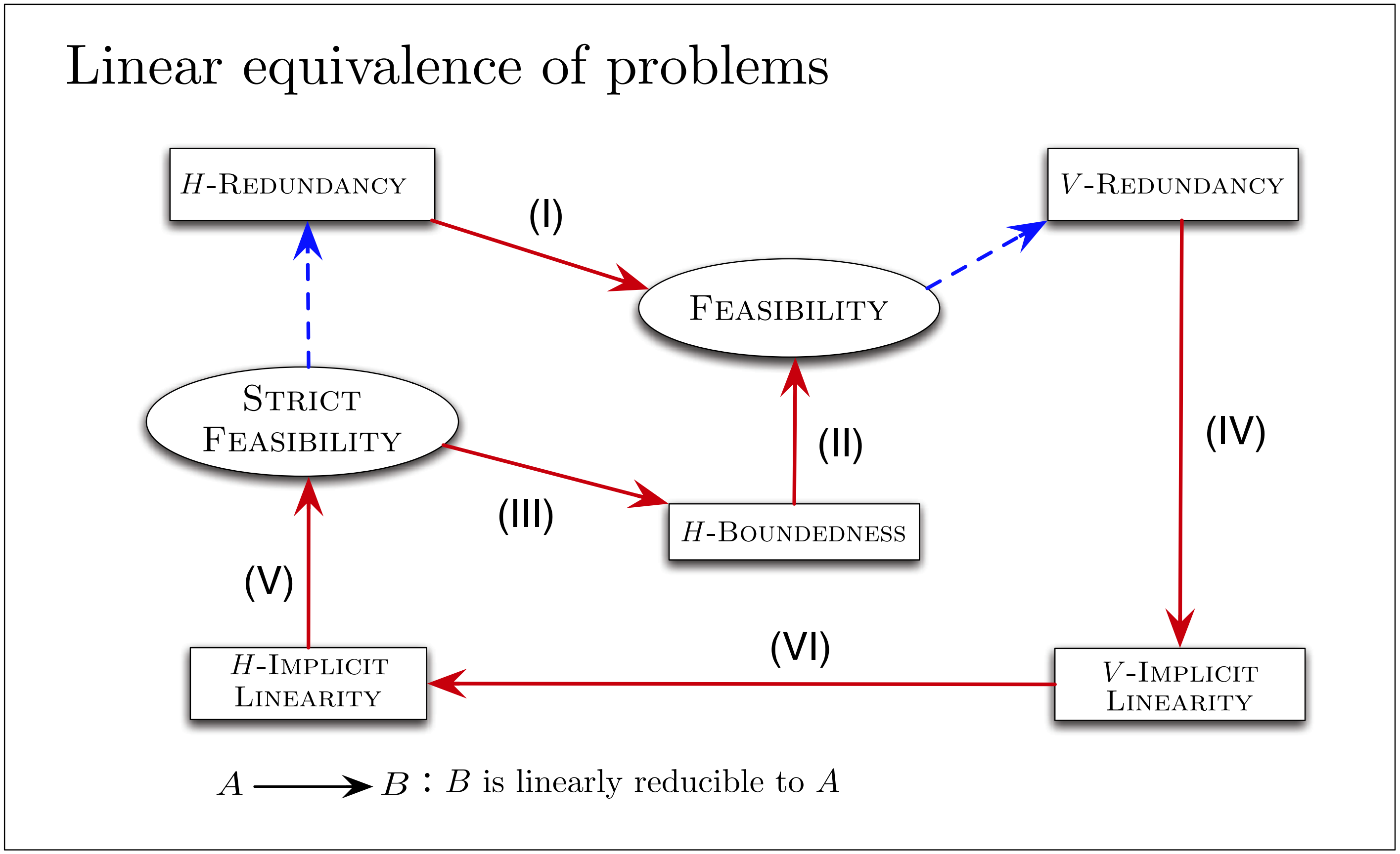
Redundancy Removal (for V-Polytopes)

Input A	**Output** $\lambda_{\text{ESS}}(A)$
 a set of $n (= 500)$ points in $d (= 3)$ space, a V-polytope	 all $n' (= 69)$ extreme points, a minimal V-representation

- In general, much easier than the representation conversion.
  (One can compute it for very large d, by solving many LPs.)

Complexity of Redundancy Removal

Lemma. (Each) Redundancy removal is as hard as LP.



***H*-Redundancy:** Given $A \in \mathbb{Q}^{m \times d}$, $b \in \mathbb{Q}^m$ and $i \in [m]$, determine whether $A_i\, x \leq b_i$ is <u>redundant</u> in the system $A\, x \leq b$.

Complexity of Redundancy Removal

By the linear equivalence lemma, the H-redundancy (or V-redundancy) checking takes time proportional to $\mathrm{LP}(m,d)$, that is, the time necessary to solve a linear program of size $m \times d$.

However, one can do better to remove all redundancies than the trivial bound $m \times \mathrm{LP}(m,d)$.

Theorem. (Clarkson '94)

An algorithm to detect all redundancies from an H(V)-representation in time $m \times \mathrm{LP}(s,d)$ exists, where $s (\leq m)$ is the number of essential inequalities (points).

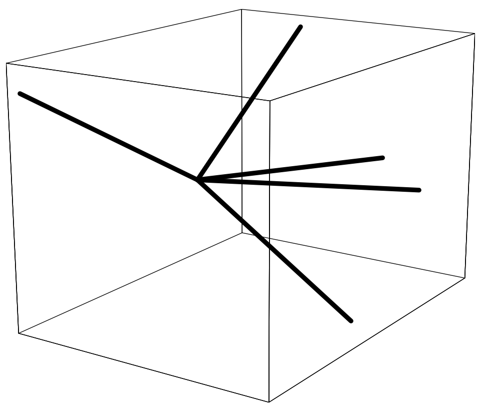
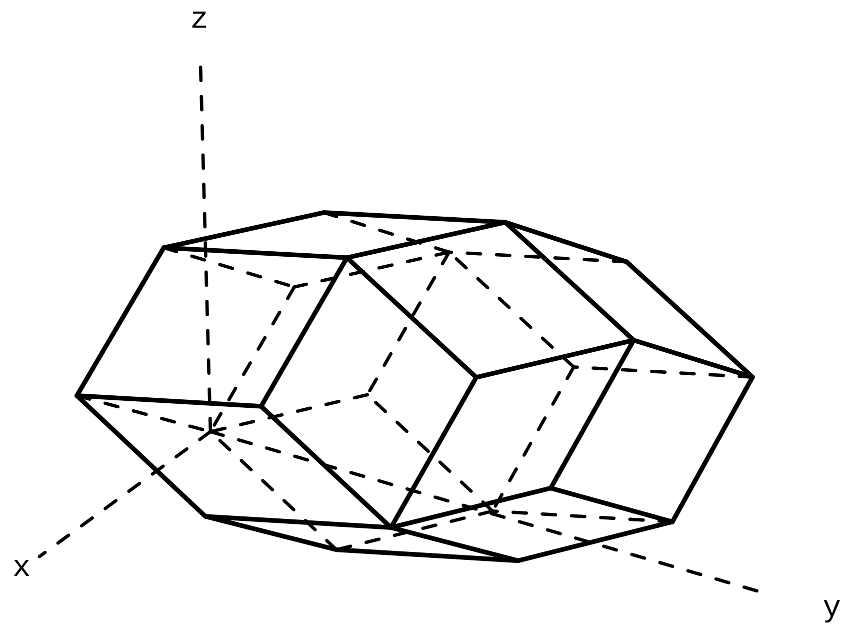
A Challenge in Redundancy Removal

Can One Do Better Than Clarkson?

Is there any algorithm to remove all redundancies from an H(V)-representation which runs faster than Clarkson's algorithm?

- Can one exploit similarities of the LP's solved by LP-based algorithms?
- Can one design a randomized algorithm which runs faster (in the expected sense)?

Zonotope Construction

Input $I_1, I_2, \ldots, I_k$	**Output** $\lambda_{\mathrm{ZO}}(I_1, I_2, \ldots, I_k)$
 $k(=5)$ line segments in dimension $d(=3)$	 all $n(=22)$ extreme points of $I_1 + I_2 + \cdots + I_k$

Arrangement and Zonotope Construction

There are different approaches.

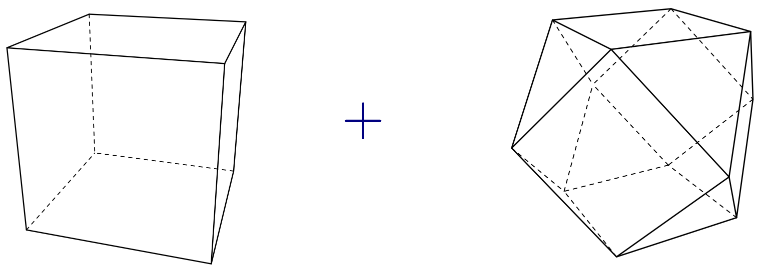
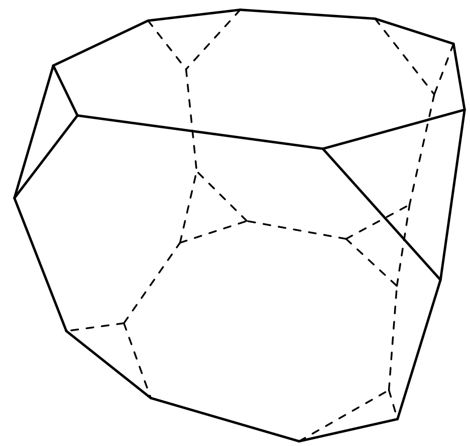
Theorem [Edelsbrunner-O'Rourke-Seidel '86].
For $d \geq 3$, there exists an incremental algorithm to generate all vertices of a zonotope given by k generators in $\mathbb{R}^d$ in $O(k^{d-1})$ time and $O(k^{d-1})$ space for fixed d.

This algorithm is worst-case optimal, but it is neither polynomial nor compact.

Theorem [Avis-KF '96 and Ferrez-KF-Liebling '01].
There exists a reverse search algorithm to generate all v vertices in $O(k\,\mathrm{LP}(k,d)\,v)$ time and $O(k\,d)$ space.

This algorithm is both polynomial and compact.

Minkowski Addition of V-Polytopes

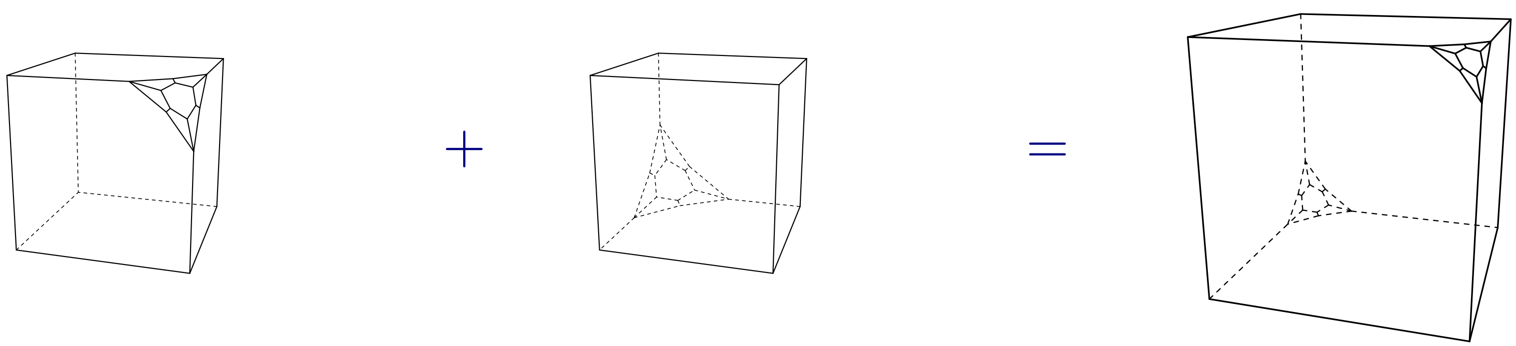
Input $V_1, V_2, \ldots, V_k$	**Output** $\lambda_{\mathrm{MI}}(V_1, V_2, \ldots, V_k)$
 $k(=2)$ V-polytopes in dimension $d(=3)$	 all $n(=24)$ extreme points, a V-representation of the sum

Minkowski-sum: $P = P_1 + P_2 := \{v_1 + v_2 : v_1 \in P_1 \text{ and } v_2 \in P_2\}$.

Complexity of Minkowski Addition of Polytopes

Sometimes, the Minkowski sum of polytopes is very simple and its vertex complexity is linear bounded by the input size.

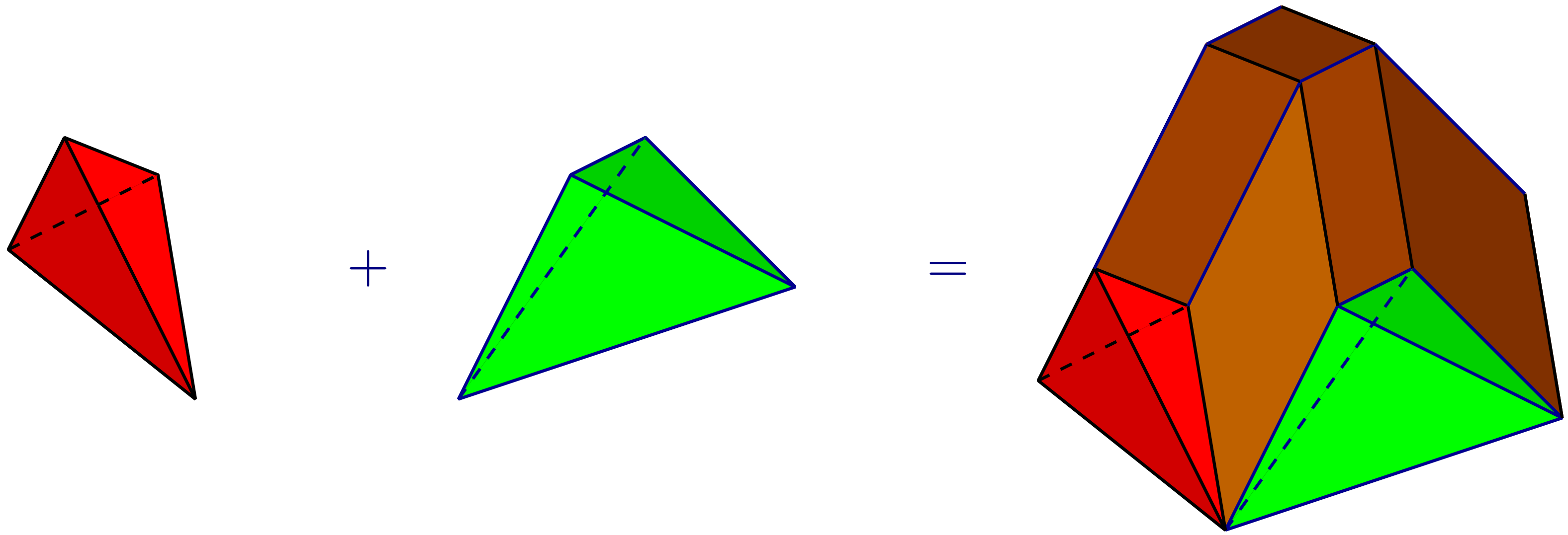
Proposition (Linearly Bounded Minkowski-Addition). For each $k \geq 2$ and $d \geq 2$, there is an infinite family of Minkowski additions for which $f_0(P_1 + P_2 + \cdots + P_k) \leq f_0(P_1) + f_0(P_2) + \cdots + f_0(P_k)$.



Complexity of Minkowski Addition of Polytopes

Theorem (Tight Upper Bound) [KF-Weibel '07 [7]].
In dimension $d \geq 3$, it is possible to choose k ($\leq d-1$) polytopes $P_1, \ldots, P_k$ so that the trivial upper bound for the number of vertices is attained by their Minkowski sum.

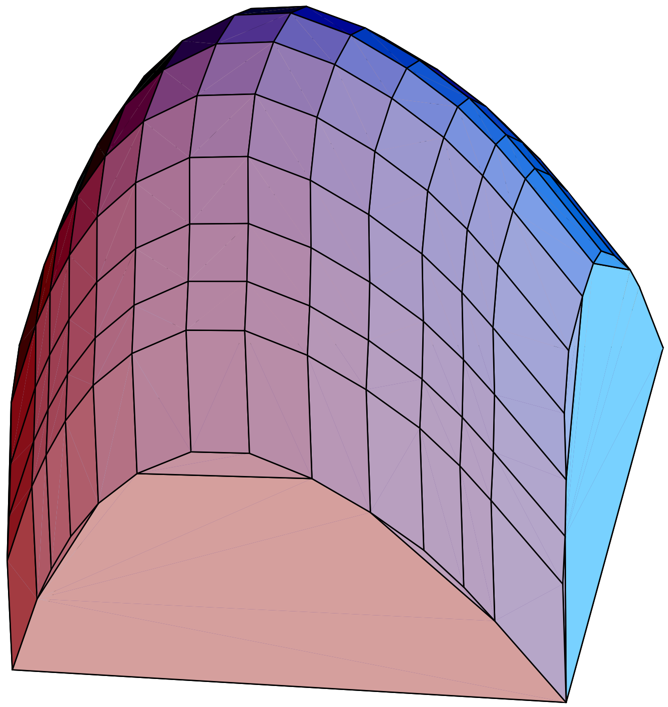
$$f_0(P_1 + P_2 + \cdots + P_k) = f_0(P_1) \times f_0(P_2) \times \cdots \times f_0(P_k).$$



Complexity of Minkowski-Addition of Polytopes

$$f_0(P_1) = f_0(P_2) = 14$$

$$f_0(P_1 + P_2) = f_0(P_1) \times f_0(P_2) = 196$$

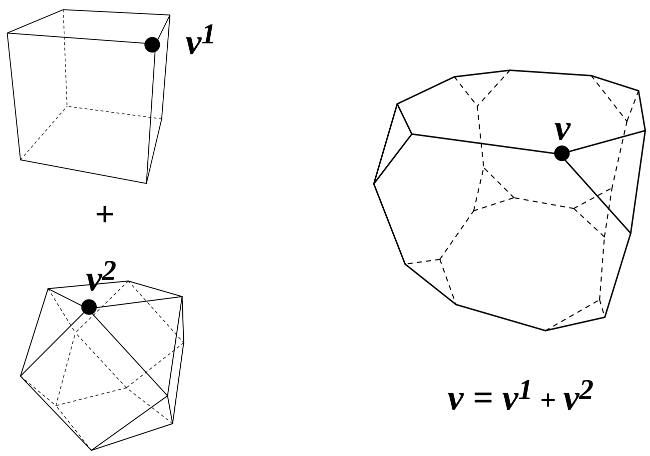


Gritzmann-Sturmfels' Alogrithm I (1993)

This is an input-polynomial algorithm for fixed k.

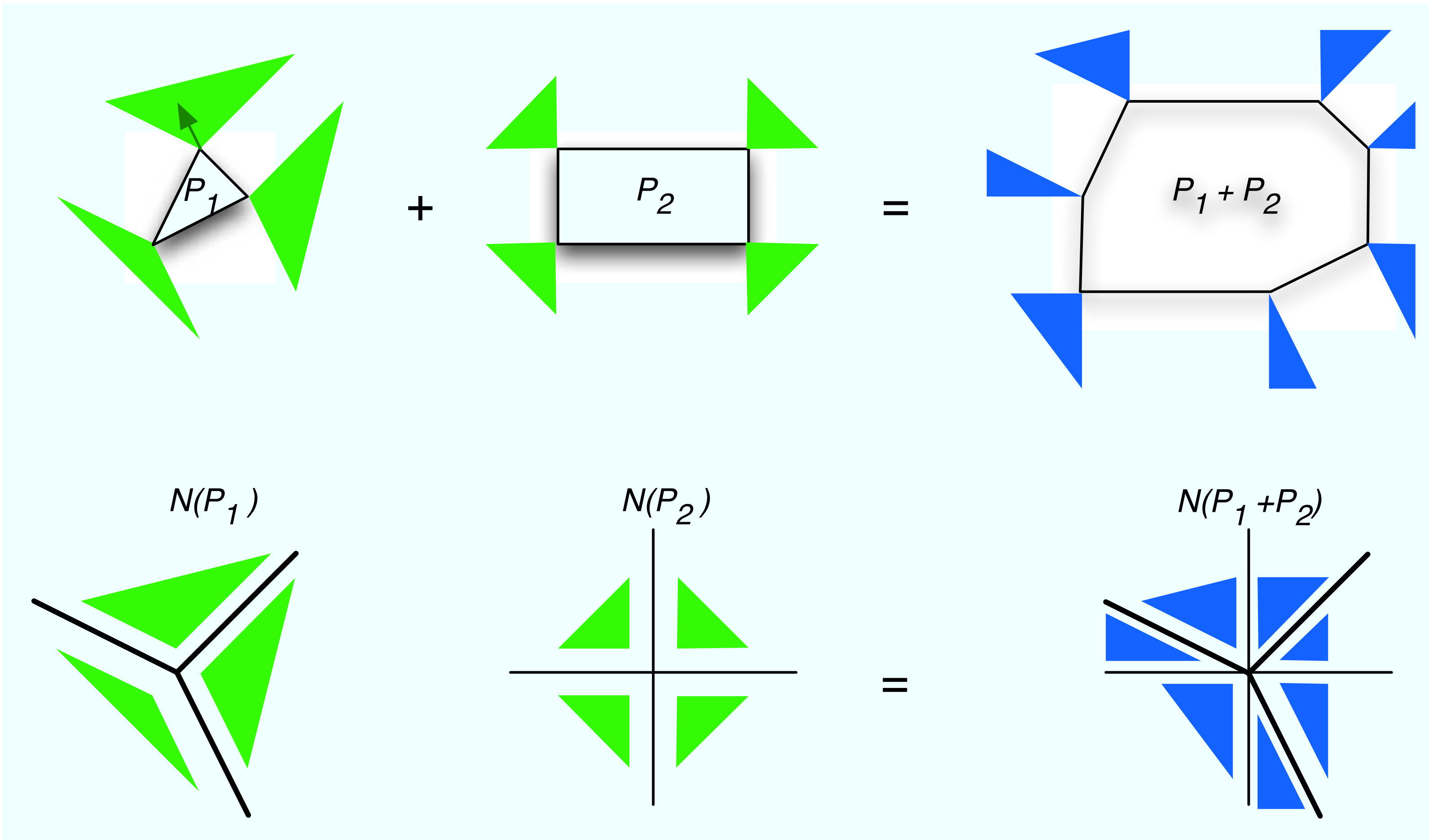
Input: $V_1, V_2, \ldots, V_k$

Algorithm: For all tuples $(v^1, v^2, \ldots, v^k)$ with $v^i \in V_i$,
decide whether $v = v^1 + v^2 + \cdots + v^k$ is extreme in $P_1 + P_2 + \cdots + P_k$.
(This can be done by solving an LP.)



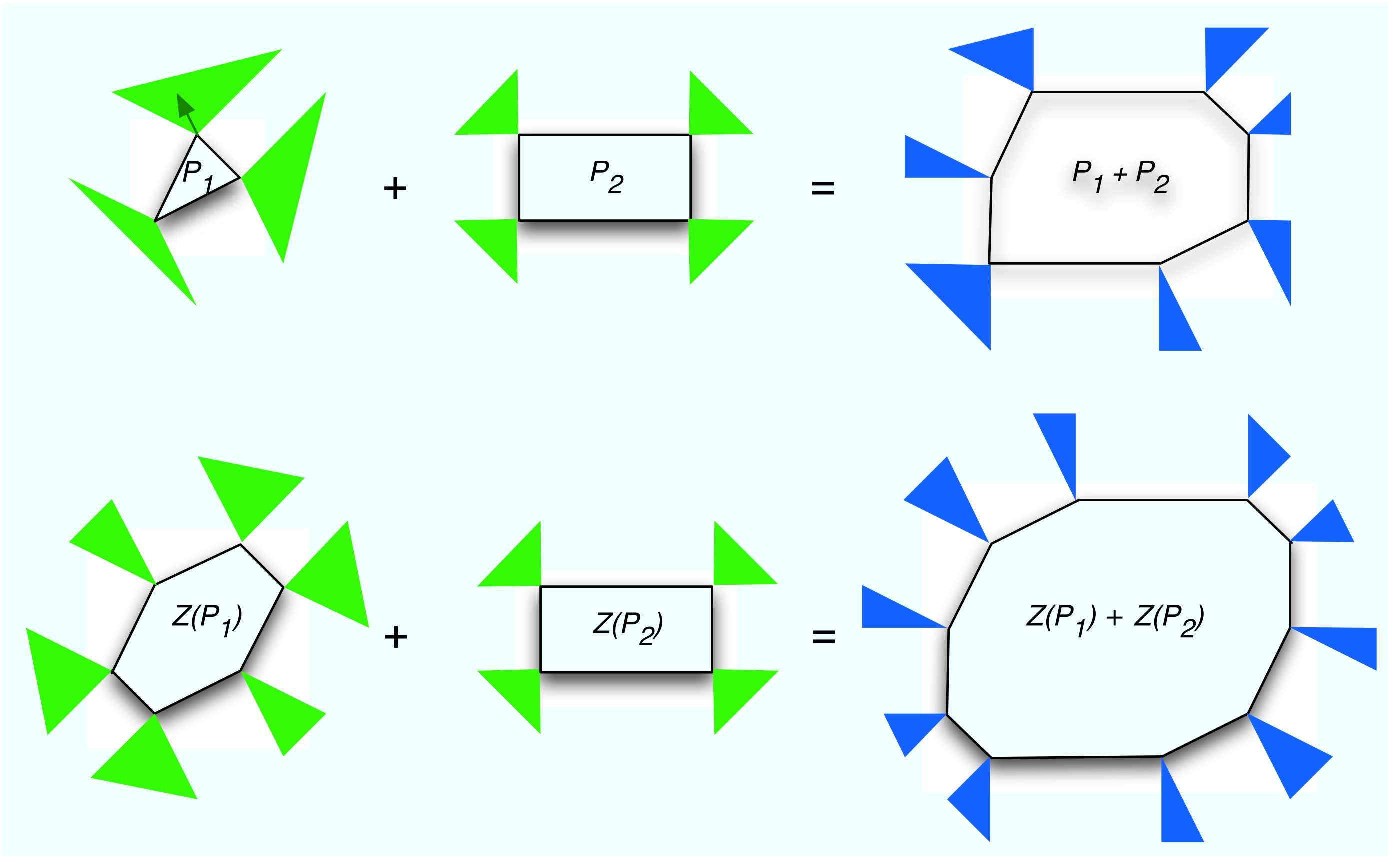
Complexity: $O(s \cdot LP(s-d, s))$, where $s = |V_1| \times |V_2| \times \cdots \times |V_k|$.

Minkowski Addition and Outer Normal Cones/Fans



Computing the Minkowski addition can be considered as superimposing the fans of outer normal cones.

Gritzmann-Sturmfels' Algorithm II (1993)

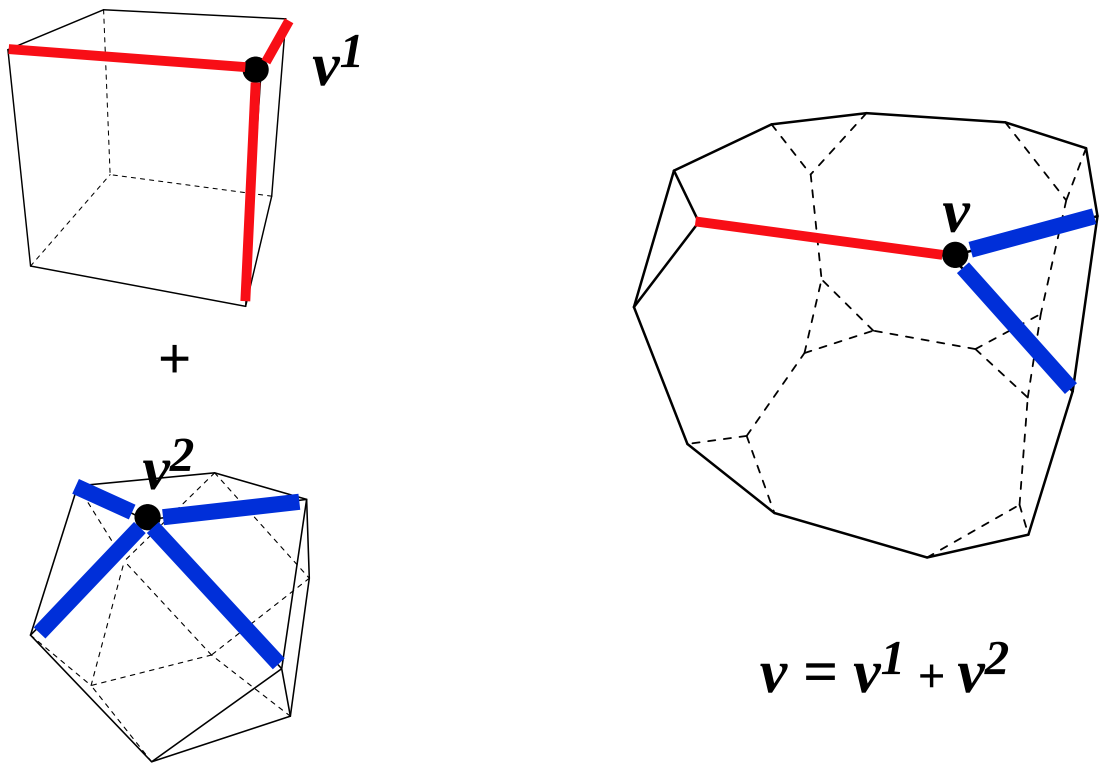


Compute $N(Z(P_1)+Z(P_2))$ by the incremental zonotope construction algorithm in $O(m^{d-1})$ time and $O(m^{d-1})$ space. Then, merge some cones to get $N(P_1+P_2)$.

New Idea: Adjacency in the Minkowski Addition

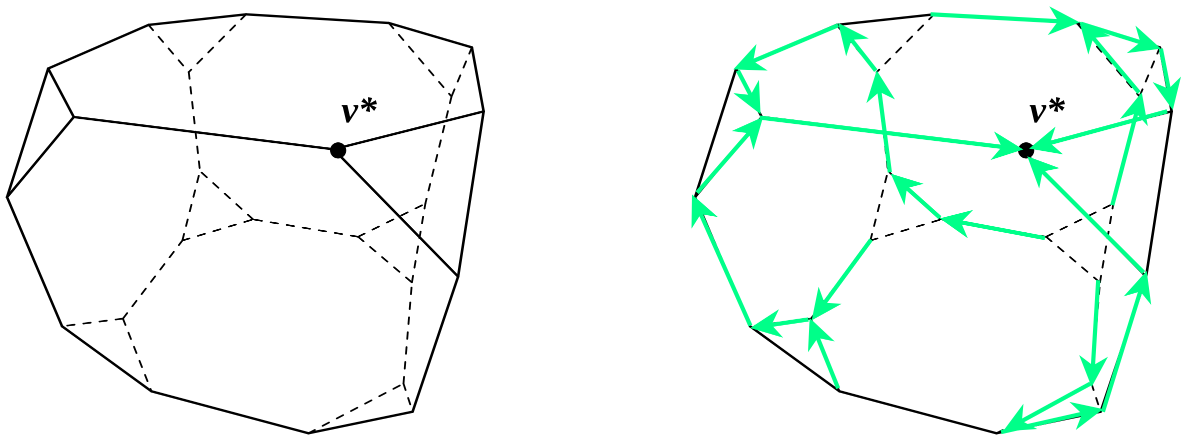
Listing all neighbors of a given vertex v is easy via LPs.

They are inherited from adjacency in the corresponding vertices of P_i's.



Reverse Search for the Minkowski Addition

Define a unique directed spanning tree rooted at any fixed vertex v^*, e,g, the simplex pivot tree (with a fixed rule).



A reverse search algorithm traces reversely the tree from the root v^* in depth-first manner, using an adjacency oracle.

Time complexity: $O(\delta LP(\delta, d) f_0(P))$, where δ is the sum of the max degrees of $G(P_i)$'s.

An Implementation of Minkowski Sum by Weibel (2005)

- A parallel implementation of the reverse search algorithm is freely available: **minksum** by Christophe Weibel.
- It is written in C++, using GMP and the rational arithmetic LP code in **cddlib** by KF.

Experiments (The sum of a simplex and its dual) on Pentium 1.7 MHz

d	cpu (sec)	cpu_init	cpu_lp	#vert	#edges	#lp	lp_size
10	4.21	0.79	1.79	110	990	704	20x11
20	91.91	16.39	51.74	420	7980	3004	40x21
30	601.61	108.28	371.06	930	26970	6904	60x31

The Hardest Problem Solved

A Minkowski sum of 9 polytopes in $\mathbb{R}^{27}$, each of which has only 6 vertices. It took about a month to generate all $2,372,583$ vertices.

A Challenge in Minkowski Sum of Polytopes

How hard is to compute the facets of a Minkowski sum?

Worst Case Behavior (Upper Bound Theorem)?

Given two polytopes P_1 and P_2 with n_1 and n_2 vertices each, what is the maximum number of facets $P_1 + P_2$ can have?

- The maximum number of vertices is $n_1 \times n_2$ (KF-Weibel 2005).
- The tight bound for facets is known for $d \leq 3$ (KF-Weibel 2006).
- This question relates closely to finding an efficient algorithm to list all facets of the sum.

Gröbner Basis Listing (Gröbner Fan Construction)

Input V	**Output** $\lambda_{\mathrm{GR}}(V)$
A polynomial ideal $I =$ $\langle b-a^2, c-a^3, d-a^6, b^3-d\rangle$ $\subset \mathbb{C}[a,b,c,d]$	$\mathcal{G}_0 = \{b-a^2, c-a^3, d-a^6\},$ $\mathcal{G}_1 = \{c^2-d, ab-c, b^2-ac, a^2-b\},$ $\mathcal{G}_2 = \{c^2-d, a^3-c, b-a^2\},$ $\vdots$ $\mathcal{G}_{11} = \{a^6-d, b-a^2, c-a^3\}.$
$n(=4)$ generating polynomials in $d(=4)$ variables	all $m(=12)$ reduced Gröbner bases

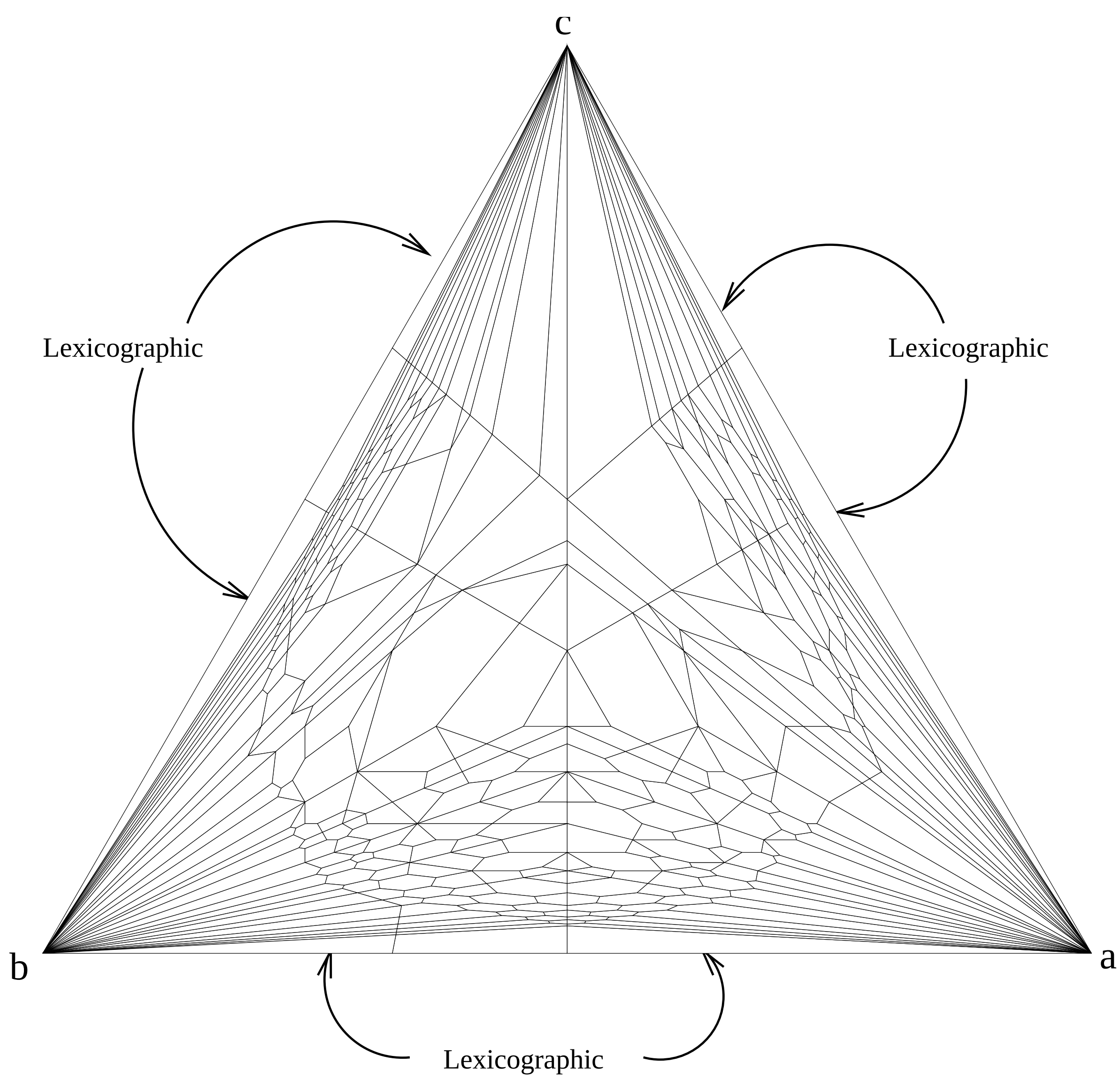
An Implementation of a Gröbner Fan Algorithm (by Jensen 2004)

- A faithful implementation of the reverse search algorithm due to KF-Jensen-Thomas, an extended version of Sturmfels' Algorithm (1995): **gfan** by Anders Jensen.
- It is written in C++, using both GMP and the rational arithmetic LP code in cddlib by KF.

A Computed Example (Example 3.9 in Sturmfels' book)

The ideal $I = \langle a^5 + b^3 + c^2 - 1, a^2 + b^2 + c - 1, a^6 + b^5 + c^3 - 1 \rangle$ has exactly 360 Göbner bases. It took 105 seconds on a laptop (1.8 GHz AMD XP-M). One third of the time is spent in the LP solving.

The Gröbner Fan of $\langle a^5+b^3+c^2-1, a^2+b^2+c-1, a^6+b^5+c^3-1\rangle$



Multiparametric LCP

(Given $M \in \mathbb{R}^{n \times n}$: S-matrix, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times s}$)

$$\textbf{LCP:} \quad \begin{aligned} & w = Mz + q, w \geq 0, z \geq 0 \\ & w^T z = 0. \end{aligned}$$

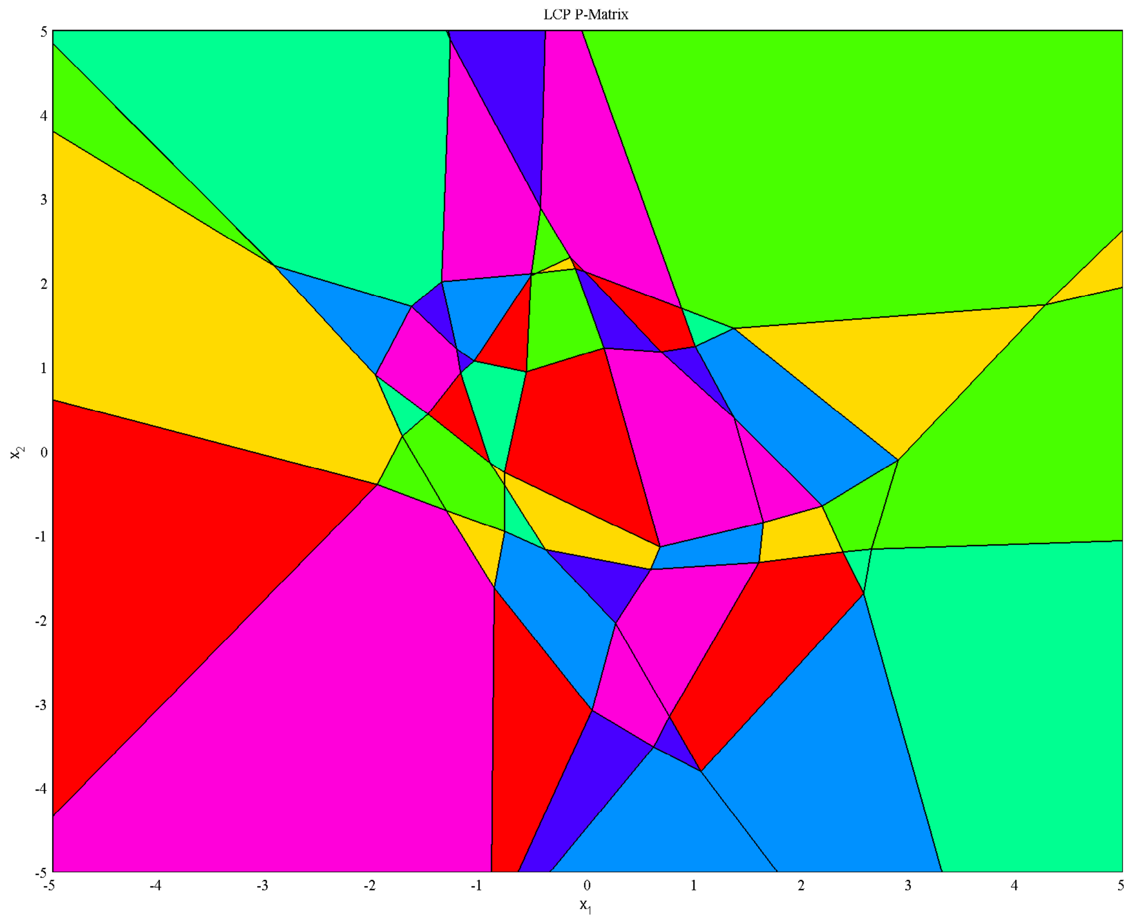
$$\textbf{pLCP}(\theta)\textbf{:} \quad \begin{aligned} & w = Mz + q + A\theta, w \geq 0, z \geq 0 \\ & w^T z = 0. \end{aligned}$$

Goal: Pre-solve pLCP(θ) for all possible $\theta \in \mathbb{R}^s$.

This amounts to partition the parameter space into full-dimensional "critical" regions each of which has a unique (lexico) optimal basis.

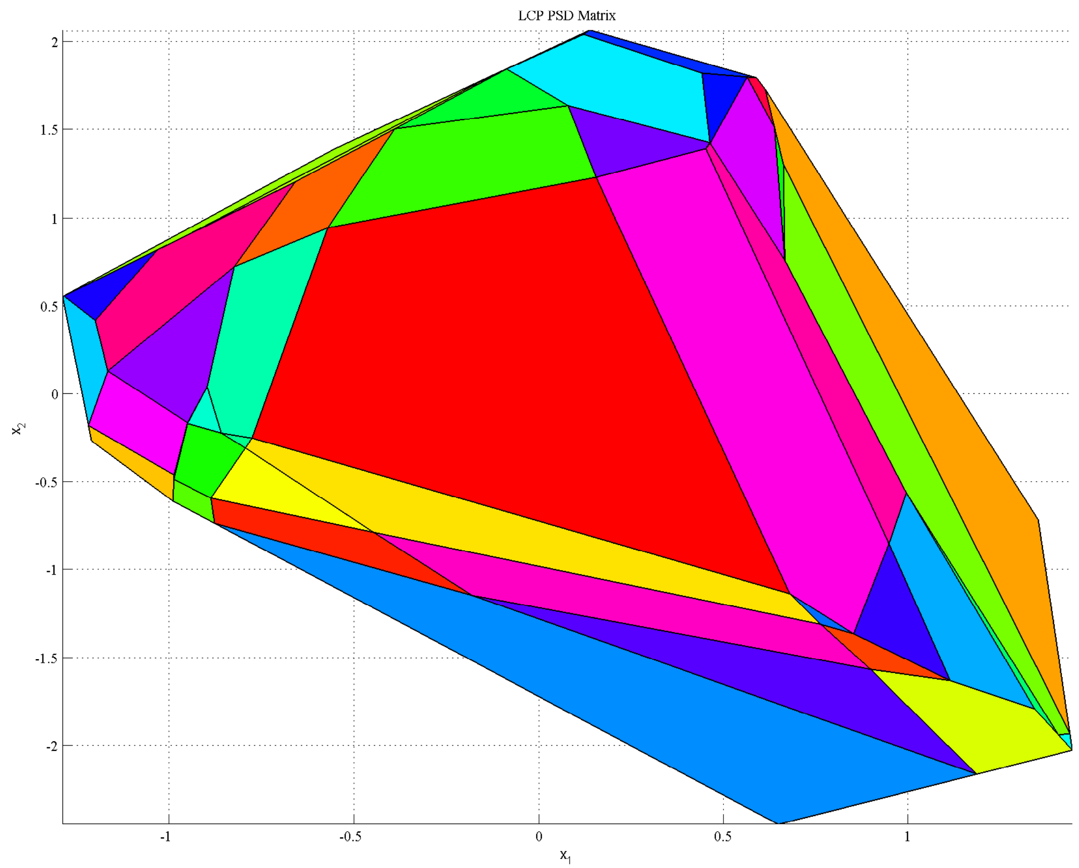
A recent work by Jones et al (2006) shows that the closure of a critical region is a polyhedron but the partition is not a cell complex in general.

The Set of All Critical Regions: A P-Matrix Example



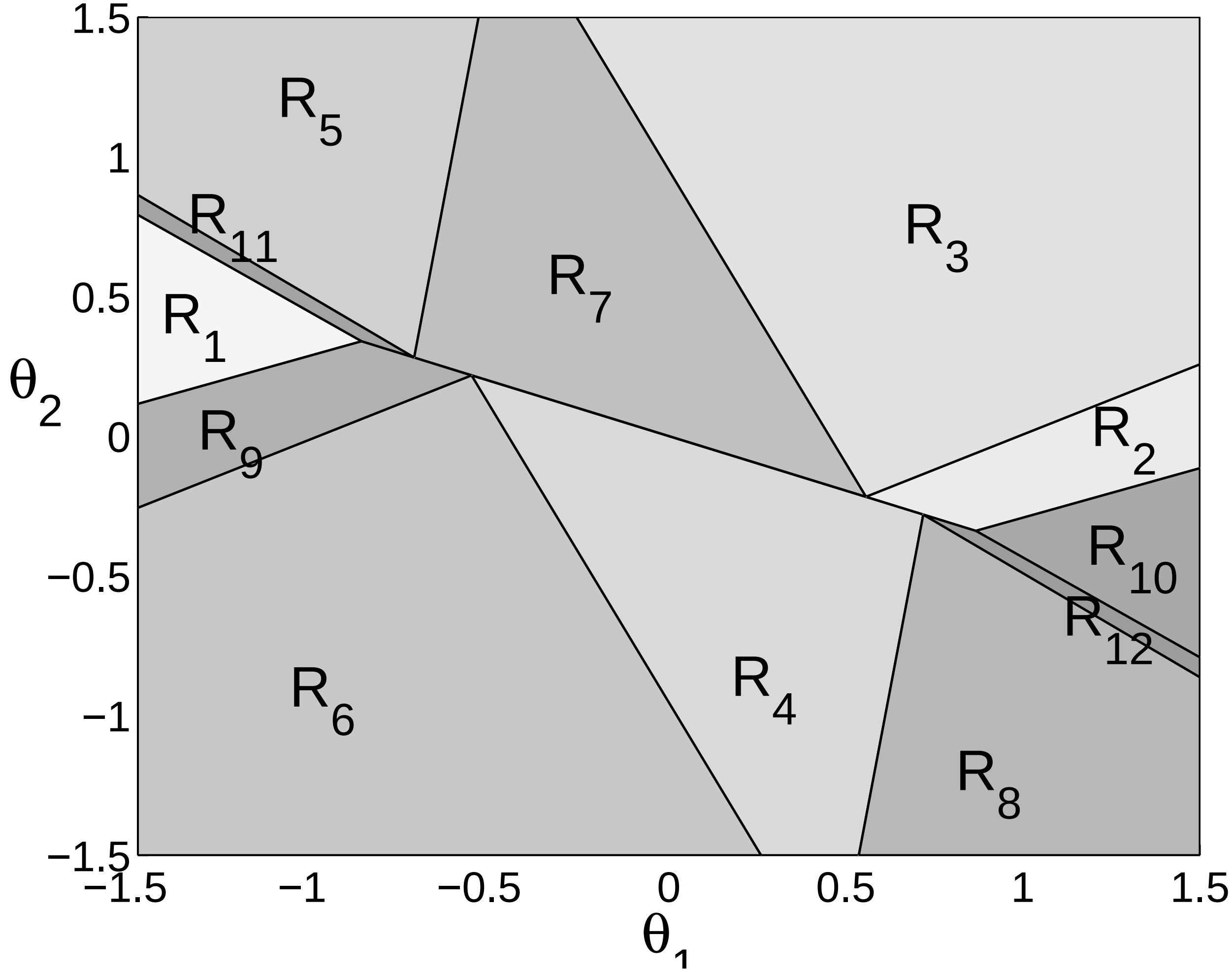
$n = 10, d = 2$. The figure generated by the MPT toolbox, ETH Zurich.

The Set of All Critical Regions: A PSD Example



$n = 10, d = 2$. The figure generated by the MPT toolbox, ETH Zurich.

The Set of All Critical Regions: A PSD Example (Not a Cell Complex)



The figure taken from Spjotvold et al. 2005.

Sufficient pLCPs are Well-Behaving

$K(M)$: the complementary range, i.e. the set of all q's such that LCP(M,q) has a solution.

Q_0: the class of all matrices M such that $K(M)$ is convex.

D: the class of fully semimonotone matrices.

Proposition: Let $M \in D \cap Q_0$ (e.g. a sufficient matrix). For all $q \in K(M)$, there exists $\delta > 0$ such that LCP$(M, q + (\varepsilon^1, \varepsilon^2, \ldots, \varepsilon^n)^T)$ has a unique feasible complementary basis for all $\varepsilon \in (0, \delta)$.

Corollary: There is a canonical map τ for sufficient pLCPs.

Columbano-KF-Jones (2009) gave an exact algorithm to compute this canonical map. It is polynomial if the problem is nondegenerate.

Concluding Remarks

- Polyhedral computation is a rapidly growing research domain with applications in science, engineering, social sciences, etc.
- The availability of open-source software packages has increased the popularity of polyhedral computation methods considerably.
- There are many challenging open problems in polyhedral computation: the PVP problem, exploiting symmetries, Minkowski H-additions, faster redundancy removal, computing polyhedral projections, counting lattice points, etc.
- Many instances in applications are too hard to solve exactly. Possibilities to approximate the output should be explored.

References

[1] D. Avis. lrs Homepage. http://cgm.cs.mcgill.ca/~avis/C/lrs.html.

[2] S. Columbano, K. Fukuda, and C. Jones. An output-sensitive algorithm for multi-parametric lcps with sufficient matrices. In D. Avis, D. Bremner, and A. Deza, editors, Polyhedral Computation, volume 48 of CRM Proceedings and Lecture Notes, pages 73–102. Amer. Math. Soc., Providence, RI, 2009.

[3] K. Fukuda. From the zonotope construction to the Minkowski addition of convex polytopes. Journal of Symbolic Computation, 38(4):1261–1272, 2004.

[4] K. Fukuda. Polyhedral computation FAQ. Both html and ps versions available from http://www.ifor.math.ethz.ch/~fukuda/fukuda.html.

[5] K. Fukuda. cdd, cddplus and cddlib homepage. http://www.ifor.math.ethz.ch/~fukuda/cdd_home/index.html.

[6] K. Fukuda, A. Jensen, and R. Thomas. Computing Gröbner fans. Mathematics of Computation, 76:2189–2212, 2007.

[7] K. Fukuda and C. Weibel. f-vectors of Minkowski additions of convex polytopes. Discrete Comput. Geom., 37:503–516, 2007.

[8] A.N. Jensen. Gfan version 0.4: A user's manual, 2009. available from http://www.math.tu-berlin.de/~jensen/software/gfan/gfan.html.

[9] C. Weibel. Minksum version 1.6.2. available from McGill University, http://www.math.mcgill.ca/~weibel/.