Efficient edge skeleton and volume computation for polytopes defined by oracles

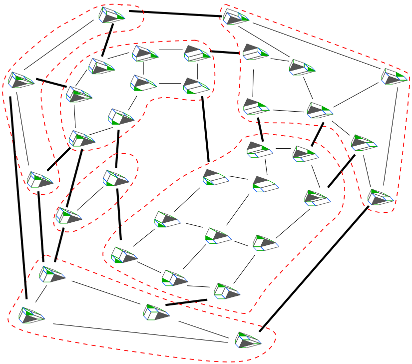
Vissarion Fisikopoulos

Joint work with I.Z. Emiris (UoA), B. Gärtner (ETHZ)

Dept. of Informatics & Telecommunications, University of Athens

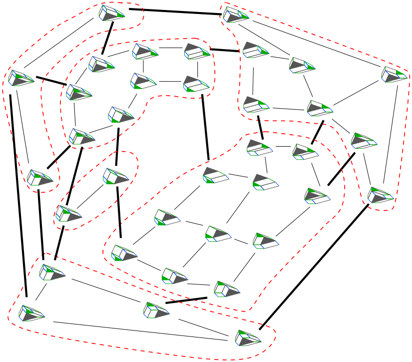
McGill, 20.Jun.2013

Main motivation: resultant polytopes



Vertices $\rightarrow$ equilavent classes of regular triangulations of a pointset's convex hull.

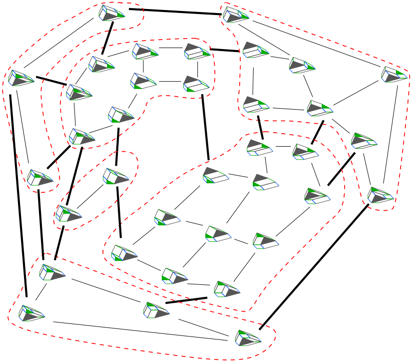
Main motivation: resultant polytopes



Vertices → equilavent classes of regular triangulations of a pointset's convex hull.

- Algorithm: [Emiris F Konaxis Peñaranda '12]
  vertex oracle + incremental construction = output-sensitive

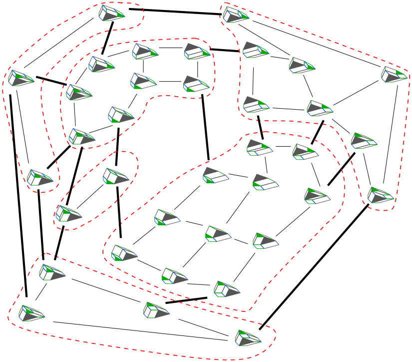
Main motivation: resultant polytopes



Vertices → equilavent classes of regular triangulations of a pointset's convex hull.

- Algorithm: [Emiris F Konaxis Peñaranda '12]
  vertex oracle + incremental construction = output-sensitive
- Software: computation in < 7 dimensions

Main motivation: resultant polytopes

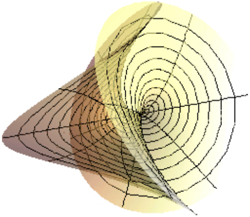


Vertices → equilavent classes of regular triangulations of a pointset's convex hull.

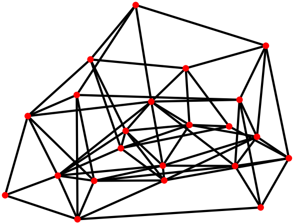
- Algorithm: [Emiris F Konaxis Peñaranda '12]
  vertex oracle + incremental construction = output-sensitive
- Software: computation in < 7 dimensions
- Q: Can we compute in dim. > 7 (edge-skeleton, volume) ?

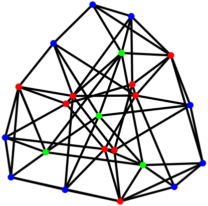
Applications

- Geometric Modeling (Implicitization) [EmirisKalinkaKonaxisLuuBa'12]



- Combinatorics of 4-d resultant polytopes [Dickenstein Emiris F '13]





Outline

Polytope Representation & Oracles

Edge Skeleton Computation

Geometric Random Walks & Volume approximation

Motivation: Resultant polytopes

Experimental Results

Outline

Polytope representation

Convex polytope $P \in \mathbb{R}^{n}$.

Explicit: Vertex-, Halfspace - representation (V_P, H_P),
Edge-sketelon (ES_P), Triangulation (T_P)

Polytope representation

Convex polytope $P \in \mathbb{R}^n$.

Explicit: Vertex-, Halfspace - representation (V_P, H_P),
Edge-sketelon (ES_P), Triangulation (T_P)

$V_P \to H_P$: convex hull problem, $H_P \to V_P$ vertex enum. problem

- open: $\exists$ total poly-time algorithm, i.e. poly(input,output)

Polytope representation

Convex polytope $P \in \mathbb{R}^{n}$.

Explicit: Vertex-, Halfspace - representation (V_P, H_P),
Edge-sketelon (ES_P), Triangulation (T_P)

$V_P \to H_P$: convex hull problem, $H_P \to V_P$ vertex enum. problem

- open: $\exists$ total poly-time algorithm, i.e. poly(input,output)

Implicit: Oracles (OPT_P, SEP_P, MEM_P)

Polytope representation

Convex polytope $P \in \mathbb{R}^n$.

Explicit: Vertex-, Halfspace - representation (V_P, H_P),
Edge-sketelon (ES_P), Triangulation (T_P)

$V_P \to H_P$: convex hull problem, $H_P \to V_P$ vertex enum. problem

- open: $\exists$ total poly-time algorithm, i.e. poly(input,output)

Implicit: Oracles (OPT_P, SEP_P, MEM_P)

We study algorithms for polytopes given by OPT_P:

- Resultant, Discriminant, Secondary polytopes
- Minkowski sums

Well-described polytope and oracles

Definition

A rational polytope $P \subseteq \mathbb{R}^n$ is well-described (with a parameter φ) if there exists an H-representation for P in which every inequality has encoding length at most φ. The encoding length of P is $\langle P \rangle = n + \varphi$.

Proposition (Grötschel et al.'93)

For a well-described polytope, we can compute OPT_P from SEP_P (and vice versa) in oracle polynomial-time. The runtime (polynomially) depends on n and φ.

Outline

Vertex enumeration with edge-directions

Given OPT_P and a superset D of the edge directions $D(P)$ of $P \subseteq \mathbb{R}^n$, compute the vertices P.

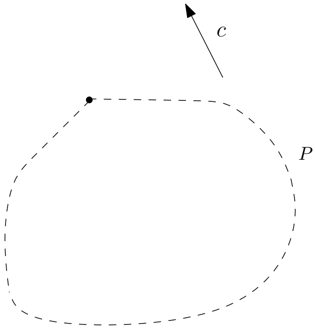
Proposition (Rothblum Onn '07)

Let $P \subseteq \mathbb{R}^n$ *given by* OPT_P, *and* $D \supseteq D(P)$. *All vertices of* P *can be computed in*

$$O(|D|^{n-1}) \text{ calls to } \mathrm{OPT}_P + O(|D|^{n-1}) \text{ arithmetic operations.}$$

- Computes the Mink. sum (zonotope) Z of the unit vectors supported on D.
- Computes an arbitrary vector v in the normal cone of each vertex of Z and calls OPT_P with v.

Edge skeleton computation with edge-directions



Input:

- OPT_P
- Edge vec. P (dir. & len.): D

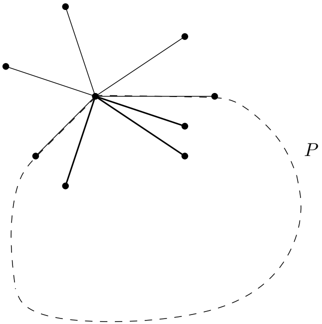
Output:

- Edge-skeleton of P

Sketch of **Algorithm**:

- Compute a vertex of P ($x = \mathrm{OPT}_P(c)$ for arbitrary $c^T \in \mathbb{R}^n$)

Edge skeleton computation with edge-directions



Input:

- OPT_P
- Edge vec. P (dir. & len.): D

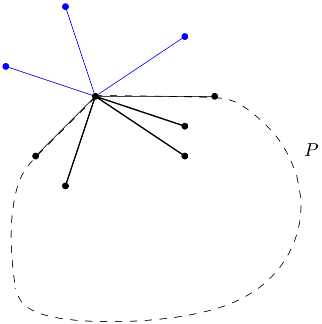
Output:

- Edge-skeleton of P

Sketch of **Algorithm**:

- Compute a vertex of P ($x = \mathrm{OPT}_P(c)$ for arbitrary $c^T \in \mathbb{R}^n$)
- Compute segments $S = \{(x, x + d)$, for all $d \in D\}$

Edge skeleton computation with edge-directions



Input:

- OPT_P
- Edge vec. P (dir. & len.): D

Output:

- Edge-skeleton of P

Sketch of **Algorithm**:

- Compute a vertex of P ($x = \mathrm{OPT}_P(c)$ for arbitrary $c^T \in \mathbb{R}^n$)
- Compute segments $S = \{(x, x + d)$, for all $d \in D\}$
- Remove from S all segments (x, y) s.t. $y \notin P$ ($\mathrm{OPT}_P \rightarrow \mathrm{SEP}_P$)

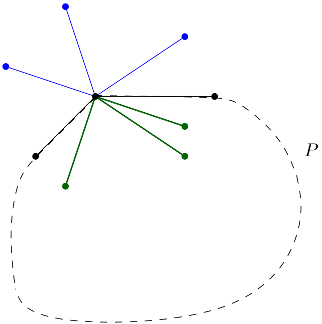
Edge skeleton computation with edge-directions

Input:

- OPT_P
- Edge vec. P (dir. & len.): D

Output:

- Edge-skeleton of P



Sketch of **Algorithm**:

- Compute a vertex of P ($x = \mathrm{OPT}_P(c)$ for arbitrary $c^T \in \mathbb{R}^n$)
- Compute segments $S = \{(x, x + d)$, for all $d \in D\}$
- Remove from S all segments (x, y) s.t. $y \notin P$ ($\mathrm{OPT}_P \to \mathrm{SEP}_P$)
- Remove from S the segments that are not extreme

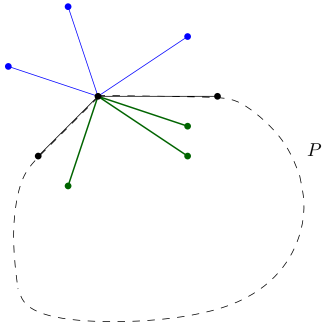
Edge skeleton computation with edge-directions

Input:

- $\mathrm{OPT}_{\mathrm{P}}$
- Edge vec. P (dir. & len.): D

Output:

- Edge-skeleton of P



Sketch of **Algorithm**:

- Compute a vertex of P ($x = \mathrm{OPT}_{\mathrm{P}}(c)$ for arbitrary $c^T \in \mathbb{R}^n$)
- Compute segments $S = \{(x, x + d)$, for all $d \in D\}$
- Remove from S all segments (x, y) s.t. $y \notin P$ ($\mathrm{OPT}_{\mathrm{P}} \rightarrow \mathrm{SEP}_{\mathrm{P}}$)
- Remove from S the segments that are not extreme
- Can be altered to work with edge directions only

Runtime of the algorithm

Theorem

Given OPT_P *and a superset of edge directions* D *of a well-described polytope* P*, the edge skeleton of* P *can be computed in oracle total polynomial-time*

$$O\left(m\left(|D|\mathbb{O}(\langle P\rangle + \langle D\rangle) + \mathbb{LP}(4n^3|D|(\langle P\rangle + \langle D\rangle))\right)\right),$$

- $\langle D\rangle$ *is the binary encoding length of the vector set* D*,*
- m *the number of vertices of* P*,*
- $\mathbb{O}(\langle P\rangle)$: *runtime of oracle conversion algorithm for* P*,*
- $\mathbb{LP}(\langle A\rangle + \langle b\rangle + \langle c\rangle)$ *runtime of max* $c^T x$ *over* $\{x : Ax \leq b\}$*.*

Workspace efficient variant by employing reverse search.

Applications

Given polytopes $P_1, \dots, P_r \subseteq \mathbb{R}^n$ signed Minkowski sum combines Minkowski sums and differences, namely

$$P = P_1 + s_2 P_2 + \cdots + s_r P_r,\ s_i \in \{-1, 1\},$$

assuming P is a polytope.

Corollary

Given OPT oracles for well-described $P_1, \dots, P_r \subseteq \mathbb{R}^n$*, and supersets of edge directions* $D_1, \dots, D_r$*, the edge skeleton of* P *can be computed in oracle total polynomial-time.*

- Similar results for resultant, secondary and discriminant polytopes.

More applications

Convex combinatorial optimization: given $\mathcal{F} \subset 2^N$ with $N = \{1, \ldots, m\}$, a vectorial weighting $w : N \to \mathbb{Q}^n$, and a convex functional $c : \mathbb{Q}^n \to \mathbb{Q}$, find $F \in \mathcal{F}$ of maximum value $c(w(F))$.

- [Rothblum Onn '04] polynomial algorithm for fixed n.

Convex integer programming: maximize a convex function over the integer hull of a polyhedron.

- [De Loera et al. '09] polynomial algorithms for many interesting cases; all edges are computed via Graver bases.

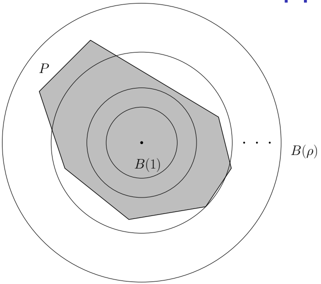
Outline

Polytope volume computation

Given polytope $\mathrm{P} \subset \mathbb{R}^n$ computing it volume is:

- $\#$-P hard for P in V- or H-representation [Dyer '88],
- open if both representations are available [Fukuda '00].

Efficient volume approximation [Dyer et.al'91]



Volume approximation of P reduces to uniform sampling from P

Proposition (Lovász et al.'04)

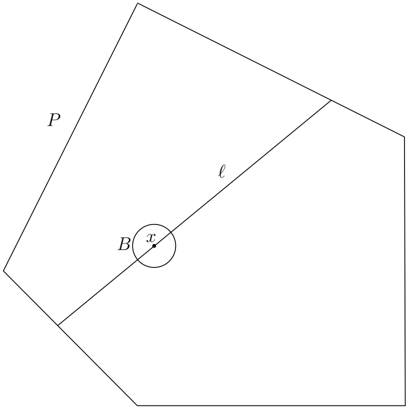
The volume of $P \subseteq \mathbb{R}^n$, *given by* MEM_P *oracle s.t.* $B(1) \subseteq P \subseteq B(\rho)$, *can be approximated with relative error* ε *and probability* $1-\delta$ *using*

$$O\left(\frac{n^4}{\varepsilon^2}\log^9\frac{n}{\varepsilon\delta} + n^4\log^8\frac{n}{\delta}\log\rho\right) = O^*(n^4)$$

oracle calls.

Note: $O^*(\cdot)$ *hides polylog factors in argument and error parameter*

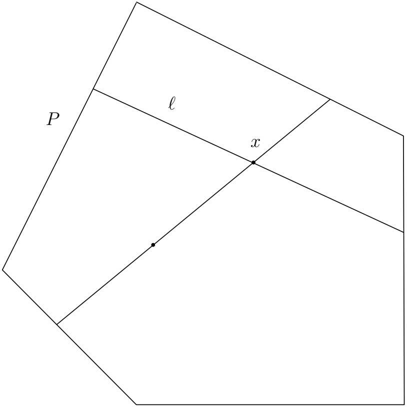
Random points in polytopes with MEM_P



Hit-and-Run walk

- line ℓ through x, uniform on $B_x(1)$
- move x to a uniform disrtibuted point on $P \cap \ell$

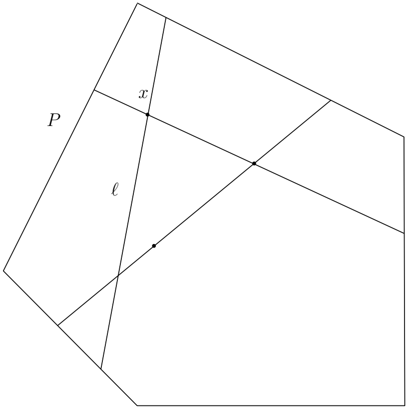
Random points in polytopes with MEM_P



Hit-and-Run walk

- line ℓ through x, uniform on $B_x(1)$
- move x to a uniform disrtibuted point on $P \cap \ell$

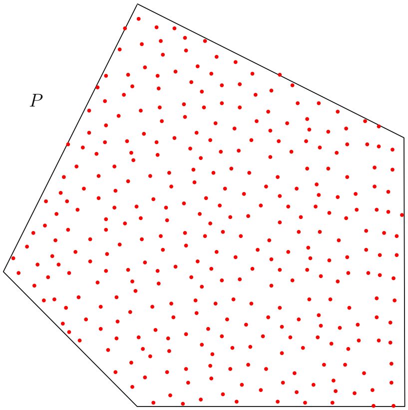
Random points in polytopes with MEM_P



Hit-and-Run walk

- line ℓ through x, uniform on $B_x(1)$
- move x to a uniform disrtibuted point on $P \cap \ell$

Random points in polytopes with MEM_P

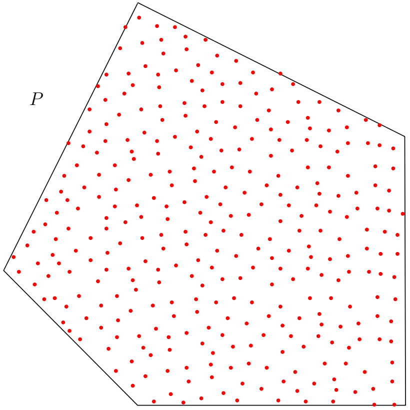


Hit-and-Run walk

- line ℓ through x, uniform on $B_x(1)$
- move x to a uniform disrtibuted point on $P \cap \ell$

x will be "uniformly disrtibuted" in P after $O(n^3)$ hit-and-run steps [Lovász98]

Random points in polytopes with $\mathrm{OPT_P}$



1. Hit-and-Run walk
   with $\mathrm{OPT} \rightarrow MEM$ in every step

Volume of polytopes given by OPT_P

Input: OPT_P, ρ: $B(1) \subseteq P \subseteq B(\rho)$

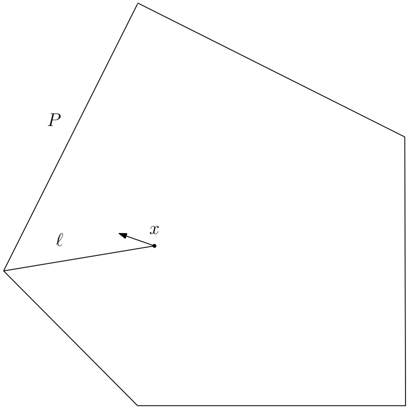
Output: ϵ-approximation $vol(P)$

- Call volume algorithm
- Each MEM_P oracle calls feasibility/optimization algorithm

Corollary

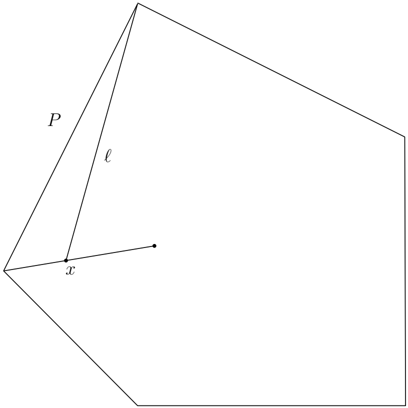
An approximation of the volume of (signed) Minkowski sums and resultant, secondary, discriminant polytopes (given by OPT *oracles) can be computed in oracle polynomial time.*

Random points in polytopes with $\mathrm{OPT_P}$ REVISITED



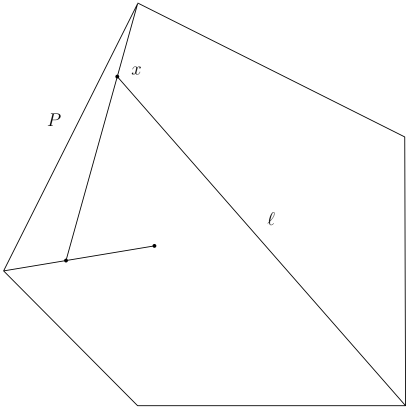
1. Hit-and-Run walk
   with $\mathrm{OPT} \rightarrow MEM$ in every step
2. Vertex walk
   - for uniform c compute $\mathrm{OPT_P}(c)$
   - segment ℓ, connect x, $\mathrm{OPT_P}(c)$
   - move x to a uniform disrtibuted point on ℓ

Random points in polytopes with $\mathrm{OPT_P}$ REVISITED



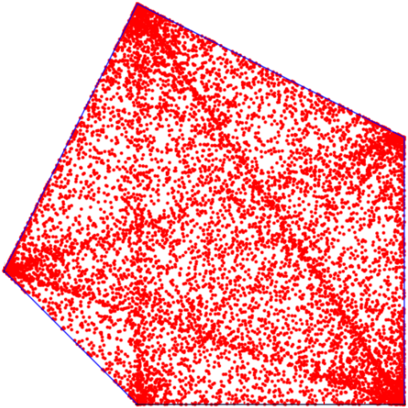
1. Hit-and-Run walk
   with $\mathrm{OPT} \to MEM$ in every step
2. Vertex walk
   - for uniform c compute $\mathrm{OPT_P}(c)$
   - segment ℓ, connect x, $\mathrm{OPT_P}(c)$
   - move x to a uniform disrtibuted point on ℓ

Random points in polytopes with $\mathrm{OPT_P}$ REVISITED



1. Hit-and-Run walk
   with $\mathrm{OPT} \to MEM$ in every step
2. Vertex walk
   - for uniform c compute $\mathrm{OPT_P}(c)$
   - segment ℓ, connect x, $\mathrm{OPT_P}(c)$
   - move x to a uniform disrtibuted point on ℓ

Random points in polytopes with OPT_P REVISITED



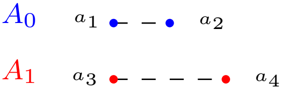
1. Hit-and-Run walk
   with $\mathrm{OPT} \to MEM$ in every step
2. Vertex walk
   - for uniform c compute $\mathrm{OPT}_\mathrm{P}(c)$
   - segment ℓ, connect x, $\mathrm{OPT}_\mathrm{P}(c)$
   - move x to a uniform disrtibuted point on ℓ

Open problem: Generate uniform points in P using OPT_P

Outline

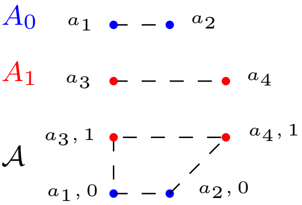
What is a resultant polytope?

- Given $n+1$ point sets $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$



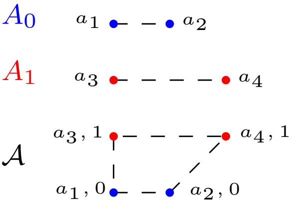
What is a resultant polytope?

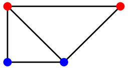
- Given $n+1$ point sets $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$
- $\mathcal{A} = \bigcup_{i=0}^{n} (A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$ where $e_i = (0, \ldots, 1, \ldots, 0) \subset \mathbb{Z}^n$



What is a resultant polytope?

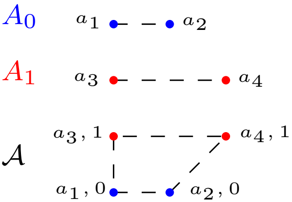
- Given $n + 1$ point sets $A_0, A_1, \dots, A_n \subset \mathbb{Z}^n$
- $\mathcal{A} = \bigcup_{i=0}^{n} (A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$ where $e_i = (0, \dots, 1, \dots, 0) \subset \mathbb{Z}^n$
- Given T a triangulation of $conv(\mathcal{A})$, a cell is a-mixed if it contains n 1-dimensional segments from A_j, $j \neq i$, and some vertex $a \in A_i$.

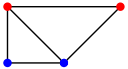




What is a resultant polytope?

- Given $n+1$ point sets $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$
- $\mathcal{A} = \bigcup_{i=0}^{n} (A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$ where $e_i = (0, \ldots, 1, \ldots, 0) \subset \mathbb{Z}^n$
- Given T a triangulation of $conv(\mathcal{A})$, a cell is a-mixed if it contains n 1-dimensional segments from A_j, $j \neq i$, and some vertex $a \in A_i$.
- $\rho_T(a) = \sum_{\substack{a-\text{mixed} \\ \sigma \in T : a \in \sigma}} \text{vol}(\sigma) \ \in \mathbb{N}, \quad a \in \mathcal{A}$

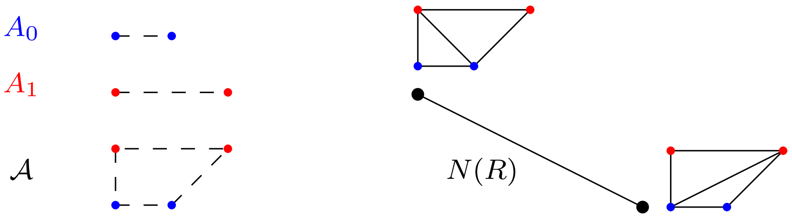




$\rho_T = (0, 2, 1, 0)$

What is a resultant polytope?

- Given $n+1$ point sets $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$
- $\mathcal{A} = \bigcup_{i=0}^{n} (A_i \times \{e_i\}) \subset \mathbb{Z}^{2n}$ where $e_i = (0, \ldots, 1, \ldots, 0) \subset \mathbb{Z}^n$
- Given T a triangulation of $conv(\mathcal{A})$, a cell is a-mixed if it contains n 1-dimensional segments from A_j, $j \neq i$, and some vertex $a \in A_i$.
- $\rho_T(a) = \sum_{\substack{a-\text{mixed} \\ \sigma \in T : a \in \sigma}} \text{vol}(\sigma) \ \in \mathbb{N}, \quad a \in \mathcal{A}$
- Resultant polytope $N(R) = conv(\rho_T : T \ \text{ triang. of } conv(\mathcal{A}))$

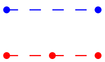


Connection with Algebra

- The Newton polytope of f, $N(f)$, is the convex hull of the set of exponents of its monomials with non-zero coefficient.
- The resultant R is the polynomial in the coefficients of a system of polynomials which vanishes if there exists a common root in the torus of the given polynomials.

A_0

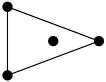
A_1



$$f_0(x) = ax^2 + b$$

$$f_1(x) = cx^2 + dx + e$$

$N(R)$



$$R(a,b,c,d,e) = ad^2b + c^2b^2 - 2caeb + a^2e^2$$

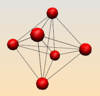
Connection with Algebra

- The Newton polytope of f, $N(f)$, is the convex hull of the set of exponents of its monomials with non-zero coefficient.
- The resultant R is the polynomial in the coefficients of a system of polynomials which vanishes if there exists a common root in the torus of the given polynomials.

A_0  $$f_0(x,y) = ax + by + c$$

A_1  $$f_1(x,y) = dx + ey + f$$

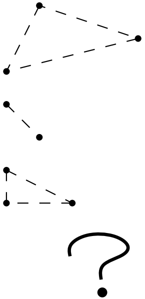
A_2  $$f_2(x,y) = gx + hy + i$$

$N(R)$  $$R(a,b,c,d,e,f,g,h,i) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

4-dimensional Birkhoff polytope

Connection with Algebra

- The Newton polytope of f, $N(f)$, is the convex hull of the set of exponents of its monomials with non-zero coefficient.
- The resultant R is the polynomial in the coefficients of a system of polynomials which vanishes if there exists a common root in the torus of the given polynomials.



A_0 $$f_0(x,y) = axy^2 + x^4y + c$$

A_1 $$f_1(x,y) = dx + ey$$

A_2 $$f_2(x,y) = gx^2 + hy + i$$

$N(R)$

NP-hard to compute the resultant in the general case

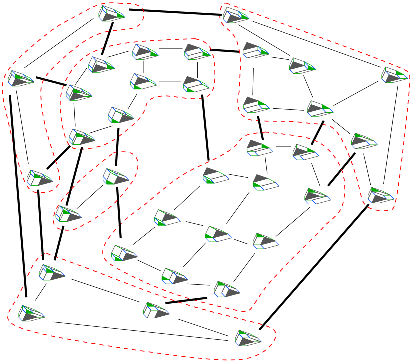
The idea of the algorithm

Input: $\mathcal{A} \in \mathbb{Z}^{2n}$ defined by $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$

Simplistic method:

- compute the secondary polytope $\Sigma(\mathcal{A})$
- many-to-one relation between vertices of $\Sigma(\mathcal{A})$ and $N(R)$ vertices

Cannot enumerate 1 representative per class by walking on secondary edges

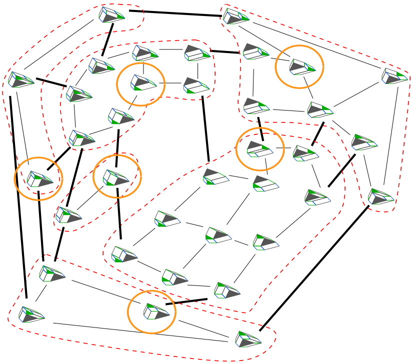


The idea of the algorithm

Input: $\mathcal{A} \in \mathbb{Z}^{2n}$ defined by $A_0, A_1, \ldots, A_n \subset \mathbb{Z}^n$

New Algorithm: [EFKP'12]

- Vertex oracle: given a direction vector compute a vertex of $N(R)$
- Output sensitive: computes only one triangulation of $\mathcal{A}$ per $N(R)$ vertex + one per $N(R)$ facet
- Computes projections of $N(R)$ or $\Sigma(\mathcal{A})$



Runtime and software

Theorem (Emiris F Konaxis Peñaranda '12)

We compute the Vertex- and Halfspace-representations of $N(R)$*, as well as a triangulation* T *of* $N(R)$*, in*

$$O^*(m^5 \, |vtx(N(R))| \cdot |T|^2),$$

where $m = \dim N(R)$*, and* $|T|$ *the number of full-dim faces of* T*.*

Respol software

- C++, CGAL (Computational Geometry Algorithms Library)
- `http://sourceforge.net/projects/respol`
- Alternative algorithm that utilizes tropical geometry (`GFan` library) [Jensen Yu '11]

How 4-d resultant polytopes look like?

(6, 15, 18, 9)
(8, 20, 21, 9)
(9, 22, 21, 8)
.
.
.
(17, 49, 47, 15)
(17, 49, 48, 16)
(17, 49, 49, 17)
(17, 50, 50, 17)
(18, 51, 48, 15)
(18, 51, 49, 16)
(18, 52, 50, 16)
(18, 52, 51, 17)
(18, 53, 51, 16)
(18, 53, 53, 18)
(18, 54, 54, 18)
(19, 54, 52, 17)
(19, 55, 51, 15)
(19, 55, 52, 16)
(19, 55, 54, 18)
(19, 56, 54, 17)
(19, 56, 56, 19)
(19, 57, 57, 19)
(20, 58, 54, 16)
(20, 59, 57, 18)
(20, 60, 60, 20)
(21, 62, 60, 19)
(21, 63, 63, 21)
(22, 66, 66, 22)

Open problem

Almost symmetric f-vector?

Outline

Polytope Representation & Oracles

Edge Skeleton Computation

Geometric Random Walks & Volume approximation

Motivation: Resultant polytopes

Experimental Results

Experiments Volume given Membership oracle

- n-cubes (table), σ=average absolute deviation, μ=average 20 experiments

n	exact vol	exact sec	# rand. points	# walk steps	vol min	vol max	vol μ	vol σ	approx sec
2	4	0.06	2218	8	3.84	4.12	3.97	0.05	0.23
4	16	0.06	2738	7	14.99	16.25	15.59	0.32	1.77
6	64	0.09	5308	38	60.85	67.17	64.31	1.12	39.66
8	256	2.62	8215	16	242.08	262.95	252.71	5.09	46.83
10	1024	388.25	11370	40	964.58	1068.22	1019.02	30.72	228.58
12	4096	–	14725	82	3820.94	4247.96	4034.39	80.08	863.72

- (the only known) implementation of [Lovász et al.'12] tested only for cubes up to $n = 8$
- no hope for exact methods in much higher than 10 dim

Experiments Volume of Minkowski sum

- Mink. sum of n-cube and n-crosspolytope, σ=average absolute deviation, μ=average over 10 experiments

n	exact vol	exact sec	# rand. points	# walk steps	vol min	vol max	vol μ	vol σ	approx sec
2	14.00	0.01	216	11	12.60	19.16	15.16	1.34	119.00
3	45.33	0.01	200	7	42.92	57.87	49.13	3.92	462.65
4	139.33	0.03	100	7	100.78	203.64	130.79	21.57	721.42
5	412.26	0.23	100	7	194.17	488.14	304.80	59.66	1707.97

- at every hit-and-run step: $\mathrm{OPT} \rightarrow \mathrm{MEM}$ (LasVegas optimization algorithm of [BertsimasVempala04])

Thank you!