Efficient volume and edge-skeleton computation for polytopes defined by oracles

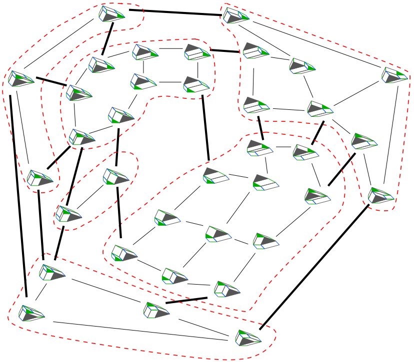
Vissarion Fisikopoulos

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

Dept. of Informatics & Telecommunications, University of Athens

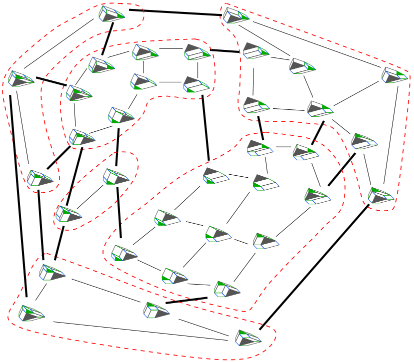
EuroCG, TU Braunschweig, 19.Mar.2013

Main motivation: resultant polytopes



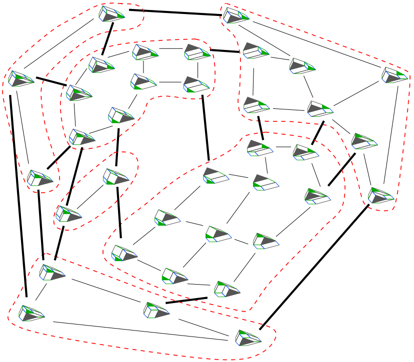
- Algorithm: [Emiris F Konaxis Peñaranda SoCG'12]
  vertex oracle + incremental construction = output-sensitive

Main motivation: resultant polytopes



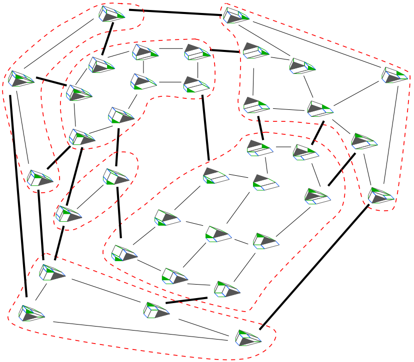
- Algorithm: [Emiris F Konaxis Peñaranda SoCG'12]
  vertex oracle + incremental construction = output-sensitive
- Software: computation in < 7 dimensions

Main motivation: resultant polytopes



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

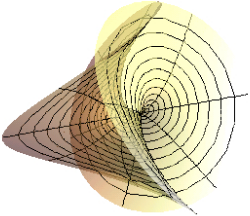
Main motivation: resultant polytopes



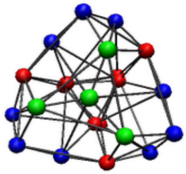
- Algorithm: [Emiris F Konaxis Peñaranda SoCG'12]
  vertex oracle + incremental construction = output-sensitive
- Software: computation in < 7 dimensions
- Q: Can we compute information when dim. > 7 ? (eg. volume)
- Hint: Can precompute *all* edge vectors, if the input is generic.

Applications

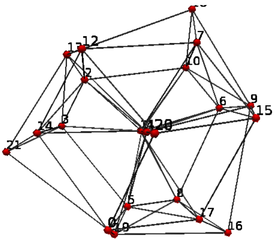
- Geometric Modeling (Implicitization) [EKKL'12]



- Combinatorics of 4-d resultant polytopes (with Emiris & Dickenstein)



(figure courtesy of M.Joswig)



Facet and vertex graph of the largest 4-dimensional resultant polytope

Outline

Polytope Representation & Oracles

Edge Skeleton Computation

Geometric Random Walks & Volume approximation

Experimental Results

Outline

Polytope representation

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

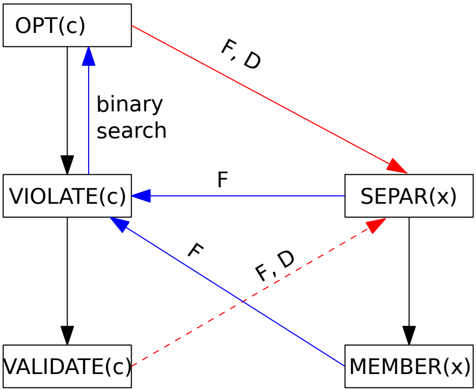
Explicit: Vertex-, Halfspace - representation (V_P, H_P),
Edge-sketelon (ES_P), Triangulation (T_P), Face lattice

Implicit: Oracles (OPT_P, MEM_P)

We study algorithms for polytopes given by OPT_P:

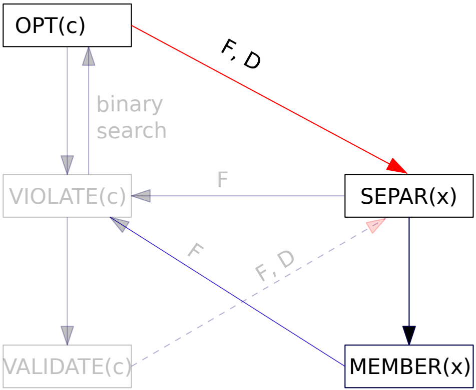
- Resultant, Discriminant, Secondary polytopes
- Minkowski sums

Oracles and duality [Grötschel et al.'93]



- $c^T, x \in \mathbb{R}^n$
- Feasibility (F)

Oracles and duality [Grötschel et al.'93]

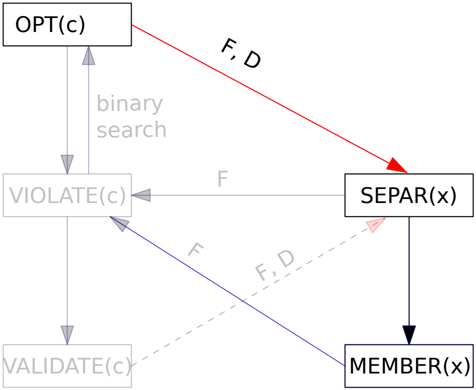


- $c^T, x \in \mathbb{R}^n$
- Feasibility (F)

(Polar) Duality (D):

$$\mathbf{0} \in \mathrm{int}(P), \quad P^* := \{c \in \mathbb{R}^n : c^T x \leq 1, \text{ for all } x \in P\} \subseteq (\mathbb{R}^n)^*$$

Oracles and duality [Grötschel et al.'93]



- $c^T, x \in \mathbb{R}^n$
- Feasibility (F)

Prop. Given OPT compute MEM in $O^*(n)$ OPT_P calls + $O^*(n^{3.38})$ arithmetic ops. utilizing algorithm of [Vaidya89]

Note: $O^*(\cdot)$ *hides log factors of* ρ/r, *where* $B(\rho) \subseteq P \subseteq B(r)$.

Outline

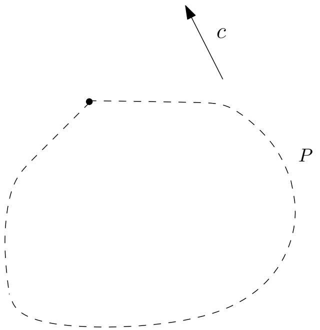
Edge skeleton computation

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 $c^T \in \mathbb{R}^n$)

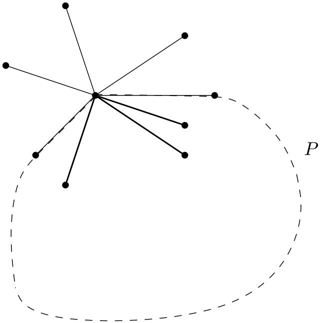
Edge skeleton computation

Input:

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

Output:

- Edge-skeleton of P



Sketch of **Algorithm**:

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

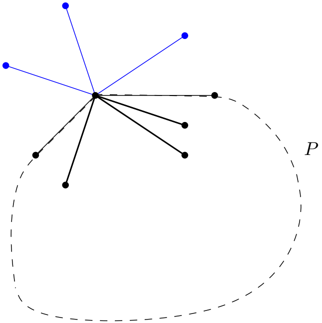
Edge skeleton computation

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. $y \notin P$ ($\mathrm{OPT}_P \rightarrow \mathrm{MEM}_P$)

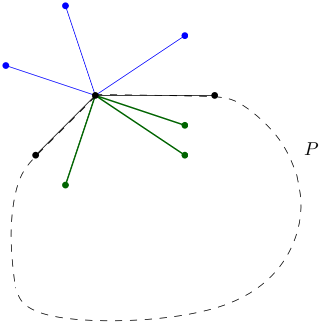
Edge skeleton computation

Input:

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

Output:

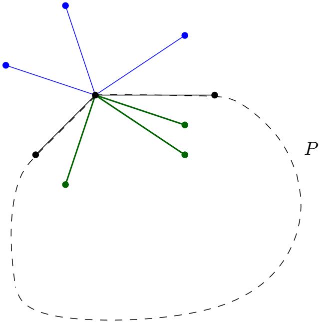
- Edge-skeleton of P



Sketch of **Algorithm**:

- Compute a vertex of P ($x = OPT_P(c)$ 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. $y \notin P$ ($OPT_P \rightarrow MEM_P$)
- Remove from S the segments that are not extreme

Edge skeleton computation



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. $y \notin P$ ($\mathrm{OPT}_P \rightarrow MEM_P$)
- Remove from S the segments that are not extreme

Open problem 1: Do not use $\mathrm{OPT}_P \rightarrow MEM_P$.

Edge skeleton computation

Proposition

[RothblumOnn07] *Let* $P \subseteq \mathbb{R}^n$ *given by* OPT_P, *and* $E \supseteq D(P)$. *All vertices of* P *can be computed in*

$$O(|E|^{n-1}) \textit{ calls to } OPT_P + O(|E|^{n-1}) \textit{ arithmetic operations.}$$

Theorem

The edge skeleton of P *can be computed in*

$O^*(m^3 n)$ *calls to* OPT_P $+O^*(m^3 n^{3.38} + m^4 n)$ *arithmetic operations*,

m*: the number of vertices of* P.

Corollary

For resultant polytopes $R \subset \mathbb{Z}^n$ *this becomes (*d *is a constant)*

$$O^*(m^3 n^{\lfloor (d/2)+1 \rfloor} + m^4 n).$$

Outline

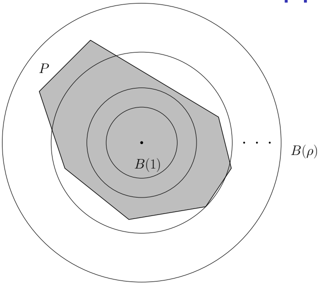
Polytope Representation & Oracles

Edge Skeleton Computation

Geometric Random Walks & Volume approximation

Experimental Results

Efficient volume approximation [Dyer et.al'91]



Volume approximation of P reduces to uniform sampling from P

Proposition (Lovász et al.'04)

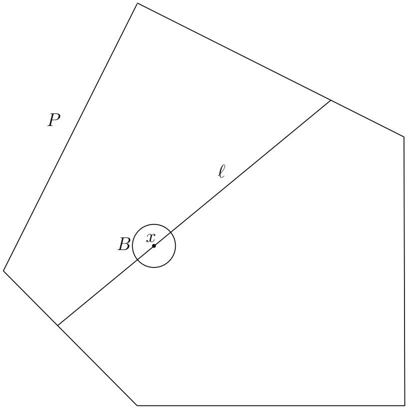
The volume of $P \subseteq \mathbb{R}^n$, *given by* MEM_P *oracle s.t.* $B(1) \subseteq P \subseteq B(\rho)$, *can be approximated with relative error* ε *and probability* $1-\delta$ *using*

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

oracle calls.

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

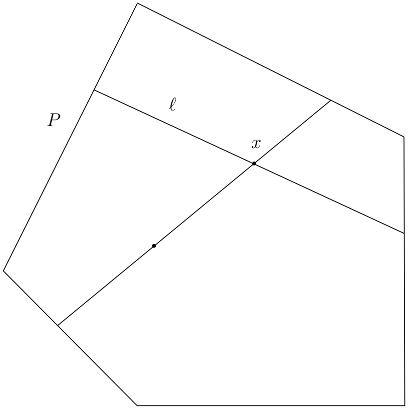
Random points in polytopes with MEM_P



Hit-and-Run walk

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

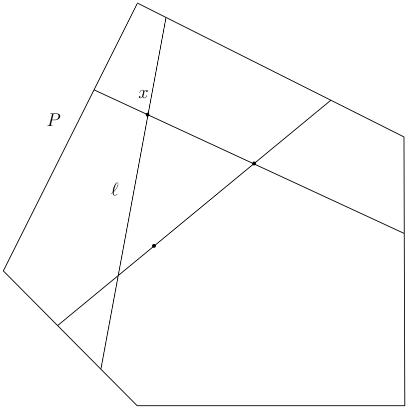
Random points in polytopes with MEM_P



Hit-and-Run walk

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

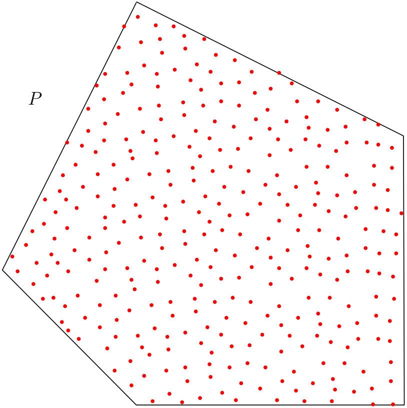
Random points in polytopes with MEM_P



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

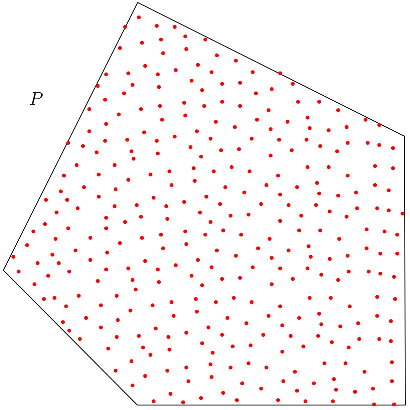


Hit-and-Run walk

- 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 $\mathrm{OPT}_{\mathrm{P}}$



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

Volume of polytopes given by OPT_P

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

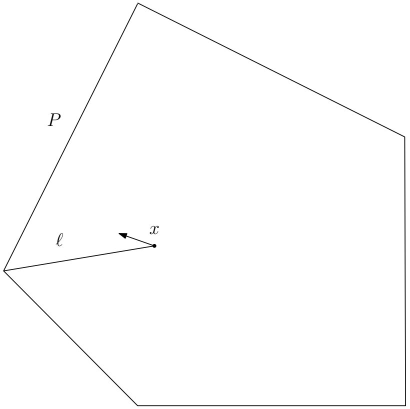
Output: ϵ-approximation $vol(P)$

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

Corollary

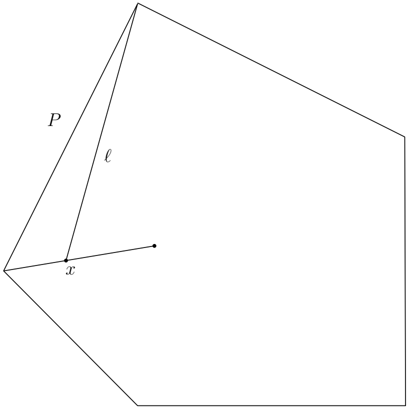
An approximation of the volume of resultant and Minkowski sum polytopes given by OPT *oracles can be computed in* $O^*(n^{\lfloor (d/2)+5 \rfloor})$ *and* $O^*(n^{7.38})$ *respectively, where* d *is a constant.*

Random points in polytopes with OPT_P REVISITED



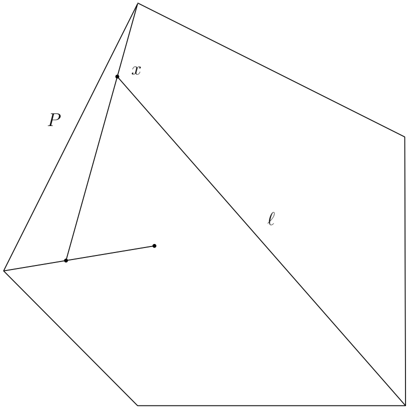
1. Hit-and-Run walk
   with $\mathrm{OPT} \to MEM$ in every step
2. Vertex walk
   - for uniform c compute $\mathrm{OPT}_\mathrm{P}(c)$
   - segment ℓ, connect x, $\mathrm{OPT}_\mathrm{P}(c)$
   - move x to a uniform disrtibuted point on ℓ

Random points in polytopes with OPT_P REVISITED



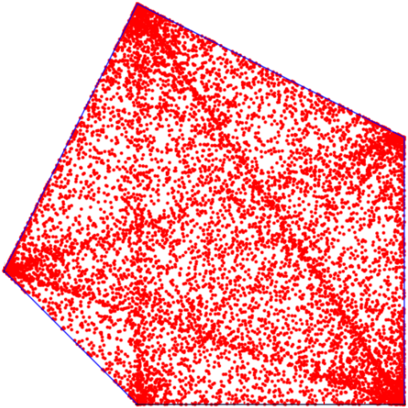
1. Hit-and-Run walk
   with $\mathrm{OPT} \to MEM$ in every step
2. Vertex walk
   - for uniform c compute $\mathrm{OPT}_\mathrm{P}(c)$
   - segment ℓ, connect x, $\mathrm{OPT}_\mathrm{P}(c)$
   - move x to a uniform disrtibuted point on ℓ

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

Random points in polytopes with 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

Open problem 2: Generate uniform points in P using OPT_P

Outline

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

Experimental evidences:

- volume up to dimension 12 within mins with $< 2\%$ error
- the minimum and maximum values bounds the exact volume

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: $\mathrm{OPT} \rightarrow \mathrm{MEM}$
- Implementation: LasVegas optimization algorithm of [BertsimasVempala04]

- slower than volume with MEM but improvements in optimization and volume implementation improve also this

Last slide !

The code

- http://sourceforge.net/projects/randgeom

Thank You !!!