Oracle-based algorithms for high-dimensional polytopes

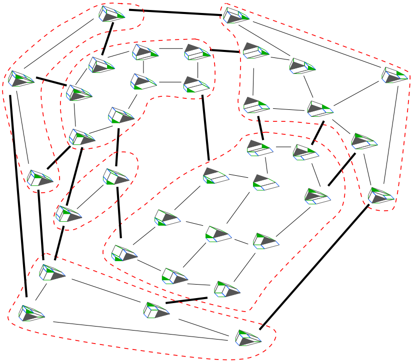
Vissarion Fisikopoulos

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

Dept. of Informatics & Telecommunications, University of Athens

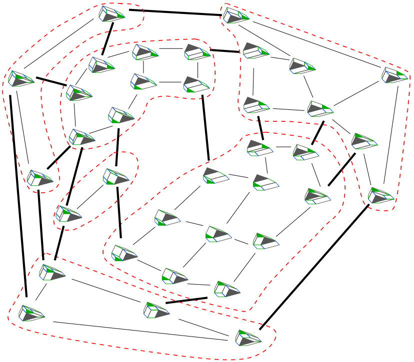
Workshop on Geometric Computing, Heraklion, Crete, 22.Jan.2013

Motivation: Secondary & Resultant polytopes



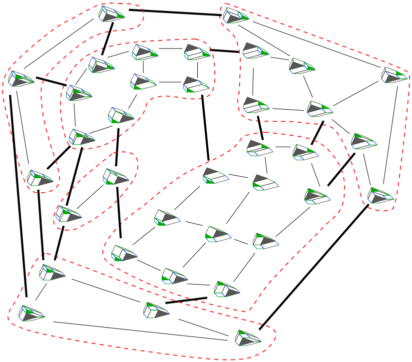
- Previous work: Oracle (optimization) & output-sensitive algorthm [EFKP SoCG'12]

Motivation: Secondary & Resultant polytopes



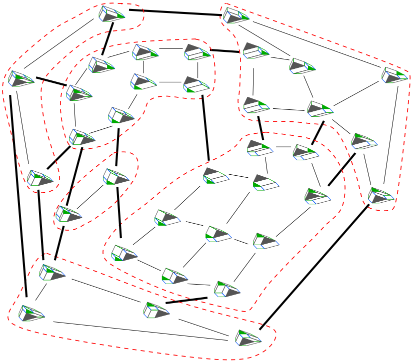
- Previous work: Oracle (optimization) & output-sensitive algorthm [EFKP SoCG'12]
- In practice: computation in < 7 dimensions

Motivation: Secondary & Resultant polytopes



- Previous work: Oracle (optimization) & output-sensitive algorthm [EFKP SoCG'12]
- In practice: computation in < 7 dimensions
- Q: Can we compute information when dim. > 7 ? eg. volume

Motivation: Secondary & Resultant polytopes



- Previous work: Oracle (optimization) & output-sensitive algorthm [EFKP SoCG'12]
- In practice: computation in < 7 dimensions
- Q: Can we compute information when dim. > 7 ? eg. volume
- Q: More polytopes given by optimization oracles ?

Outline

Outline

Polytope Representation & Oracles

Edge Skeleton Computation

Geometric Random Walks: Optimization & Volume computation

Experimental Results

Polytope representation

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

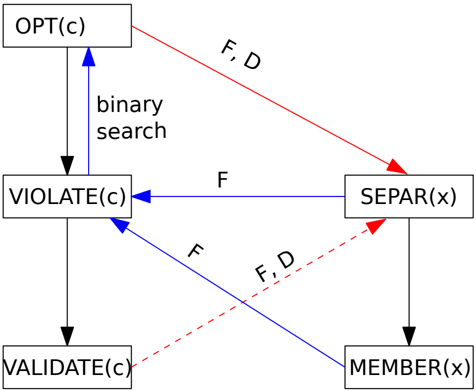
Explicit: Vertex-, Halfspace - representation (V_P, H_P),
Edge-sketelon (ES_P), Triangulation (T_P), Face lattice

Implicit: Oracles $(OPT_P$, SEP_P, $MEM_P)$

Motivation-Applications

- Resultant, Discriminant, Secondary polytopes
- (Generalized) Minkowski sums

Oracles and duality [Grötschel et al.'93]

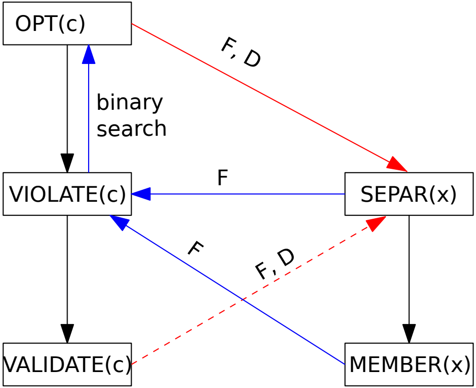


- $c^T, x \in \mathbb{R}^n$
- Feasibility (F)

(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)

(Polar) Duality (D):

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

Given OPTIMIZATION compute SEPARATION.

Polytope change of representation

Problem	Algorithm	Complexity
$V_P \to H_P$	Convex hull	EXP
$OPT_P \to T_P$	Incremental [EFKP'12]	P(in,out)
Feasibility	Ellipsoid [Kha'79], Las Vegas [BV'04]	P_{bit}, ZPP
OPT_P + {edge dir.} $\to ES_P$	Incremental [EFG'12]	P_{bit}(in,out)
$MEM_P \to$ ϵ-approx $vol(P)$	Monte-Carlo [Dyer et.al'91,LV'04]	BPP

Polytope change of representation

Problem	Algorithm	Complexity
$V_P \rightarrow H_P$	Convex hull	EXP
$OPT_P \rightarrow T_P$	Incremental [EFKP'12]	P(in,out)
Feasibility	Ellipsoid [Kha'79], Las Vegas [BV'04]	P_{bit}, ZPP
$OPT_P +$ {edge dir.} $\rightarrow ES_P$	Incremental [EFG'12]	P_{bit}(in,out)
$MEM_P \rightarrow$ ϵ-approx $vol(P)$	Monte-Carlo [Dyer et.al'91,LV'04]	BPP

Our contribution: Theory & Implementation

Outline

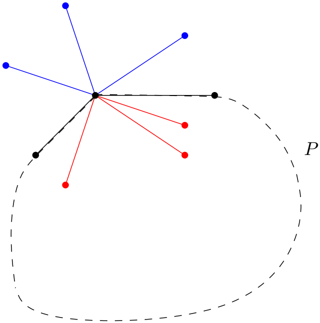
Edge skeleton computation

Input:

- OPT_P
- Edge directions of P: 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

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*

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

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

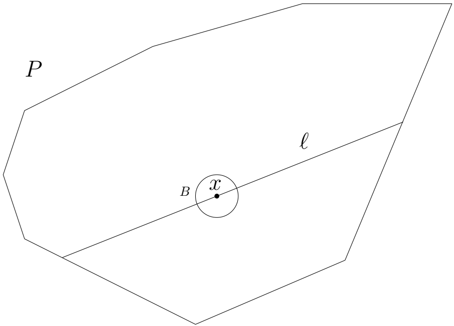
Polytope Representation & Oracles

Edge Skeleton Computation

Geometric Random Walks: Optimization & Volume computation

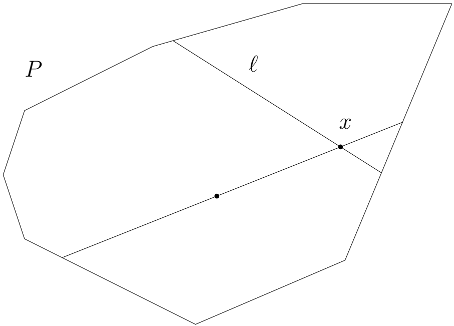
Experimental Results

Random points with Hit-and-Run



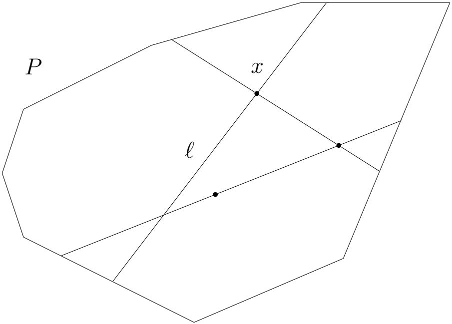
- line ℓ through x, uniform on $B_x(1)$
- move x to a uniform disrtibuted point on $P \cap \ell$

Random points with Hit-and-Run



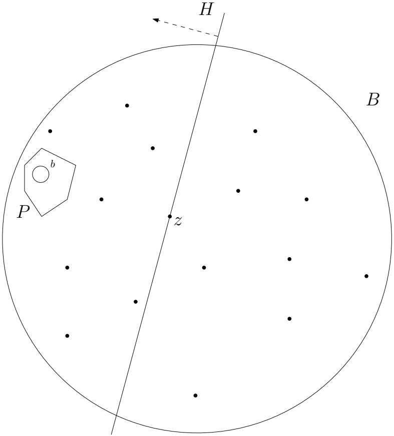
- line ℓ through x, uniform on $B_x(1)$
- move x to a uniform disrtibuted point on $P \cap \ell$

Random points with Hit-and-Run



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

Optimization using random walks [BV'04]



Optimization reduces to Feasibility:

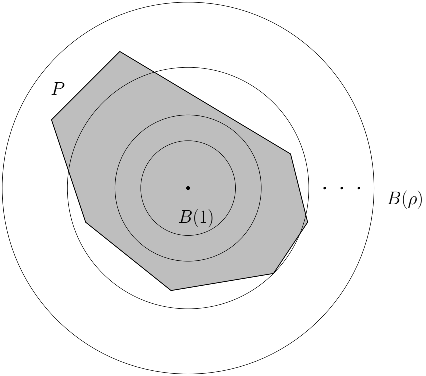
Input: $\mathcal{SEP}_P$, B, $L = \lg \frac{\text{radius}(B)}{\text{radius}(b)}$

Output: $z \in P \subseteq \mathbb{R}^n$ or P is empty

1. Compute N random points $y_1, \ldots, y_N$ uniform in B;
2. Let $z \leftarrow \frac{1}{N} \sum_{i=1}^{N} y_i$; $H \leftarrow \mathcal{SEP}_P(z)$;
3. If $z \in P$ return z, else $B \leftarrow B \cap H$;
4. Repeat steps 1-3, $2nL$ times; Report P is empty;

Complexity: $O^*(n)$ oracle calls + $O^*(n^7)$ arithm. oper.

Volume computation using random walks [Dyer et.al'91]



Input: MEM_P, ρ:
$B(1) \subseteq P \subseteq B(\rho) \subseteq \mathbb{R}^n$

Output: ϵ-approximation $vol(P)$

1. $P_i := P \cap B(2^{i/n})$, $i = 0 : \lceil n \lg \rho \rceil$;
   $P_0 = B(1)$, $P_{n \lg \rho} = P$
2. Generate rand. point in P_0
3. Generate rand. points in P_i and count how many fall in P_{i-1}

$$vol(P) = vol(P_0) \prod_{i=1}^{m} \frac{vol(P_i)}{vol(P_{i-1})}$$

Complexity [Lovász et al.'04]: $O^*(n^4)$ oracle calls

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.*

Outline

Experiments Optimization

- n-cubes (table), n-crosspolytopes, skinny crosspolytopes
- M: multipoint walk,   H: Hit-and-Run walk

n	# rand. points	# walk steps	Alg. O1 M(sec)	Alg. O1 H(sec)	Alg. O2 M(sec)	Alg. O2 H(sec)	Alg. O3 M(sec)	Alg. O3 H(sec)
2	4	0	0.02	0.05	0.01	0.01	0.01	0.006
4	38	0	0.59	1.53	0.10	0.08	0.10	0.119
6	96	1	5.54	13.23	0.47	0.84	0.99	0.727
8	172	4	61.40	73.94	4.33	5.34	9.82	4.527
10	265	10	306.20	357.88	26.64	17.22	74.86	16.44
11	316	14	559.97	853.04	54.71	36.95	112.57	55.60

- Efficient computation ($<$ 1min) up to dimension 11 using Hit-and-Run

Experiments Volume given Membership oracle

- n-cubes (table), n-crosspolytopes, σ=average absolute deviation, μ=average over 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 $n = 8$
- volume up to dimension 12 within mins with $< 2\%$ error
- no hope for exact methods in much higher than 10 dim
- 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

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

Future work - Open problems

1. describe an *efficient* random walk procedure for P given by OPT instead of MEM
2. P of special case (e.g. Minkowski sum, resultant, secondary polytope)
3. volume computation in the polar dual and *Mahler volume*
4. describe *all* edge directions of a resultant polytope

References

The code

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

Thank You !