Algorithms for high-dimensional polytopes defined by oracles

Vissarion Fisikopoulos

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

Dept. of Informatics & Telecommunications, University of Athens

Motivation: Secondary & Resultant polytopes

Algorithm: [EFKP SoCG'12]
vertex oracle + incremental construction = output-sensitive

Software: computation in $<7$ dimensions

Q: Can we compute information when dim. $>7$? eg. volume

Q: More polytopes given by optimization oracles?
Motivation: Secondary & Resultant polytopes

- Algorithm: [EFKP SoCG’12]
  vertex oracle + incremental construction = output-sensitive
Motivation: Secondary & Resultant polytopes

- **Algorithm:** [EFKP SoCG’12]
  vertex oracle + incremental construction = output-sensitive
- **Software:** computation in < 7 dimensions

Q: Can we compute information when dim. > 7? eg. volume
Q: More polytopes given by optimization oracles?
Motivation: Secondary & Resultant polytopes

- Algorithm: [EFKP SoCG'12]
  vertex oracle + incremental construction = output-sensitive
- Software: computation in $< 7$ dimensions
- Q: Can we compute information when dim. $> 7$? eg. volume
Motivation: Secondary & Resultant polytopes

- **Algorithm:** [EFKP SoCG’12]
  - vertex oracle + incremental construction = output-sensitive
- **Software:** computation in \(<7\) dimensions
- **Q:** Can we compute information when dim. \(>7\) ? eg. volume
- **Q:** More polytopes given by optimization oracles ?
Applications

▶ Geometric Modeling (Implicitization) [EKKL’12]

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

Facet and vertex graph of the largest 4-dimensional resultant polytope (figure courtesy of M.Joswig)
How 4-d resultant polytopes look like?

(6, 15, 18, 9)  
(8, 20, 21, 9)  
(9, 22, 21, 8)  
  
(17, 48, 45, 14)  
(17, 48, 46, 15)  
(17, 48, 47, 16)  
(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, 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: Optimization & Volume computation

Experimental Results
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]

(Polar) Duality (D):

\[ 0 \in \text{int}(P), \quad P^* := \{ c \in \mathbb{R}^n : c^T x \leq 1, \text{ for all } x \in P \} \subseteq (\mathbb{R}^n)^* \]
Oracles and duality [Grötschel et al.’93]

(Polar) Duality (D):

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

Given OPTIMIZATION compute SEPARATION.
## Polytope change of representation

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_P \rightarrow H_P$</td>
<td>Convex hull</td>
<td>EXP</td>
</tr>
<tr>
<td>Feasibility</td>
<td>Ellipsoid [Kha’79],</td>
<td>P&lt;sub&gt;bit&lt;/sub&gt;, ZPP</td>
</tr>
<tr>
<td></td>
<td>Las Vegas [BV’04]</td>
<td></td>
</tr>
<tr>
<td>$\text{OPT}_P + {\text{edge dir.}} \rightarrow \text{ES}_P$</td>
<td>Incremental [EFG’12]</td>
<td>P&lt;sub&gt;bit&lt;/sub&gt;(in,out)</td>
</tr>
<tr>
<td>$\text{MEM}_P \rightarrow \epsilon$-approx $\text{vol}(P)$</td>
<td>Monte-Carlo</td>
<td>BPP</td>
</tr>
<tr>
<td></td>
<td>[Dyer et.al’91,LV’04]</td>
<td></td>
</tr>
</tbody>
</table>
## Polytope change of representation

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_P \rightarrow H_P$</td>
<td>Convex hull</td>
<td>EXP</td>
</tr>
<tr>
<td>Feasibility</td>
<td>Ellipsoid [Kha’79], Las Vegas</td>
<td>$P_{bit}$, $ZPP$</td>
</tr>
<tr>
<td></td>
<td>[BV’04]</td>
<td></td>
</tr>
<tr>
<td>$OPT_P + {\text{edge dir.}} \rightarrow ES_P$</td>
<td>Incremental [EFG’12]</td>
<td>$P_{bit}(\text{in, out})$</td>
</tr>
<tr>
<td>$MEM_P \rightarrow \epsilon$-approx $\text{vol}(P)$</td>
<td>Monte-Carlo</td>
<td>BPP</td>
</tr>
<tr>
<td></td>
<td>[Dyer et.al’91, LV’04]</td>
<td></td>
</tr>
</tbody>
</table>

Our contribution: Theory & Implementation
Outline

Polytope Representation & Oracles

Edge Skeleton Computation

Geometric Random Walks: Optimization & Volume computation

Experimental Results
Edge skeleton computation

Input:
- $\text{OPT}_P$
- Edge directions of $P$: $D$

Output:
- Edge-skeleton of $P$

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

Input:
- \(\text{OPT}_P\)
- Edge directions of \(P\): \(D\)

Output:
- Edge-skeleton of \(P\)

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

Open problem: Do not use \(\text{OPT}_P \rightarrow \text{SEP}_P\).
**Edge skeleton computation**

**Proposition**

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

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

**Theorem**

*The edge skeleton of $P$ can be computed in*

\[ O^*(m^3n) \text{ calls to } \text{OPT}_P + O^*(m^3n^{3.38} + m^4n) \text{ arithmetic operations,} \]

\[ m: \text{ the number of vertices of } P. \]

**Corollary**

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

\[ O^*(m^3n^{[(d/2)+1]} + m^4n). \]
Outline

Polytope Representation & Oracles

Edge Skeleton Computation

Geometric Random Walks: Optimization & Volume computation

Experimental Results
Random points in polytopes with \( \text{SEP}_P \)

**Hit-and-Run walk**

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

\( x \) will become “random” after \( O(n^3) \) hit-and-run steps \([\text{Lovász98}]\)
Random points in polytopes with $\text{SEP}_P$

**Hit-and-Run walk**
- line $\ell$ through $x$, uniform on $B_x(1)$
- move $x$ to a uniform distributed point on $P \cap \ell$

$x$ will become “random” after $O(n^3)$ hit-and-run steps \cite{Lovász98}
Random points in polytopes with $\text{SEP}_P$

Hit-and-Run walk

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

$x$ will become “random” after $O(n^3)$ hit-and-run steps [Lovász98]
Random points in polytopes with \( \text{OPT}_P \)

1. Hit-and-Run walk
   with \( \text{OPT} \rightarrow \text{SEP} \) in every step
2. Vertex walk
   - for uniform \( c \) compute \( \text{OPT}_P (c) \)
   - segment \( \ell \), connect \( x, \text{OPT}_P (c) \)
   - move \( x \) to a uniform distributed point on \( \ell \)

Open problem: Analyse Vertex walk (or a similar walk).
Random points in polytopes with $\text{OPT}_P$

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

2. Vertex walk
   - for uniform $c$ compute $\text{OPT}_P(c)$
   - segment $\ell$, connect $x$, $\text{OPT}_P(c)$
   - move $x$ to a uniform distributed point on $\ell$

Open problem: Analyse Vertex walk (or a similar walk).
Random points in polytopes with $\text{OPT}_P$

1. Hit-and-Run walk  
   with $\text{OPT} \rightarrow \text{SEP}$ in every step
2. Vertex walk  
   - for uniform $c$ compute $\text{OPT}_P(c)$  
   - segment $\ell$, connect $x$, $\text{OPT}_P(c)$  
   - move $x$ to a uniform distributed point on $\ell$

Open problem: Analyse Vertex walk (or a similar walk).
Optimization reduces to Feasibility:

**Input:** \( \text{SEP}_P, B, L = \log \frac{\text{radius}(B)}{\text{radius}(b)} \)

**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 \( z \leftarrow \frac{1}{N} \sum_{i=1}^{N} y_i \); \( H \leftarrow \text{SEP}_P(z) \);
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: \( \text{MEM}_P, \rho: \quad B(1) \subseteq P \subseteq B(\rho) \subseteq \mathbb{R}^n \)

Output: \( \epsilon \)-approximation \( \text{vol}(P) \)

1. \( P_i := P \cap B\left(\frac{2^i}{n}\right), \quad i = 0 : \lceil n \log \rho \rceil; \quad P_0 = B(1), \quad P_{n \log \rho} = P \)

2. Generate rand. point in \( P_0 \)

3. Generate rand. points in \( P_i \) and count how many fall in \( P_{i-1} \)

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

Complexity [Lovász et al.'04]: \( O^*(n^4) \) oracle calls
Volume of polytopes given by $\text{OPT}_p$

**Input:** $\text{OPT}_p$, $\rho$: $B(1) \subseteq P \subseteq B(\rho)$

**Output:** $\epsilon$-approximation $\text{vol}(P)$

- Call volume algorithm
- Each $\text{MEM}_P$ oracle calls feasibility/optimization algorithm

**Corollary**

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

Polytope Representation & Oracles

Edge Skeleton Computation

Geometric Random Walks: Optimization & Volume computation

Experimental Results
Experiments Optimization

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

<table>
<thead>
<tr>
<th>n</th>
<th># rand. points</th>
<th># walk steps</th>
<th>Alg. O1</th>
<th>Alg. O2</th>
<th>Alg. O3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>M(sec)</td>
<td>H(sec)</td>
<td>M(sec)</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>0</td>
<td>0.02</td>
<td>0.05</td>
<td>0.01</td>
</tr>
<tr>
<td>4</td>
<td>38</td>
<td>0</td>
<td>0.59</td>
<td>1.53</td>
<td>0.10</td>
</tr>
<tr>
<td>6</td>
<td>96</td>
<td>1</td>
<td>5.54</td>
<td>13.23</td>
<td>0.47</td>
</tr>
<tr>
<td>8</td>
<td>172</td>
<td>4</td>
<td>61.40</td>
<td>73.94</td>
<td>4.33</td>
</tr>
<tr>
<td>10</td>
<td>265</td>
<td>10</td>
<td>306.20</td>
<td>357.88</td>
<td>26.64</td>
</tr>
<tr>
<td>11</td>
<td>316</td>
<td>14</td>
<td>559.97</td>
<td>853.04</td>
<td>54.71</td>
</tr>
</tbody>
</table>

- Efficient computation (< 1min) up to dimension 11 using Hit-and-Run
Experiments Volume given Membership oracle

- $n$-cubes (table), $n$-crosspolytopes, $\sigma =$ average absolute deviation, $\mu =$ average over 20 experiments

<table>
<thead>
<tr>
<th>$n$</th>
<th>exact $\text{vol}$</th>
<th>exact sec</th>
<th>$#$ rand. points</th>
<th>$#$ walk steps</th>
<th>vol min</th>
<th>vol max</th>
<th>vol $\mu$</th>
<th>vol $\sigma$</th>
<th>approx $\text{vol}$</th>
<th>approx sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>0.06</td>
<td>2218</td>
<td>8</td>
<td>3.84</td>
<td>4.12</td>
<td>3.97</td>
<td>0.05</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>0.06</td>
<td>2738</td>
<td>7</td>
<td>14.99</td>
<td>16.25</td>
<td>15.59</td>
<td>0.32</td>
<td>1.77</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>0.09</td>
<td>5308</td>
<td>38</td>
<td>60.85</td>
<td>67.17</td>
<td>64.31</td>
<td>1.12</td>
<td>39.66</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>2.62</td>
<td>8215</td>
<td>16</td>
<td>242.08</td>
<td>262.95</td>
<td>252.71</td>
<td>5.09</td>
<td>46.83</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1024</td>
<td>388.25</td>
<td>11370</td>
<td>40</td>
<td>964.58</td>
<td>1068.22</td>
<td>1019.02</td>
<td>30.72</td>
<td>228.58</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4096</td>
<td>–</td>
<td>14725</td>
<td>82</td>
<td>3820.94</td>
<td>4247.96</td>
<td>4034.39</td>
<td>80.08</td>
<td>863.72</td>
<td></td>
</tr>
</tbody>
</table>

- (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, $\sigma=$average absolute deviation, $\mu=$average over 10 experiments

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

- 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 $\text{OPT}$ instead of $\text{MEM}$

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
Last slide!

The code

- [http://sourceforge.net/projects/randgeom](http://sourceforge.net/projects/randgeom)