polymake: software for polytope constructions in linear and integer optimization

Michael Joswig

TU Berlin

CO@Work, ZIB, 29 September 2015

joint w/ polymake team
1 polymake Basics
   Solving an integer linear program

2 One Special Feature
   Highly symmetric integer programs
   The core point method
   Computational results

3 Convex Hull Experiments
   Some rules of thumb

4 Epilogue
polymake Overview
most recent version 2.14 of March 2015

• software for research in mathematics
  • geometric combinatorics: convex polytopes, matroids, . . .
  • linear/combinatorial optimization
  • toric/tropical geometry
  • combinatorial topology

• open source, GNU Public License
  • supported platforms: Linux, FreeBSD, MacOS X
  • about 150,000 uloc (C++, Perl, C, Java)
  • interfaces to many other software systems

• co-authored (since 1996) w/ Ewgenij Gawrilow
  • contributions by Benjamin Assarf, Simon Hampe, Katrin Herr, Silke Horn, Lars Kastner, Georg Loho, Benjamin Lorenz, Andreas Paffenholz, Julian Pfeifle, Thomas Rehn, Thilo Rörig, Benjamin Schröter, André Wagner and others

www.polymake.org
The Basic Definition

A \textit{(convex) polytope} is the convex hull of finitely many points (in $\mathbb{R}^d$).

- $=$ intersection of finitely many closed halfspaces (if bounded)
- $=$ set of feasible points of a linear program (if bounded for all choices of linear objective functions)
A (convex) polytope is the convex hull of finitely many points (in $\mathbb{R}^d$).

- $=$ intersection of finitely many closed halfspaces (if bounded)
- $=$ set of feasible points of a linear program (if bounded for all choices of linear objective functions)
- conversion from points to inequalities (or vice versa) conceptually simple but still has its challenges
Example: Knapsack Problem

\[
\begin{align*}
\text{max} & \quad \sum_{i=1}^{d} u_i x_i \\
\text{s.t.} & \quad \sum_{i=1}^{d} w_i x_i \leq b \\
\end{align*}
\]

- \( d = \# \) items
- \( u_i = \) utility of item \( i \)
- \( w_i = \) weight of item \( i \)
- \( b = \) total weight bound

\( x_i \in \mathbb{N} \quad \text{for all } i \in [d] \)
Algorithm Overview (Selection)

- convex polytopes, polyhedra and fans
  - convex hulls: cdd, lrs, normaliz, ppl, beneath-and-beyond
  - Voronoi diagrams, Delone decompositions
  - Hasse diagrams of face lattices
  - $\leadsto$ lattice polytopes/toric varieties

- optimization
  - Hilbert bases: normaliz, 4ti2
  - Gomory–Chvátal closures
  - enumerating integer points: LattE, bounding box/by projection

- simplicial complexes

- tropical geometry

- graphs, matroids, permutation groups, …
One Special Feature
The Setup

We consider linear programs $\text{LP}(A, b, c)$ of the form

$$\begin{align*}
\text{max} & \quad c^\top x \\
\text{s.t.} & \quad Ax \leq b, \ x \in \mathbb{R}^d
\end{align*}$$

where $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^d$.

Assumptions:

- $P(A, b) := \{ x \in \mathbb{R}^d \mid Ax \leq b \}$ not empty
- Optimal solution exists, that is, $\text{LP}(A, b, c)$ bounded
- $c \neq 0$
The Setup

We consider linear programs $\text{LP}(A, b, c)$ of the form

$$\begin{align*}
\text{max} & \quad c^\top x \\
\text{s.t.} & \quad Ax \leq b, \; x \in \mathbb{R}^d
\end{align*}$$

where $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^d$.

Assumptions:

- $P(A, b) := \{ x \in \mathbb{R}^d \mid Ax \leq b \}$ not empty
- optimal solution exists, that is, $\text{LP}(A, b, c)$ bounded
- $c \neq 0$

Notation: $\text{ILP}(A, b, c)$ if additionally $x \in \mathbb{Z}^d$ required
Symmetric Integer Linear Programs

Definition

symmetry of ILP\((A, b, c)\) = linear automorphism of LP\((A, b, c)\)

- which acts on signed standard basis \(\{\pm e_1, \pm e_2, \ldots, \pm e_d\}\) of \(\mathbb{R}^d\)
as signed permutation

Facts:

- signed permutations: \(\mathbb{O}_{\mathbb{Z}} \cong \mathbb{Z}_2 \wr \text{Sym}(d) = (\mathbb{Z}_2)^d \rtimes \text{Sym}(d)\)
Symmetric Integer Linear Programs

Definition

symmetry of \( \text{ILP}(A, b, c) = \text{linear automorphism of LP}(A, b, c) \)
- which acts on signed standard basis \( \{\pm e_1, \pm e_2, \ldots, \pm e_d\} \) of \( \mathbb{R}^d \)
as signed permutation

Facts:

- signed permutations: \( O_n \mathbb{Z} \cong \mathbb{Z}_2 \wr \text{Sym}(d) = (\mathbb{Z}_2)^d \rtimes \text{Sym}(d) \)
- group of combinatorial automorphisms of standard cube/cross polytope
Symmetric Integer Linear Programs

Definition

symmetry of ILP($A, b, c$) = linear automorphism of LP($A, b, c$)

- which acts on signed standard basis \{±$e_1, ±e_2, \ldots, ±e_d$\} of $\mathbb{R}^d$
  as signed permutation

Facts:

- signed permutations: $O_n\mathbb{Z} \cong \mathbb{Z}_2 \wr \text{Sym}(d) = (\mathbb{Z}_2)^d \rtimes \text{Sym}(d)$
- group of combinatorial automorphisms of standard cube/cross polytope

Margot 2002; Friedman 2007; Kaibel & Pfetsch 2008; Ostrowski & al. 2011; …
The Core Point Method

Consider ILP($A, b, c$) as above. Assume that the entire group Sym($d$) acts as symmetries.

Theorem (Bödi, J. & Herr 2013)

Then the ILP can be solved to optimality in $O(md^2)$ time.
The Core Point Method

Consider ILP\((A, b, c)\) as above. Assume that the entire group \(\text{Sym}(d)\) acts as symmetries.

**Theorem (Bödi, J. & Herr 2013)**

Then the ILP can be solved to optimality in \(O(md^2)\) time.

- Herr, Rehn & Schürmann 2013: extension of core point algorithm solves MIPLIB 2010 problem toll-like w/ polymake and Gurobi
## Computational Results

### Wild Input

<table>
<thead>
<tr>
<th>$d$</th>
<th>time LP (s)</th>
<th>time IP (s)</th>
<th>time LP (s)</th>
<th>time IP (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>0.06</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.17</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>6</td>
<td>0.05</td>
<td>0.74</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>7</td>
<td>0.13</td>
<td>2.71</td>
<td>0.09</td>
<td>0.13</td>
</tr>
<tr>
<td>8</td>
<td>0.62</td>
<td>10.15</td>
<td>0.24</td>
<td>0.38</td>
</tr>
<tr>
<td>9</td>
<td>2.08</td>
<td>42.06</td>
<td>0.69</td>
<td>1.03</td>
</tr>
<tr>
<td>10</td>
<td>8.02</td>
<td>135.51</td>
<td>1.86</td>
<td>2.89</td>
</tr>
</tbody>
</table>
Convex Hull Experiments
Example: Max-Cut

- combinatorial optimization problem on $\Gamma = (V, E)$ finite graph

$$\max \sum_{s \in S, t \in T, \{s, t\} \in E} w(s, t)$$

- maximum over all partitions $S \sqcup T = V$
- $w$ = weight function on $E$
- each cut $S \sqcup T$ gives rise to a subset of $E$, which can be encoded by its characteristic vector
  - $\rightsquigarrow 0/1$-polytope

Barahona & al. 1988; Avis, Imai & Ito 2008; Bonato & al. 2014;...
Example: Max-Cut

- combinatorial optimization problem on $\Gamma = (V, E)$ finite graph

$$\max \sum_{s \in S, t \in T, \{s,t\} \in E} w(s, t)$$

- maximum over all partitions $S \sqcup T = V$
- $w = \text{weight function on } E$
- each cut $S \sqcup T$ gives rise to subset of $E$, which can be encoded by its characteristic vector
  - $\rightsquigarrow 0/1$-polytope

- goal: determine facets of the cut polytopes

Barahona & al. 1988; Avis, Imai & Ito 2008; Bonato & al. 2014; . . .
DEMO
Facets of Cut Polytopes

variable dimension

\[ d = k + 6 \]
\[ n = 2^{k+5} = \# \text{ cuts} \]
\[ m = 2d + 8 = 2k + 20 \]

Barahona 1983: facets known if no \( K_5 \)-minor
Knapsack Integer Hulls
fixed dimension, variable right hand side

\[ a_1 = 2, \ a_2 = 3, \ a_i = a_{i-2} + a_{i-1} \]

\[ F_d(b) = \{ x \in \mathbb{R}^d_{\geq 0} \mid a^\top x \leq b \} \]

- \( d = 5 \)
- \( n = 1366, 3173, 6509, 12182, 21245, 35025, 55157 \)
- \( m = 12, 15, 12, 12, 8, 13, 15 \)
Voronoi Diagrams of Random Points in a Box
variable dimension, variable number of points
DEMO
Some Rules of Thumb

1. If you do not know anything about your input, try double description.
   - cdd, ppl, nmz
2. Do use double description for computing the facets of 0/1-polytopes.
   - cdd, ppl
3. On random input beneath-and-beyond often behaves very well.
   - bb
4. Use reverse search for partial information and non-degenerate input.
   - lrs
Epilogue
