Introduction to Porta and Polymake

CO@Work Berlin

Christian Raack and Axel Werner

09/22/2009

Berlin Mathematical School

DFG Research Center MATHEON
Mathematics for key technologies

Preliminaries

Goals:

- Repetition of polyhedral theory
- Usage of software
  - `Porta`
  - `Polymake` (+ `JavaView`)
- Combinatorial optimization and polyhedra

You will find all the examples + tutorial on the virtual machine

`~/COatWork-Data/0922/porta`

`~/COatWork-Data/0922/polymake`

Outline

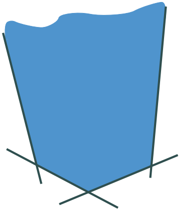
Outline

H-description of Polyhedra

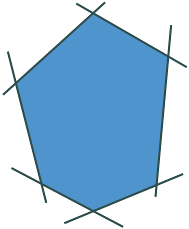
Halfspace Description

- Polyhedron: Intersection of finitely many halfspaces:

$$P(A,b) = \{x \in \mathbb{R}^d \ : \ Ax \leq b\}$$
$$A \in \mathbb{R}^{m \times d},\ b \in \mathbb{R}^m.$$



- Polytope: bounded polyhedron

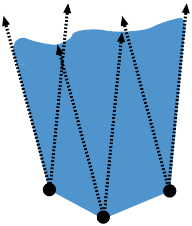


V-description of Polyhedra

Vertex description

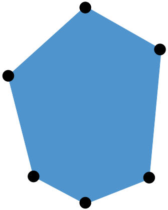
- Polyhedron: $P(A,b)$ can be written as

$$P(A,b) = \mathrm{conv}(V) + \mathrm{cone}(E)$$
$$V = \{v_1, \dots, v_k\}, E = \{e_1, \dots, e_\ell\}$$



- Polytope: $P(A,b)$ can be written as

$$P(A,b) = \mathrm{conv}(V)$$

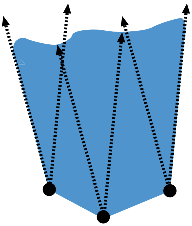


V-description of Polyhedra

Vertex description

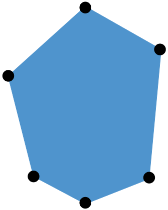
- Polyhedron: $P(A,b)$ can be written as

$$P(A,b) = \mathrm{conv}(V) + \mathrm{cone}(E)$$
$$V = \{v_1, \dots, v_k\}, E = \{e_1, \dots, e_\ell\}$$



- Polytope: $P(A,b)$ can be written as

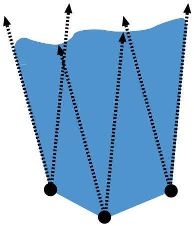
$$P(A,b) = \mathrm{conv}(V)$$



- H- and V-description are not unique !
- There exist non-redundant descriptions.

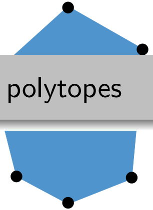
V-description of Polyhedra

Vertex description



- Polyhedron: $P(A,b)$ can be written as

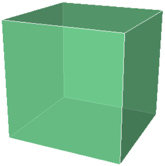
$$P(A,b) = \mathrm{conv}(V) + \mathrm{cone}(E)$$
$$V = \{v_1, \ldots, v_k\}, E = \{e_1, \ldots, e_\ell\}$$



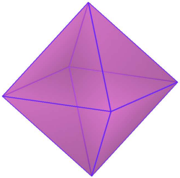
In this lecture we will mainly consider full-dimensional polytopes

$$P(A,b) = \mathrm{conv}(V)$$

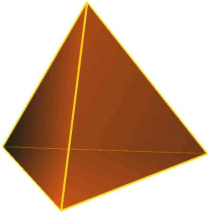
Examples: Platonic Solids



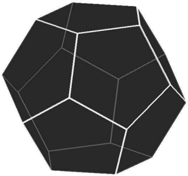
Cube



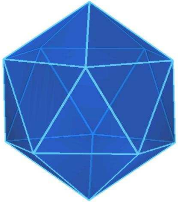
Octahedron



Tetrahedron



Dodecahedron



Icosahedron

Regular polytopes – visualization with `Polymake`/JavaView

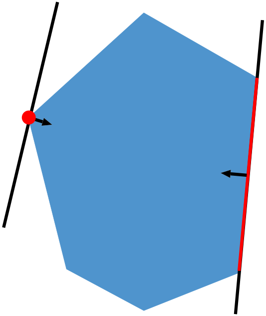
Faces

P polyhedron in $\mathbb{R}^d$.

- Face of polyhedron P:

$$F = \{x \in P \ : \ a^{\mathsf{T}}x = c\},$$

for valid inequality $a^{\mathsf{T}}x \leq c$
with $a \in \mathbb{R}^d$, $c \in \mathbb{R}$



Face F is a

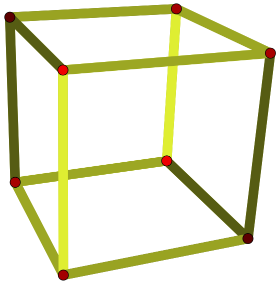
- vertex if $\dim F = 0$
- edge if $\dim F = 1$
- facet if $\dim F = \dim(P) - 1$
- Empty set and P are trivial faces

Example: d-cube

d-cube:

$$C_d := \{x \in \mathbb{R}^d \; : \; -1 \leq x_i \leq 1, \; i = 1, \ldots, d\}$$
$$= \mathrm{conv}(\{-1, 1\}^d)$$

- 2^d vertices
- $d \cdot 2^{d-1}$ edges
- $2 \cdot d$ facets

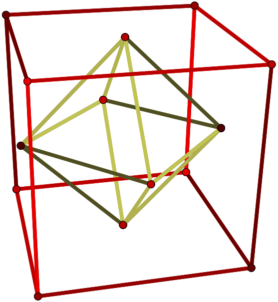


Example: d-cube

d-cube:

$$C_d := \{x \in \mathbb{R}^d \ : \ -1 \leq x_i \leq 1, \ i = 1, \ldots, d\}$$
$$= \mathrm{conv}(\{-1, 1\}^d)$$

- 2^d vertices
- $d \cdot 2^{d-1}$ edges
- $2 \cdot d$ facets

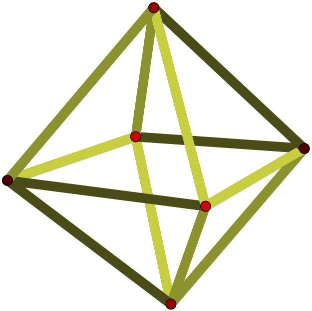


Example: d-crosspolytope

d-crosspolytope:

$$\begin{aligned} C_d^{\Delta} :=& \operatorname{conv}\{e_1, -e_1, \ldots, e_d, -e_d\} \\ =& \{x \in \mathbb{R}^d \ : \ a^{\mathsf{T}} x \leq 1, \ a \in \{-1,1\}^d\} \end{aligned}$$

- 2^d facets
- $d \cdot 2^{d-1}$ ridges
- $2 \cdot d$ vertices



Problems related to polyhedra

- Given H-description, how to derive a V-description ?
- Given V-description, how to derive an H-description ?
- What is the dimension of my polyhedron ?
- What is the dimension of the face defined by my inequality ?
- How many vertices/facets (faces of dimension n) has my polyhedron ?
- How can I visualize my polyhedron ?

... Porta and Polymake come into play

Porta and Polymake – Overview

	Porta	Polymake
Dimension (polyhedra,faces)	✓	✓
H → V	✓	✓
V → H	✓	✓
Enumeration of integral solutions	✓	(✓)
Advanced polyhedral properties (Combinatorics, Topology)	–	✓
Visualization	–	✓
Scripting, Interactive User Interface	–	✓

- So why using `Porta` ? : It is easy to use! Input/Output is "handy"
- `Polymake` is more complex, for visualization you need `JavaView`
- Transformation: `porta2poly`, `lp2poly`, `lp2porta`

Outline

Porta – Input/Output files

Porta developers:

- **Thomas Christof** (University Heidelberg)
- Andreas Löbel (ZIB)

Porta commands:

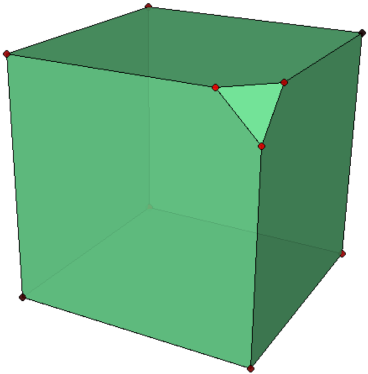
- **`traf`** – Tranforms H $\longrightarrow$ V and V $\longrightarrow$ H
- **`fmel`** – Projects your polyhedron
- **`vint`** – Enumerates all integral points in your polyhedron
- `dim`, `fctp`, `posie`, `iespo`, `iespo`, `portsort`

For help type:

```
>  man porta (man traf, man fmel, man vint)
```

All computations in rational arithmetic

Porta – Input/Output files



- 3-cube with one vertex being cut off
- 7 facets
- 10 vertices

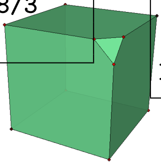
Porta – Input/Output files

- H-description: cube_cut.ieq

```
DIM=3
VALID_POINT
     0     0     0
INEQUALITIES_SECTION
(1) +x1                <=    1
(2) +x1                >=   -1
(3)      +x2           >=   -1
(4)      +x2           <=    1
(5)            +x3     <=    1
(6)            +x3     >=   -1
(7) +x1 + x2 + x3     <=  8/3
END
```

- V-description: cube_cut.poi

```
DIM = 3
CONV_SECTION
( 1)  -1    1    1
( 2)   1   -1    1
( 3)   1    1   -1
( 4)  -1   -1    1
( 5)  -1    1   -1
( 6)   1   -1   -1
( 7)  -1   -1   -1
( 8) 2/3    1    1
( 9)   1  2/3    1
(10)   1    1  2/3
END
```



Porta – **traf** – Transformation

traf is the central command of Porta

H-description $\longrightarrow$ V-description (ieq $\longrightarrow$ poi):

```
> traf cube_cut.ieq
```

Output: file cube_cut.ieq.poi

V-description $\longrightarrow$ H-description (poi $\longrightarrow$ ieq):

```
> traf cube_cut.poi
```

Output: file cube_cut.poi.ieq

Transformation yields **non-redundant** systems

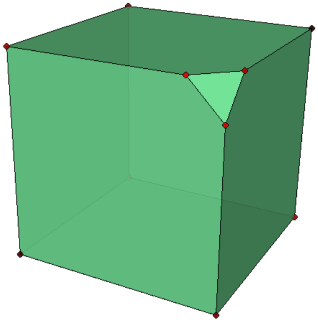
- All vertices
- All extreme rays

- All facet defining inequalities
- Description of affine hull

Porta – **traf** – Transformation

Which vertex belongs to which facet ?

```
>  traf -v cube_cut.ieq
```



```
         \  facet
          \
vertex    1234567
--------------------------
1        | ...**.* :  3
2        | *...*.* :  3
3        | *..*..* :  3
4        | .*.**.. :  3
5        | *.*.*.. :  3
6        | *..*.*. :  3
7        | .**.*.. :  3
8        | .*.*.*. :  3
9        | *.*..*. :  3
10       | .**..*. :  3
           ..............
#        | 5445543
```

Outline

Porta – **fmel** – Fourier Motzkin Elimination

How to transform a V-description to an H-description ?

Given, $v_1, \ldots, v_k \in \mathbb{R}^d$, describe $P = \mathrm{conv}\{v_1, \ldots, v_k\}$ by inequalities in a higher-dimensional space. Use definition of convex hull:

$$\begin{aligned} x &= \lambda_1 v_1 + \cdots + \lambda_k v_k \\ 1 &= \lambda_1 + \cdots + \lambda_k \\ 0 &\leq \lambda_1, \ldots, \lambda_k \end{aligned}$$

(the variables are x and $\lambda_1, \ldots, \lambda_k$).

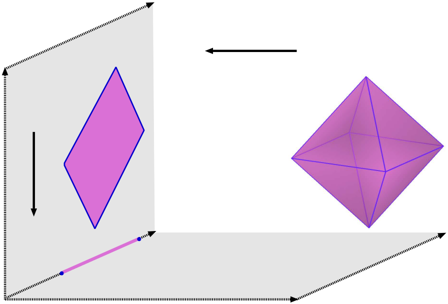
- Then project out $\lambda_1, \ldots, \lambda_k$ by Fourier-Motzkin Elimination (FMEL).
- Idea: Combine $+-$ pairs of inequalities to eliminate λ_1 $(, \lambda_2, \ldots, \lambda_k)$

Porta – **fmel** – Fourier Motzkin Elimination

Some facts:

- The projection of a polyhedron is a polyhedron
- FMEL may square the number of inequalities in every step (typically many redundant inequalities in between)
- Resulting number of facets can be exponential (compared to number of vertices in V-description)

$V \longrightarrow H$:
Consecutive orthogonal projection using FMEL



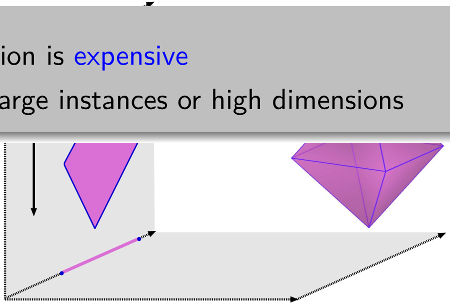
Porta – **fmel** – Fourier Motzkin Elimination

Some facts:

- The projection of a polyhedron is a polyhedron
- FMEL may square the number of inequalities in every step (typically many redundant inequalities in between)
- Resulting number of facets can be exponential (compared to number of vertices in V-description)

> - Fourier-Motzkin-Elimination is expensive
> - **traf** cannot be used for large instances or high dimensions

Consecutive orthogonal projection using FMEL



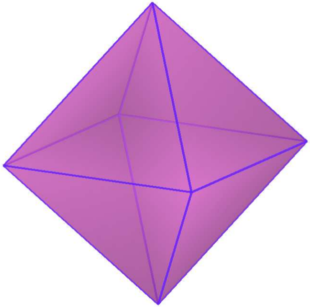
Porta – **fmel** – Fourier Motzkin Elimination

- Explicit projection with `Porta`: command **fmel**

Example: Lets project the octahedron "down" (eliminate x_1, x_2):

octahedron.ieq

```
DIM=3
ELIMINATION_ORDER
1 2 0
INEQUALITIES_SECTION
 +x1 + x2 + x3 <=   1
 -x1 + x2 + x3 <=   1
 +x1 - x2 + x3 <=   1
 +x1 + x2 - x3 <=   1
 -x1 - x2 + x3 <=   1
 +x1 - x2 - x3 <=   1
 -x1 + x2 - x3 <=   1
 -x1 - x2 - x3 <=   1
END
```



Porta – **fmel** – Fourier Motzkin Elimination

- Explicit projection with `Porta`: command **fmel**

Example: Lets project the octahedron "down" (eliminate x_1, x_2):

octahedron.ieq

```
DIM=3
ELIMINATION_ORDER
1 2 0
INEQUALITIES_SECTION
 +x1 + x2 + x3 <=   1
 -x1 + x2 + x3 <=   1
 +x1 - x2 + x3 <=   1
 +x1 + x2 - x3 <=   1
 -x1 - x2 + x3 <=   1
 +x1 - x2 - x3 <=   1
 -x1 + x2 - x3 <=   1
 -x1 - x2 - x3 <=   1
END
```

We type

```
> fmel octahedron.ieq
```

and get octahedron.ieq.ieq

```
DIM = 3
INEQUALITIES_SECTION
( 1)      <= 1
( 2) +x3 <= 1
( 3) +x3 <= 1
( 4) -x3 <= 1
( 5) -x3 <= 1
END
```

Porta – **fmel** – Fourier Motzkin Elimination

- Explicit projection with `Porta`: command **fmel**

Example: Lets project the octahedron "down" (eliminate x_1, x_2):

octahedron.ieq

```
DIM=3
ELIMINATION_ORDER
```

We type

Observe that **fmel** produces redundant inequalities.

```
 +x1 + x2 + x3 <=   1
 -x1 + x2 + x3 <=   1
 +x1 - x2 + x3 <=   1
 +x1 + x2 - x3 <=   1
 -x1 - x2 + x3 <=   1
 +x1 - x2 - x3 <=   1
 -x1 + x2 - x3 <=   1
 -x1 - x2 - x3 <=   1
END
```

```
DIM = 3
INEQUALITIES_SECTION
( 1)      <= 1
( 2) +x3 <= 1
( 3) +x3 <= 1
( 4) -x3 <= 1
( 5) -x3 <= 1
END
```

Porta – **fmel** – Fourier Motzkin Elimination

```
> fmel -c octahedron.ieq
```

switching off redundancy checks

```
DIM = 3
INEQUALITIES_SECTION
(  1)      <= 1
(  2)      <= 1
(  3)      <= 1
(  4) +x3 <= 1
(  5) +x3 <= 1
(  6) +x3 <= 1
(  7) -x3 <= 1
(  8) -x3 <= 1
(  9) -x3 <= 1
( 10) +x3 <= 2
( 11) +x3 <= 2
( 12) +x3 <= 2
( 13) +x3 <= 2
( 14) -x3 <= 2
( 15) -x3 <= 2
```

Porta – **fmel** – Fourier Motzkin Elimination

Side remark

- Elimination order influences the generation of redundant inequalities
- Changing the order my speed up `fmel` and `traf`

```
> traf cube_cut.poi
```

```
PORTA - a POlyhedron Representation Transformation Algorithm
 . . .
input file cube_cut.poi o.k.
 . . .
FOURIER - MOTZKIN - ELIMINATION:
 . . .
sum of inequalities over all iterations :     41
 . . .
number of inequalities :    7
. . .
output written to file cube_cut.poi.ieq
```

Porta – **fmel** – Fourier Motzkin Elimination

Side remark

- Elimination order influences the generation of redundant inequalities
- Changing the order my speed up `fmel` and `traf`

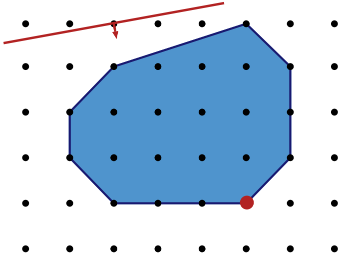
```
> traf -o cube_cut.poi  # use elimination heuristic
```

```
PORTA - a POlyhedron Representation Transformation Algorithm
 . . .
input file cube_cut.poi o.k.
 . . .
FOURIER - MOTZKIN - ELIMINATION:
 . . .
sum of inequalities over all iterations :     37
. . .
number of inequalities :    7
. . .
output written to file cube_cut.poi.ieq
```

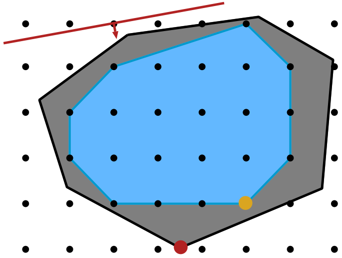
Outline

Integral Polyhedra and Relaxations

- Integer Programming =
  Optimization over integral polytopes
- Convex hull of integral solutions
- H-Description (facets) typically unknown
- V-Description (set of solutions) too large



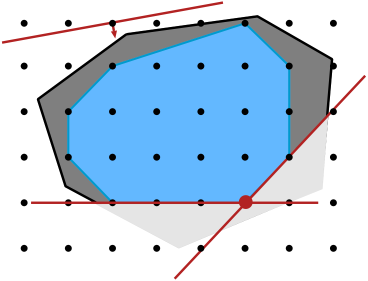
Integral Polyhedra and Relaxations



- Integer Programming = Optimization over integral polytopes
- Convex hull of integral solutions
- H-Description (facets) typically unknown
- V-Description (set of solutions) too large
- Use a Relaxation:
  - (LP) relaxation $P = P(A,b) = \{x \ : \ Ax \leq b\}$
  - Convex hull of integral solutions $P_I = \mathrm{conv}\{x \ : \ Ax \leq b, x \in \mathbb{Z}\}$

Integral Polyhedra and Relaxations

- Integer Programming = Optimization over integral polytopes
- Convex hull of integral solutions
- H-Description (facets) typically unknown
- V-Description (set of solutions) too large
- Use a Relaxation:



  - (LP) relaxation $P = P(A, b) = \{x \;:\; Ax \leq b\}$
  - Convex hull of integral solutions $P_I = \mathrm{conv}\{x \;:\; Ax \leq b, x \in \mathbb{Z}\}$

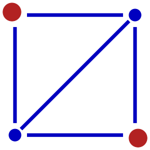
A key concept:

  - Find "some interesting" facets. Use them as cutting planes.
  - Relax and Cut

The Stable Set Polytope

- Graph $G = (V, E)$
- $S \subseteq V$ is a **stable set** if no two vertices $u, v \in S$ are connected by an edge.
- Incidence vector $\chi^S \in \{0,1\}^V$ with $\chi^S_v = 1$ iff $v \in S$
- Stable set polytope

$$STAB(G) \quad := \mathrm{conv}\{\chi^S \;:\; S \subseteq V \text{ stable}\}$$

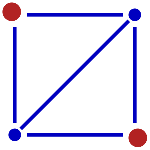


$$\begin{matrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{matrix}$$

The Stable Set Polytope

- Graph $G = (V, E)$
- $S \subseteq V$ is a **stable set** if no two vertices $u, v \in S$ are connected by an edge.
- Incidence vector $\chi^S \in \{0,1\}^V$ with $\chi^S_v = 1$ iff $v \in S$
- Stable set polytope

$$\begin{aligned} STAB(G) \quad &:= \mathrm{conv}\{\chi^S \ : \ S \subseteq V \text{ stable}\} \\ &= \mathrm{conv}\{x \in \{0,1\}^V \ : \ x_u + x_v \leq 1 \text{ for all } uv \in E\} \end{aligned}$$

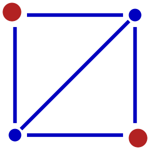


0	1	0	0	0	1
0	0	1	0	0	0
0	0	0	1	0	1
0	0	0	0	1	0

The Stable Set Polytope

- Graph $G = (V, E)$
- $S \subseteq V$ is a **stable set** if no two vertices $u, v \in S$ are connected by an edge.
- Incidence vector $\chi^S \in \{0,1\}^V$ with $\chi^S_v = 1$ iff $v \in S$
- Stable set polytope

$$\begin{aligned} STAB(G) &:= \mathrm{conv}\{\chi^S \;:\; S \subseteq V \text{ stable}\} \\ &= \mathrm{conv}\{x \in \{0,1\}^V \;:\; x_u + x_v \leq 1 \text{ for all } uv \in E\} \\ FSTAB(G) &:= \mathrm{conv}\{x \in [0,1]^V \;:\; x_u + x_v \leq 1 \text{ for all } uv \in E\} \end{aligned}$$



0	1	0	0	0	1
0	0	1	0	0	0
0	0	0	1	0	1
0	0	0	0	1	0

Porta – **vint** – Hunting for Facets

How can I use `Porta` to find facets of $STAB(G)$??

There are two options:

1. Enumerate all vertices (all solutions) (by hand)
   - Write them to stable-set.poi
   - Call

   ```
   > traf stable_set.poi
   ```
2. Use your LP relaxation
   - Write it to stable-set-relax.ieq
   - Call

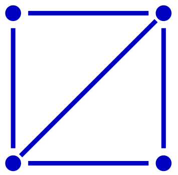
   ```
   > vint stable-set-relax.ieq     # enum. integral solutions
   > traf stable-set-relax.poi
   ```

Study the resulting H-description and generalize the facets

Porta – **vint** – Hunting for Facets

stable-set-relax.ieq

```
DIM =4
LOWER_BOUNDS
0 0 0 0
UPPER_BOUNDS
1 1 1 1
INEQUALITIES_SECTION
(1) x1          >= 0
(2)    x2       >= 0
(3)       x3    >= 0
(4)          x4 >= 0
(5) x1+x2       <= 1
(6)    x2+x3    <= 1
(7)       x3+x4 <= 1
(8) x1      +x4 <= 1
(9)    x2   +x4 <= 1
END
```



Porta – **vint** – Hunting for Facets

stable-set-relax.ieq

```
DIM =4
LOWER_BOUNDS
0 0 0 0
UPPER_BOUNDS
1 1 1 1
INEQUALITIES_SECTION
(1) x1          >= 0
(2)    x2       >= 0
(3)       x3    >= 0
(4)          x4 >= 0
(5) x1+x2       <= 1
(6)    x2+x3    <= 1
(7)       x3+x4 <= 1
(8) x1       +x4 <= 1
(9)    x2    +x4 <= 1
END
```

0. Check for Integrality

```
> traf stable-set-relax.ieq
```

stable-set-relax.ieq.poi

```
DIM = 4
CONV_SECTION
(1)   0   0   0   0
(2)   0 1/2 1/2 1/2
(3) 1/2 1/2   0 1/2
(4) 1/2 1/2 1/2 1/2
(5)   0   0   0   1
(6)   0   0   1   0
(7)   0   1   0   0
(8)   1   0   0   0
(9)   1   0   1   0
END
```

Porta – **vint** – Hunting for Facets

stable-set-relax.ieq

```
DIM =4
LOWER_BOUNDS
0 0 0 0
UPPER_BOUNDS
1 1 1 1
INEQUALITIES_SECTION
(1) x1             >= 0
(2)     x2         >= 0
(3)         x3     >= 0
(4)             x4 >= 0
(5) x1+x2          <= 1
(6)     x2+x3      <= 1
(7)         x3+x4  <= 1
(8) x1         +x4 <= 1
(9)     x2     +x4 <= 1
END
```

1. Enumeration

```
> vint stable-set-relax.ieq
```

stable-set-relax.poi

```
DIM =  4
CONV_SECTION
(1) 0 0 0 0
(2) 0 0 0 1
(3) 0 0 1 0
(4) 0 1 0 0
(5) 1 0 0 0
(6) 1 0 1 0
END
```

Porta – **vint** – Hunting for Facets

stable-set-relax.ieq

```
DIM =4
LOWER_BOUNDS
0 0 0 0
UPPER_BOUNDS
1 1 1 1
INEQUALITIES_SECTION
(1) x1             >= 0
(2)    x2          >= 0
(3)       x3       >= 0
(4)          x4    >= 0
(5) x1+x2          <= 1
(6)    x2+x3       <= 1
(7)       x3+x4    <= 1
(8) x1       +x4   <= 1
(9)    x2    +x4   <= 1
END
```

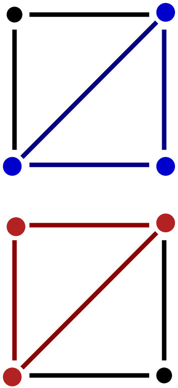
2. Transformation

```
> traf stable-set-relax.poi
```

stable-set-relax.poi.ieq

```
DIM = 4
INEQUALITIES_SECTION
(1)  x1            >= 0
(2)     x2         >= 0
(3)        x3      >= 0
(4)           x4   >= 0
(5)    +x2+x3+x4   <= 1
(6) +x1+x2    +x4  <= 1
END
```

Maximal cliques



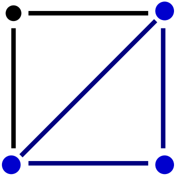
2. Transformation

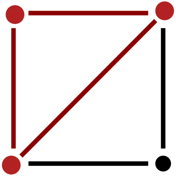
```
> traf stable-set-relax.poi
```

stable-set-relax.poi.ieq

```
DIM = 4
INEQUALITIES_SECTION
(1)  x1              >= 0
(2)     x2           >= 0
(3)        x3        >= 0
(4)           x4     >= 0
(5)    +x2+x3+x4     <= 1
(6) +x1+x2   +x4     <= 1
END
```

Maximal cliques





3. Generalize your observations

Theorem

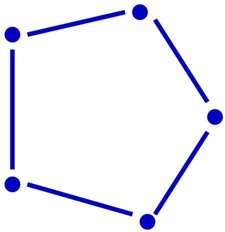
Let $G = (V, E)$ and $S \subseteq V$.
Inequality

$$\sum_{v \in S} x_v \leq 1$$

defines a facet of $STAB(G)$ if and only if S is a maximal clique.

Porta – **vint** – Hunting for Facets

2. Transformation



Odd hole inequalities

$$\sum_{v \in C} x_v \leq \frac{|C|-1}{2}$$

with C being the vertices of an odd cycle without chords are valid for $STAB(G)$.

```
> traf stable-set-relax.poi
```

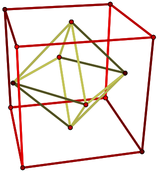
stable-set-relax.poi.ieq

```
DIM = 5
INEQUALITIES_SECTION
( 1) -x1                <= 0
( 2)     -x2            <= 0
( 3)         -x3        <= 0
( 4)             -x4    <= 0
( 5)                 -x5 <= 0
( 6) +x1+x2             <= 1
( 8)     +x2+x3         <= 1
( 7)         +x3+x4     <= 1
( 6)             +x4+x5 <= 1
( 9) +x1             +x5 <= 1
(11) +x1+x2+x3+x4+x5 <= 2
END
```

Porta – Conclusion

Porta commands:

- **traf** – Tranforms H $\longrightarrow$ V and V $\longrightarrow$ H
- **fmel** – Projects your polyhedron
- Elimination order can be crucial (-o option for **traf**)
- **vint** – Enumerates all integral points in your polyhedron
- Combine **vint** and **traf** to find facets of integral polytopes



Outline

Polymake and JavaView

Polymake

- Developed by Michael Joswig and Ewgenij Gawrilow at TU Berlin
- Interactive shell ($\geq$ v 2.95) and scripting based on perl
- Analyze polytopes, polyhedra, simplicial complexes, ...
- Convex hull computation (as porta), linear programming, visualization, more mathematics (combinatorics, geometry, topology)
- www.math.tu-berlin.de/polymake

JavaView

- Developed by Konrad Polthier, Klaus Hildebrandt, ... at TU Berlin
- 3D geometry viewer and a mathematical visualization software.
- Web-integration