Orbitwise polyhedral representation conversion

David Bremner

New Brunswick

21 August 2010
Outline

Introduction
  Polyhedra
  Representation Transformation
  Symmetries

Algorithmic Approaches
  Decomposition
  Symmetrization of standard techniques
  Fundamental Domain

Remarks
Outline

Introduction
  Polyhedra
    Representation Transformation
    Symmetries

Algorithmic Approaches
  Decomposition
  Symmetrization of standard techniques
  Fundamental Domain

Remarks
**H- and V-representations**

**H-representations**

Polyhedron \( \mathcal{H}(P) = \{ Ax \geq b \} \)

Cone \( MH(P) = \{ Ax \geq 0 \} \)

**V-representations**

Polyhedron \( \mathcal{V}(P) = \text{conv } X + \text{pos } Y \)

Cone \( \text{pos } Y \)
H- and V-representations

H-representations

Polyhedron $\mathcal{H}(P) = \{ Ax \geq b \}$
Cone $\mathcal{M}(P) = \{ Ax \geq 0 \}$

V-representations

Polyhedron $\mathcal{V}(P) = \text{conv } X + \text{pos } Y$
Cone $\text{pos } Y$
Homogenization

Vertex \( x \) \((x, 1)\)

Ray \( r \) \((r, 0)\)

Constraint \( a^T x \geq b \) \((a, -b)^T (x, x_{d+1}) \geq 0\)

Unified representation for polytopes, polyhedra

Simplified polarity
Homogenization

Vertex $x$ $(x, 1)$

Ray $r$ $(r, 0)$

Constraint $a^T x \geq b$ $ (a, -b)^T (x, x_{d+1}) \geq 0$

Unified representation for polytopes, polyhedra

Simplified polarity
Homogenization

Vertex $x$ $(x, 1)$
Ray $r$ $(r, 0)$
Constraint $a^T x \geq b$ $(a, -b)^T (x, x_{d+1}) \geq 0$

Unified representation for polytopes, polyhedra
Simplified polarity
Homogenization

Vertex \( x \) \((x, 1)\)

Ray \( r \) \((r, 0)\)

Constraint \( a^T x \geq b \) \((a, -b)^T(x, x_{d+1}) \geq 0\)

Unified representation for polytopes, polyhedra

Simplified polarity
Homogenization

Vertex $x$ $(x, 1)$

Ray $r$ $(r, 0)$

Constraint $a^T x \geq b$ $(a, -b)^T (x, x_{d+1}) \geq 0$

Unified representation for polytopes, polyhedra

Simplified polarity
Outline

Introduction
- Polyhedra
- Representation Transformation
- Symmetries

Algorithmic Approaches
- Decomposition
- Symmetrization of standard techniques
- Fundamental Domain

Remarks
The problem

Facet generation

Given $Y \in \mathbb{R}^{n \times d}$, find minimal $A$ such that $(\text{pos } Y) = \{ Ax \geq 0 \}$

Extreme ray generation

Given $A \in \mathbb{R}^{m \times d}$ find minimal $Y$ such that $(\text{pos } Y) = \{ Ax \geq 0 \}$

- Equivalent by cone polarity (i.e. commutative inner product)
The problem

Facet generation

Given $Y \in \mathbb{R}^{n \times d}$, find minimal $A$ such that $(\text{pos } Y) = \{ Ax \geq 0 \}$

Extreme ray generation

Given $A \in \mathbb{R}^{m \times d}$ find minimal $Y$ such that $(\text{pos } Y) = \{ Ax \geq 0 \}$

▶ Equivalent by cone polarity (i.e. commutative inner product)
The problem

Facet generation

Given $Y \in \mathbb{R}^{n \times d}$, find minimal $A$ such that $(\text{pos } Y) = \{ Ax \geq 0 \}$

Extreme ray generation

Given $A \in \mathbb{R}^{m \times d}$ find minimal $Y$ such that $(\text{pos } Y) = \{ Ax \geq 0 \}$

- Equivalent by cone polarity (i.e. commutative inner product)
The "standard" difficulties.

- **Degeneracy**
  Non-degenerate input, or output means solvable in polynomial time

- **Intermediate Size**
  Performance of incremental methods is often good, but sometimes very bad.

- **"Fat" lattices**
  Another kind of intermediate result blowup
The "standard" difficulties.

- **Degeneracy**
  
  Non-degenerate input, or output means solvable in polynomial time

- **Intermediate Size**
  
  Performance of incremental methods is often good, but sometimes very bad.

- **"Fat" lattices**
  
  Another kind of intermediate result blowup
The "standard" difficulties.

- **Degeneracy**
  Non-degenerate input, or output means solvable in polynomial time

- **Intermediate Size**
  Performance of incremental methods is often good, but sometimes very bad.

- **"Fat" lattices**
  Another kind of intermediate result blowup
The "standard" difficulties.

- **Degeneracy**
  Non-degenerate input, or output means solvable in polynomial time

- **Intermediate Size**
  Performance of incremental methods is often good, but sometimes very bad.

- **"Fat" lattices**
  Another kind of intermediate result blowup
Outline

Introduction
  Polyhedra
  Representation Transformation
  Symmetries

Algorithmic Approaches
  Decomposition
  Symmetrization of standard techniques
  Fundamental Domain

Remarks
### Symmetries

<table>
<thead>
<tr>
<th><strong>Type</strong></th>
<th><strong>Preserves</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Combinatorial</td>
<td>Face Lattice</td>
</tr>
<tr>
<td>Basic</td>
<td>facet defining ray sets</td>
</tr>
<tr>
<td>Projective</td>
<td>Cone</td>
</tr>
<tr>
<td>Linear</td>
<td>Cone</td>
</tr>
<tr>
<td>Restricted ($\text{Aut}(X)$)</td>
<td>Generators</td>
</tr>
</tbody>
</table>
Symmetries

Type
Combinatorial
Basic
Projective
Linear
Restricted ($\text{Aut}(X)$)

Preserves
Face Lattice
facet defining ray sets
Cone
Cone
Generators
**Definition**

For \((G, \Omega)\), \(R \subseteq \Omega\) is called a *representative set* if it contains at least one element from each \(G\)-orbit of \(\Omega\). We write \(\text{Rep}_G(\Omega)\) or just \(\text{Rep}(\Omega)\) to denote some \(G\)-representative set for \(\Omega\).

**Facet generation**

Given \(Y \in \mathbb{R}^{n \times d}\), find \(\text{Rep}(A)\) for minimal \(A\) such that \((\text{pos } Y) = \{ Ax \geq 0 \}\)

**Extreme ray generation**

Given \(A \in \mathbb{R}^{m \times d}\) find \(\text{Rep}(Y)\) for minimal \(Y\) such that \((\text{pos } Y) = \{ Ax \geq 0 \}\)
Orbitwise Representation Conversion

Definition

For \((G, \Omega)\), \(R \subseteq \Omega\) is called a representative set if it contains at least one element from each \(G\)-orbit of \(\Omega\). We write \(\text{Rep}_G(\Omega)\) or just \(\text{Rep}(\Omega)\) to denote some \(G\)-representative set for \(\Omega\).

Facet generation

Given \(Y \in \mathbb{R}^{n \times d}\), find \(\text{Rep}(A)\) for minimal \(A\) such that 
\((\text{pos } Y) = \{Ax \geq 0\}\)

Extreme ray generation

Given \(A \in \mathbb{R}^{m \times d}\) find \(\text{Rep}(Y)\) for minimal \(Y\) such that 
\((\text{pos } Y) = \{Ax \geq 0\}\)
Orbitwise Representation Conversion

Definition
For \((G, \Omega)\), \(R \subseteq \Omega\) is called a representative set if it contains at least one element from each \(G\)-orbit of \(\Omega\). We write \(\text{Rep}_G(\Omega)\) or just \(\text{Rep}(\Omega)\) to denote some \(G\)-representative set for \(\Omega\).

Facet generation
Given \(Y \in \mathbb{R}^{n \times d}\), find \(\text{Rep}(A)\) for minimal \(A\) such that \((\text{pos } Y) = \{Ax \geq 0\}\)

Extreme ray generation
Given \(A \in \mathbb{R}^{m \times d}\) find \(\text{Rep}(Y)\) for minimal \(Y\) such that \((\text{pos } Y) = \{Ax \geq 0\}\)
Outline

Introduction
  Polyhedra
  Representation Transformation
  Symmetries

Algorithmic Approaches
  Decomposition
  Symmetrization of standard techniques
  Fundamental Domain

Remarks
Adjacency Decomposition

**Input:** \( V = \mathcal{V}(\mathcal{P}) \) and group
\( G \leq \text{Aut}(V) \)

**Output:** \( \mathcal{F} = \text{Rep}(\mathcal{H}(\mathcal{P})) \)

\[ T \leftarrow \{ F \} \text{ with } F \text{ a facet of } \mathcal{P}. \]

\[ \mathcal{F} \leftarrow \emptyset. \]

**while** \( \exists F \in T \) **do**

\[ (\mathcal{F}, T) \leftarrow (\mathcal{F} \cup \{ F \}, T \setminus \{ F \}) \]

**foreach** facet \( H \) of \( F \) **do**

Let \( F' \) s.t. \( F \cap F' = H \).

**if** \( F' \not\in G(\mathcal{F} \cup T) \) **then**

\[ T \leftarrow T \cup \{ F' \}. \]

**end if**

**end for**

**end while**
Adjacency Decomposition

**Input:** $V = \mathcal{V}(\mathcal{P})$ and group $G \leq \text{Aut}(V)$

**Output:** $\mathcal{F} = \text{Rep}(\mathcal{H}(\mathcal{P}))$

1. $\mathcal{T} \leftarrow \{F\}$ with $F$ a facet of $\mathcal{P}$.
2. $\mathcal{F} \leftarrow \emptyset$.
3. **while** $\exists F \in \mathcal{T}$ **do**
   - $(\mathcal{F}, \mathcal{T}) \leftarrow (\mathcal{F} \cup \{F\}, \mathcal{T} \setminus \{F\})$
   - **foreach** facet $H$ of $F$ **do**
     - Let $F'$ s.t. $F \cap F' = H$.
     - **if** $F' \notin G(\mathcal{F} \cup \mathcal{T})$ **then**
       - $\mathcal{T} \leftarrow \mathcal{T} \cup \{F'\}$.
     - **end if**
   - **end for**
4. **end while**
Adjacency Decomposition

Input: $V = \mathcal{V}(\mathcal{P})$ and group $G \leq \text{Aut}(V)$
Output: $\mathcal{F} = \text{Rep}(\mathcal{H}(\mathcal{P}))$

$\mathcal{T} \leftarrow \{F\}$ with $F$ a facet of $\mathcal{P}$.
$\mathcal{F} \leftarrow \emptyset$.

while $\exists F \in \mathcal{T}$ do

$(\mathcal{F}, \mathcal{T}) \leftarrow (\mathcal{F} \cup \{F\}, \mathcal{T} \setminus \{F\})$

foreach facet $H$ of $F$ do

Let $F'$ s.t. $F \cap F' = H$.
if $F' \notin G(\mathcal{F} \cup \mathcal{T})$ then

$\mathcal{T} \leftarrow \mathcal{T} \cup \{F'\}$.
endif
end for
end while
Incidence Decomposition

**Input:** $V = V(\mathcal{P})$ and group $G \leq \text{Aut}(V)$

**Output:** $\mathcal{F} = \text{Rep}(\mathcal{H}(\mathcal{P}))$

\[
\mathcal{F} \leftarrow \emptyset.
\]
\[
\mathcal{R} \leftarrow \text{Rep}(V(\mathcal{P})).
\]
\[
\text{for } r \in \mathcal{R} \text{ do}
\]
\[
\text{for } f \in \mathcal{F} \text{ do}
\]
\[
\text{for } F \in \mathcal{F}_r \text{ do}
\]
\[
\text{if } F \notin G\mathcal{F} \text{ then}
\]
\[
\mathcal{F} \leftarrow \mathcal{F} \cup \{F\}.
\]
\[
\text{end if}
\]
\[
\text{end for}
\]
\[
\text{end for}
\]
\[
\text{end for}
\]
Incidence Decomposition

Input: $V = V(\mathcal{P})$ and group $G \leq \text{Aut}(V)$

Output: $\mathcal{F} = \text{Rep}(\mathcal{H}(\mathcal{P}))$

$\mathcal{F} \leftarrow \emptyset$.

$\mathcal{R} \leftarrow \text{Rep}(V(\mathcal{P}))$.

for $r \in \mathcal{R}$ do

for $F \in \mathcal{F}_r$ do

if $F \notin G\mathcal{F}$ then

$\mathcal{F} \leftarrow \mathcal{F} \cup \{F\}$.

end if

end for

end for
## Testing for $G$-equivalency

### Invariants

- **Combinatorial invariants:**
  - rank, cardinality
  - set-orbit-intersection size
  - heuristic tests on adjacency matrices.

- **Metric invariants:** the set of inner products
  
  Depends on restricted automorphism

### Tests

- Directly using backtracking (Permlib, GAP)
- Reduce to graph isomorphism (nauty, bliss)
Testing for $G$-equivalency

**Invariants**

- Combinatorial invariants;
  - rank, cardinality
  - set-orbit-intersection size
  - heuristic tests on adjacency matrices.

- Metric invariants: the set of inner products
  
  **Depends on restricted automorphism**

**Tests**

- Directly using backtracking (Permlib, GAP)
- Reduce to graph isomorphism (nauty, bliss)
Outline

Introduction
  Polyhedra
  Representation Transformation
  Symmetries

Algorithmic Approaches
  Decomposition
  Symmetrization of standard techniques
  Fundamental Domain

Remarks
Incremental Construction (Projection)

▶ Reorder so first $d$ vectors form basis

\[
V = \begin{bmatrix}
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\]

▶ Lift $V$ to a simplex in $\mathbb{R}^n$

\[
V_n = \begin{bmatrix}
V_{1\ldots d} & 0_{d\times n-d} \\
V_{d+1\ldots n} & I_{n-d} \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

▶ Find $G_n = \text{Stab} \{ d + 1 \ldots n \} = \text{Stab} \{ 4 \} = \{ () , (2,3) \}$

▶ Find $G_n$-orbits of facets and ridges of $P_n$. 4 orbits of ridges:

\[
\langle \{ 1,2 \} \rangle = \{ \{ 1,2 \}, \{ 1,3 \} \}
\]

\[
\langle \{ 2,4 \} \rangle = \{ \{ 2,4 \}, \{ 3,4 \} \}
\]

\[
\langle \{ 2,3 \} \rangle = \{ \{ 2,3 \} \}
\]

\[
\langle \{ 1,4 \} \rangle = \{ \{ 1,4 \} \}
\]
Incremental Construction (Projection)

- Reorder so first $d$ vectors form basis

$$V = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Lift $V$ to a simplex in $\mathbb{R}^n$

$$V_n = \begin{bmatrix} V_1 \ldots d & 0_{d \times n-d} \\ V_{d+1} \ldots n & I_{n-d} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

- Find $G_n = \text{Stab}\{d+1 \ldots n\} = \text{Stab}\{4\} = \{(), (2,3)\}$

- Find $G_n$-orbits of facets and ridges of $P_n$. 4 orbits of ridges:

$$\langle \{1,2\} \rangle = \{\{1,2\}, \{1,3\}\}$$

$$\langle \{2,4\} \rangle = \{\{2,4\}, \{3,4\}\}$$

$$\langle \{2,3\} \rangle = \{\{2,3\}\}$$

$$\langle \{1,4\} \rangle = \{\{1,4\}\}$$
Incremental Construction (Projection)

- Reorder so first $d$ vectors form basis

$$V = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Lift $V$ to a simplex in $\mathbb{R}^n$

$$V_n = \begin{bmatrix} V_{1\ldots d} & 0_{d \times n-d} \\ 0_{d+1\ldots n} & I_{n-d} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

- Find $G_n = \text{Stab} \{ d + 1 \ldots n \} = \text{Stab} \{ 4 \} = \{ (), (2, 3) \}$

- Find $G_n$-orbits of facets and ridges of $P_n$. 4 orbits of ridges:

  - $\langle \{ 1, 2 \} \rangle = \{ \{ 1, 2 \}, \{ 1, 3 \} \}$
  - $\langle \{ 2, 4 \} \rangle = \{ \{ 2, 4 \}, \{ 3, 4 \} \}$
  - $\langle \{ 2, 3 \} \rangle = \{ \{ 2, 3 \} \}$
  - $\langle \{ 1, 4 \} \rangle = \{ \{ 1, 4 \} \}$
Incremental Construction (Projection)

- Reorder so first $d$ vectors form basis

$$V = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Lift $V$ to a simplex in $\mathbb{R}^n$

$$V_n = \begin{bmatrix} V_{1\ldots d} & 0_{d \times n-d} \\ V_{d+1\ldots n} & I_{n-d} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

- Find $G_n = \text{Stab}\{d + 1 \ldots n\} = \text{Stab}\{4\} = \{(), (2,3)\}$

- Find $G_n$-orbits of facets and ridges of $P_n$. 4 orbits of ridges:

$$\langle \{1,2\} \rangle = \{ \{1,2\}, \{1,3\} \}$$
$$\langle \{2,4\} \rangle = \{ \{2,4\}, \{3,4\} \}$$
$$\langle \{2,3\} \rangle = \{ \{2,3\} \}$$
$$\langle \{1,4\} \rangle = \{ \{1,4\} \}$$
Incremental Construction (Projection)

\[ G_n = \{ (), (2, 3) \} \]

\[ \langle \{ 1, 2 \} \rangle = \{ \{ 1, 2 \}, \{ 1, 3 \} \} \]
\[ \langle \{ 2, 4 \} \rangle = \{ \{ 2, 4 \}, \{ 3, 4 \} \} \]
\[ \langle \{ 2, 3 \} \rangle = \{ \{ 2, 3 \} \} \]
\[ \langle \{ 1, 4 \} \rangle = \{ \{ 1, 4 \} \} \]

Project representatives from each orbit to get \( P_{n-1} \).
Here we discard the second two orbits as redundant.

Compute \( G_{n-1} = \) \[ \text{Stab}\{ d + 1 \ldots n - 1 \} = \text{Stab}\{\} \]

Transform \( G_n \) orbits into \( G_{n-1} \) orbits
Here we fuse the first two orbits.
Incremental Construction (Projection)

$G_n = \{ (), (2,3) \}$

$G_n$-orbits of ridges:

$\langle \{1,2\} \rangle = \{ \{1,2\}, \{1,3\} \}$
$\langle \{2,4\} \rangle = \{ \{2,4\}, \{3,4\} \}$
$\langle \{2,3\} \rangle = \{ \{2,3\} \}$
$\langle \{1,4\} \rangle = \{ \{1,4\} \}$

- Project representatives from each orbit to get $P_{n-1}$.
  Here we discard the second two orbits as redundant.

- Compute $G_{n-1} =$
  $\text{Stab} \{d + 1 \ldots n - 1\} = \text{Stab} \emptyset$

- Transform $G_n$ orbits into $G_{n-1}$ orbits
  Here we fuse the first two orbits.
Incremental Construction (Projection)

\[ G_n = \{ (), (2, 3) \} \]

\[ \langle \{ 1, 2 \} \rangle = \{ \{ 1, 2 \}, \{ 1, 3 \} \} \]
\[ \langle \{ 2, 4 \} \rangle = \{ \{ 2, 4 \}, \{ 3, 4 \} \} \]
\[ \langle \{ 2, 3 \} \rangle = \{ \{ 2, 3 \} \} \]
\[ \langle \{ 1, 4 \} \rangle = \{ \{ 1, 4 \} \} \]

- Project representatives from each orbit to get \( P_{n-1} \).
  Here we discard the second two orbits as redundant.

- Compute \( G_{n-1} = \text{Stab} \{ d + 1 \ldots n - 1 \} = \text{Stab} \emptyset \)

- Transform \( G_n \) orbits into \( G_{n-1} \) orbits
  Here we fuse the first two orbits.
Incremental Construction (Projection)

\[ G_n = \{ (), (2, 3) \} \]

\[ \langle \{ 1, 2 \} \rangle = \{ \{ 1, 2 \}, \{ 1, 3 \} \} \]
\[ \langle \{ 2, 4 \} \rangle = \{ \{ 2, 4 \}, \{ 3, 4 \} \} \]
\[ \langle \{ 2, 3 \} \rangle = \{ \{ 2, 3 \} \} \]
\[ \langle \{ 1, 4 \} \rangle = \{ \{ 1, 4 \} \} \]

- Project representatives from each orbit to get \( P_{n-1} \).
  Here we discard the second two orbits as redundant.
- Compute \( G_{n-1} = \text{Stab} \{ d + 1 \ldots n - 1 \} = \text{Stab} \emptyset \)
- Transform \( G_n \) orbits into \( G_{n-1} \) orbits
  Here we fuse the first two orbits.
Incremental Construction (Projection)

\[ G_n = \{ (), (2,3) \} \]

- Project representatives from each orbit to get \( P_{n-1} \).
  Here we discard the second two orbits as redundant.

- Compute \( G_{n-1} = \text{Stab}\{d+1 \ldots n-1\} = \text{Stab}\emptyset \)

- Transform \( G_n \) orbits into \( G_{n-1} \) orbits
  Here we fuse the first two orbits.
Pivoting

pivot $C' = C \setminus \{l\} \cup \{e\}$ such that $C'$ is a basis.

basis graph nodes = bases, edges = pivots
Pivoting

**pivot** \( C' = C \setminus \{l\} \cup \{e\} \)
such that \( C' \) is a basis.

**basis graph** nodes = bases, edges = pivots
Pivoting

pivot \quad C' = C \setminus \{l\} \cup \{e\}

such that $C'$ is a basis.

basis graph \quad \text{nodes} = \text{bases}, \text{edges} = \text{pivots}
Pivoting

pivot \[ C' = C \setminus \{l\} \cup \{e\} \]
such that \( C' \) is a basis.

basis graph
nodes = bases, edges = pivots
Pivoting

**pivot**  
\[ C' = C \setminus \{l\} \cup \{e\} \]  
such that \( C' \) is a basis.

**basis graph**  
nodes = bases, edges = pivots
Algorithmic Approaches Symmetrization of standard techniques

Orbitwise perturbation

Proposition

Let $V \subseteq \mathbb{R}^d$. Let $H \leq \overline{\text{Aut}}(V)$, $V_1, \ldots, V_k$ the orbits of $V$ under $H$, and $u$ be a fixed point for $H$,

There exists $1 \gg \varepsilon_1 \gg \varepsilon_2 \gg \cdots \gg \varepsilon_k > 0$ such that

$V' = \bigcup_j (V_j \pm \varepsilon_j u)$ is a valid perturbation of $V$ and $H \leq \overline{\text{Aut}}(V')$.
Algorithmic Approaches  Symmetrization of standard techniques

Orbitwise perturbation

Proposition

Let $V \subset \mathbb{R}^d$. Let $H \leq \overline{\text{Aut}}(V)$, $V_1, \ldots, V_k$ the orbits of $V$ under $H$, and $u$ be a fixed point for $H$,

There exists $1 \gg \varepsilon_1 \gg \varepsilon_2 \gg \cdots \gg \varepsilon_k > 0$ such that

$V' = \bigcup_j (V_j \pm \varepsilon_j u)$ is a valid perturbation of $V$ and $H \leq \overline{\text{Aut}}(V')$. 

David Bremner (New Brunswick)  Orbitwise representation conversion  21 August 2010  20 / 27
Orbitwise perturbation

**Proposition**

- Let $V \subset \mathbb{R}^d$. Let $H \leq \overline{\text{Aut}}(V)$, $V_1, \ldots, V_k$ the orbits of $V$ under $H$, and $u$ be a fixed point for $H$,

- There exists $1 \gg \varepsilon_1 \gg \varepsilon_2 \gg \cdots \gg \varepsilon_k > 0$ such that

- $V' = \bigcup_j (V_j \pm \varepsilon_j u)$ is a valid perturbation of $V$ and $H \leq \overline{\text{Aut}}(V')$. 
Face Lattice

Extend and Canonicalize

\[ \text{Ext}(x) \] non-isomorphic extensions of \( \hat{x} \).

Solve LP

\[ f(x) \] Optional parent function, to allow memoryless enumeration.

\[ f(\hat{y}_1) = \hat{x} \quad f(\hat{y}_2) \neq \hat{x} \quad f(\hat{y}_t) = \hat{x} \]

Ext \( \hat{x} \)
Algorithmic Approaches  
Symmetrization of standard techniques

Face Lattice

Extend and Canonicalize

$\text{Ext}(x)$ non-isomorphic extensions of $\hat{x}$.

Solve LP

$f(x)$ Optional parent function, to allow memoryless enumeration.

\[
\begin{align*}
\hat{y}_1 &\quad f(\hat{y}_1) = \hat{x} \\
\hat{y}_2 &\quad f(\hat{y}_2) \neq \hat{x} \\
\ldots & \\
\hat{y}_t &\quad f(\hat{y}_t) = \hat{x} \\
\end{align*}
\]
Outline

Introduction
- Polyhedra
- Representation Transformation
- Symmetries

Algorithmic Approaches
- Decomposition
- Symmetrization of standard techniques
- Fundamental Domain

Remarks
Definition

Let $G$ be the symmetry group of $S \subseteq \mathbb{R}^n$. $F \subseteq S$ is a fundamental domain for $S$ if

$$S = \bigcup_{g \in G} g(F)$$

and the regions $g(F)$ are interior disjoint.
Definition

Let $G$ be the symmetry group of $S \subseteq \mathbb{R}^n$. $F \subseteq S$ is a fundamental domain for $S$ if

$$S = \bigcup_{g \in G} g(F)$$

and the regions $g(F)$ are interior disjoint.
Voronoi Diagrams

\[ D(p, q) := \{ x \mid d(p, x) \leq d(q, x) \} \]

\[ VR(p, S) := \bigcap_{q \in P \setminus \{p\}} D(p, q) \]

**Theorem (Ehrlich and Im Hof 1979)**

Let \( G \) be a group acting on \( \mathbb{R}^n \). Let \( p \) be a "generic" point fixed only by the identity of \( G \). Then \( VR(p, \text{orbit}(p)) \) is a fundamental domain.
Voronoi Diagrams

\[
D(p, q) := \{ x \mid d(p, x) \leq d(q, x) \}
\]

\[
VR(p, S) := \bigcap_{q \in P \setminus \{p\}} D(p, q)
\]

**Theorem (Ehrlich and Im Hof 1979)**

Let \( G \) be a group acting on \( \mathbb{R}^n \). Let \( p \) be a "generic" point fixed only by the identity of \( G \). Then \( VR(p, \text{orbit}(p)) \) is a fundamental domain.
Voronoi Diagrams

\[ D(p, q) := \{ x \mid d(p, x) \leq d(q, x) \} \]
\[ VR(p, S) := \bigcap_{q \in P \setminus \{ p \}} D(p, q) \]

**Theorem (Ehrlich and Im Hof 1979)**

Let \( G \) be a group acting on \( \mathbb{R}^n \). Let \( p \) be a “generic” point fixed only by the identity of \( G \). Then \( VR(p, \text{orbit}(p)) \) is a fundamental domain.
Bisectors

\[ B(u, v) = \{ x \mid d(x, u) = d(x, v) \} \]

**Proposition**

Let \( u \) and \( v \) be two vectors in the same orbit. Every orbit of extreme rays has a representative on the \( u \)-side of \( B(u, v) \).
Bisectors

\[ B(u, v) = \{ x \mid d(x, u) = d(x, v) \} \]

A modified representation transformation problem

Given \( \mathcal{H} \)-representations for cone \( P \) and \( F \).
Find \( \mathcal{V}(P) \cap F \)

Proposition

Let \( u \) and \( v \) be two vectors in the same orbit. Every orbit of extreme rays has a representative on the \( u \) side of \( B(u, v) \).
Bisectors

\[ B(u, v) = \{ x \mid d(x, u) = d(x, v) \} \]

Proposition

Let \( u \) and \( v \) be two vectors in the same orbit. Every orbit of extreme rays has a representative on the \( u \) side of \( B(u, v) \).
Remarks

**Adaptive algorithms** Recursive decomposition, choice of algorithm. Misses good probing method for incremental method.

**Good Subgroups** Perturbations and semi-generic points can both be defined in terms of subgroups.

**Discrete Geometry** What polytopes admit transitive triangulations? How many isometry classes of full-dimensional simplices does the cube have?
Adaptive algorithms  Recursive decomposition, choice of algorithm. Misses good probing method for incremental method.

Good Subgroups  Perturbations and semi-generic points can both be defined in terms of subgroups.

Discrete Geometry  What polytopes admit transitive triangulations? How many isometry classes of full-dimensional simplices does the cube have?
Remarks

Adaptive algorithms Recursive decomposition, choice of algorithm. Misses good probing method for incremental method.

Good Subgroups Perturbations and semi-generic points can both be defined in terms of subgroups.

Discrete Geometry What polytopes admit transitive triangulations? How many isometry classes of full-dimensional simplices does the cube have?

Fundamental Domain Undergraduate project of Thea Gegenberg, ongoing with Achill Schürmann and Gordon Williams. C++ prototype using bliss, cddlib, gmpxx.