An Algorithm for Constraint/Generator Removal from Double Description of Polyhedra

Sergey Bastrakov [a], Nikolai Zolotykh [a, b]

[a] Lobachevsky State University of Nizhni Novgorod
[b] HSE Branch in Nizhny Novgorod

NET 2015

Contents

1. Convex Polyhedra
2. Double Description Method
3. Constraint/Generator Removal

Convex Polyhedra

Definition

Convex polyhedron in $\mathbb{R}^d$ is a set of solutions to a finite system of linear inequalities:

$$P = \{x \in \mathbb{R}^d : Ax \geq b\}$$

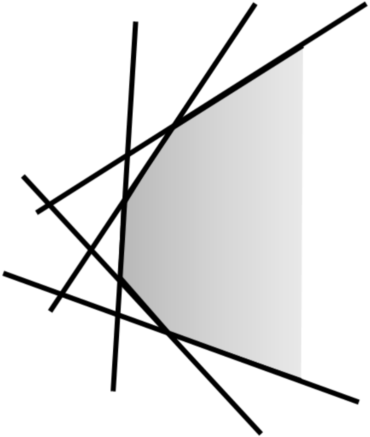
Each polyhedron can be represented in two ways:

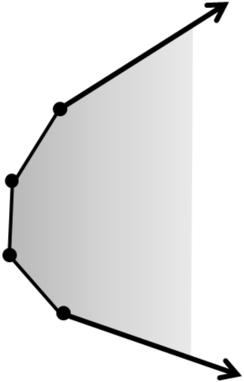
- **Facet representation** as a set of constraints: $P = \{x \in \mathbb{R}^d : Ax \geq b\}$
- **Vertex representation** as a set generators: $P = \mathrm{conv}(v_1, v_2, \ldots, v_n) + \mathrm{cone}(u_1, u_2, \ldots, u_m)$

Example: Facet and Vertex Representations

Facet representation:
$P = \{x \in \mathbb{R}^2 : Ax \geq b\}$

Vertex representation:
$P = \mathrm{conv}\,(v_1, v_2, v_3, v_4) + \mathrm{cone}\,(u_1, u_2)$





Converting Between Representations

- **Vertex enumeration problem**: given facet representation, find vertex representation:

$$Ax \geq b \longrightarrow \begin{array}{c} v_1, v_2, \ldots, v_n \\ u_1, u_2, \ldots, u_m \end{array}$$

- **Facet enumeration problem**: given vertex representation, find facet representation:

$$\begin{array}{c} v_1, v_2, \ldots, v_n \\ u_1, u_2, \ldots, u_m \end{array} \longrightarrow Ax \geq b$$

Facet/vertex enumeration problems can be reduced to one another using duality $\Rightarrow$ essentially the same **problem of computing dual representation**

Applications

Vertex and facet representations are equivalent descriptively, but not computationally

- Graphical applications in 2D and 3D
- Cutting plane algorithms, e.g. in ILP and DC programming
- Polyhedral method of loop nest optimization in compilers (e.g. Graphite for gcc, Polly for LLVM)
- Dynamic Groebner basis construction algorithms
- Many others

Double Description Method

- **DDM** [Motzkin, etc., 1953] is one of the most widely used algorithms for computing dual representation
- Operates with polyhedral cones. Homogenization:

$$P = \left\{x \in \mathbb{R}^d : Ax \geq b\right\} \longrightarrow$$
$$C = \left\{(x_0, x) \in \mathbb{R}^{d+1} : Ax - bx_0 \geq 0, x_0 \geq 0\right\}$$

- Input: $C = \left\{x \in \mathbb{R}^d : Ax \geq 0\right\}$
- Output: $\{u_1, u_2, \ldots, u_n\} : C = \mathrm{cone}\left(u_1, u_2, \ldots, u_n\right)$
- Incremental algorithm: start with a subsystem of inequalities, process one inequality per step.

Step of DDM

- Input: $C = \{x : Ax \geq 0\} = \mathrm{cone}\,(U)$
- Output: $U' : \{x \in C : ax \geq 0\} = \mathrm{cone}\,(U')$
- Classify extreme rays of C onto three groups:

$$U_+ = \{u \in U : au > 0\} - \text{feasible}$$

$$U_0 = \{u \in U : au = 0\} - \text{boundary}$$

$$U_- = \{u \in U : au < 0\} - \text{infeasible}$$

- For each edge $(u_+, u_-) \in U_+ \times U_-$ make a combination $u_\pm = (-au_-)u_+ + (au_+)u_-$
- $U' = U_+ \cup U_0 \cup U_\pm$, where $U_\pm$ is all combinations

Adjacency Tests

- The key operation is finding all edges (u_+, u_-).
- It is done using the incidence sets J_+, J_-: indices of inequalities that are satisfied as equalities
- Pairwise tests:
  - Necessary condition $|A(J_+ \cap J_-)| \geq \mathrm{rank}(A) - 2$
  - Algebraic test: $\mathrm{rank}(A(J_+ \cap J_-)) = \mathrm{rank}(A) - 2$
  - Combinatorial test:

$$\nexists w \in U : w \neq u_+, w \neq u_-, J_+ \cap J_- \subseteq J_w$$

- Using trees to avoid pairwise enumeration
  - k-d trees: Terzer & Stelling'2008
  - We use a modification of their approach

qskeleton

- Open source implementation of DDM and Fourier-Chernikov elimination algorithm
- Hybrid DDM with ideas of Quickhull [Bastrakov & Zolotykh, 2011]
- Graph modification, k-d trees for adjacency in DDM and Fourier-Chernikov elimination
- Support for parallel computing
- `https://github.com/sbastrakov/qskeleton/`

Constraint/Generator Removal Problem

- Given irreducible facet and vertex representations (double description) of a polyhedron P
- **Constraint removal problem**: find vertex representation of a polyhedron Q defined by a subset of constraints of P.
- **Generator removal problem**: find facet representation of a polyhedron Q generated by a subset of generators of P.
- The problems are dual to one another, consider generator removal problem
- Important for polyhedral loop optimization

Constraint Removal Algorithms

Naive algorithm

- Solve vertex representation problem for Q
- Ignore the given double description of P

Incremental algorithm

- Suggested by Amato, Scozzari & Zaffanella'2014
- Main idea: solve vertex representation problem for a subset of constraints of Q using DDM

We present a **new algorithm**

- Does not involve solving vertex representation problem
- Uses facet and ray adjacency information of P

A new constraint removal algorithm

- Use homogenization to transform from polyhedron to a polyhedral cone
- Remove constraints one by one:

$$\{x : Ax \geq 0, \alpha x \geq 0\} \longrightarrow \{x : Ax \geq 0\}$$

- Due to irreducibility, each inequality defines a facet of the cone

Removing a Constraint

Input: $C = \{Ax \geq 0, ax \geq 0\} = \mathrm{cone}\,(U)$
Output: $U' : C' = \{Ax \geq 0\} = \mathrm{cone}\,(U')$

- $U_+ := \{u \in U : au > 0\}\,, U_0 := \emptyset, U_- := \emptyset$
- $F_{adj} := \{$facets of C adjacent to the removed facet$\}$
- $E_{adj} := \{$edges of C that intersect the removed facet$\}$
- For each $(u, v) \in E_{adj} : au = 0, av > 0$
  - Consider a ray that is a continuation to $ax < 0$
  - If it intersects a facet from F_{adj}, add the first intersection point to U_-
  - Otherwise add u to U_0
- Return $U_+ \cup U_0 \cup U_-$

Proof of Correctness (Sketch)

- Consider inverse operation: a step of DDM from $C' = \{Ax \geq 0\}$ to $C = \{Ax \geq 0, ax \geq 0\}$
- During DDM $U' = U'_+ \cup U'_0 \cup U'_-$
- Need to show that $U_+ = U'_+$, $U_0 = U'_0$, $U_- = U'_-$
- $U_+ = U'_+$ is obvious
- Proof of $U_0 = U'_0$ and $U_- = U'_-$ relies on rules for making combinations in DDM, adjacency conditions and irreducibility of the double description of C.

Complexity

$$C = \left\{x \in \mathbb{R}^d : Ax \geq 0\right\} = \mathrm{cone}\left(u_1, u_2, \ldots, u_n\right), A \in \mathbb{R}^{m \times d}$$

Suppose we find F_{adj} and E_{adj} using pairwise Algebraic test, all arithmetic operations take $O(1)$

Theorem

The complexity of removing a constraint of C is $O(m^2 n^2)$ for every fixed d.

Theorem

The complexity of sequential removing k constraints of C is $O(k\, m^{2+2\lfloor (d-1)/2 \rfloor})$ for every fixed d.

Comparison to the Incremental Algorithm

- Complexity of the proposed algorithm:
  - $O(m^2 n^2)$ for removing a single constraint
  - $O(k\, m^{2+2\lfloor (d-1)/2 \rfloor})$ for removing k constraints
- Incremental algorithm mainly performs DDM on F_{adj}.
- For standard DDM with pairwise Algebraic adjacency test the complexity of the incremental algorithm is

$$O(|F_{adj}|^{2+2\lfloor (d-1)/2 \rfloor})$$

- If $|F_{adj}| \approx m$ the naive algorithm is probably the best

Computational Evaluation

Polyhedron	$\mathbf{n_{del}}$	$\mathbf{T_{naive}}$ [s]	$\mathbf{T_{incr.}}$ [s]	$\mathbf{T_{proposed}}$ [s]
ccc6	1	4.2	**0.01**	**0.01**
ccc6	10	3.8	2.3	**1.6**
sampleh8	1	35.6	**5.2**	**4.8**
sampleh8	10	42.9	12.5	**7.3**
trunc10	1	1.5	**0.1**	**0.1**
trunc10	10	6.1	6.2	**4.3**
C4(20) $\times$ C4(20)	1	101.2	13.6	**2.4**
C4(20) $\times$ C4(20)	10	82.7	56.2	**45.7**
sphere6	1	15.4	**0.3**	0.7
sphere6	10	14.8	**2.5**	5.1

Summary & Future Work

- A new algorithm for constraint/generator removal
- Possible enhancements:
  - Use trees to speed up adjacency tests
  - Use some accelerating data structure to find the first intersection points between continuations of edges and facets of F_{adj}
- Computational evaluation results:
  - The proposed algorithm is competitive to the incremental
  - Naive algorithm is worse, but not nearly as bad as in Amato, Scozzari & Zaffanella paper