Polyhedral Representation Conversion

John MacLaren Walsh, Ph.D.

April 27, 2014

1 Reading References


• K. Fukuda, The double description revisited.


2 Software References

Please have a look at some of the links below.

• cddlib and cddplus. Efficient implementation of the double descriptions method. Also includes other useful polyhedral computation routines.

• lrs and lrslib Efficient implementation of lexicographic reverse search. Also includes other useful polyhedral computation routines.

• QSopt Popular and useful software for linear programming.

• QSopt.ex A useful rational implementation of simplex.

• Polco Parallel java implementation of double descriptions.

• Skeleton: Implementation of Double Descriptions Method.

• Qhull. Calculate the extreme points of the convex hull of a series of points.

• CPLEX a longstanding popular commercial linear optimization solver bought by IBM in 2009.

3 Overview

We saw in the first two lectures in the course that polyhedra have two types of representations: one that is descriptive, and another that is generative. The descriptive representation, also known as the inequality representation, describes the polyhedron as the set of simultaneous solutions to a finite collection of linear inequalities

\[ \mathcal{P} = \{ x \in \mathbb{R}^N | Hx \leq h \} \] (1)

The generative representation, also known as the extremal representation, describes the polyhedron as the convex hull of a finite number of extreme points \( \mathcal{S} = \{ s_1, \ldots, s_K \} \) and extreme directions \( \mathcal{T} = \{ t_1, \ldots, t_L \} \)

\[ \mathcal{P} = \text{conv}(\mathcal{S}) + \text{conic}(\mathcal{T}) \] (2)
Now that we have seen that closed convex sets have the same descriptions, with the additional degree of flexibility that the sets of linear inequalities/extreme points/extreme directions may be infinite, we are lead to wonder how to pass between these representations.

Previously we had shown that the extreme points and extreme directions could be determined from the inequality representation. In particular, provided the matrix $H \in \mathbb{R}^{M \times N}$ appearing in the inequality description had $M \geq N$ and was full rank so that the polyhedron was line free, at an extreme point, $N$ linearly independent inequalities held with equality, and the remaining inequalities were also obeyed. Similarly, assuming again that the polyhedron is line-free, at an extreme point, $N - 1$ linearly independent inequalities were equal to zero in the recession cone $\{ x \in \mathbb{R}^N \mid Hx \leq 0 \}$, and the remaining linearly independent inequalities to these inequalities were strictly less than zero.

However, merely searching every possible subset of $N$ or $N - 1$ linearly independent inequalities and checking whether the remaining inequalities obey the required conditions does not form a practical method for representation conversion as it involves $\binom{M}{N}$ or $\binom{M}{N-1}$ tests, which can yield a prohibitively large complexity.

In this document we will discuss two methods of efficiently passing between representations of polyhedra: the double descriptions method, and lexicographic reverse search. Before introducing these methods we will introduce several other key facts about polyhedra.

These facts will show that it suffices to consider the problem of representation conversion between cones (passing between their inequality description and extreme ray description).

4 Cones are Enough: Homogenization

Let’s begin by supposing we have an inequality description of a polyhedron $\mathcal{P} = \{ x \in \mathbb{R}^N \mid Hx \leq h \}$ which is not a cone, so that $h \neq 0$. We can embed this polyhedron in a cone $\mathcal{C}$ of one higher dimension defined as

$$\mathcal{C} = \left\{ \begin{bmatrix} x_0 \\ x \end{bmatrix} \in \mathbb{R}^{N+1} \left| \begin{bmatrix} 1 \\ -h \\ H \end{bmatrix} \begin{bmatrix} x_0 \\ x \end{bmatrix} \leq 0 \right. \right\}$$

(3)

This cone is called the homogenization of $\mathcal{P}$. Observe that

$$\mathcal{C} \cap \left\{ \begin{bmatrix} x_0 \\ x \end{bmatrix} \in \mathbb{R}^{N+1} \mid x_0 = 1 \right\} = \left\{ \begin{bmatrix} 1 \\ x \end{bmatrix} \mid x \in \mathcal{P} \right\}$$

(4)

Now, suppose that $s_k$ is an extreme point of the original polyhedron $\mathcal{P}$. Observe that the vector $[1, s_k^T]^T$ will obey $N$ linearly independent inequalities in the $N + 1$ dimensional cone with equality (the same inequalities that held with equality for the point to be a an extreme point in $\mathcal{P}$). Furthermore, any other inequalities that are linearly independent of these inequalities will hold with strict inequality, so that $[1, s_k^T]^T$ is in fact also an extreme ray of $\mathcal{C}$. To see why, observe that if there was an inequality that held with equality that was linearly independent on these $N$ inequalities, then the collection of coefficients in these $N + 1$ linearly independent inequalities would form an invertible matrix. But since these matrix would be invertible, the only solution that could make all of these $N + 1$ linearly independent inequalities hold with equality (to zero!) would be $0$ the vector of all zeros. However, $[1, s_k^T]^T \neq 0$. Hence, if $s_k$ is an extreme point of $\mathcal{P}$, $[1, s_k^T]^T$ is an extreme ray of $\mathcal{C}$.

Next, suppose that $t_\ell$ is an extreme ray of the original polyhedron $\mathcal{P}$. Observe that $[0, t_\ell^T]^T$ will be an extreme ray of $\mathcal{C}!$ Indeed, the $N - 1$ linearly independent inequalities in the recession cone $Hx \leq 0$ which held with equality also hold with equality in

$$\begin{bmatrix} -1 \\ -h \\ H \end{bmatrix} \begin{bmatrix} 0 \\ t_\ell \end{bmatrix} \leq 0$$

(5)

and additionally the first inequality in this matrix also holds with equality, so there are $N$ linearly independent inequalities that hold with equality and any other inequalities linearly independent from these inequalities also hold with strict inequality (since $t_\ell$ was an extreme ray of $\mathcal{P}$).

Observe also that these are the only extreme rays of the cone $\mathcal{C}$. Hence, finding the extreme points and extreme rays of $\mathcal{P}$ is equivalent to finding the extreme rays of the homogenization cone $\mathcal{C}$.

5 One Direction is Enough: Polars

Next, we show that we only need to consider the problem of converting the inequality description of a cone to its extreme ray representation (and not vice versa).
The polar $H^\circ$ of a convex cone $H$ is the coefficients of all linear inequalities that it obeys

$$H^\circ = \{ y \in \mathbb{R}^{N+1} | y^T x \leq 0, \forall x \in H \}$$

(6)

The polar of a polyhedral cone is also a polyhedral cone has an inequality description whose coefficients are the extreme rays of the original polyhedral code, and an extreme ray representation which is the coefficients of the inequalities of the original cone.

Indeed, suppose the extreme rays of $H$ are $\{d_1, \ldots, d_P\}$ so that

$$\left\{ \sum_{p=1}^P \gamma_p d_p \bigg| \gamma_p \geq 0 \forall p \in \{1, \ldots, P\} \right\}$$

Then the polar of $H$ is

$$H^\circ = \left\{ y \in \mathbb{R}^{N+1} \bigg| \sum_{p=1}^P \gamma_p y^T d_p \leq 0, \forall \gamma_p \geq 0 \forall \{1, \ldots, P\} \right\}$$

(8)

$$= \left\{ y \in \mathbb{R}^{N+1} \bigg| y^T \left( \sum_{p=1}^P \gamma_p d_p \right) \leq 0, \forall \gamma_p \geq 0 \forall \{1, \ldots, P\} \right\}$$

(9)

$$= \left\{ y \in \mathbb{R}^{N+1} \bigg| y^T \left( \sum_{p=1}^P \gamma_p d_p \right) \leq 0, \forall \gamma_p \geq 0 \forall \{1, \ldots, P\} \right\}$$

(10)

which shows that the inequalities of the polar cone are the extreme rays of the original cone. Additionally, since $Hx \leq 0$ implies $y^T x \leq 0$ if and only if $\gamma^T H = y^T$ for some $\gamma \geq 0$. This shows that the polar cone

$$H^\circ = \left\{ \sum_{m=1}^M \gamma_m h_m \bigg| \gamma_m \geq 0 \forall m \in \{1, \ldots, M\} \right\}$$

(11)

has extreme rays that are the inequalities of the original cone.

This shows that the polar of the polar of a polyhedral cone is the original polyhedral cone. For a convex cone, the polar of the polar is the closure of the original convex cone.

Because polars swap roles between rays and inequalities, we observe that it suffices to consider representation conversion between inequalities and extreme rays. This is because if we want to convert between the extreme ray representation and an inequality representation, we can simply interpret the extreme rays as inequalities (the polar) with their coefficients, do representation conversion on the polar to get its extreme rays, then re-interpret the extreme rays as inequalities (taking the polar again) with the same coefficients.

6 Methods for Representation Conversion

We will present two methods for representation conversion, the double descriptions method and the lexicographic reverse search method.

6.1 Double Descriptions Method

Double descriptions works by considering cones associated with larger and larger subsets of the inequalities of $H$, and keeping track of the way their extreme rays evolve as new inequalities from $H$ are added. In this manner, at each iteration, double descriptions calculates the extreme ray representation of a potentially tighter (certainly no looser) outer bound to $H$.

To begin, find a $N \times N$ sub matrix $H_A$ of $N$ linearly independent rows of $H$ from the cone $H = \{ x \in \mathbb{R}^N | Hx \leq 0 \}$.

The first cone we consider is $H_1 = \{ x \in \mathbb{R}^N | H_A x \leq 0 \}$ (obviously $H \subseteq H_1$ since it involves all of the inequalities in $H_1$ and then some more). Recalling the definition of an extreme ray, we observe that the directions of extreme rays $r_1$ of $H_1$ can be scaled to obey $H_A r_1 = -e_1$, where $e_1$ is a column of the $N \times N$ identity matrix, since a ray must obey $N - 1$ of the rows with equality and the remaining row with strict inequality, and the strict inequality can be scaled by scaling the ray. Hence, the extreme rays $R_1$ of $H_1$ are the columns of $-H_A^{-1}$.

Next, we consider how to add a new inequality to a double descriptions pair. Suppose $R_k = \{ r^{(k)}_\ell | \ell \in \{1, \ldots, L_k\} \}$ are the extreme rays of $H_k = \{ x \in \mathbb{R}^N | H_A x \leq 0 \}$ where $H_B$ is assumed full rank (by starting with $H_1$ we ensured that it is). We would like to determine how to find the extreme rays $R_{k+1}$ of the cone $H_{k+1} = \{ x \in \mathbb{R}^N | H_{A \cup \{n\}} x \leq 0 \}$ from the rays $R_k$ of $H_k$ and the new inequality $h_n = H_{(n)}$.  

3
To do this, partition the rays \( R_k \) up into three parts \( R_k = R_k^+ \cup R_k^0 \cup R_k^- \) where

\[
R_k^+ = \left\{ r^{(k)}_\ell \mid h_n r^{(k)}_\ell > 0 \right\}, \quad R_k^0 = \left\{ r^{(k)}_\ell \mid h_n r^{(k)}_\ell = 0 \right\}, \quad R_k^- = \left\{ r^{(k)}_\ell \mid h_n r^{(k)}_\ell < 0 \right\}.
\] (12)

The rays in \( R_k^- \) and \( R_k^0 \) are still rays in \( R_{k+1} \), since they still obey the conditions for a ray in the new cone. Some new rays may be added, which are made rays by virtue of having \( h_n r = 0 \), if both \( R_k^+ \) and \( R_k^- \) are non-empty. Because it must have \( h_n r = 0 \) to be a new ray, and it must be expressed as a conic combination of two rays (and no more) from \( R_k^+ \) to have a chance to be extremal in \( R_{k+1} \), any new ray in \( R_{k+1} \) that was not in \( R_k \) must be capable of being written as being proportional to

\[
r^{(k+1)}_\ell = r^{(k)}_a + \gamma r^{(k)}_b
\]

with \( r^{(k)}_a \in R_k^- \) and \( r^{(k)}_b \in R_k^+ \) and \( \gamma = -\frac{h_n r^{(k)}_a}{h_n r^{(k)}_b} \). In fact, for the new ray to be extremal in the new cone \( H_{k+1} \) it is necessary for \( r^{(k)}_a \) and \( r^{(k)}_b \) to have been adjacent (part of a 2 dimensional face) of the previous cone \( H_k \).

This set of new rays of the form (13) is written as \( R^+_{k+1} \), and so extreme rays of \( H_{k+1} \) are (the extremal subset \( R_{k+1} \)) of

\[
R_k^- \cup R_k^0 \cup R_{k+1}^+
\]

(14)

We will discuss how to find the extremal subset of rays from a list of candidate rays in §7.

### 6.2 Lexicographic Reverse Search

The second method of representation conversion we discussed is lexicographic reverse search.

This method was motivated by observing that, when initialized at any feasible basis associated with any extreme point of the constraint set, the simplex algorithm moves between a sequence of feasible bases, each associated with an extreme point, of monotone non-increasing cost. With a pivot rule that avoids cycling, then, the path of simplex through these extreme points forms a directed acyclic graph. The idea of the reverse search method is to, given an inequality description of a polyhedron, construct a “cost” for a linear program over it such that there are a unique extreme point attaining the optimum. Then, starting from this optimum cost, we simply run the simplex algorithm in reverse in a depth-first manner, reverse-pivoting along a branch until there are no longer any other bases which would pivot to the current base (a valid reverse pivot). When we can no longer reverse pivot, we forward pivot until there is a new reverse pivot that we have not yet selected, then select that next available reverse pivot we have not yet selected. The beauty of the depth-first nature of the reverse search method is that we do need to keep track of the bases we have already visited to search the entire graph, since all the memory necessary can be stored simply by considering where we forward pivoted from in the most recent step and incrementing.

Lexicographic reverse search implements several improvements over reverse search. In particular, by using a lexicographic pivot rule instead of the least subscript rule, we can ensure that the algorithm only visits the subset of lex-positive bases. Since every extreme point or extreme ray is involved in at least one lex-positive basis, visiting this subset of bases can lead to an overall reduction of the number of pivots, which is helpful in severely degenerate polyhedra for which there are a large number of bases associated with the same extreme point/ray. The second improvement that lexicographic reverse search implements is based on the fact that each extreme point/ray is associated with a unique lex-min basis. As such, when we arrive at a basis, we will not have to check whether or not we have already listed the extreme point associated with the current basis, instead we can simply check whether or not the current basis is lex-min. If it is, then we output the extreme point, if not, we do not.

This was a non-technical introduction to the algorithm, which is all we had time for during the 1/4 of a lecture we had to cover it. Please read the related papers by D. Avis for more details.

### 6.3 Numerical Precision

Since the simplex algorithm travels between extreme points which each are solutions to linear systems involving a full rank subset of the inequalities, if the inequalities all involve rational coefficients, then the extreme points all have rational coefficients. This means that, if the coefficients of the polyhedron and the cost are read in as rational numbers (rather than floating point numbers), we can exclusively work with rational numbers and avoid any loss in numerical precision. The implementation of lexicographic reverse search utilizes this capability, as does the QSopt-exact variant of the popular QSopt simplex linear programming solver.

Additionally, since double descriptions involves calculations which involve solving linear systems and linear equations involving the coefficients of the inequality description, it too can be implemented with a representation of the coefficients
and the extreme points/ extreme rays as rational numbers. The implementation CDD involves the capability to use both rational and real representations.

### 6.4 Double Descriptions vs Lexicographic Reverse Search

Since we have two methods of solving one problem, one is left to wonder which one is better than the other. This is a complicated question, as really neither method is superior to the other, rather they each of classes of polyhedra for which they outperform the other on. Because LRS is simplex based, it tends to perform poorly on severely degenerate polyhedra. CDD, on the other hand, suffers from the fact that the intermediate polyhedra can have substantially more extreme rays than the ultimate polyhedron.

One of the significant merits of lexicographic reverse search is that it easily lends itself to parallelism, since different processes/processors can be given different branches of the reverse search tree to work on. Double descriptions can be parallelized as well, but the computations are less amenable to massive parallelism.

Another merit of lexicographic reverse search over double descriptions is that its output comes incrementally (i.e. as it proceeds it creates extremal representations of tighter and tighter inner bounds to the polyhedron), while the intermediate extreme rays in the double descriptions method need not be extreme rays of the ultimate polyhedron, and hence these rays are not capable of being listed until the end.

### 7 Other Important Problems in Polyhedral Computation

Finally, we mentioned two other seminal problems in polyhedral computation in addition to linear programming and representation conversion. In particular, redundancy removal, in which unnecessary/redundant inequalities or rays are removed, and projection, in which some of the variables are removed from consideration, were mentioned.

We briefly discussed that a naive method of determining whether or not the inequality $h_0x \leq a_0$ is redundant in the polyhedron \( \begin{bmatrix} h_0 \\ H \end{bmatrix} x \leq \begin{bmatrix} a_0 \\ a \end{bmatrix} \) is redundant, is to solve the following linear program

\[
\max_{x: Hx \leq a} h_0x
\]

If the answer is $\leq a_0$, then the inequality is redundant, if the answer is $> a_0$, the inequality is not. We mentioned that another more intelligent method of redundancy removal is Clarkson’s method, which can be read about in Fukuda’s polyhedral computation notes. We noted that through polarity and homogenization (i.e. by putting a 1 in front of every point and 0 in front of every ray), we could do redundancy removal for the convex hull of a series of points plus the conic hull of a series of rays, leaving only the extreme points and extreme rays.