Polyhedral Computation

Jayant Apte
ASPITRG

Part I

The Three Problems of Polyhedral Computation

- Representation Conversion
  - H-representation of polyhedron

    $\Updownarrow$

  - V-Representation of polyhedron
- Projection
  - $\{P|P \in R^d\} \rightarrow \{P'|P' \in R^{(d-n)}\}$
- Redundancy Removal
  - Compute the minimal representation of a polyhedron

Outline

- **Preliminaries**
  - Representations of convex polyhedra, polyhedral cones
  - Homogenization - Converting a general polyhedron in $\mathbb{R}^d$ to a cone in $\mathbb{R}^{d+1}$
  - Polar of a convex cone
- **The three problems**
  - Representation Conversion
    - Review of LRS
    - Double Description Method
  - Projection of polyhedral sets
    - Fourier-Motzkin Elimination
    - Block Elimination
    - Convex Hull Method (CHM)
  - Redundancy removal
    - Redundancy removal using linear programming

Outline for Part 1

- Preliminaries
  - Representations of convex polyhedra, polyhedral cones
  - Homogenization – Converting a general polyhedron in $\mathbb{R}^d$ to a cone in $\mathbb{R}^{d+1}$
  - Polar of a convex cone
- The first problem
  - Representation Conversion
    - Review of LRS
    - Double Description Method

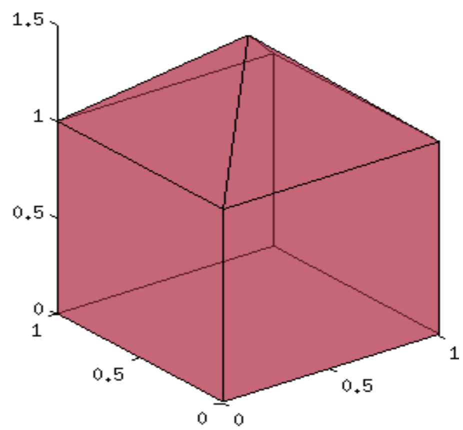
Preliminaries

Convex Polyhedron

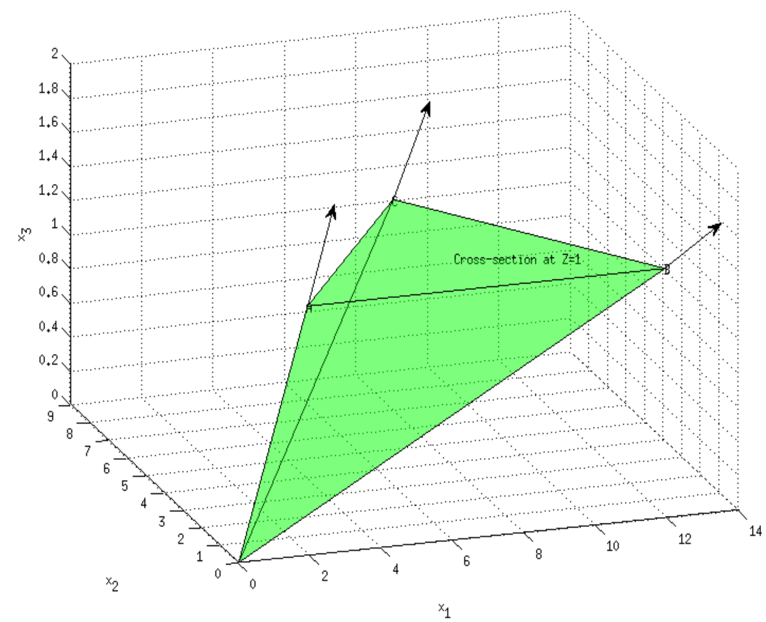
- A subset P of R^d
- The set of solutions to a finite system of linear inequalities
- Called convex polytope if it is a convex polyhedron and bounded

Examples of polyhedra

Bounded- Polytope



Unbounded - polyhedron



H-Representation of a Polyhedron

- The halfspace or inequality representation
- Polyhedron $\mathcal{P}$ is the set $x \in \mathbb{R}^n$ obeying a system of linear inequalities i.e.
- $\mathcal{P} := \{x \in \mathbb{R}^n | Hx \leq h\}$
- Can be written as $P(H, h)$
- $H \in \mathbb{R}^{m \times n}, h \in \mathbb{R}^m$ and the inequality is understood to hold elementwise
- Assumption: Polyhedron admits no lines.

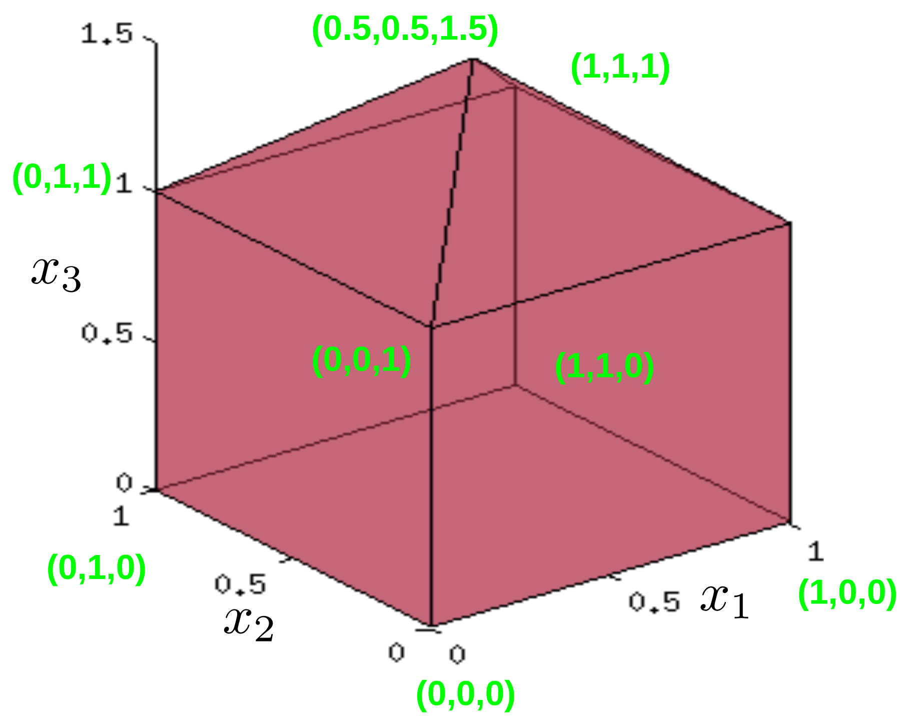
V-Representation of a Polyhedron

- $\mathcal{P} = conv\mathcal{S} + cone\mathcal{T}$
- $\mathcal{S}$ is the finite set of extreme points, $\mathcal{T}$ is the finite set of extreme directions(scaled to unit length).
- In other words, any point $x \in \mathcal{P}$ can be represented as,

  $x = \sum_{j=1}^{J} \beta_j s_j + \sum_{k=1}^{K} \gamma_k t_k$
- J is the number of extreme points, K is the number of extreme directions, $\alpha_i \in \mathbb{R}, \beta_j \geq 0, \forall j, \sum_{j=1}^{J} \beta_j = 1, \gamma_k \geq 0.$

Example

$$
\begin{aligned}
x_1 &\leq 1 \\
x_2 &\leq 1 \\
-x_1 &\leq 0 \\
-x_2 &\leq 0 \\
-x_3 &\leq 0 \\
x_2 + x_3 &\leq 2 \\
-x_2 + x_3 &\leq 1 \\
x_1 + x_3 &\leq 2 \\
-x_1 + x_3 &\leq 1
\end{aligned}
$$



Switching between the two representations : Representation Conversion Problem

- H-representation $\longrightarrow$ V-representation: The Vertex Enumeration problem
- Methods:
  - Reverse Search, Lexicographic Reverse Search
  - Double-description method
- V-representation $\longrightarrow$ H-representation

  The Facet Enumeration Problem
- Facet enumeration can be accomplished by using polarity and doing Vertex Enumeration

Polyhedral Cone

- A special polyhedron

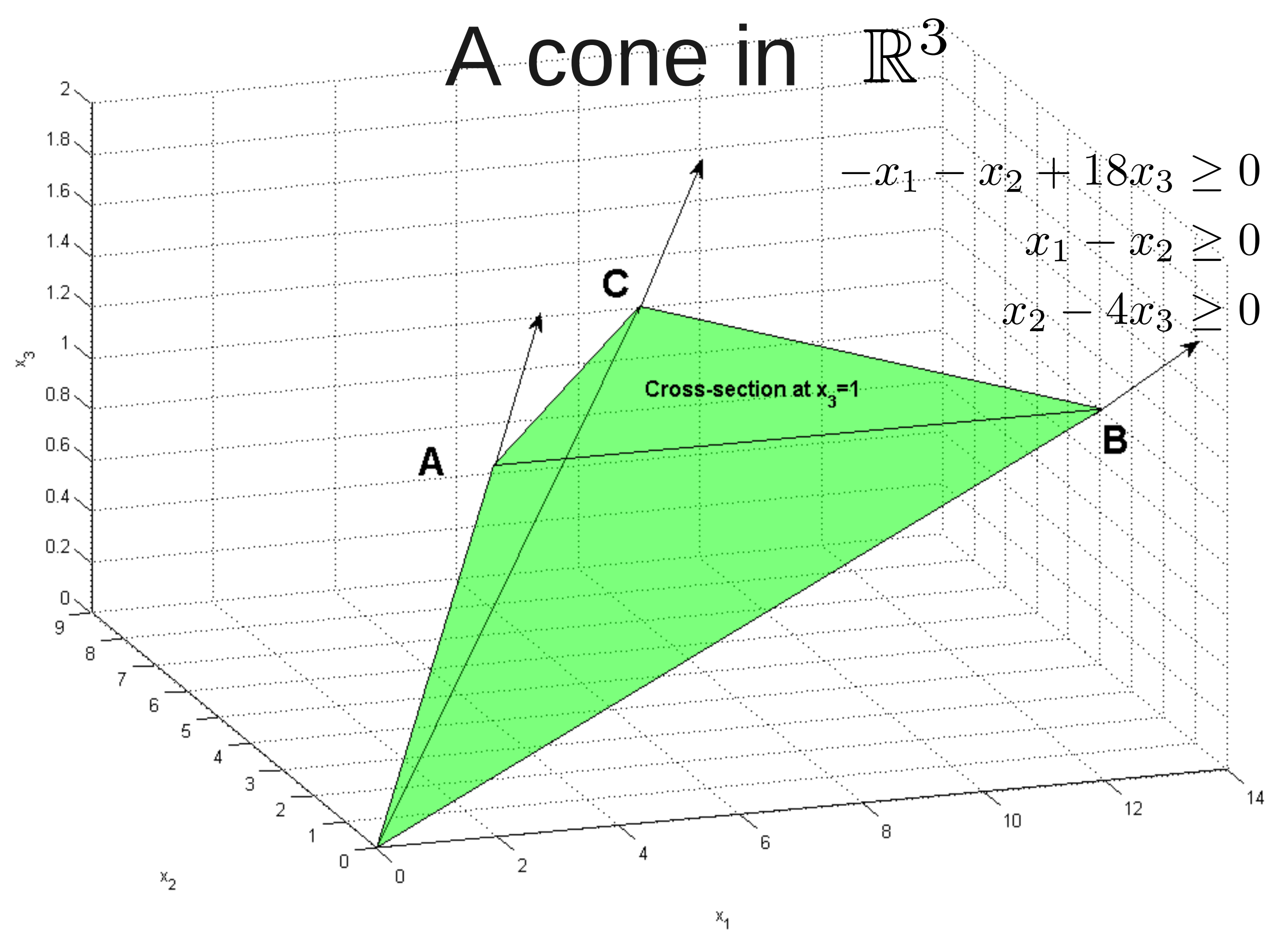
- Represented as:

- H representation:
  $C = \{x| \langle a_j, x \rangle \leq 0$ for $j = 1, 2, .., m\}$ i.e.
  $C = C(A, 0)$

- V-representation
  $C = cone(r_1, ...r_n)$
  $= \{x|x = \sum_{i=1}^{n} \mu_j r_j, \mu_j \geq 0, j = 1, ...n\}$

A cone in $\mathbb{R}^3$

$$-x_1 - x_2 + 18x_3 \geq 0$$
$$x_1 - x_2 \geq 0$$
$$x_2 - 4x_3 \geq 0$$



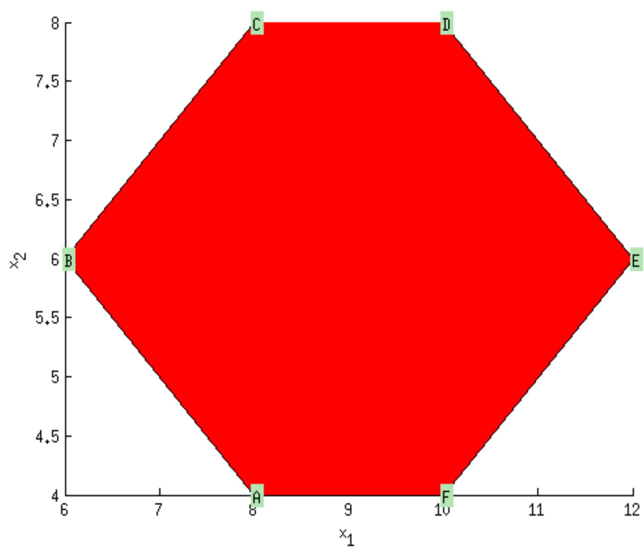
Homogenization

- Homogenization converts:
  Any polyhedron in R^d $\rightarrow$ Pointed cone in $R^{(d+1)}$
- This way, we can consider polytopes/polyhedra (bounded/unbounded) to be cones in $+1$ dimention
- Entire theory henceforth is developed for pointed polyhedral cones

H-polyhedra

- If $P = P(A, z)$ is and $\mathcal{H}$-polyhedron, we define:
- $C(P) := P\left(\begin{pmatrix} 0 & -1 \\ A & -z \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)$ which is an $\mathcal{H}$-polyhedron(a cone) in R^{d+1}
- If P is defined as $a_i x \leq z_i$, $C(P)$ is defined by inequalities $a_i x - z_i x_{d+1} \leq 0$ and $x_{d+1} \geq 0$.
- And $P = \{x \in R^d : \begin{pmatrix} x \\ 1 \end{pmatrix} \in C(P)\}$
- Conversely, if $P = P(B, 0)$ is an $\mathcal{H}$-polyhedron in R^{d+1}, then $\{x \in R^d : \begin{pmatrix} x \\ 1 \end{pmatrix} \in P\}$ is an $\mathcal{H}$-polyhedron as well

Example(d=2,d+1=3)

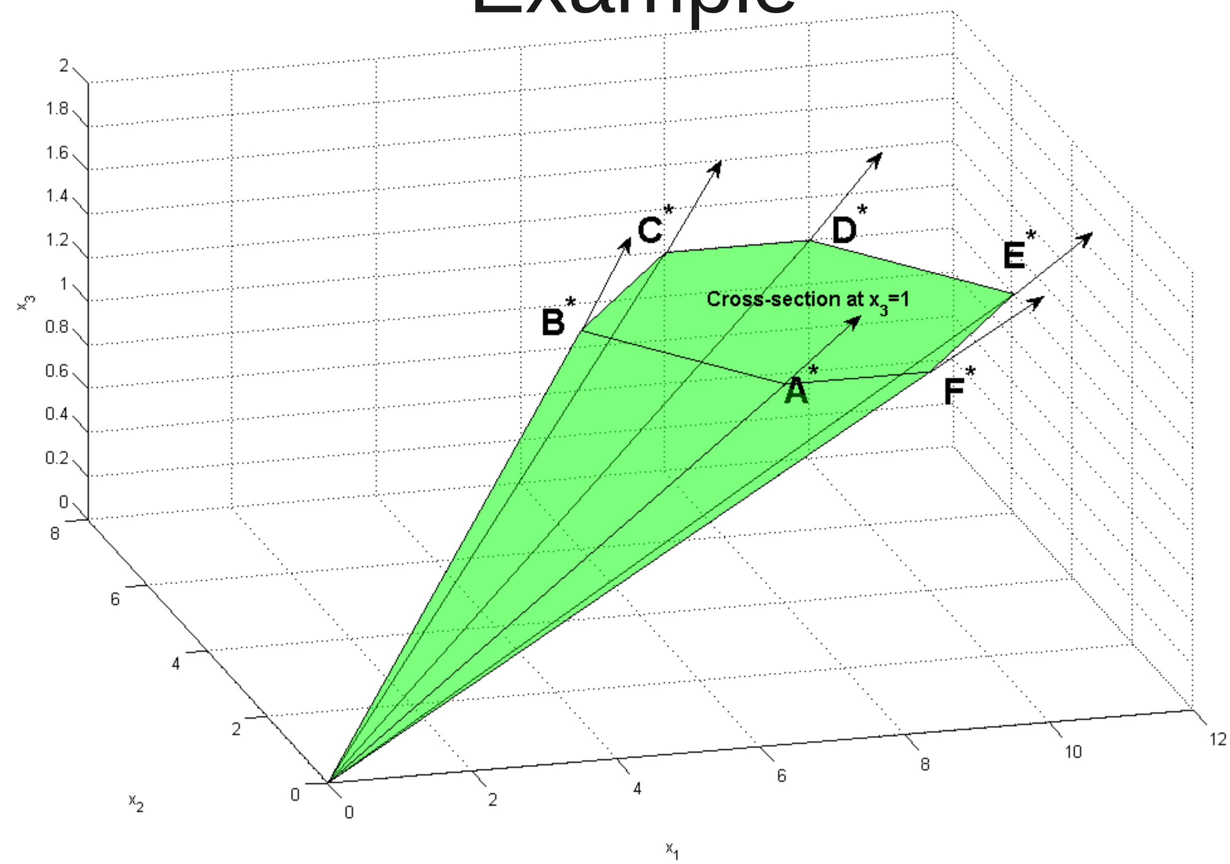


$$
\begin{aligned}
x_1 + x_2 &\le 18 \\
x_1 - x_2 &\le 6 \\
x_2 &\le 8 \\
-x_1 + x_2 &\le 0 \\
-x_1 - x_2 &\le -12 \\
-x_2 &\le -4
\end{aligned}
$$

$$\Downarrow$$

$$
\begin{aligned}
x_1 + x_2 - 18x_3 &\le 0 \\
x_1 - x_2 - 6x_3 &\le 0 \\
x_2 - 8x_3 &\le 0 \\
-x_1 + x_2 &\le 0 \\
-x_1 - x_2 + 12x_3 &\le 0 \\
-x_2 + 4x_3 &\le 0 \\
-x_3 &\le 0
\end{aligned}
$$

Example



V-polyhedra

- If $P = conv(V) + cone(Y)$ is and $\mathcal{V}$-polyhedron, we define:
- $C(P) := cone \begin{pmatrix} V & Y \\ 1 & 0 \end{pmatrix}$
  which is a $\mathcal{V} - polyhedron$ in R^{d+1}
- Conversely, if C= cone(W) is any cone in R^{d+1} generated by vectors w_i with $w_{i(d+1)} \geq 0$, then
  $\{x \in R^d : \begin{pmatrix} x \\ 1 \end{pmatrix} \in C\}$ is a $\mathcal{V}$-polyhedron

Polar of a convex cone

- One notion of duality.
- If $\{r_i | i \in I\}$ are extreme rays of a closed convex cone C, then C consists of all non-negative combinations x of the r_i's and,
  $C^\circ = \{y | \forall i \in I, \langle r_i, y \rangle \leq 0\}$ is called polar of C
- $C^{\circ\circ} = C$

Polar of a convex cone

- One notion of duality.

- If $\{r_i | i \in I\}$ are extreme rays of C, then C consists of all non-negative combinations x of the r_i's then,
  $C^\circ = \{y | \forall i \in I, \langle r_i, y \rangle \leq 0\}$ Looks like an H-representation!!!
  is called polar of C

- $C^{\circ\circ} = C$ Our ticket back to the original cone!!!

Polar: Intuition

Indicator Function

- For a convex set in R^n the indicator function $\delta(.|C)$ of C is given as:
  $\delta(x|C) = \begin{cases} 0, & \text{if } x \in C. \\ +\infty, & \text{if x} \notin \text{C.} \end{cases}$

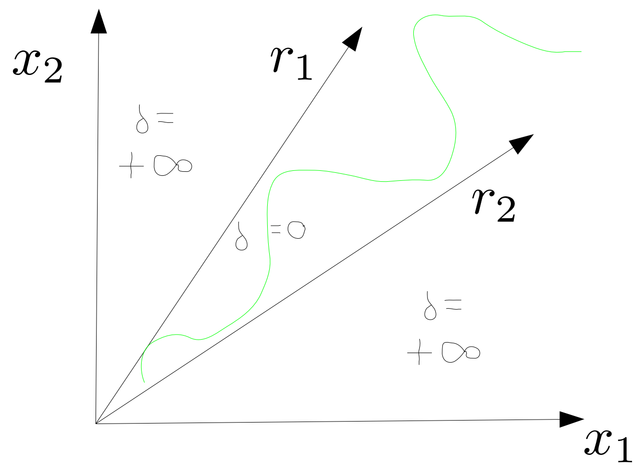
Positive Homogenoeus Function

- A function f on R^n is said to be positive homogeneous(of degree 1) if for every x one has $f(\lambda x) = \lambda f(x) \quad 0 < \lambda < \infty$

Conjugate of a convex function defined over R^n

- It is defined as:
  $f^*(y) = \sup_{x \in R^n} < y, x > -f(x)$
- For the indicator function of the convex set C defined above, it is given as,
  $\delta^*(y) = \sup_{x \in C} < y, x >$

Polar: Intuition

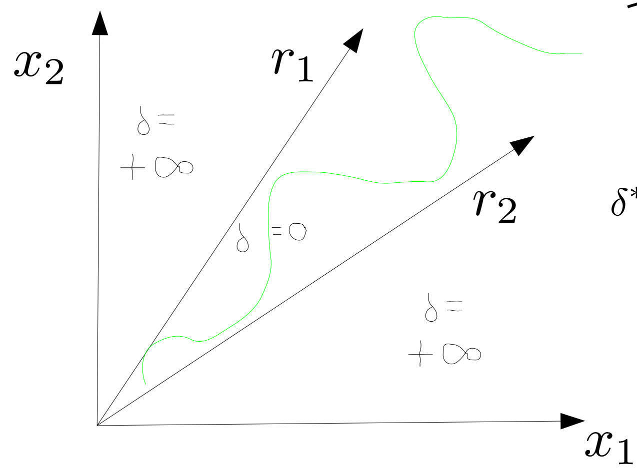


Take conjugate

$$\delta^*(y) = \sup_{x \in C} \langle y, x \rangle$$
$$= \begin{cases} \infty \text{ if } \langle y, x \rangle > 0 \text{ for some } x \in C \\ 0 \text{ if } \langle y, x \rangle \leq 0, \forall x \in C \end{cases}$$

- In first case if we find $\langle y, x \rangle > 0$ for some $x \in C$, we can always scale x to get higher value of the inner product because, for a cone, if $x \in C$, then, $\alpha x \in C, \alpha \geq 0$
- In second case 0 is the obvious supremum.

Polar: Intuition



Take conjugate

$$\delta^*(y) = \sup_{x \in C} \langle y, x \rangle = \begin{cases} \infty \text{ if } \langle y, x \rangle > 0 \text{ for some } x \in C \\ 0 \text{ if } \langle y, x \rangle \leq 0 \end{cases}$$

The set $C^\circ = \{y | \forall x \in C, \langle y, x \rangle \leq 0\}$ so obtained is called the polar of C

Polar: Intuition

Every closed convex cone has a positive homogeneous indicator function associated with it

$$\Downarrow$$

The conjugate of this indicator function is also a positive homogeneous indicator function

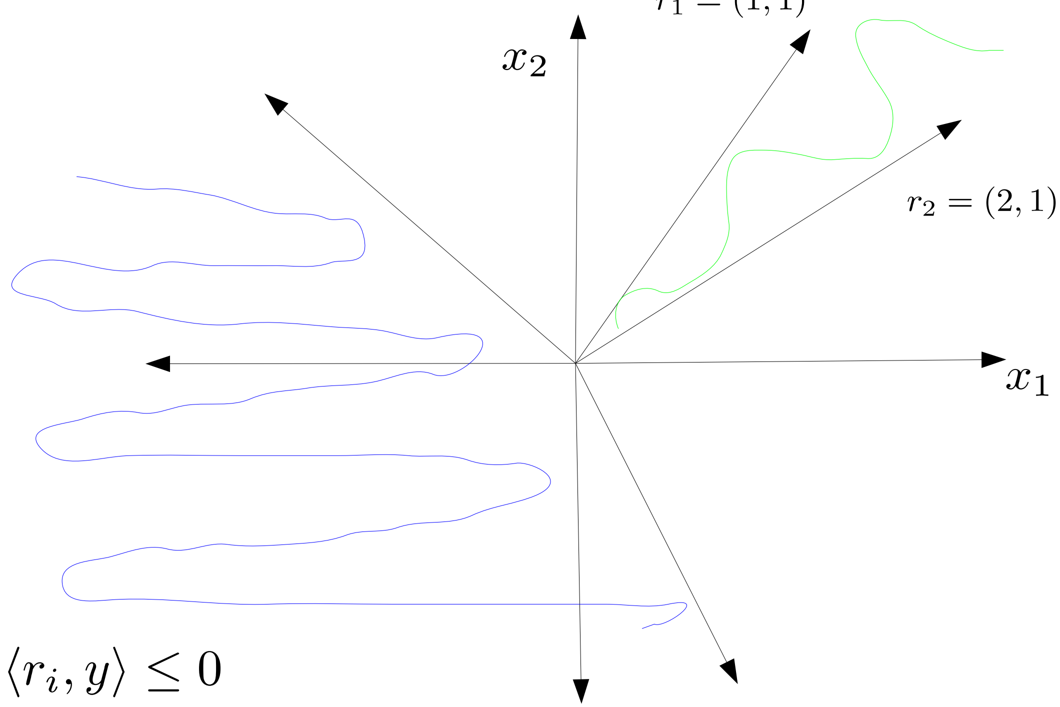
$$\Downarrow$$

The set associated with it is also a cone and is called polar of C and denoted as C°

Use of Polar

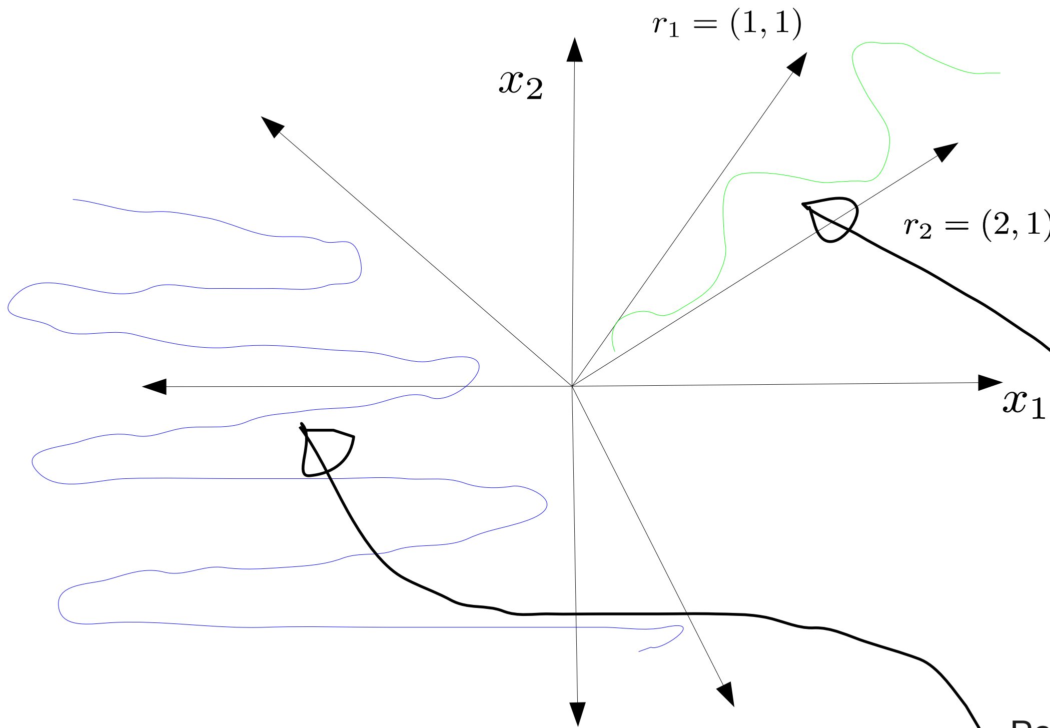
- Representation Conversion
  - No need to have two different algorithms i.e. for $H \rightarrow V$ and $V \rightarrow H$.
- Redundancy removal
  - No need to have two different algorithms for removing redundancies from H-representation and V-representation
- Safely assume that input is always an H-representation for both these problems

Polarity: Intuition



$\langle r_i, y \rangle \leq 0$
$x_1 + x_2 \leq 0$
$2x_1 + x_2 \leq 0$
(Plotting in same space as the original cone)

Polar: Intuition



$$\langle r_i, y \rangle \leq 0$$
$$x_1 + x_2 \leq 0$$
$$2x_1 + x_2 \leq 0$$

Then take polar again to get facets of this cone

Perform Vertex Enumeration on this cone.

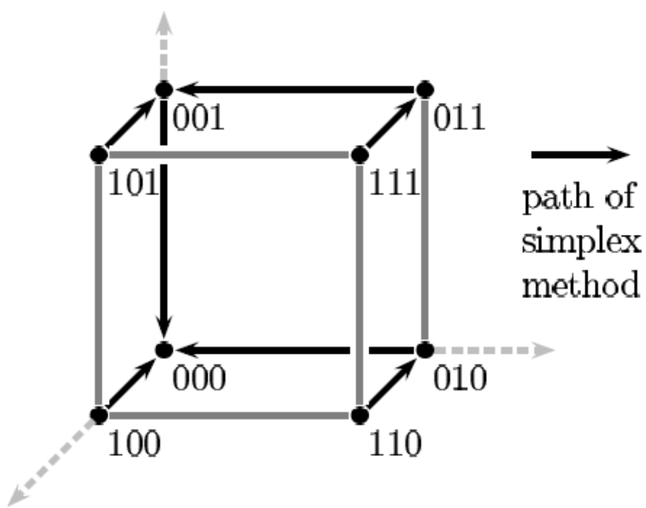
Problem #1
Representation Conversion

Review of Lexicographic Reverse Search

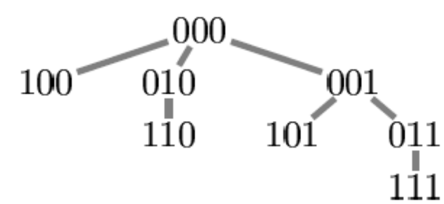
Reverse Search: High Level Idea

- Start with dictionary corresponding to the optimal vertex
- Ask yourself 'What pivot would have landed me at this dictionary if i was running simplex?'
- Go to that dictionary by applying reverse pivot to current dictionary
- Ask the same question again
- Generate the so-called 'reverse search tree'

Reverse Search on a Cube



(a) The "simplex tree" induced by the objective $(-\sum x_i)$.



(b) The corresponding reverse search tree.

Ref.David Avis, *lrs: A Revised Implementation of the Reverse Search Vertex Enumeration Algorithm*

Double Description Method

- First introduced in:

  " Motzkin, T. S.; Raiffa, H.; Thompson, G. L.; Thrall, R. M. (1953). "The double description method". *Contributions to the theory of games. Annals of Mathematics Studies*. Princeton, N. J.: Princeton University Press. pp. 51–73"

- The primitive algorithm in this paper is very inefficient.

  (How? We will see later)

- Several authors came up with their own efficient implementation(viz. Fukuda, Padberg)
- Fukuda's implementation is called *cdd*

Some Terminology

- A pair (A, R) is said to be a double description pair (DD pair) if the relationship:
- $Ax \geq 0$ iff $x = R\lambda$ for some $\lambda \geq 0$ holds
- Column size of A= Row Size of $R = d$
- Provides two different descriptions of the same object: A *Polyhedral Cone*, formally defined as:
- A set $P(A)$ represented by A as: $P(A) = \{x \in \mathbb{R}^d : Ax \geq 0\}$ and is simualteneously represented by R as: $\{x \in \mathbb{R}^d : x = R\lambda \text{ for some } \lambda \geq 0\}$
- A is called the representation matrix, while R is called the generator matrix.

Double Description Method: The High Level Idea

- An *Incremental* Algorithm
- Starts with certain subset of rows of H-representation of a cone $Ax \geq 0$ to form initial H-representation
- Adds rest of the inequalities one by one constructing the corresponding V-representation every iteration
- Thus, constructing the V-representation *incrementally.*

How it works?

Initialization:

- Let $K \subset \{1, ...m\}$ i.e. the row indices of A
- Let A_K denote the submatrix of A consisting of rows indexed by K
- Suppose we have already found a generating matrix R of $P(A_K)$ i.e. (A_K, R) is a DD pair

Iteration:

- Given a DD pair (A_K, R), select any row index $i \notin K$ and construct a DD pair (A_{K+i}, R') using the DD pair (A_K, R)

Termination:

- If $A = A_K$, we are done.

How it works?

Initialization:

- Let $K \subset \{1, ...m\}$ i.e. the row indices of A
- Let A_K denote the submatrix of A consisting of rows indexed by K
- Suppose we have already found a generating matrix R of $P(A_K)$ i.e. (A_K, R) is a DD pair

Iteration:

- Given a DD pair (A_K, R), select any row index $i \notin K$ and construct a DD pair (A_{K+i}, R') using the DD pair (A_K, R)

Termination:

- If $A = A_K$, we are done.

Initialization

- method 1:
  Find a DD pair (A_K, R) when $|K| = 1$.

- method 2:
  Select a maximal submatrix A_K of A consisting of linearly independent rows of A.
  The vectors r_j's are obtained by solving the system:
  $A_K R = I$ where I is $|K| \times |K|$
  $A_K x \geq 0 \leftrightarrow \mathrm{x} = \mathrm{A}_K^{-1}\lambda, \lambda \geq 0$

Initialization

- method 1: Very trivial and inefficient
  Find a DD pair (A_K, R) when $|K| = 1$.

- method 2:
  Select a maximal submatrix A_K of A consisting of linearly independent rows of A.
  The vectors r_j's are obtained by solving the system:
  $A_K R = I$ where I is $|K| \times |K|$
  $A_K x \geq 0 \leftrightarrow \mathrm{x} = \mathrm{A}_K^{-1}\lambda, \lambda \geq 0$

Example: Initialization

Consider the problem of performing vertex enumeration on the polyhedron represented as follows:

$$-x_1 - x_2 + 18x_3 \geq 0 \tag{1}$$
$$-x_1 + x_2 + 6x_3 \geq 0 \tag{2}$$
$$x_1 - x_2 + 8x_3 \geq 0 \tag{3}$$
$$x_1 - x_2 \geq 0 \tag{4}$$
$$x_1 + x_2 - 12x_3 \geq 0 \tag{5}$$
$$x_2 - 4x_3 \geq 0 \tag{6}$$

Example: Initialization

An Example:
Consider the problem of performing vertex enumeration on the polyhedron represented as follows:

$$-x_1 - x_2 + 18x_3 \geq 0 \tag{1}$$
$$-x_1 + x_2 + 6x_3 \geq 0 \tag{2}$$
$$x_1 - x_2 + 8x_3 \geq 0 \tag{3}$$
$$x_1 - x_2 \geq 0 \tag{4}$$
$$x_1 + x_2 - 12x_3 \geq 0 \tag{5}$$
$$x_2 - 4x_3 \geq 0 \tag{6}$$

Example : Initialization

$$A_{\{1,4,6\}} = \begin{array}{rrr} -1 & -1 & 18 \\ 1 & -1 & 0 \\ 0 & 1 & -4 \end{array}$$

$$A^{-1}_{\{1,4,6\}} = R = \begin{array}{ccc} \text{A} & \text{B} & \text{C} \\ 4.0000 & 14.0000 & 9.0000 \\ 4.0000 & 4.0000 & 9.0000 \\ 1.0000 & 1.0000 & 1.0000 \end{array}$$

How it works?

Initialization:

- Let $K \subset \{1, ...m\}$ i.e. the row indices of A
- Let A_K denote the submatrix of A consisting of rows indexed by K
- Suppose we have already found a generating matrix R of $P(A_K)$ i.e. (A_K, R) is a DD pair

Iteration:

- Given a DD pair (A_K, R), select any row index $i \notin K$ and construct a DD pair (A_{K+i}, R') using the DD pair (A_K, R)

Termination:

- If $A = A_K$, we are done.

Iteration: Insert a new constraint

- The newly inserted inequality $A_i x \geq 0$ partitions the space $\mathbb{R}^d$ into three parts:

$$H_i^+ = \{x \in \mathbb{R}^d : A_i x > 0\}$$
$$H_i^0 = \{x \in \mathbb{R}^d : A_i x = 0\}$$
$$H_i^- = \{x \in \mathbb{R}^d : A_i x < 0\}$$

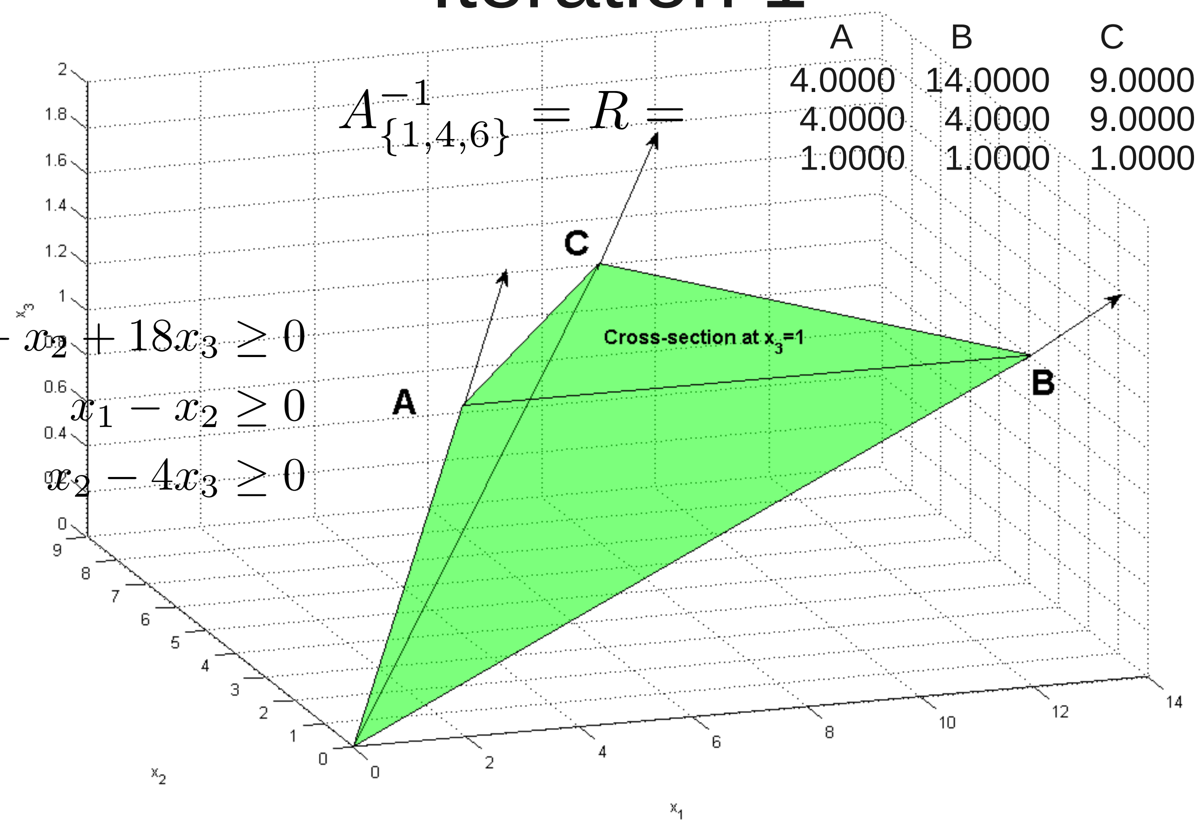
- Let J be the set of column indices of R. The rays $r_j (j \in J)$ are accordingly partitioned as:

$$J^+ = \{j \in J : r_j \in H_i^+\}$$
$$J^0 = \{j \in J : r_j \in H_i 0\}$$
$$J^- = \{j \in J : r_j \in H_i^-\}$$

Iteration 1

$$A_{\{1,4,6\}}^{-1} = R = \begin{array}{ccc} A & B & C \\ 4.0000 & 14.0000 & 9.0000 \\ 4.0000 & 4.0000 & 9.0000 \\ 1.0000 & 1.0000 & 1.0000 \end{array}$$

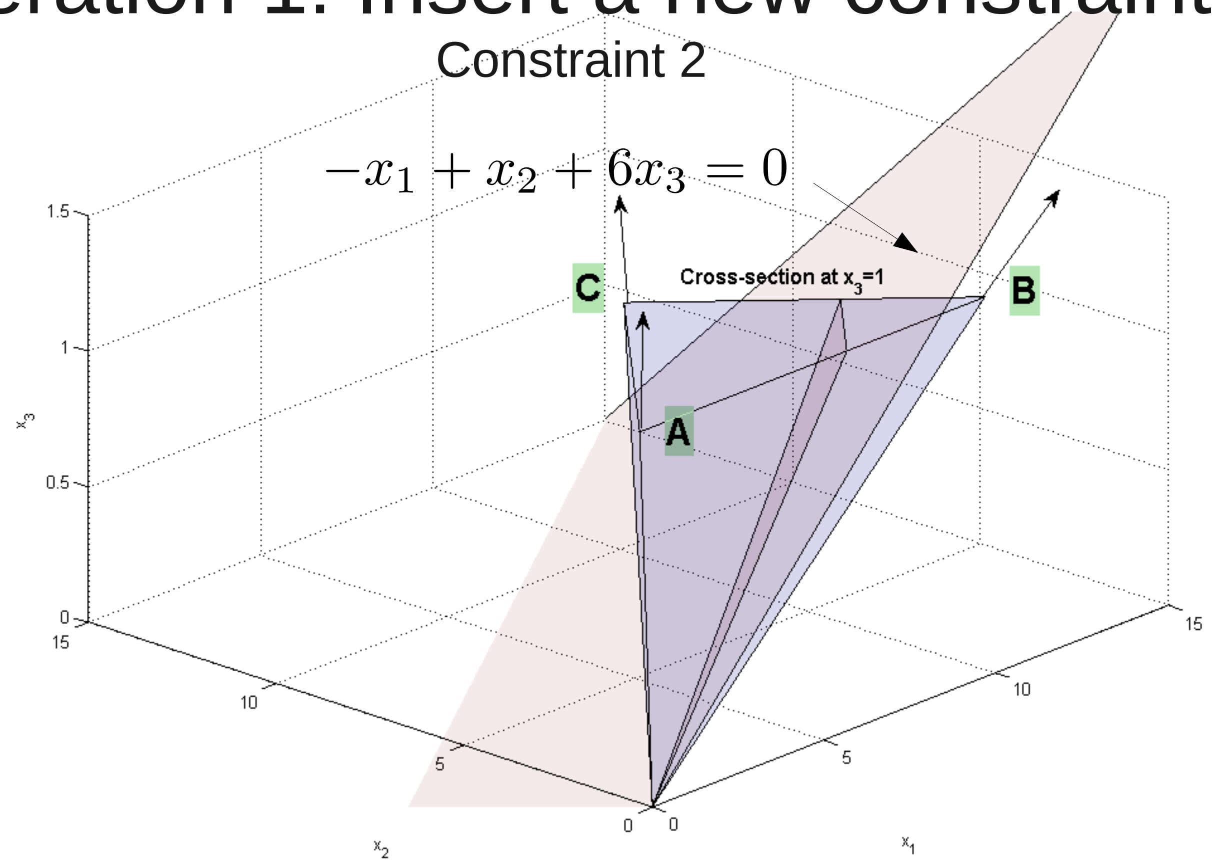
$$-x_1 - x_2 + 18x_3 \geq 0$$
$$x_1 - x_2 \geq 0$$
$$x_2 - 4x_3 \geq 0$$



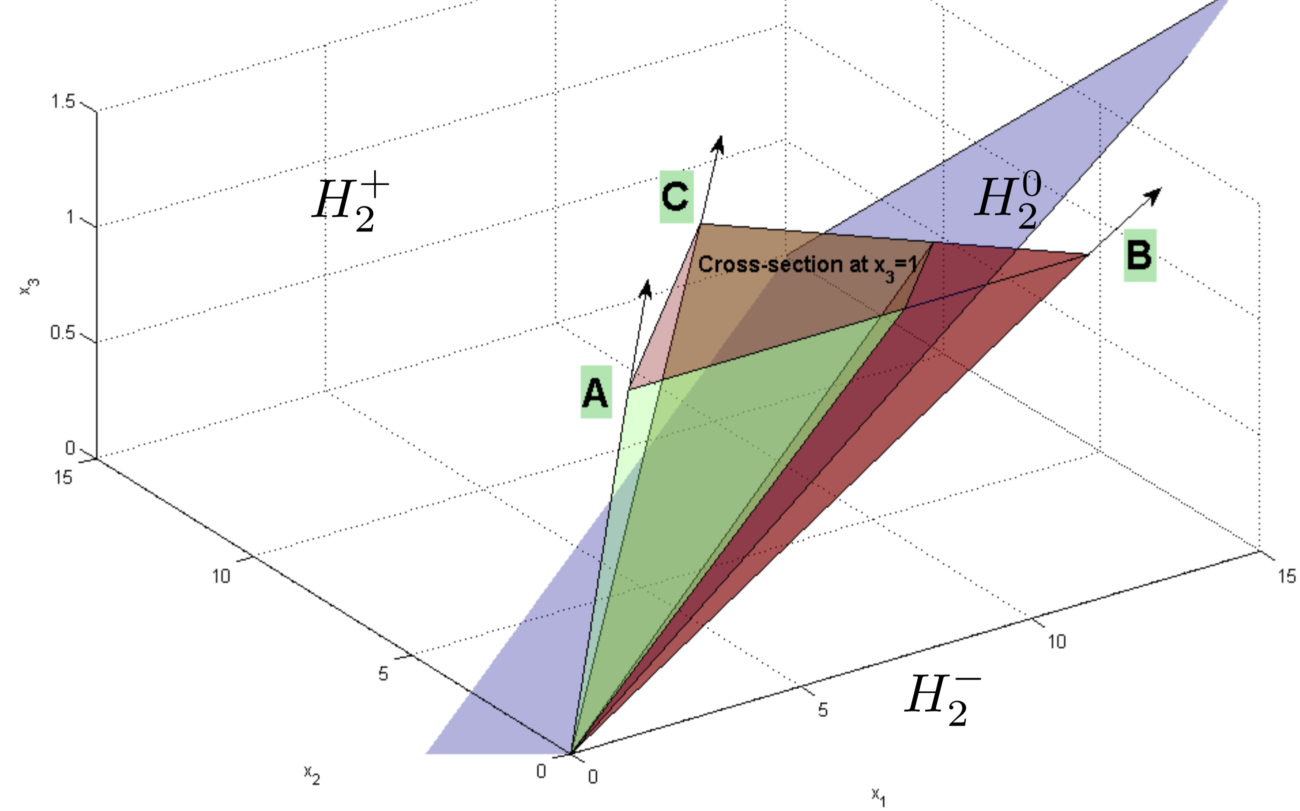
Iteration 1: Insert a new constraint

Constraint 2

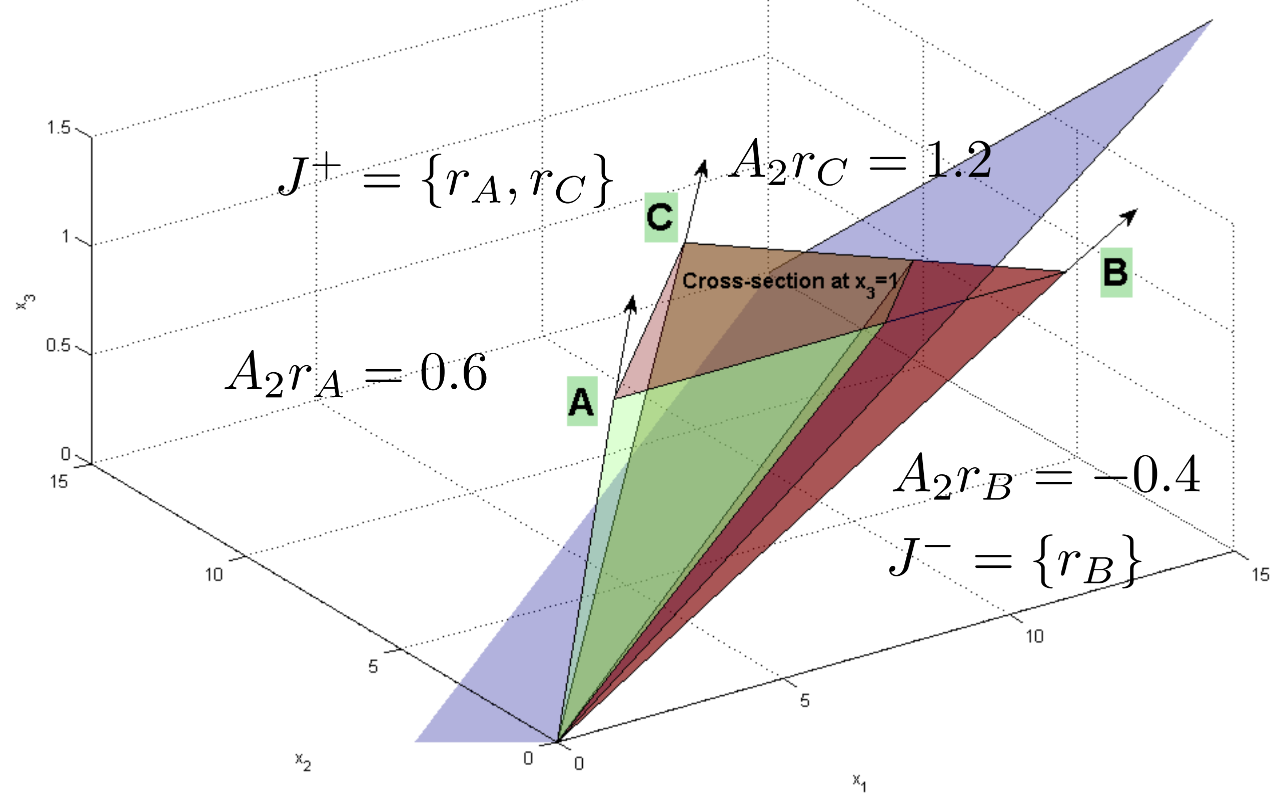
$$-x_1 + x_2 + 6x_3 = 0$$



Iteration 1: Insert a new constraint



Iteration 1: Insert a new constraint



Main Lemma for DD Method

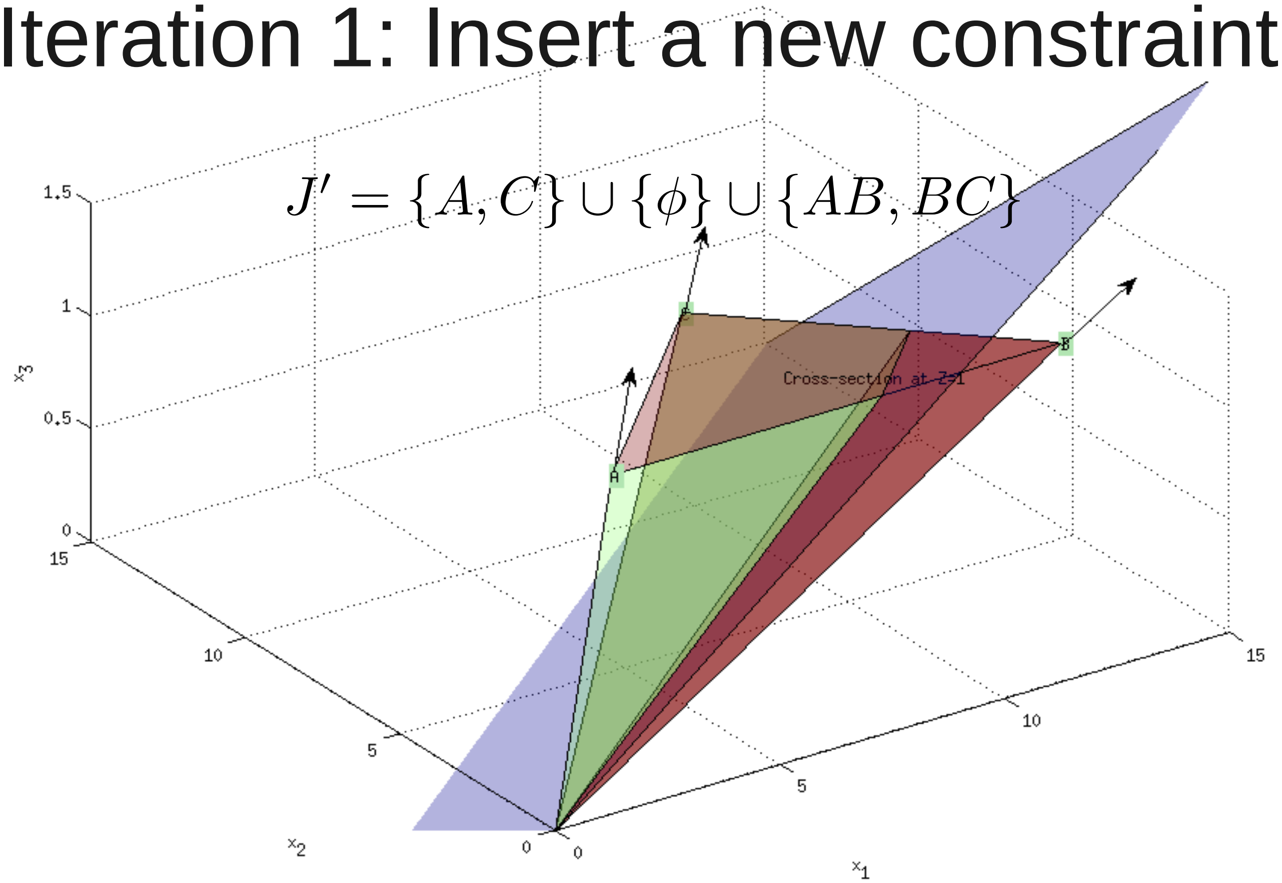
Let (A_K, R) be a DD pair and let i be the new row index of A not in K. Then the pair (A_{K+i}, R') is a DD pair, where R' is the $d \times |J'|$ matrix with column vectors $r_j (j \in J')$ defined by,

$$J' = J^+ \cup J^0 \cup (J^+ \times J^-), \text{and}$$

$$r_{jj'} = (A_i r_j) r_{j'} - (A_i r_{j'}) r_j \text{ for each } (j, j') \in J^+ \times J^-$$

Iteration 1: Insert a new constraint

$$J' = \{A, C\} \cup \{\phi\} \cup \{AB, BC\}$$

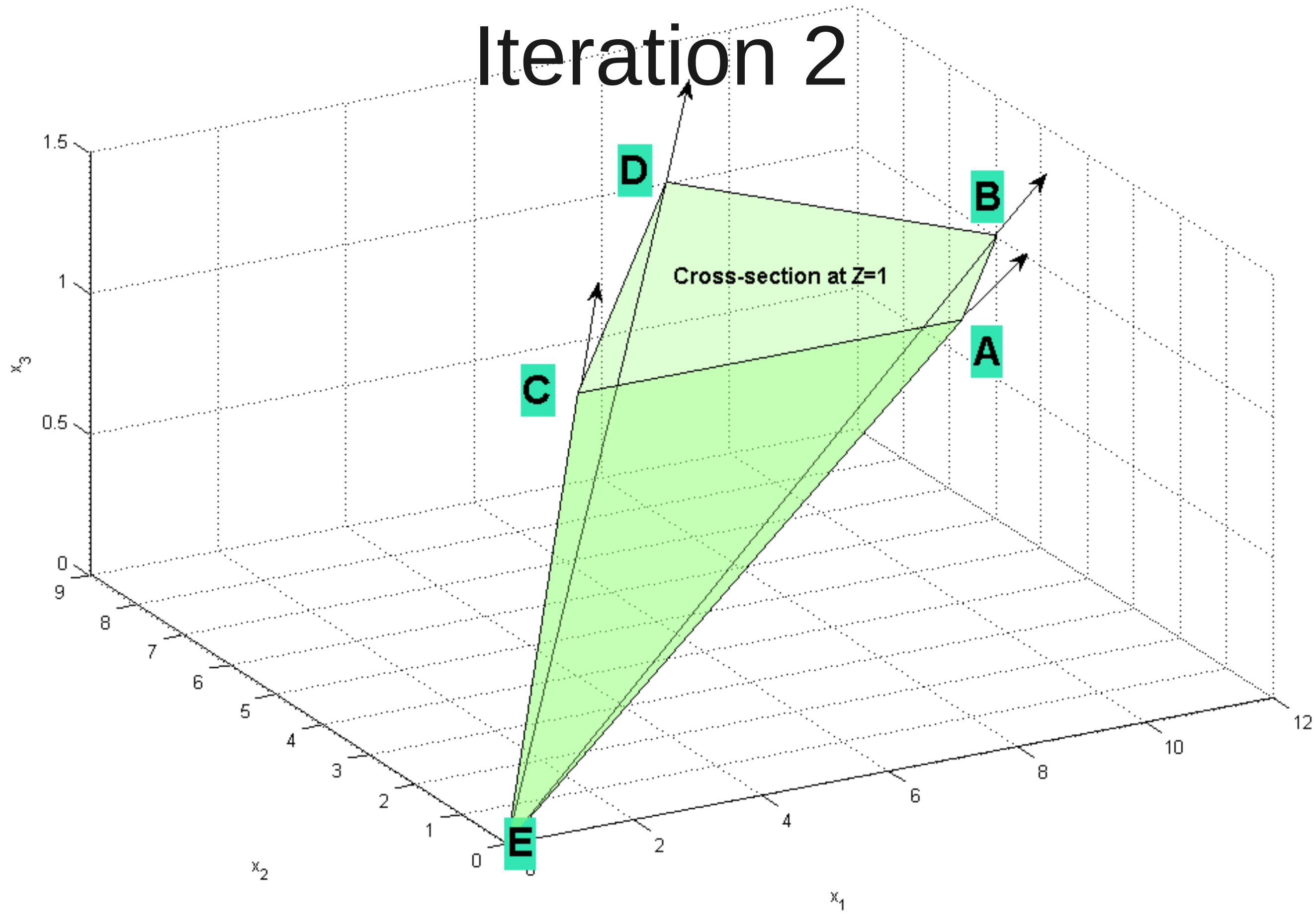


Iteration 1: Get the new DD pair

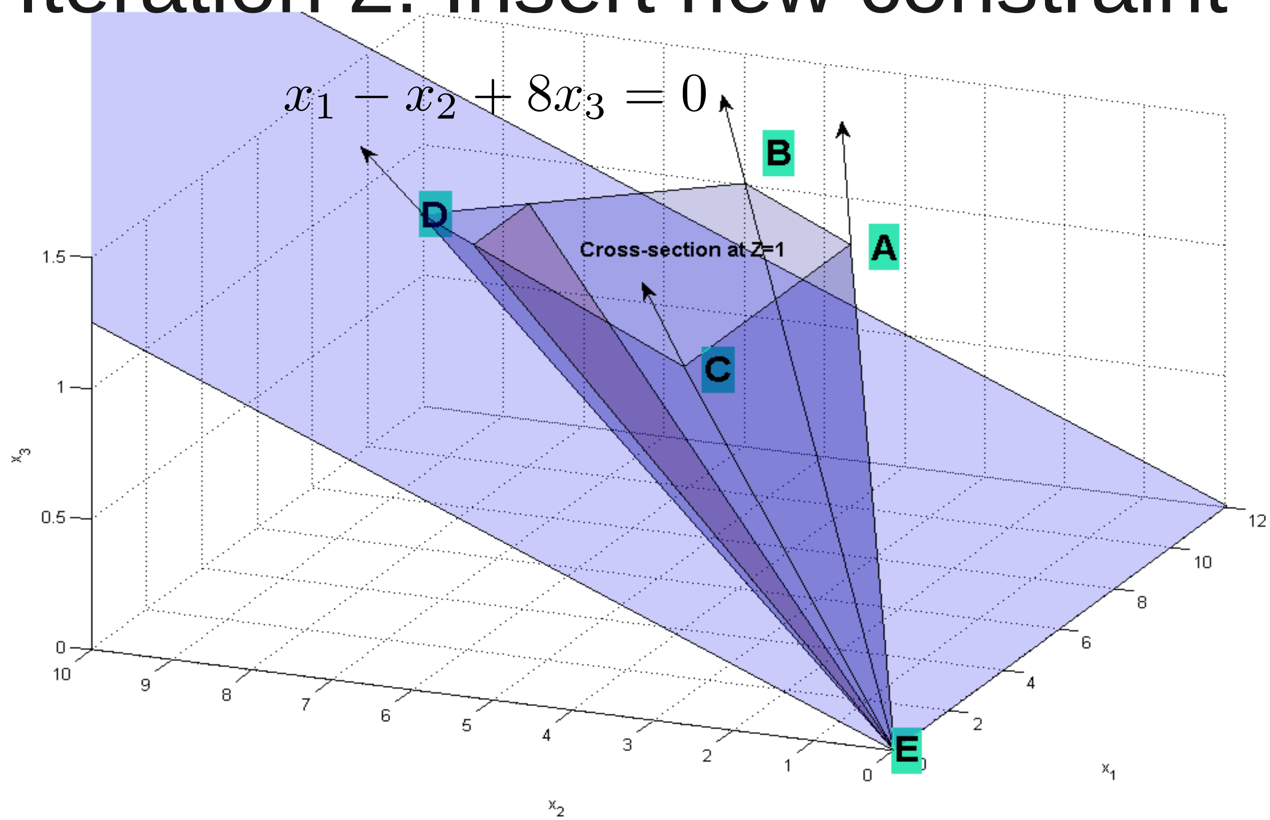
$$A_{\{1,2,4,6\}} \quad \begin{array}{rrr} -1 & -1 & 18 \\ 1 & -1 & 0 \\ 0 & 1 & -4 \\ -1 & 1 & 6 \end{array}$$

$$R_{\{1,2,4,6\}} \quad \begin{array}{cccc} \mathrm{r}_{AB} & r_{BC} & r_{A} & \mathrm{r}_{C} \\ 10.0000 & 12.0000 & 4.0000 & 9.0000 \\ 4.0000 & 6.0000 & 4.0000 & 9.0000 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{array}$$

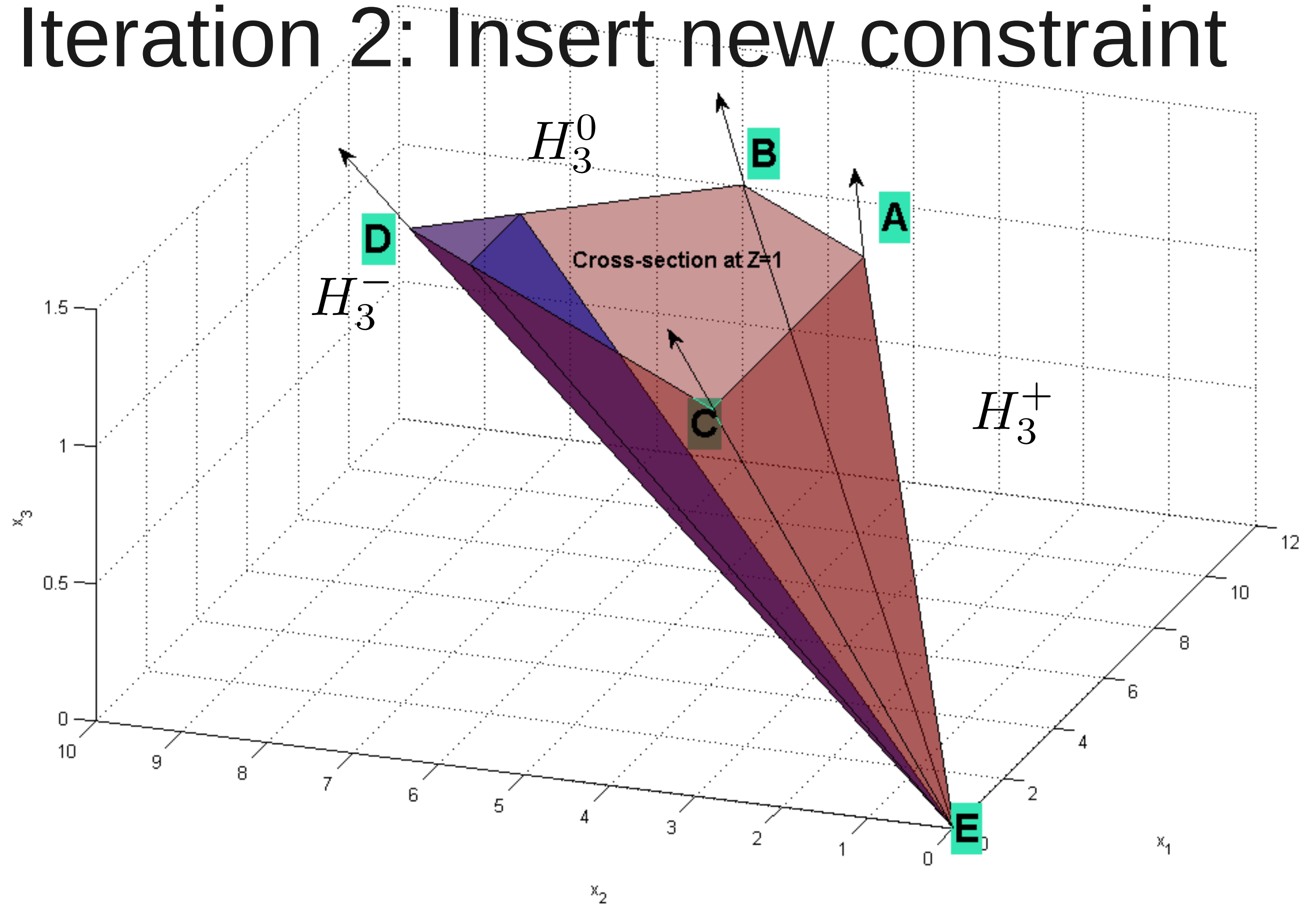
Iteration 2



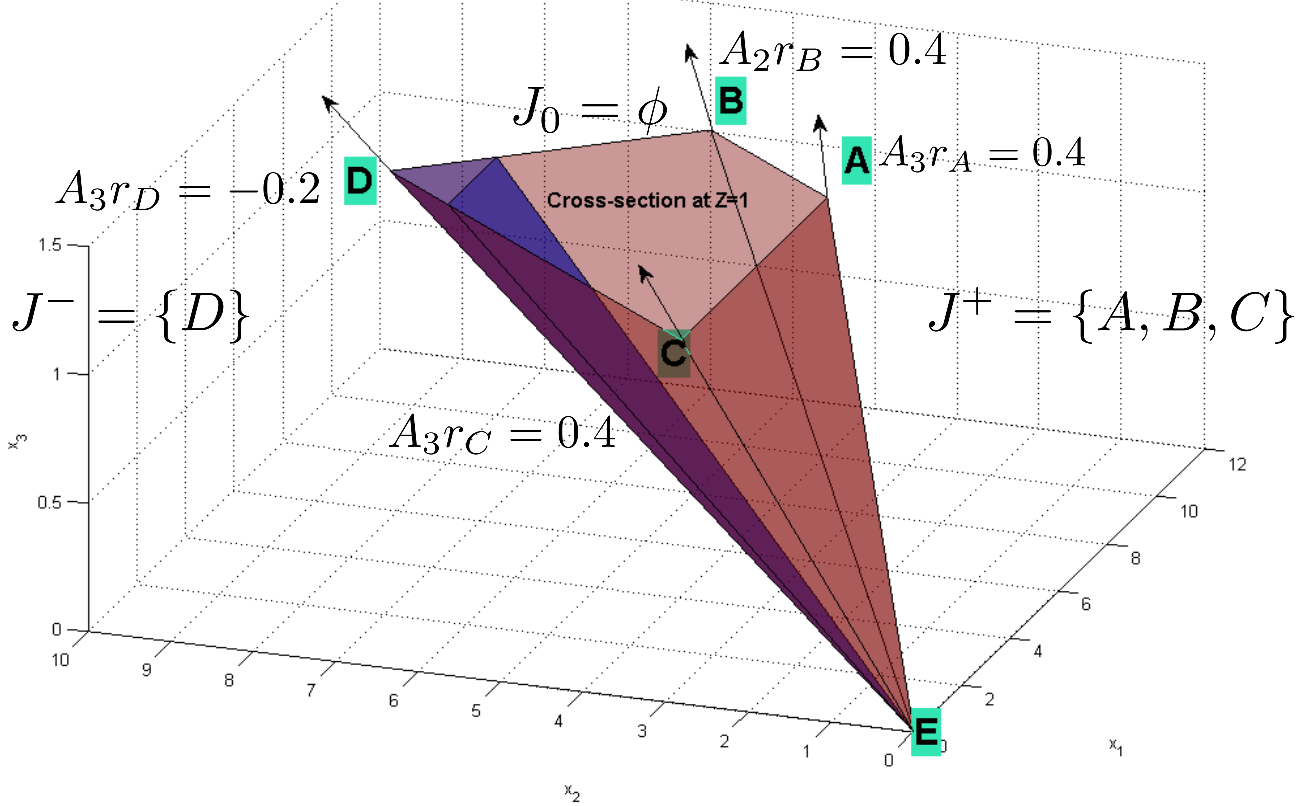
Iteration 2: Insert new constraint



Iteration 2: Insert new constraint



Iteration 2: Insert new constraint



Get the new DD pair

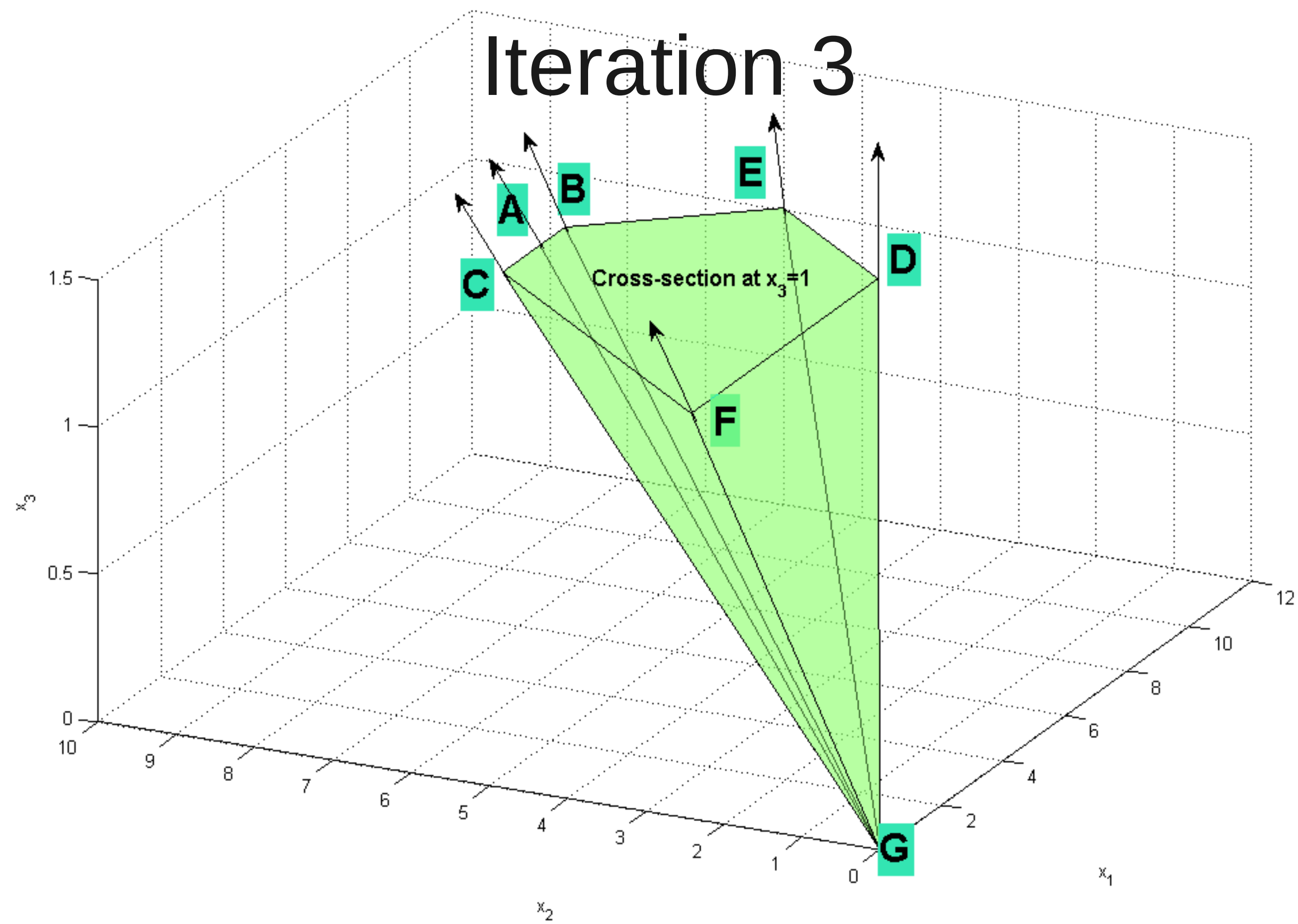
$A_{\{1,2,3,4,6\}}$

-1	-1	18
1	-1	0
0	1	-4
-1	1	6
0	-1	8

$R_{\{1,2,3,4,6\}}$

r_{AD}	r_{BD}	r_{CD}	r_A	r_B	r_C
9.2000	10.0000	8.0000	10.0000	12.0000	4.0000
8.0000	8.0000	8.0000	4.0000	6.0000	4.0000
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Iteration 3



A
B
C
D
E
F
G
Cross-section at x_3=1
x_1
x_2
x_3
0
0.5
1
1.5
0
1
2
3
4
5
6
7
8
9
10
2
4
6
8
10
12

Get the new DD pair

$A_{\{1,2,3,4,5,6\}}$

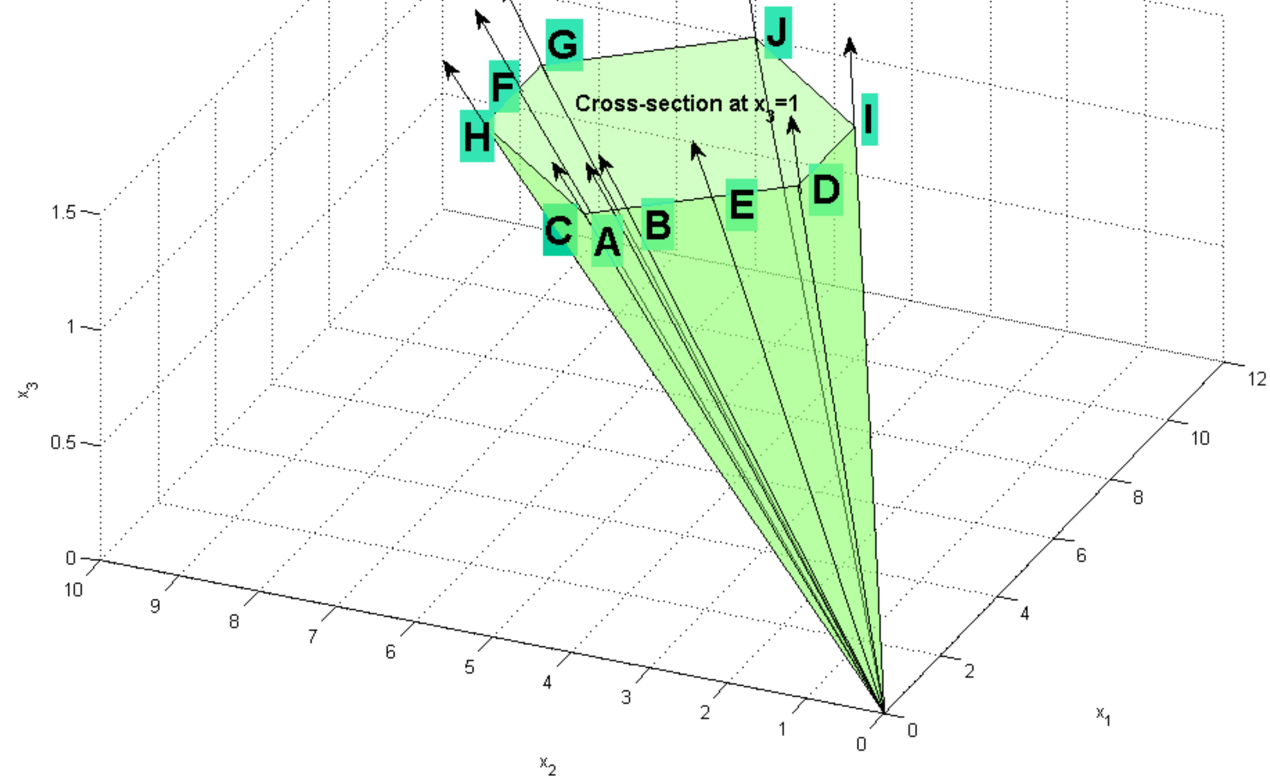
-1	-1	18
1	-1	0
0	1	-4
-1	1	6
0	-1	8
1	1	-12

$R_{\{1,2,3,4,5,6\}}$

A	B	C	D	E	F	G	H	I	J
6.2609	6.4000	6.0000	8.0000	7.2000	9.2000	10.0000	8.0000	10.0000	12.0000
5.7391	5.6000	6.0000	4.0000	4.8000	8.0000	8.0000	8.0000	4.0000	6.0000
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Termination

$$A_K = A$$



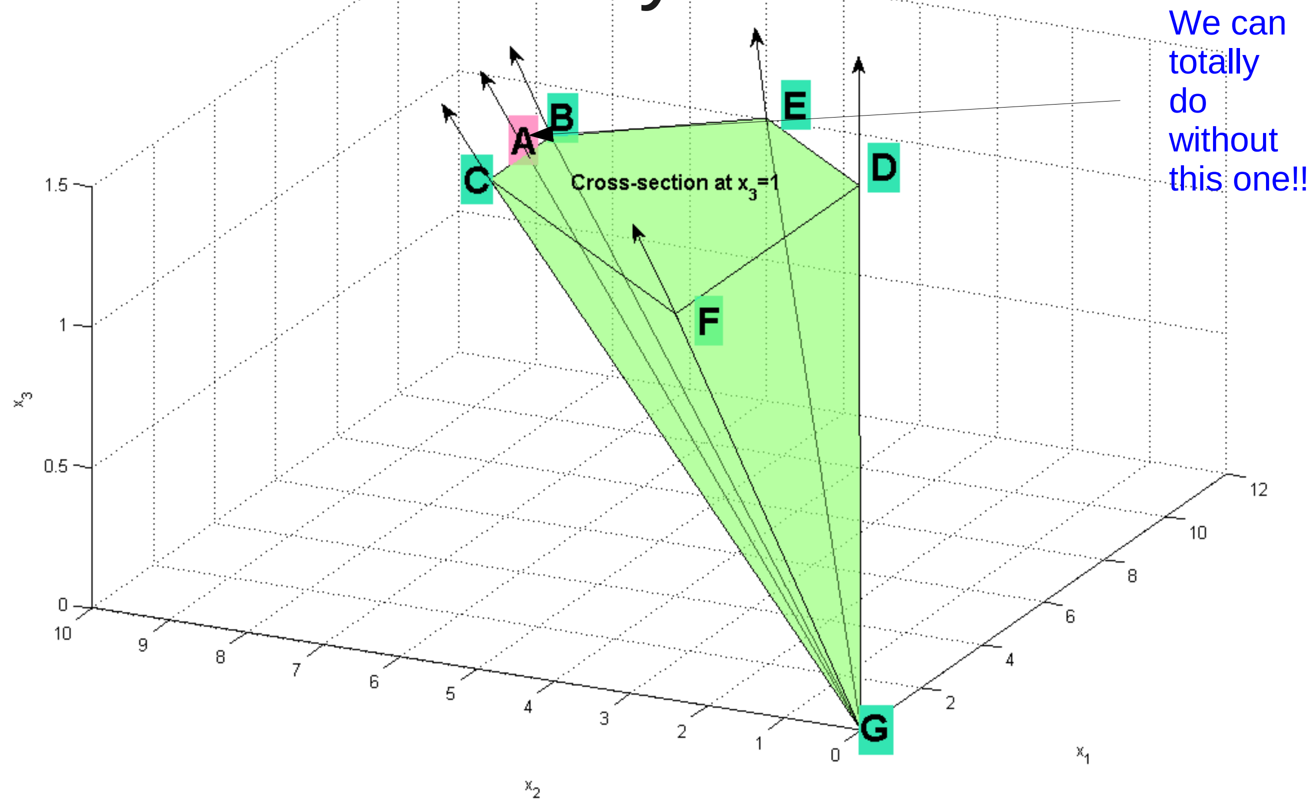
Efficiency Issues

- Is the implementation I just described good enough?

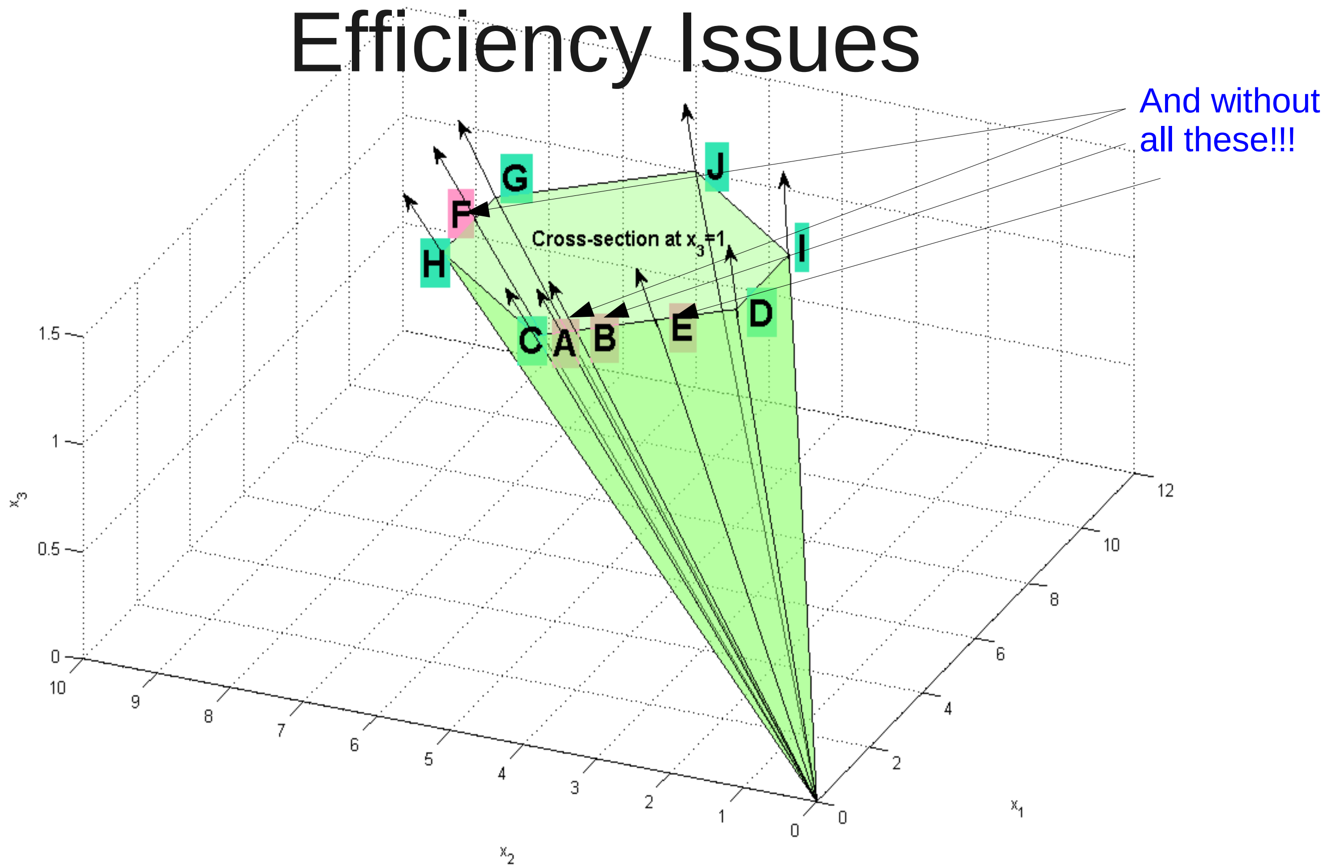
Efficiency Issues

- Is the implementation I just described good enough?
  - Hell no!
    - The implementation just described suffers from profusion of redundancy

Efficiency Issues



Efficiency Issues



Efficiency Issues

- In worst case an iteration can start with n extreme rays and end with $(\frac{n}{2})^2$ extreme rays.
- Hence, the number extreme rays can very soon grow out of hand.
- A straightforward implementation is quite useless.
- Redundancy removal for n extreme rays is equivalent to solving n linear programs which is also not a very exciting prospect.
- Hence, we focus on **Not letting redundant extreme rays to be created** in first place
- Fukuda's main contributions are in that direction

The Primitive DD method

procedure DoubleDescriptionMethod(A);
begin
    Obtain any initial DD pair (A_K, R)
    while $K \neq \{1, 2, ..., m\}$ **do**
    begin
        Select any index i from $\{1, 2, ..., m\}$;
        Construct a DD pair (A_{K+i}, R') from (A_K, R);
        $R := R'$; $K := K + i$
    end
    Output R
begin

What to do?

- Add some more structure
- Strengthen the *Main Lemma*

Add some more structure

Some definitions

ray of P

- r is said to be a *ray* of P if $r \neq 0$ and $\alpha r \in P \forall \alpha > 0$
- If r and r' are such that $r = \alpha r'$ for some positive number α, we say $r \simeq r'$

zero set/active set

- For any vector $x \in P$, we define the *zero set* or *active set* $Z(x)$ as the set of inequality indices i such that $A_i x = 0$

Proposition 4. (Fukuda)

- Let r be a ray of P, $\bar{F} := \{x : A_{Z(r)}x = 0\}, F := \bar{F} \cap P$ and $rank(A_{Z(r)}) = d - k$
  then

  (a) $rank(A_{Z(r)\cup\{i\}}) = d - k + 1 \forall i \neq Z(r)$;
  (b) F contains k linearly independent rays;
  (c) If $k \geq 2$ then r is a non-negative combination of two distinct rays r_1 and r_2 with
  $rank(A(Z(r_i))) > d - k, i = 1, 2$

Proposition 7. (Fukuda)

- Let r and r' be distinct rays of P. Then the following statements are equivalent:
  (a) r and r' are adjacent extreme rays;
  (b) r and r' are extreme rays and rank of the matrix $A_{Z(r)(r')}$ is $d-2$
  (c) if r'' is a ray with $Z(r'') \supset Z(r) \cap Z(r')$ then either $r'' \simeq r$ or $r'' \simeq r'$;

Proposition 7. (Fukuda)

- Let r and r' be distinct rays of P. Then the following statements are equivalent:

  (a) r and r' are adjacent extreme rays;

  (b) r and r' are extreme rays and rank of the matrix $A_{Z(r)(r')}$ is $d-2$

  (c) if r'' is a ray with $Z(r'') \supset Z(r) \cap Z(r')$ then either $r'' \simeq r$ or $r'' \simeq r'$;

Algebraic Characterization of adjacency

Proposition 7. (Fukuda)

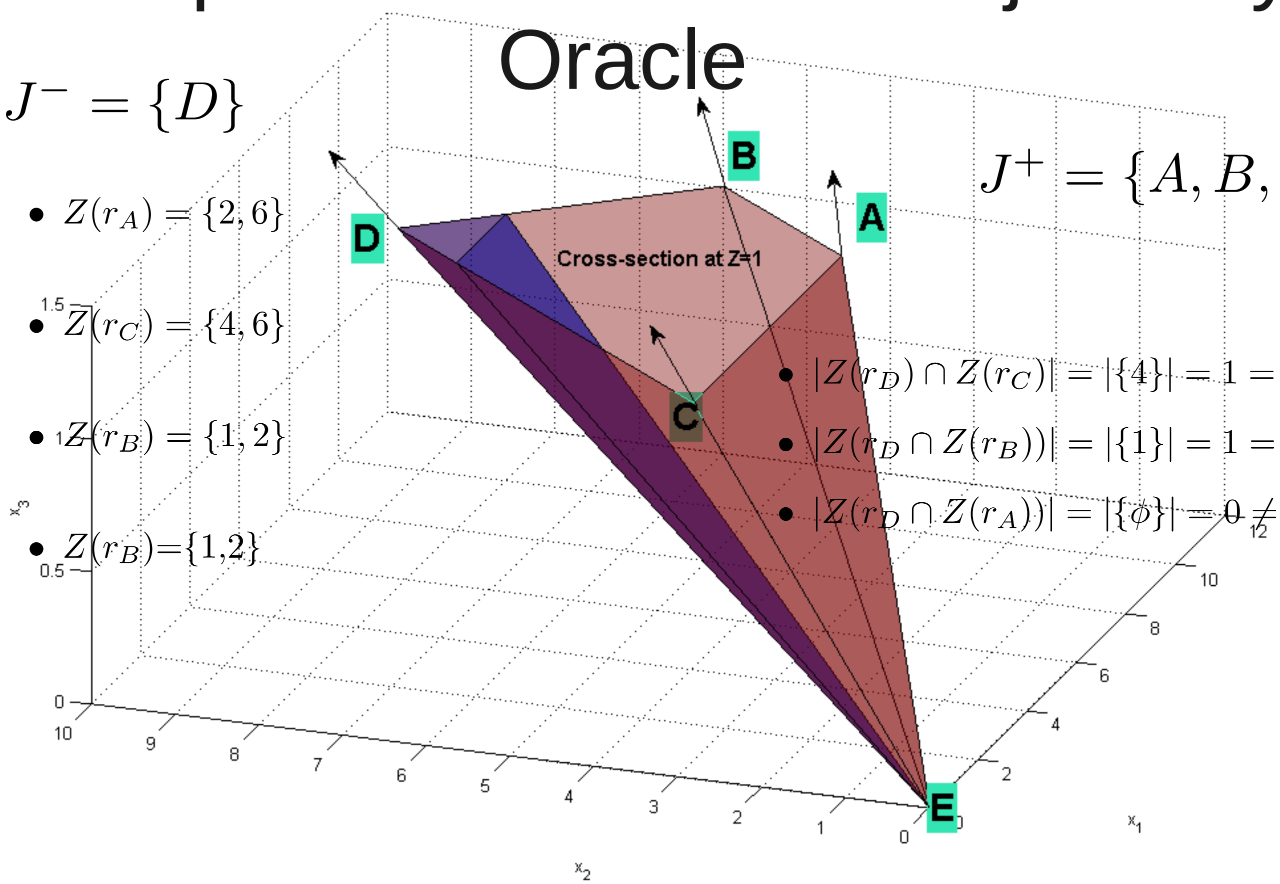
- Let r and r' be distinct rays of P. Then the following statements are equivalent:
  (a) r and r' are adjacent extreme rays;
  (b) r and r' are extreme rays and rank of the matrix $A_{Z(r)(r')}$ is $d-2$
  (c) if r'' is a ray with $Z(r'') \supset Z(r) \cap Z(r')$ then either $r'' \simeq r$ or $r'' \simeq r'$;

Combinatorial Characterization of adjacency
(Combinatorial Oracle)

Example: Combinatorial Adjacency Oracle

$J^- = \{D\}$

$J^+ = \{A, B, C\}$

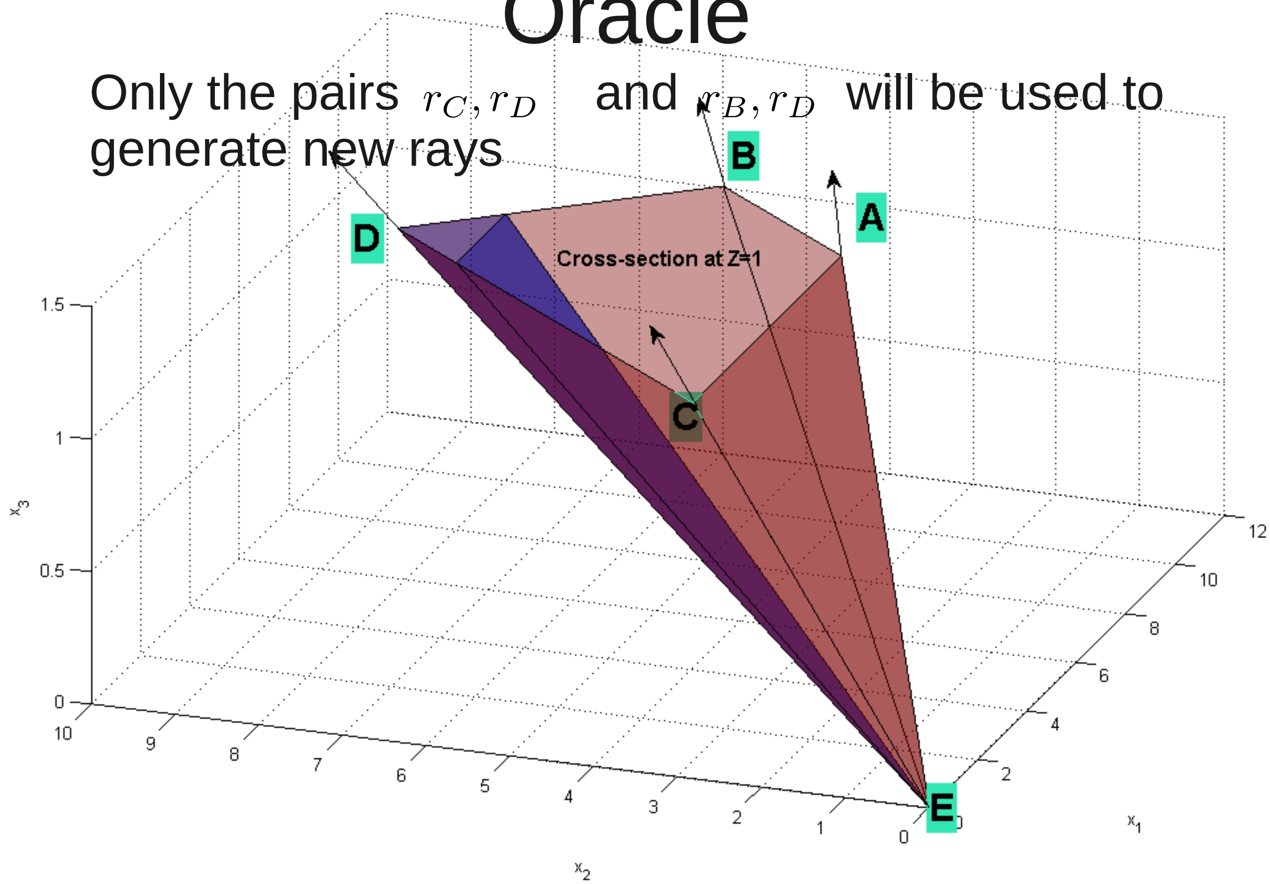


- $Z(r_A) = \{2, 6\}$
- $Z(r_C) = \{4, 6\}$
- $Z(r_B) = \{1, 2\}$
- $Z(r_B)$={1,2}

- $|Z(r_D) \cap Z(r_C)| = |\{4\}| = 1 = d - 2$
- $|Z(r_D \cap Z(r_B))| = |\{1\}| = 1 = d - 2$
- $|Z(r_D \cap Z(r_A))| = |\{\phi\}| = 0 \neq d - 2$

Example: Combinatorial Adjacency Oracle

Only the pairs r_C, r_D and r_B, r_D will be used to generate new rays



Strengthened Main Lemma for DD Method

Let (A_K, R) be a DD pair and let i be the new row index of A not in K. Then the pair (A_{K+i}, R') is a DD pair, where R' is the $d \times |J'|$ matrix with column vectors $r_j (j \in J')$ defined by,

$J' = J^+ \cup J^0 \cup Adj$,

$Adj = \{(j, j') \in J^+ \times J^- : r_j$ and $r_{j'}$ are adjacent in $P(A_K)$ $\}$, and

$r_{jj'} = (A_i r_j) r_{j'} - (A_i r_{j'}) r_j$ for each $(j, j') \in Adj$

Furthermore, if R is minimal generating matrix for $P(A_K)$ then R' is a minimal generating matrix for $P(A_{K+i})$

Procedural Description

procedure DDMethodStandard(A);
begin
    Obtain any initial DD pair (A_K, R)
    while $K \neq \{1, 2, ..., m\}$ **do**
    begin
        Select any index i from $\{1, 2, ..., m\}$;
        Construct a DD pair (A_{K+i}, R') from (A_K, R);
          /*by using Strengthened Main Lemma */
        $R := R'$; $K := K + i$
    end
    Output R
begin

Part 2

- Projection of polyhedral sets
  - Fourier-Motzkin Elimination
  - Block Elimination
  - Convex Hull Method (CHM)
- Redundancy removal
  - Redundancy removal using linear programming

Projection of a Polyhedra

- Consider an $\mathcal{H}$-polyhedron $P = P(A, z) \subseteq R^d$
- We want to project to $\{x \in R^d : x_k = 0\} \equiv R^{d-1}$ along the x_k axis
- We define:

$$\begin{aligned} proj_k(P) &:= \{x - x_k e_k : x \in P\} && (1) \\ &= \{x \in R^d : x_k = 0, \exists y \in R : x + y e_k \in P\} && (2) \end{aligned}$$

- This is projection of P in the direction of e_k
- The set $proj_k(P)$ is contained in the hyperplane $H_k = \{x \in R^d : x_k = 0\}$

Fourier-Motzkin Elimination

- Named after Joseph Fourier and Theodore Motzkin

How it works?

- We start with an $\mathcal{H}$-polyhedron $P = P(A, z) \subseteq R^d$
- Suppose we want to eliminate the variable x_k
- Consider coefficients of x_k in our system of inequalities, and assume that $a_{ik} > 0$ and $a_{jk} < 0$
- Then the respective inequalities can be written as,
  $a_i x \leq z_i \rightarrow a_{ik}x_k \leq a_{ik}x_k - a_i x + z_i$
  and
  $a_j x \leq z_j \rightarrow (-a_{jk}x_k) \geq -a_{jk}x_k + a_j x - z_j$

How it works? Contd...

- Multiply these equations by $-a_{jk}$ and a_{ik} respectively
  $-a_{jk}a_{ik}x_k \leq -a_{jk}a_{ik}x_k - a_i x - a_{jk}z_i$
  and
  $-a_{jk}a_{ik}x_k \geq -a_{jk}a_{ik}x_k + a_j a_{ik}x - a_{ik}z_j$

- These equations form upper bound and lower bound respectively on $-a_{jk}a_{ik}x_k$

- Combining the two we get,
  $a_{ik}a_j + (-a_{jk}a_j)x \leq a_{ik}z_j + (-a_{jk})z_j$

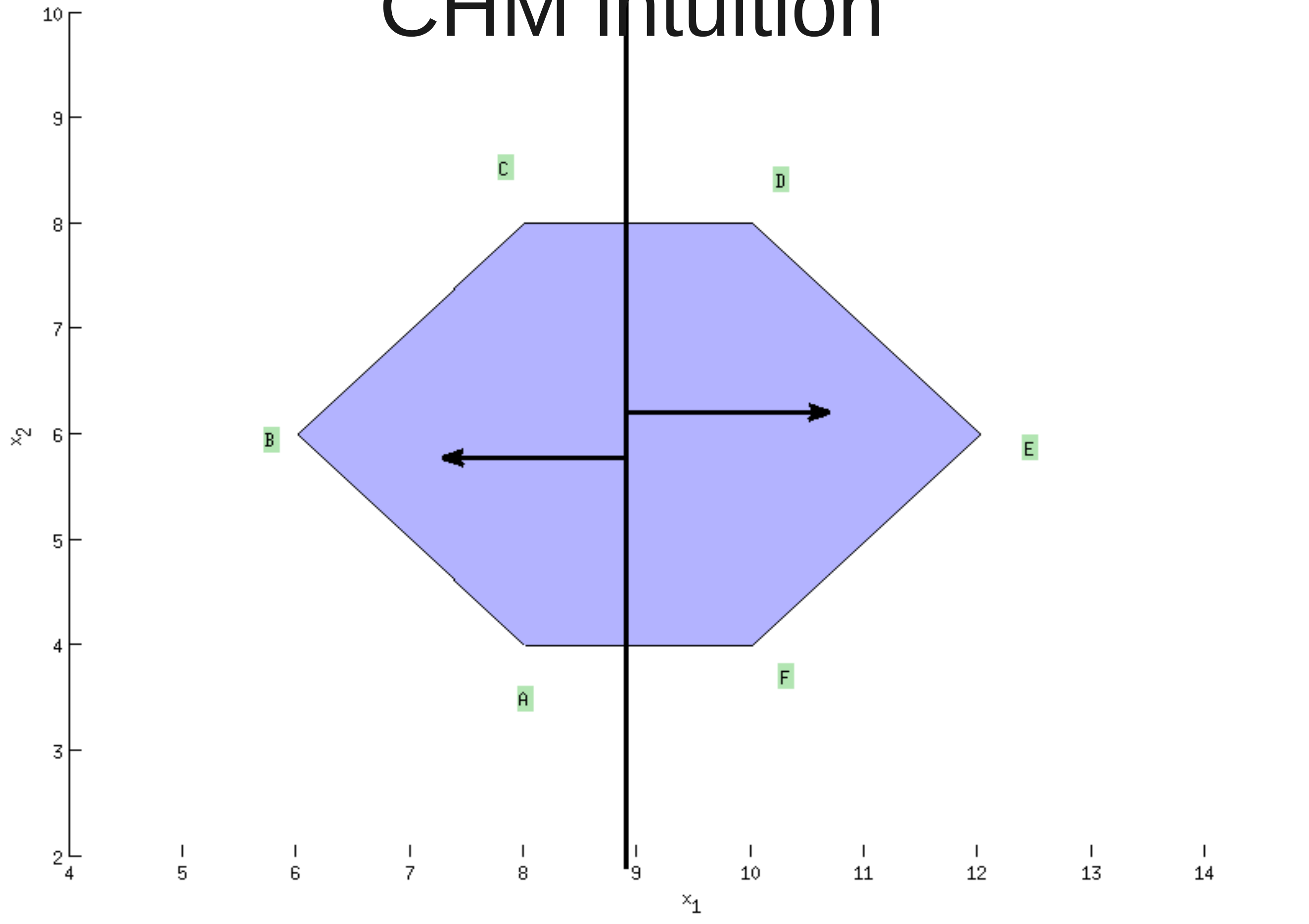
Efficiency of FM algorithm

- The number of inequalities goes beyond tractable limits within few elimination steps
- IF a has m rows, then $A^{\backslash k}$ may have as many as $\lfloor \frac{m^2}{4} \rfloor$ rows
- FM-elimination creates $O(m^2)$ new inequalities
- Useful only as a simple and elegant method that is easy to understand
- There have been efforts to introduce heuristics, as discussed in the paper by Lassez et al.

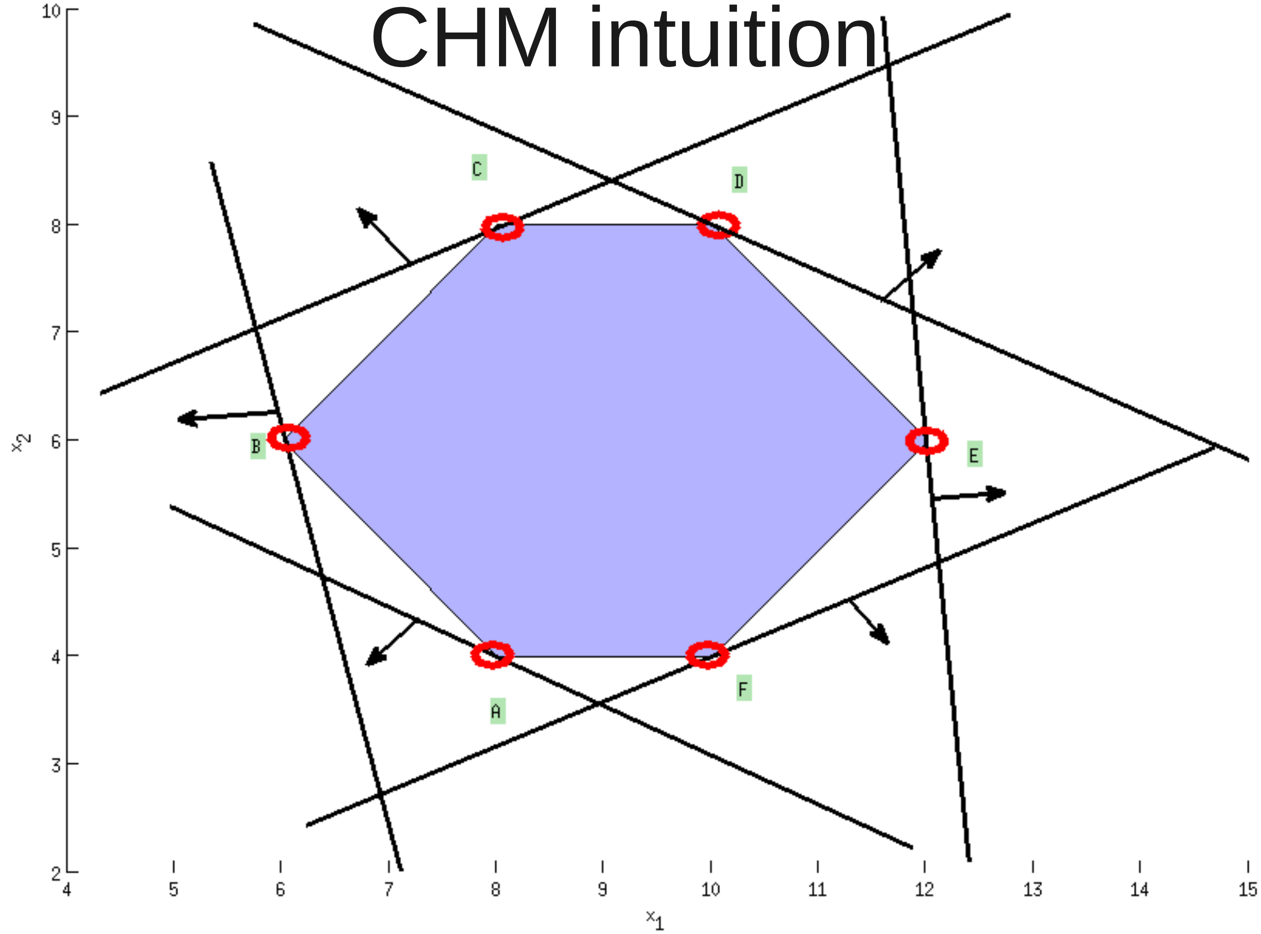
Convex Hull Method(CHM)

- First appears in “C. Lassez and J.-L. Lassez, Quantifier elimination for conjunctions of linear constraints via aconvex hull algorithm, *IBM Research Report*, T.J. Watson Research Center (1991)”
- Found to be better than most other existing algorithms when dimension of projection is small
- Cited by Weidong Xu, Jia Wang, Jun Sun in their ISIT 2008 paper “A Projection Method for Derivation of Non-Shannon-Type Information Inequalities”

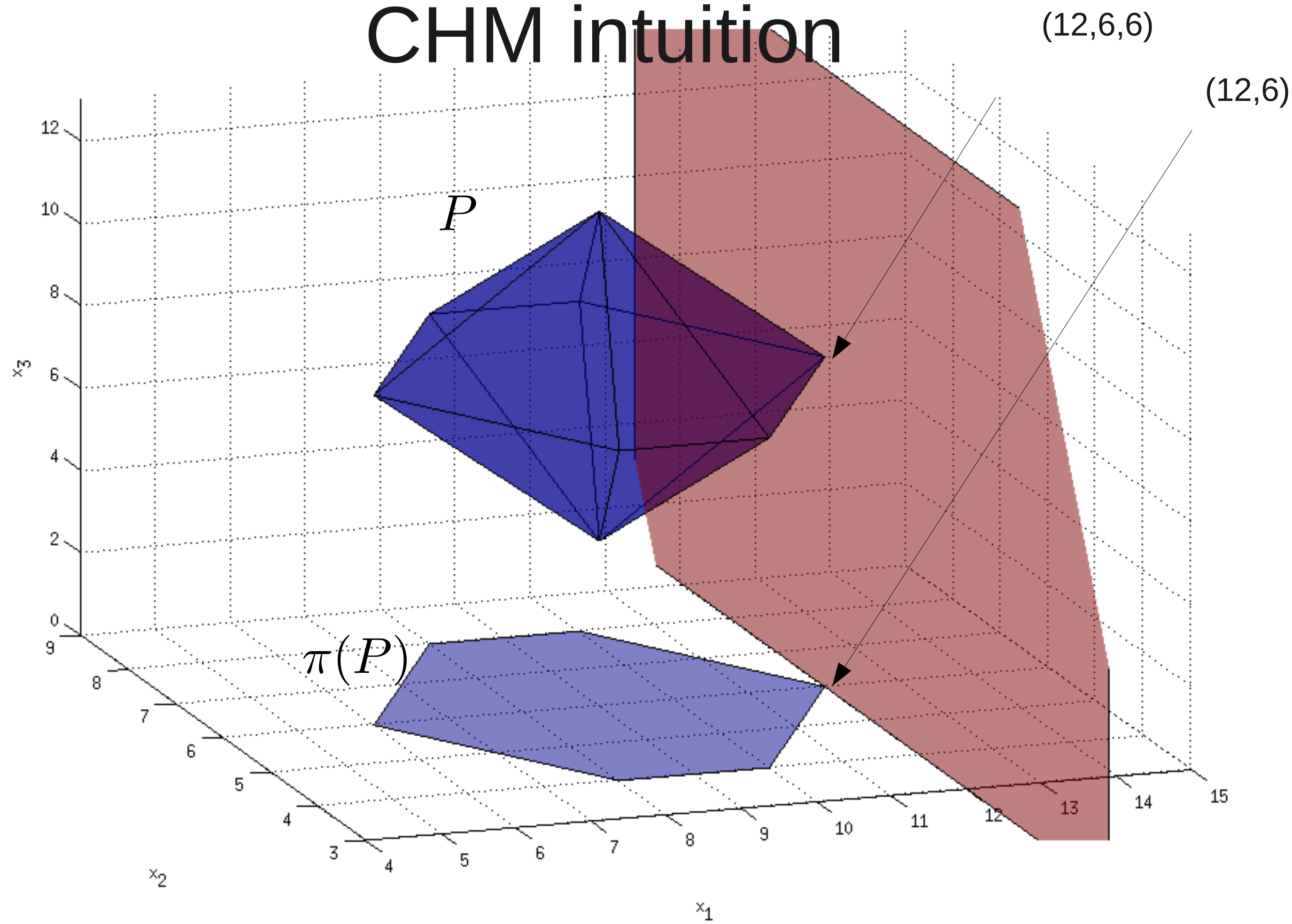
CHM intuition



CHM intuition

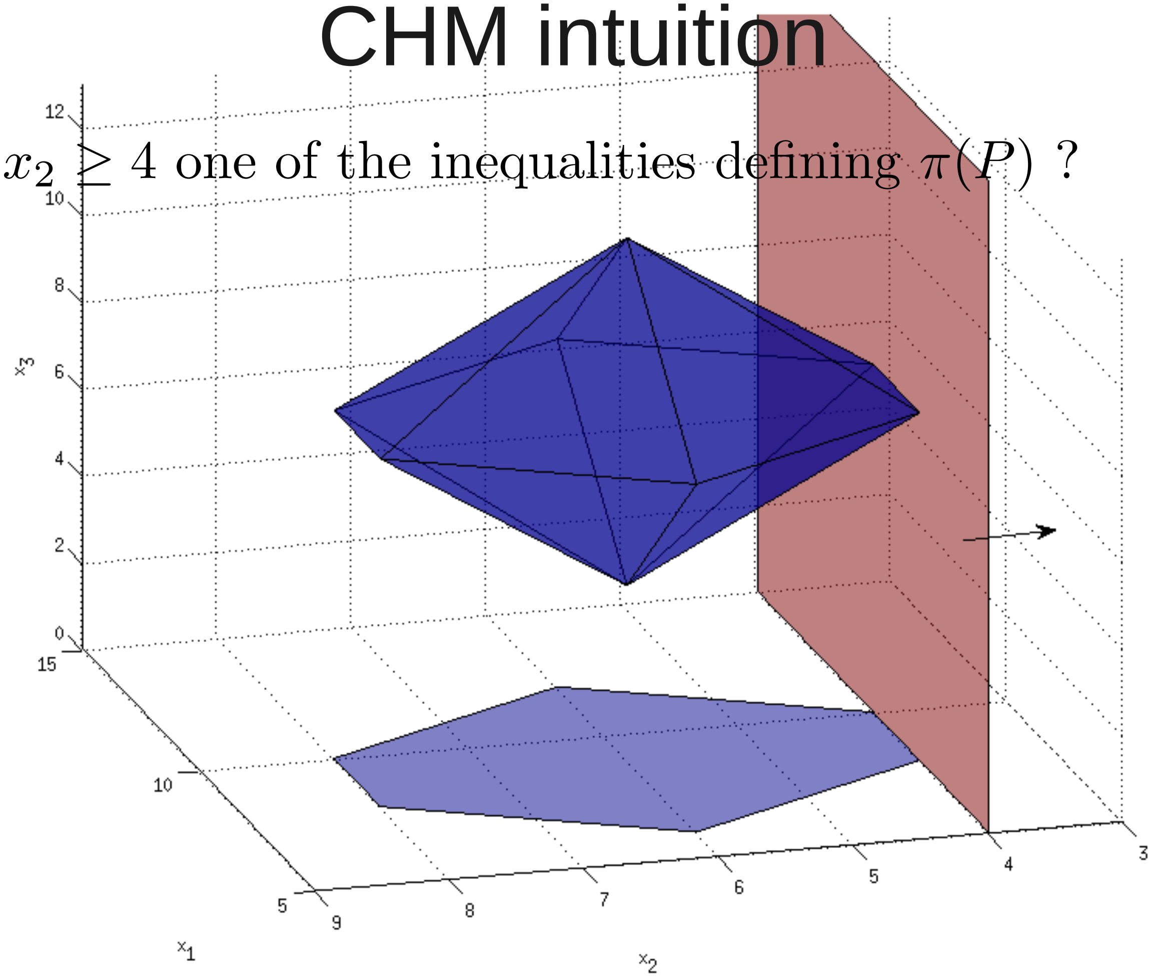


CHM intuition



CHM intuition

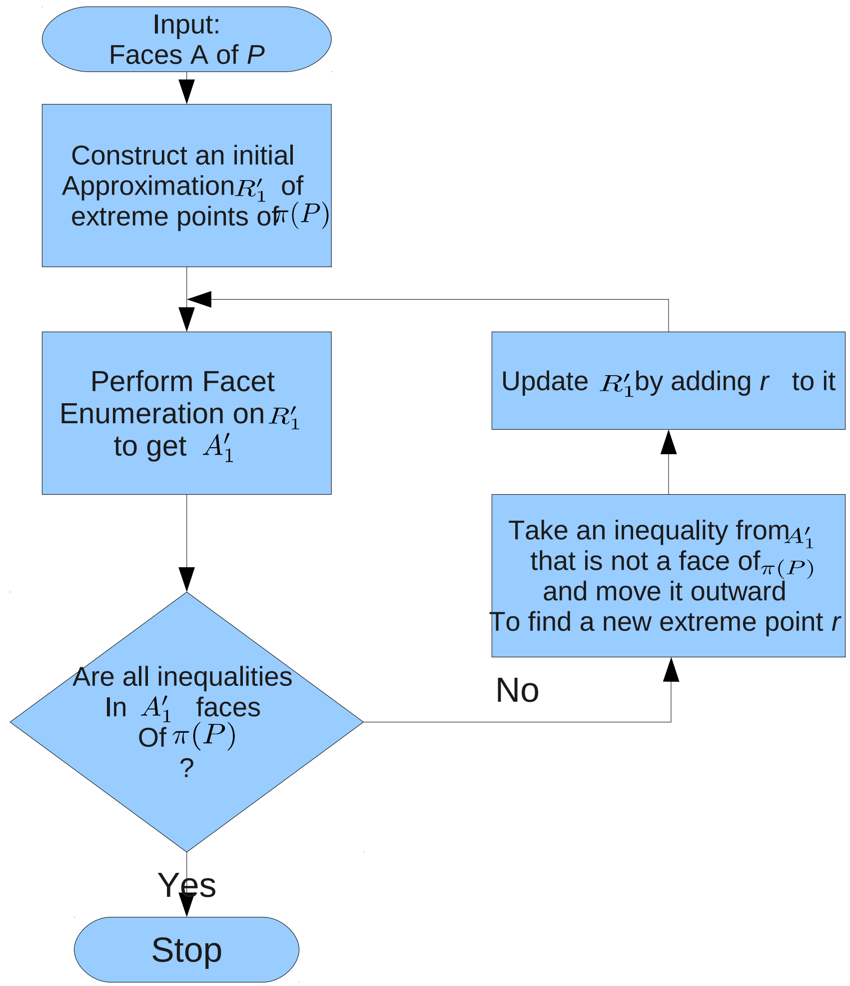
Is $x_2 \geq 4$ one of the inequalities defining $\pi(P)$?



Moral of the story

- We can make decisions about $\pi(P)$ without actually having its H-representation.
- It suffices to have P, the original polyhedron
- We run linear programs on P to make these decisions

Convex Hull Method: Flowchart

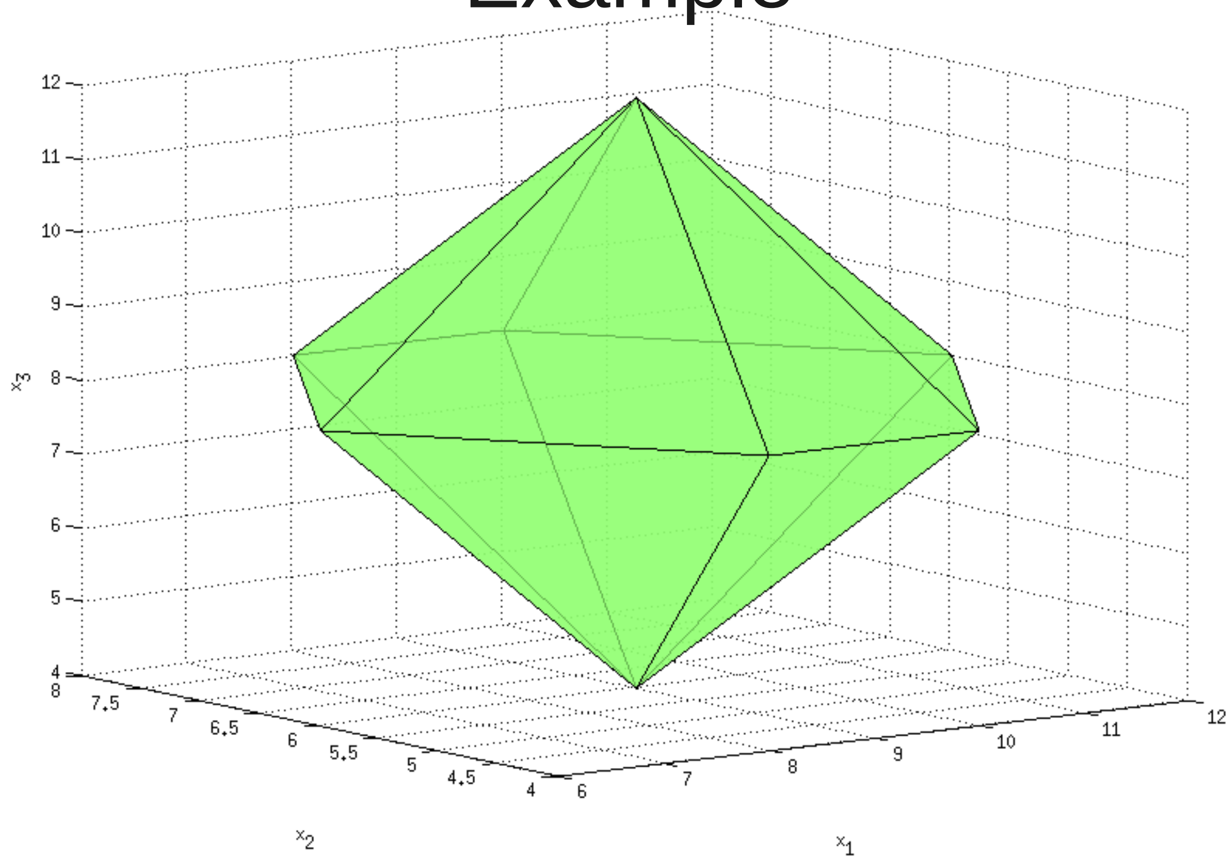


Example

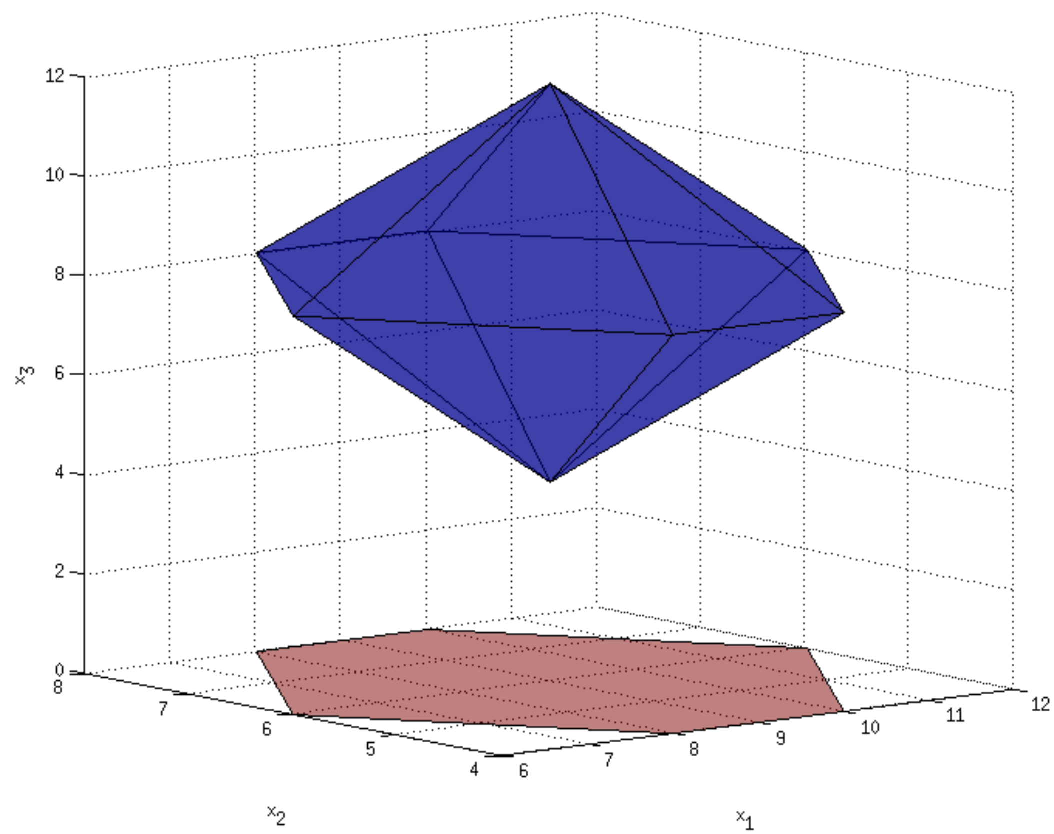
Project P onto x_1, x_2
3D$\rightarrow 2D$

$$-16 + 0x_1 + 2x_2 + 1x_3 \geq 0 \quad (1)$$
$$-72 + 4x_1 + 4x_2 + 3x_3 \geq 0 \quad (2)$$
$$0 + 0x_1 + 2x_2 - 1x_3 \geq 0 \quad (3)$$
$$-24 + 4x_1 + 4x_2 - 3x_3 \geq 0 \quad (4)$$
$$0 - 4x_1 + 4x_+ 3x_3 \geq 0 \quad (5)$$
$$48 - 4x_1 + 4x_2 - 3x_3 \geq 0 \quad (6)$$
$$48 - 4x_1 - 4x_2 + 3x_3 \geq 0 \quad (7)$$
$$8 + 0x_1 - 2x_2 + 1x_3 \geq 0 \quad (8)$$
$$-24 + 4x_1 - 4x_2 + 3x_3 \geq 0 \quad (9)$$
$$24 + 0x_1 - 2x_2 - 1x_3 \geq 0 \quad (10)$$
$$24 + 4x_1 - 4x_2 - 3x_3 \geq 0 \quad (11)$$
$$96 - 4x_1 - 4x_2 - 3x_3 \geq 0 \quad (12)$$

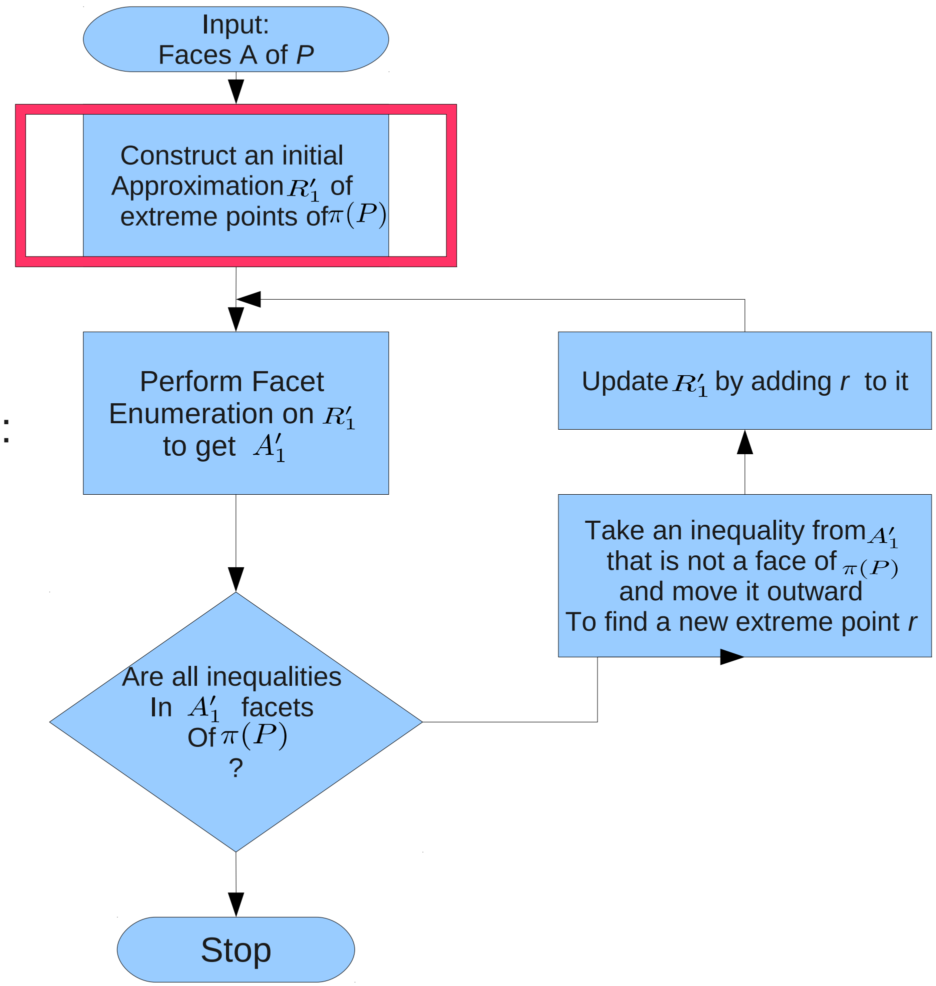
Example



Example



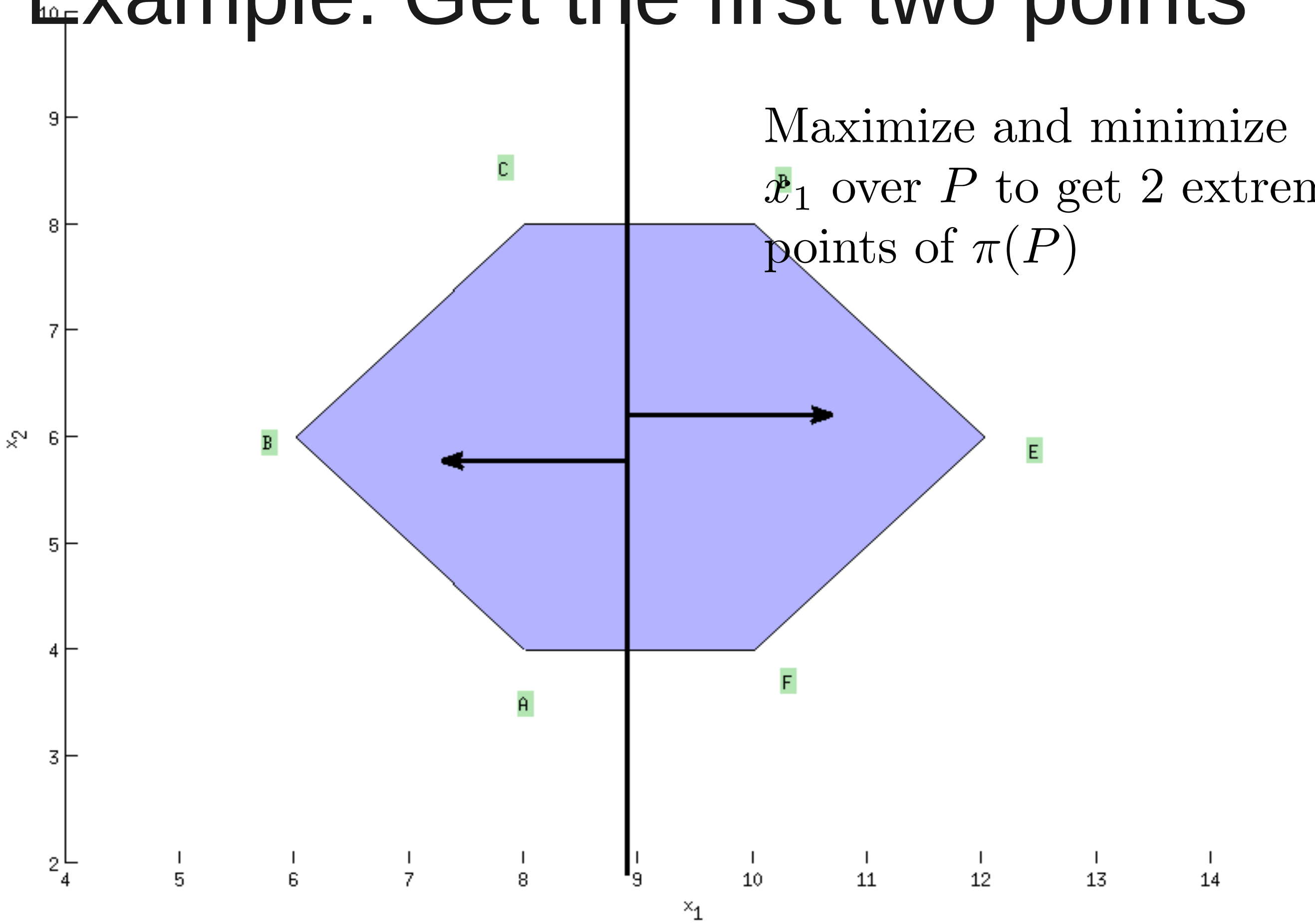
Convex Hull Method: Flowchart



How to get the extreme points of initial approximation?

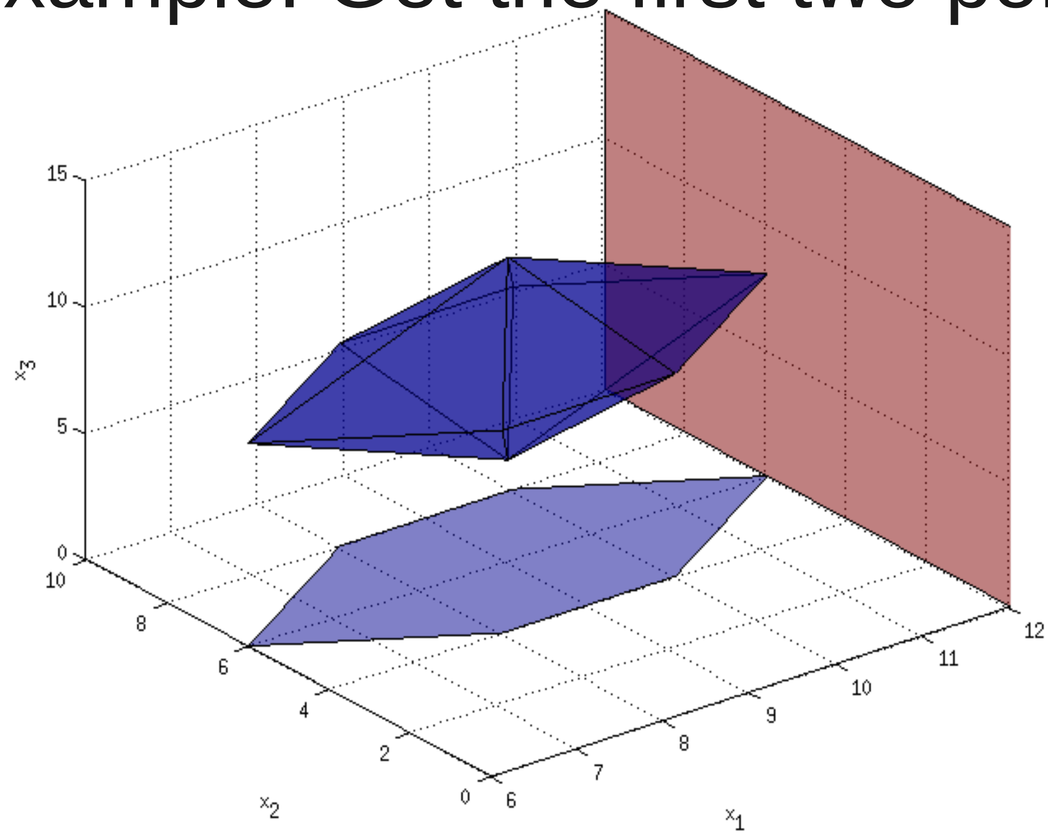
- We need $d + 1$ points to have full dimentional convex hull
- Get first two points by maximizing and minimizing x_1
- Get rest of the points by running linear programs on initial set of constraints

Example: Get the first two points

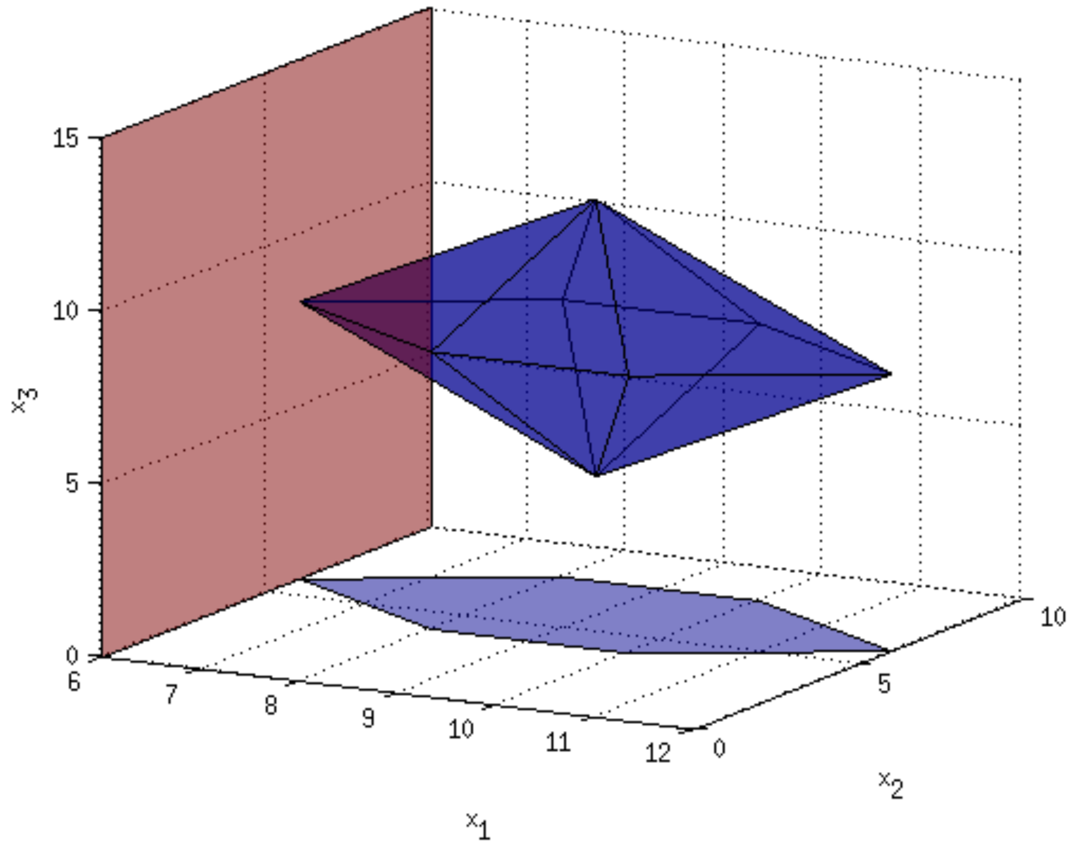


Maximize and minimize x_1 over P to get 2 extreme points of $\pi(P)$

Example: Get the first two points

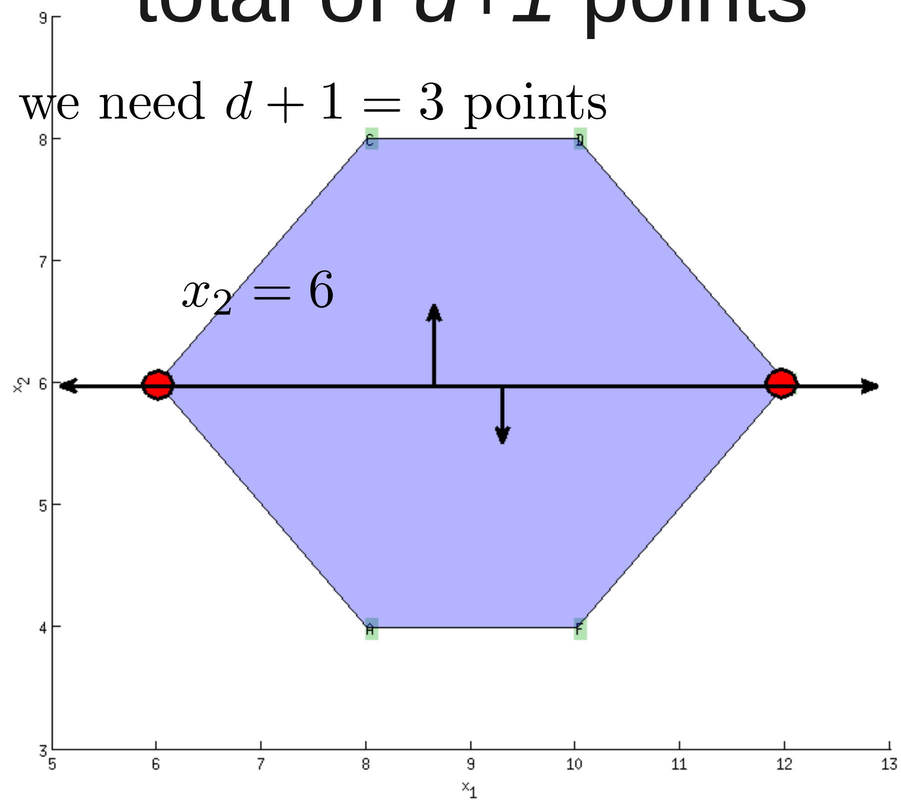


Example: Get the first two points

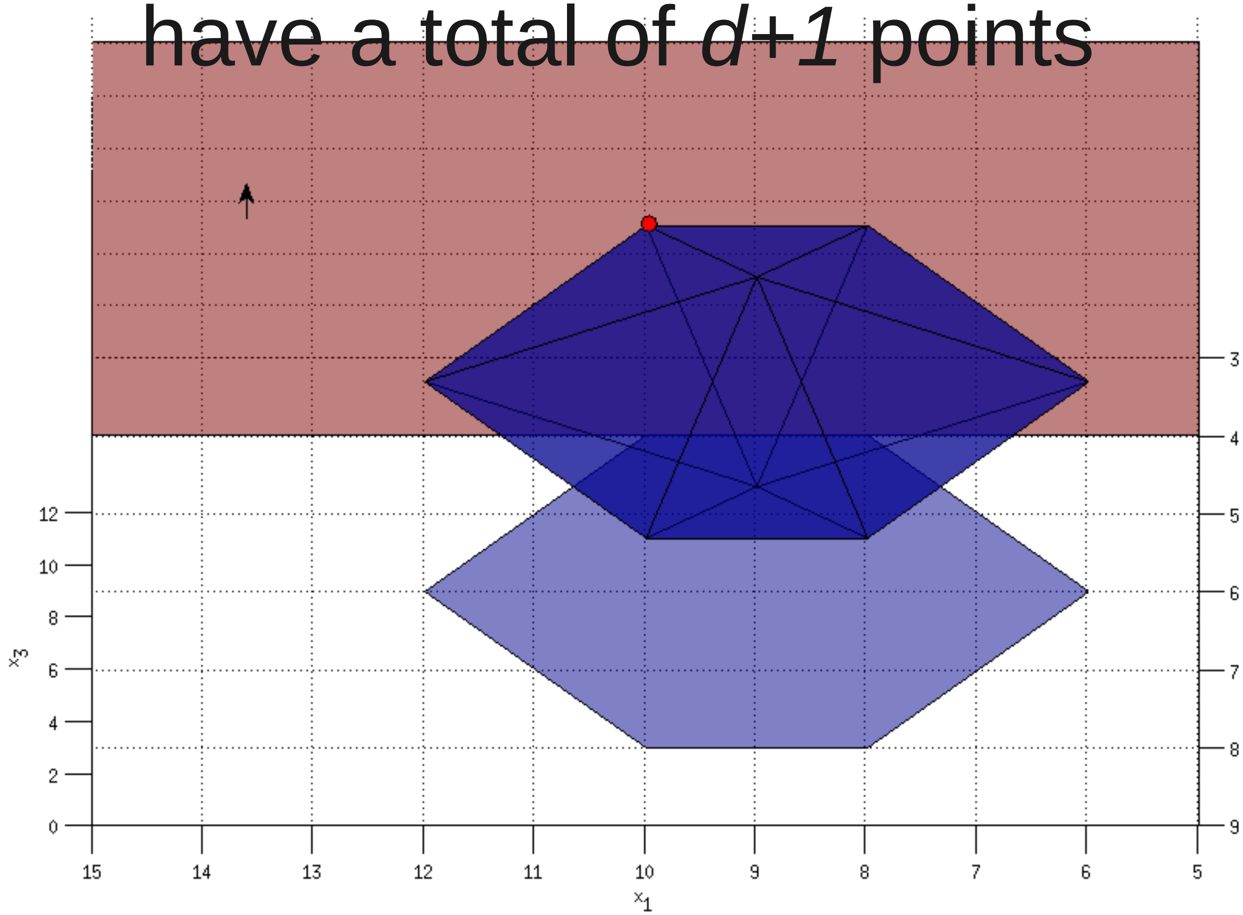


Get rest of the points so you have a total of *d+1* points

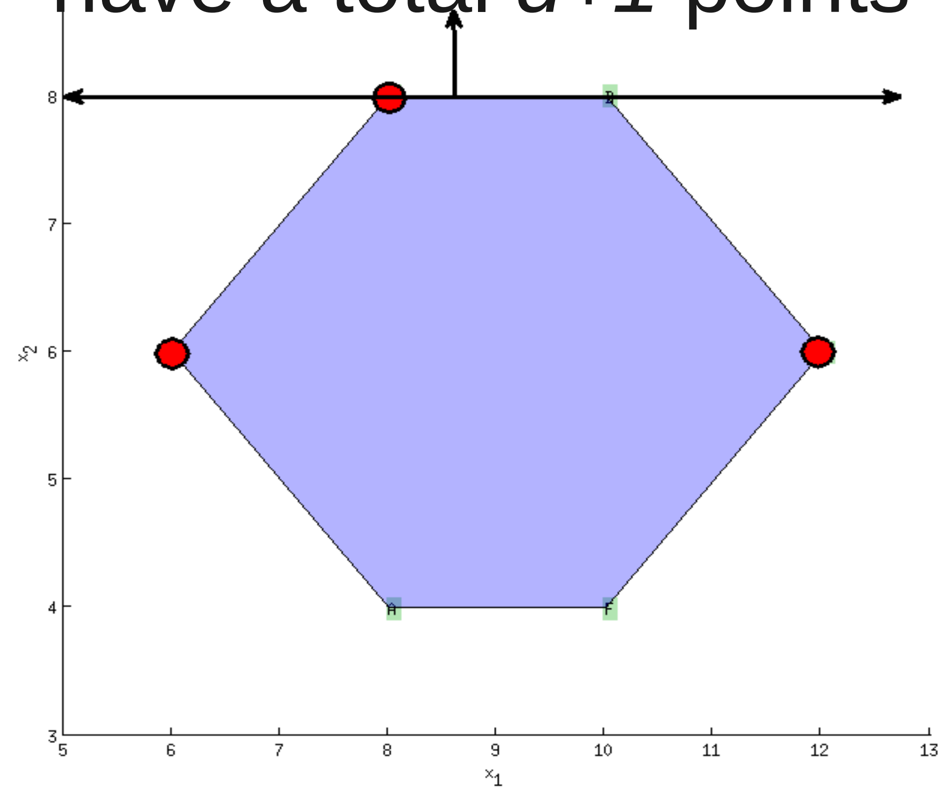
$d = 2$ so we need $d + 1 = 3$ points



Get the rest of the points so you have a total of *d+1* points



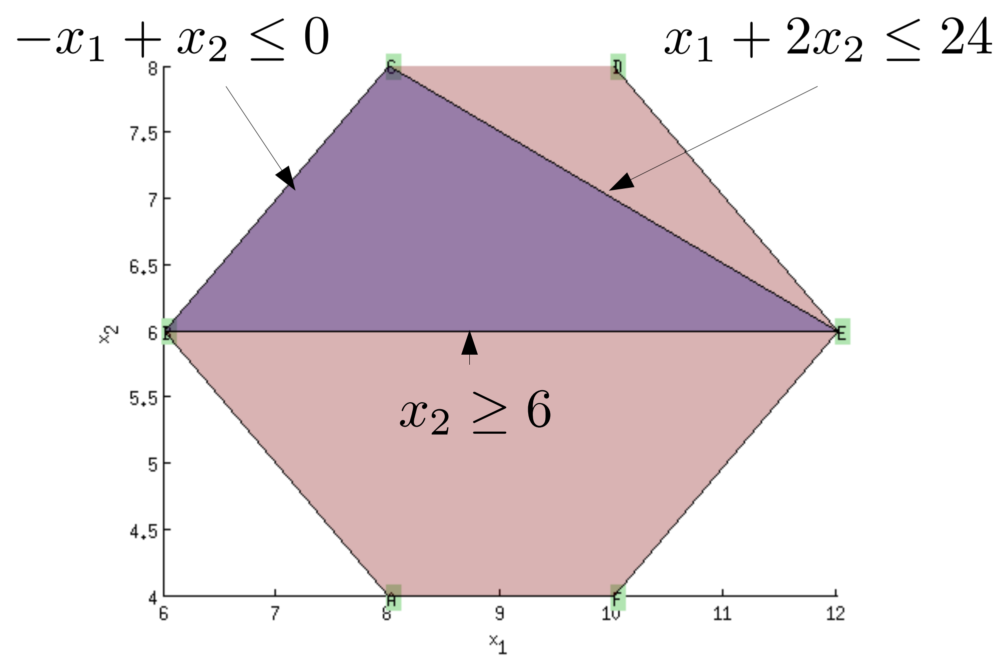
Get the rest of the points so you have a total *d+1* points



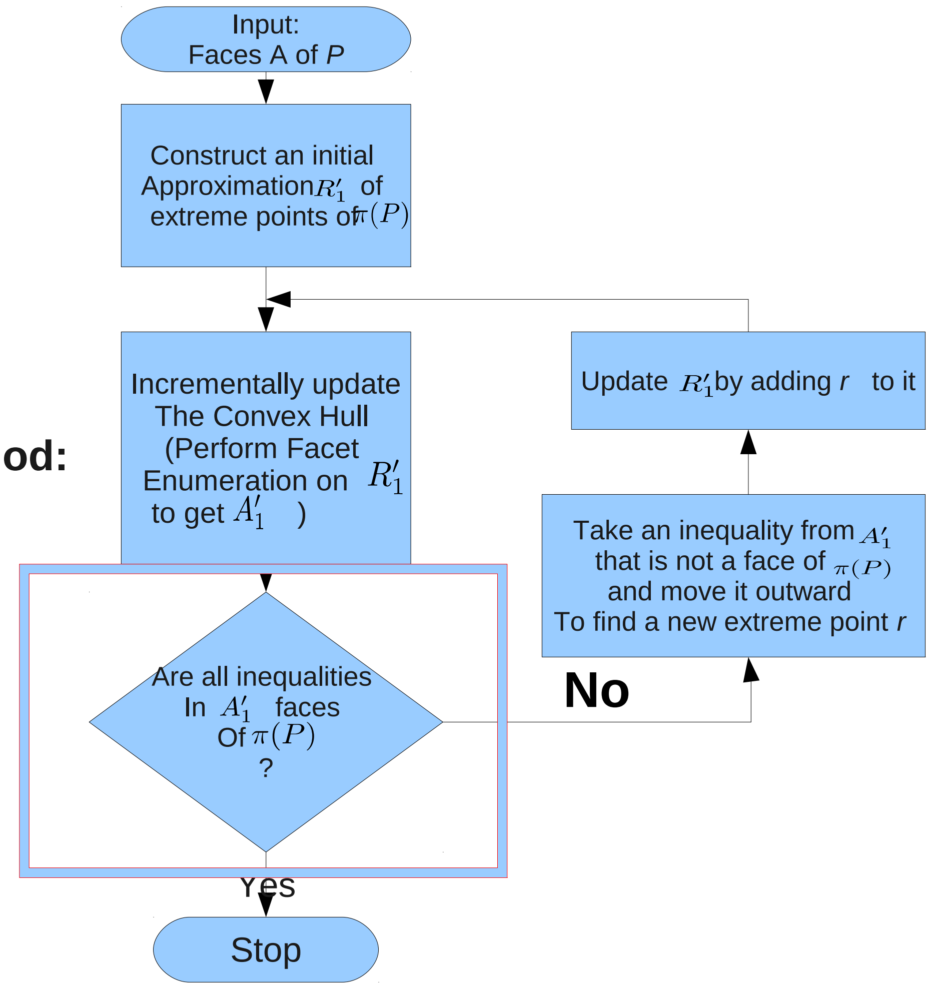
How to get the initial facets of the projection?

procedure *initial_hull(E)*;
begin
  Let CH=ϕ
  for each $p \in E$ **do**
    Compute $\sum_{j=1}^{d} \alpha_j x_j = \alpha$ the equation of the hyperplane defined by $E - \{p\}$
    Let $h = \sum_{j=1}^{d} \alpha_j x_j$
    If $h(p) \geq \alpha$
      then $CH = CH \cup \{-h \leq -\alpha\}$
      else $CH = CH \cup \{h \leq \alpha\}$
    end
end

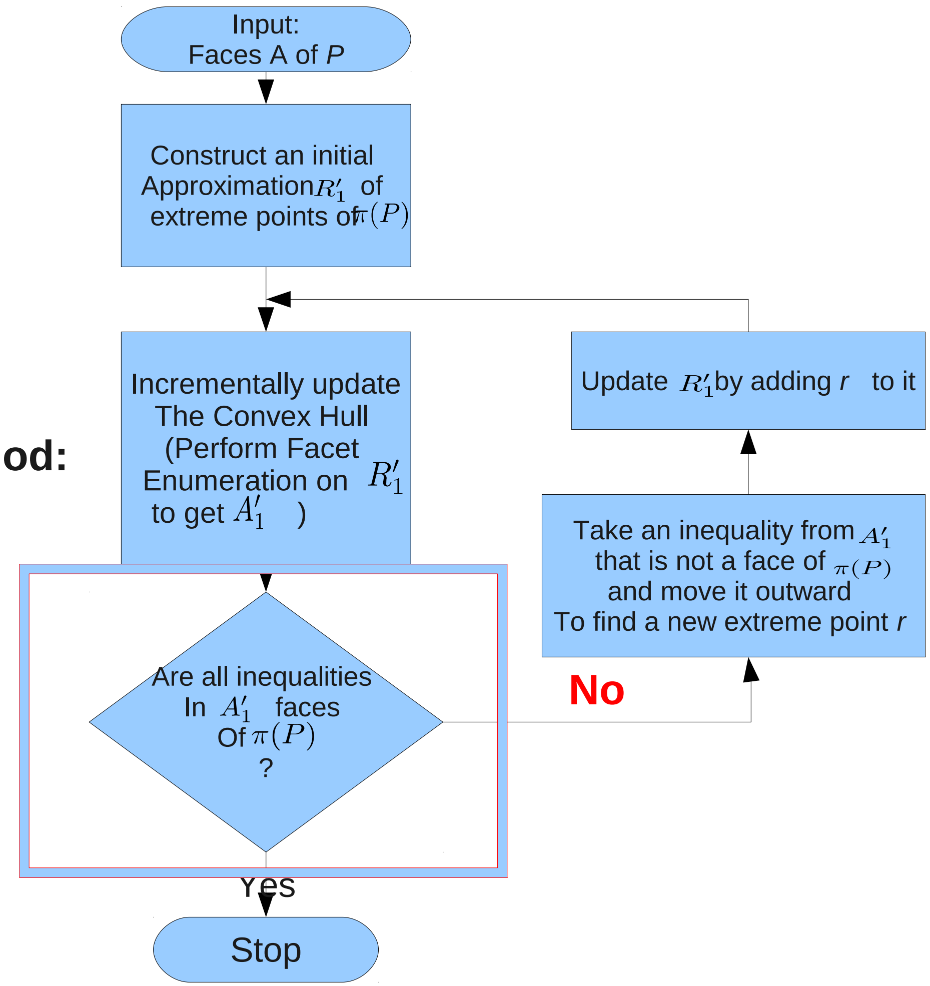
How to get the initial facets of the projection?



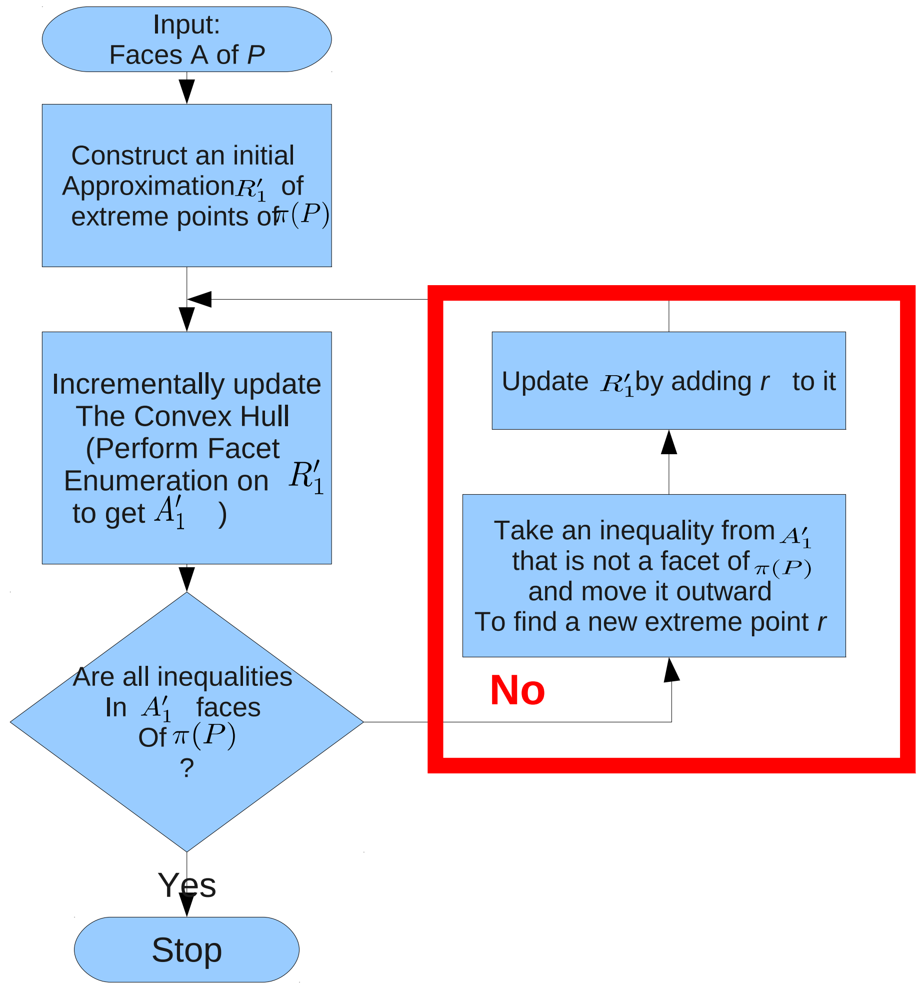
Convex Hull Method: Flowchart



Convex Hull Method: Flowchart



Convex Hull Method: Flowchart



Incremental refinement

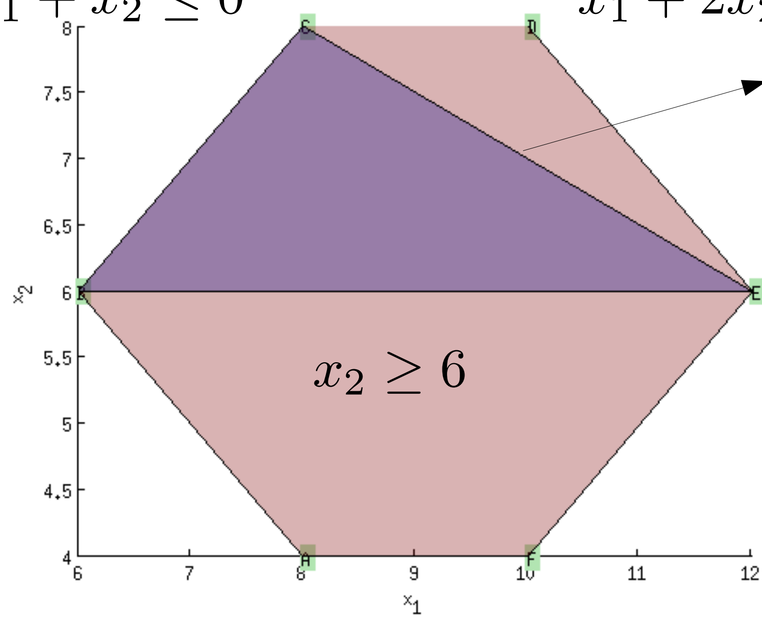
- Find a facet in current approximation of that is not actually the facet of $\pi(P)$
- How to do that?

given $\{h_i \leq \alpha_i\}$,
  Maximize h_i over P
  If $Max(h_i) = \alpha_i$
  Then Label $\{h_i \leq \alpha_i\}$ as *terminal*
  else Move it outward to find a new extreme point

Incremental refinement

$-x_1 + x_2 \leq 0$

$x_1 + 2x_2 \leq 24$

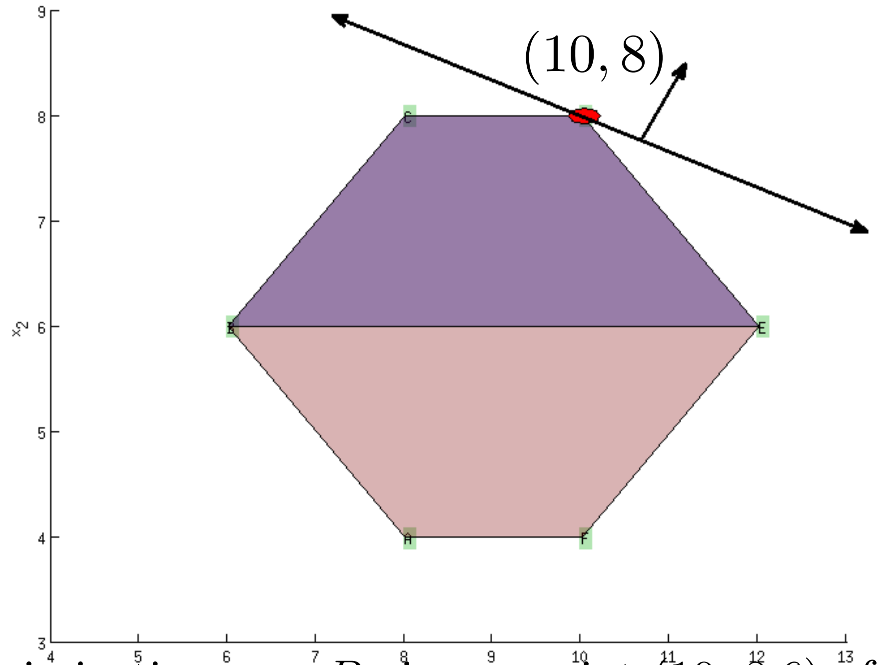


This inequality is Not a facet of $\pi(P)$

max $x_1 + 2x_2$ over P

$x_2 \geq 6$

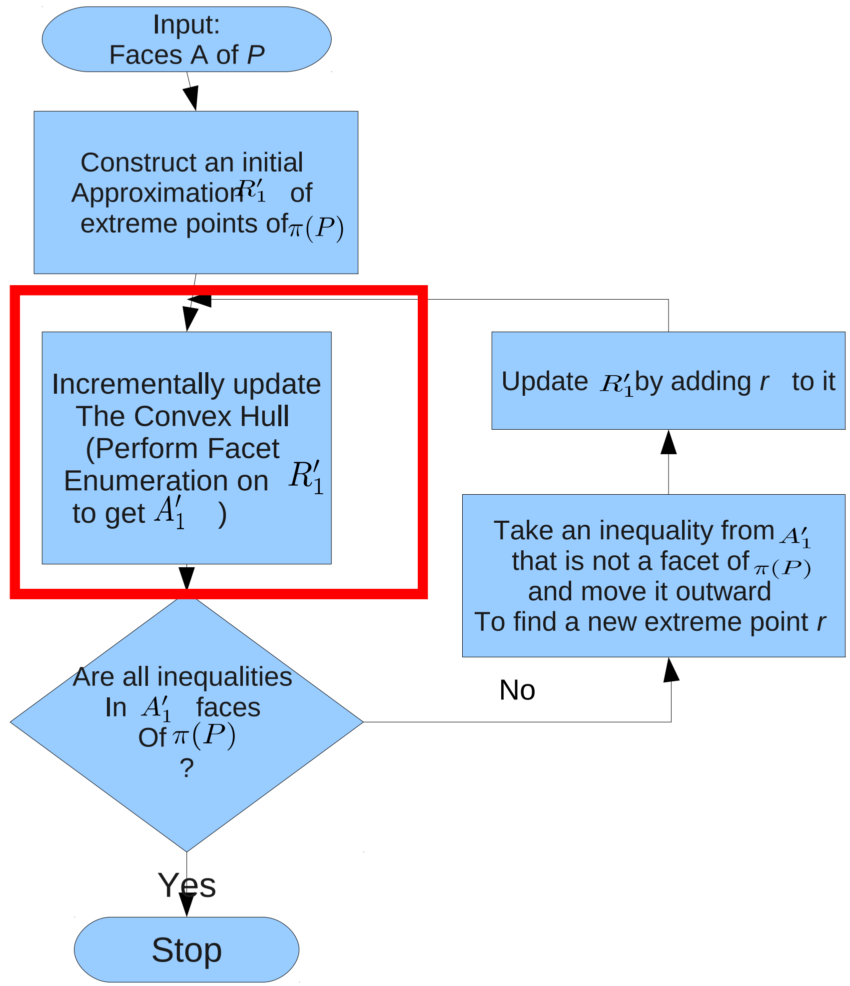
Incremental refinement



Maximization over P gives a point (10, 8,6) of P

This point corresponds to the point (10,8) of $\pi(P)$

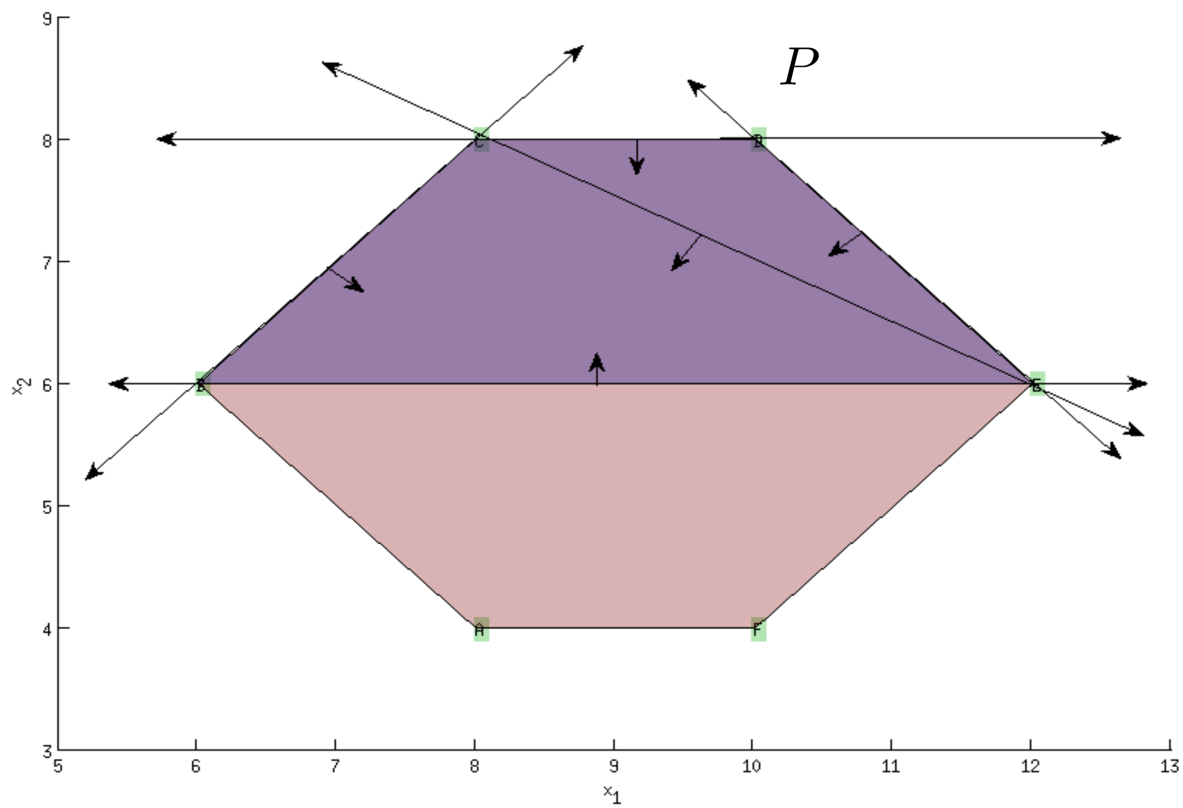
Convex Hull Method: Flowchart



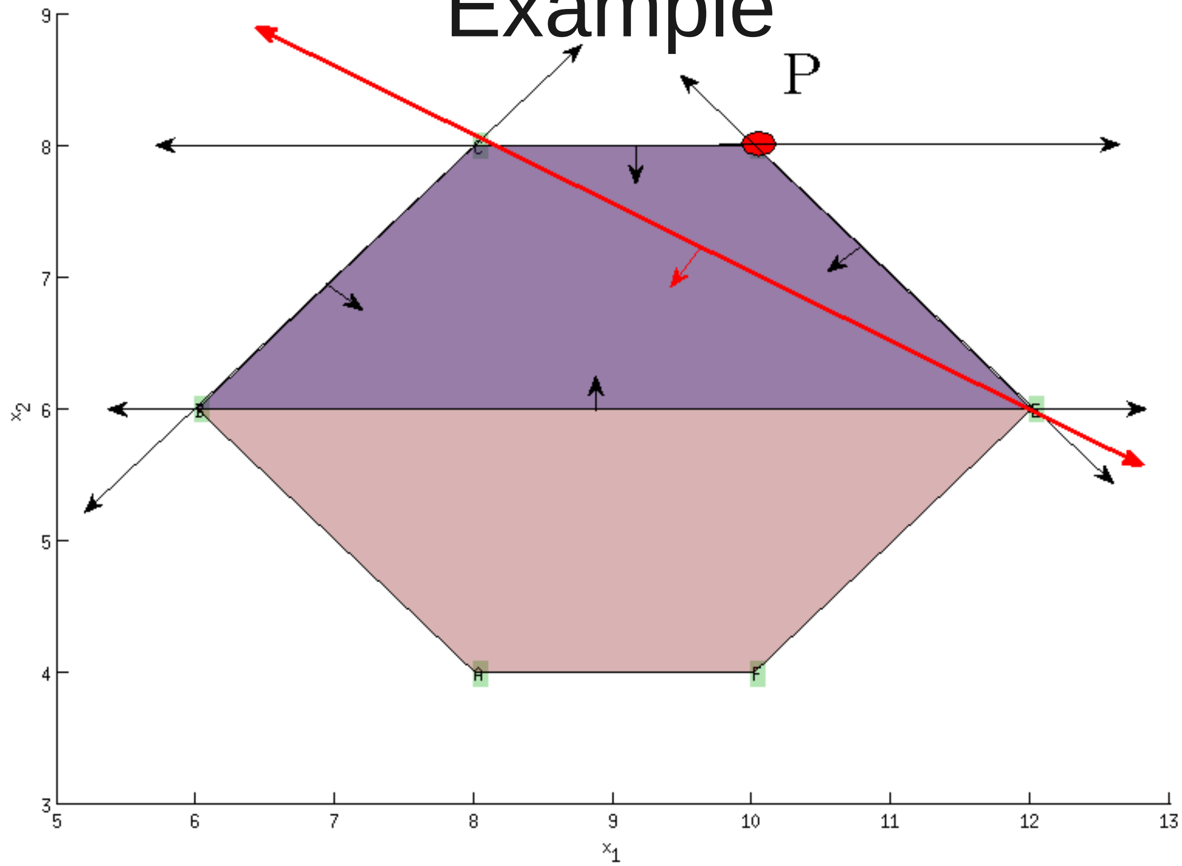
How to update the H-representation to accommodate new point ?

procedure *update_convex_hull(new_pt)*;
begin
  for each $C \in CH$ s.t. $p \notin C$ **do**
    Generate all subsets SE of $d-1$ extreme points in C
    for each SE **do**
      If $SE \cup \{p\}$ determines a unique hyperplane $\sum_i \alpha_i x_i = 0$
        then let $h = \sum_i \alpha_i x_i$
      If $\forall q \in E, h(q) \leq \alpha_0$
        then $C = \sum_i \alpha_i x_i \leq \alpha_0$
      else if $\forall q \in E, h(q) \geq \alpha_0$
        then $C = -\sum_i \alpha_i x_i \leq -\alpha_0$
      else $C = \phi$
      Let $CH = CH \cup \{C\}$
    end
  $\forall C \in CH$ **If** $p \notin C$ **then** $CH = CH - \{C\}$
  end
end

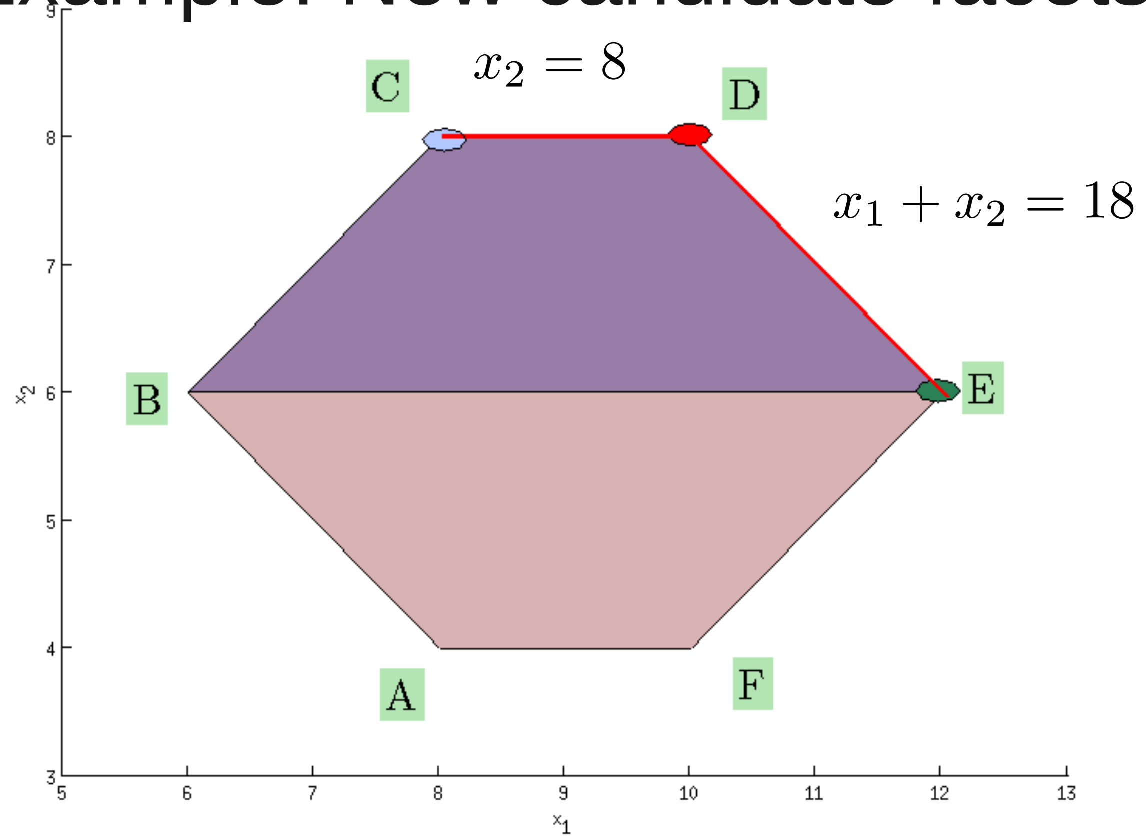
Example



Example



Example: New candidate facets

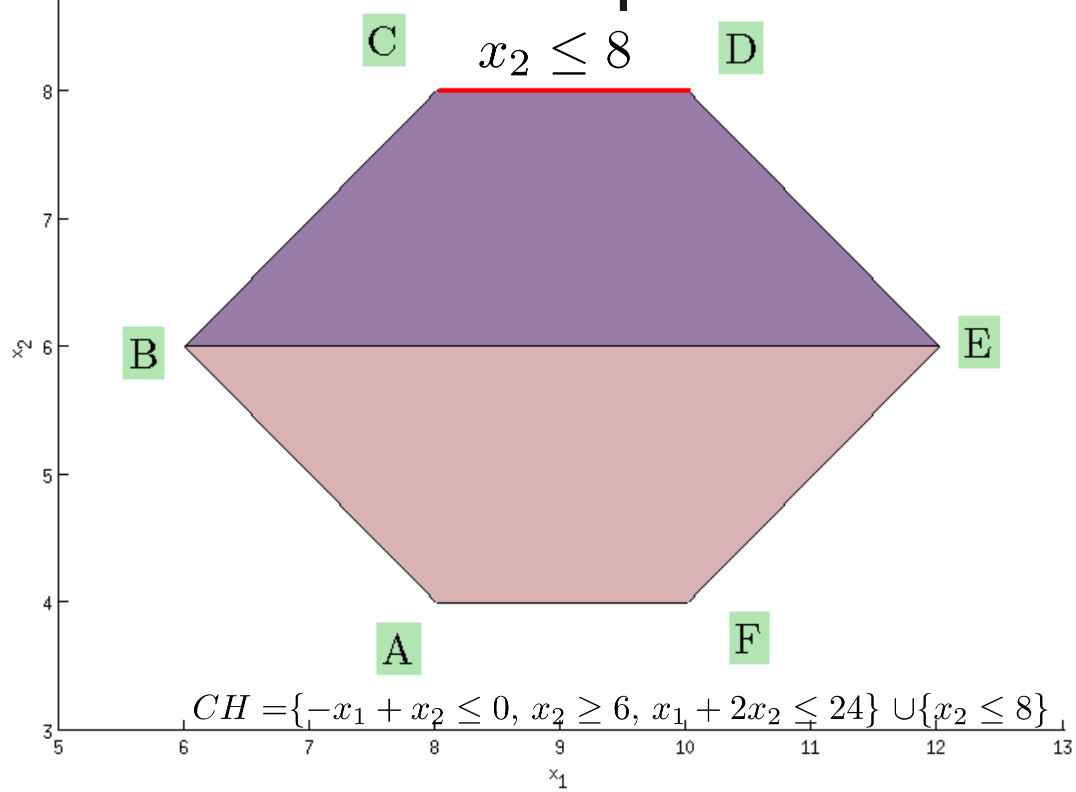


How to update the H-representation to accommodate new point ?

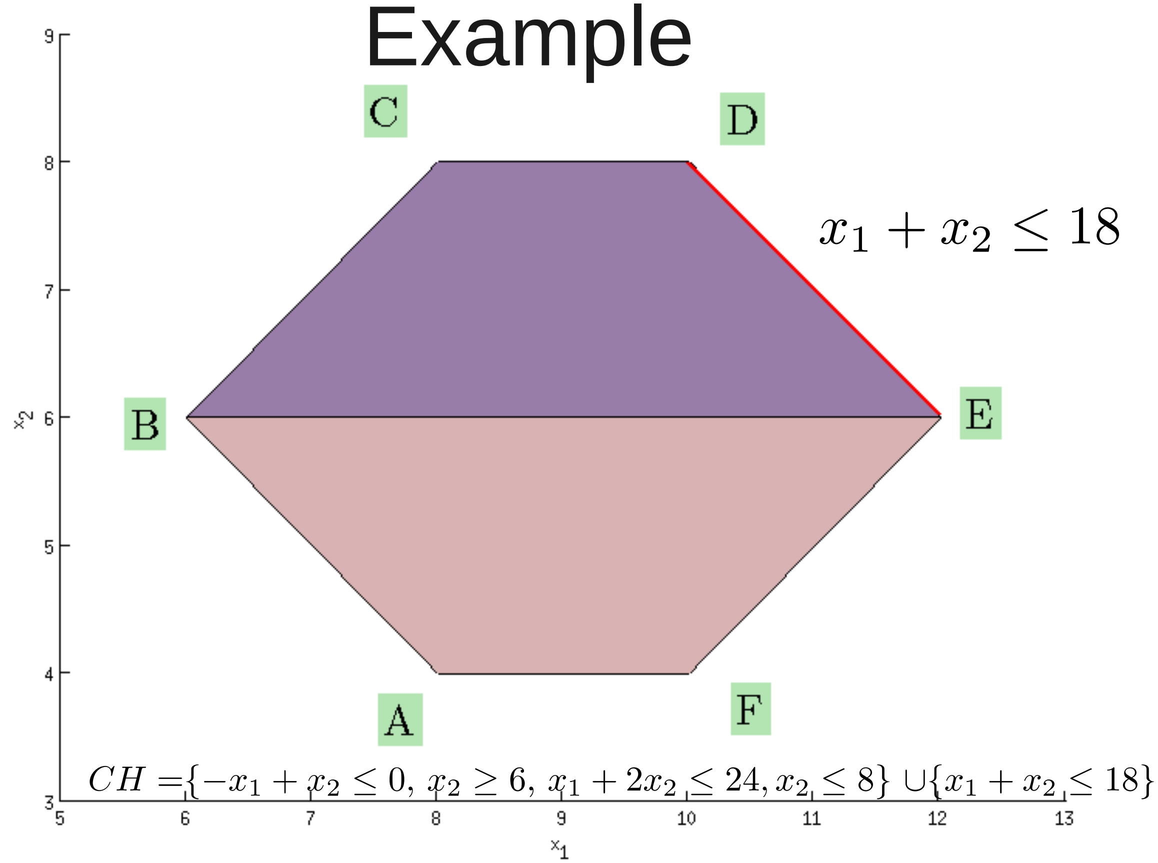
procedure *update_convex_hull(new_pt)*;
begin
    for each $C \in CH$ s.t. $p \notin C$ **do**
        Generate all subsets SE of $d-1$ extreme points in C
        for each SE **do**
            If $SE \cup \{p\}$ determines a unique hyperplane $\sum_i \alpha_i x_i = 0$
              then let $h = \sum_i \alpha_i x_i$
            If $\forall q \in E, h(q) \leq \alpha_0$
              then $C = \sum_i \alpha_i x_i \leq \alpha_0$
            else if $\forall q \in E, h(q) \geq \alpha_0$
              then $C = -\sum_i \alpha_i x_i \leq -\alpha_0$
            else $C = \phi$
            Let $CH = CH \cup \{C\}$
        end
    $\forall C \in CH$ **If** $p \notin C$ **then** $CH = CH - \{C\}$
    end
end

Determine validity of
The candidate facet
And also its sign

Example


C
$x_2 \leq 8$
D
B
E
A
F
$CH = \{-x_1 + x_2 \leq 0,\ x_2 \geq 6,\ x_1 + 2x_2 \leq 24\} \cup \{x_2 \leq 8\}$
x2
x1
3
4
5
6
7
8
9
10
11
12
13

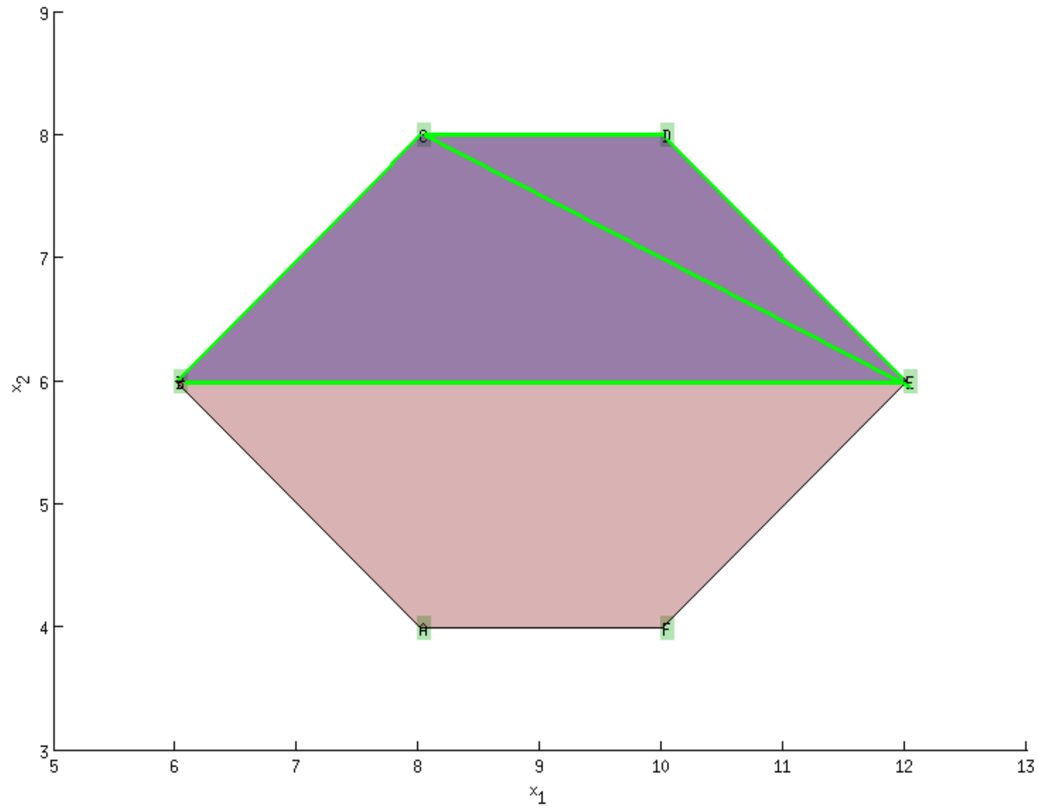
Example


C
D
$x_1 + x_2 \leq 18$
B
E
A
F
$CH = \{-x_1 + x_2 \leq 0,\ x_2 \geq 6,\ x_1 + 2x_2 \leq 24, x_2 \leq 8\} \cup \{x_1 + x_2 \leq 18\}$
x_1
x_2

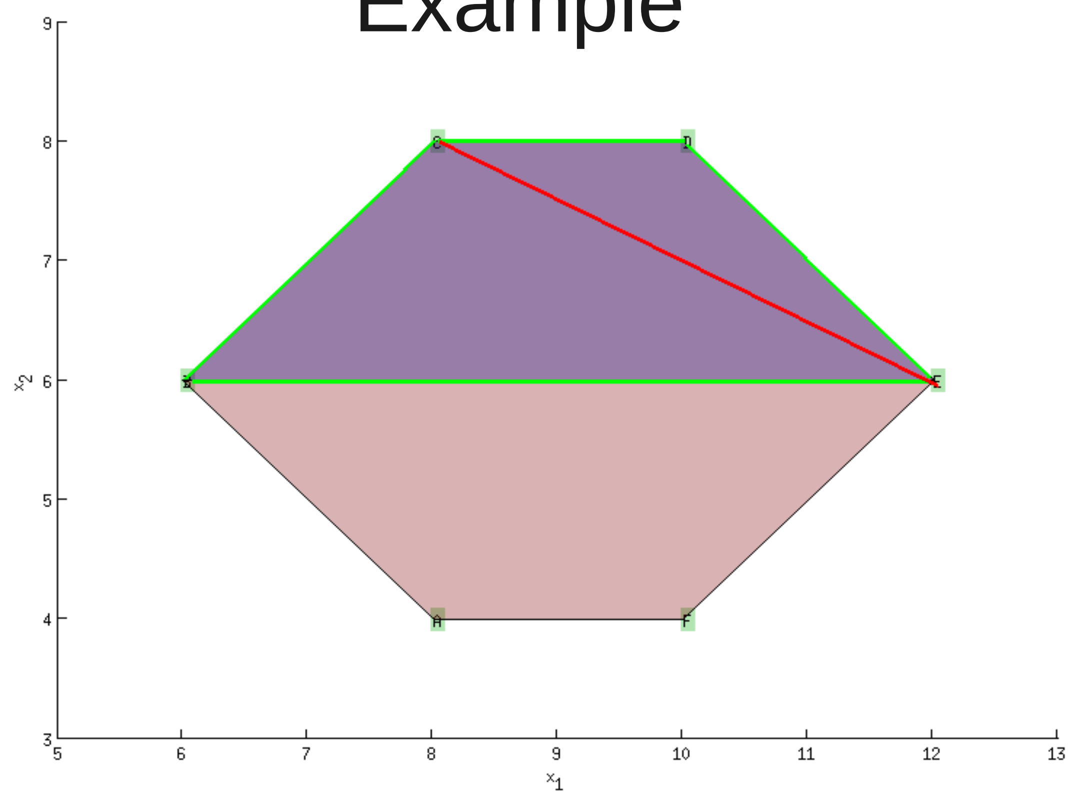
How to update the H-representation to accommodate new point ?

```
procedure update_convex_hull(new_pt);
begin
    for each C ∈ CH s.t. p ∉ C do
        Generate all subsets SE of d − 1 extreme points in C
        for each SE do
            If SE ∪ {p} determines a unique hyperplane ∑_i α_i x_i = 0
               then let h = ∑_i α_i x_i
            If ∀q ∈ E, h(q) ≤ α_0
               then C = ∑_i α_i x_i ≤ α_0
            else if  ∀q ∈ E, h(q) ≥ α_0
               then C = −∑_i α_i x_i ≤ −α_0
            else C = φ
            Let CH = CH ∪ {C}
        end
    ∀C ∈ CH If p ∉ C then CH = CH − {C}
    end
end
```

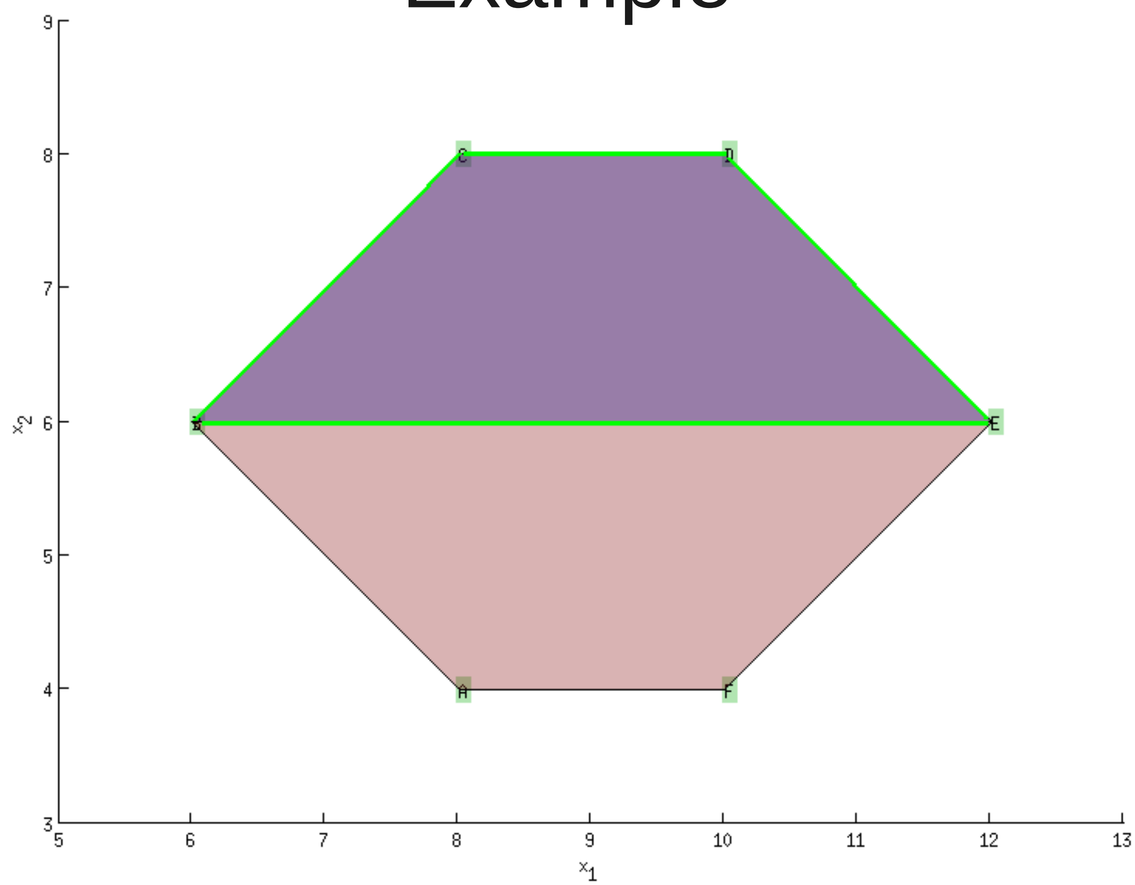
Example



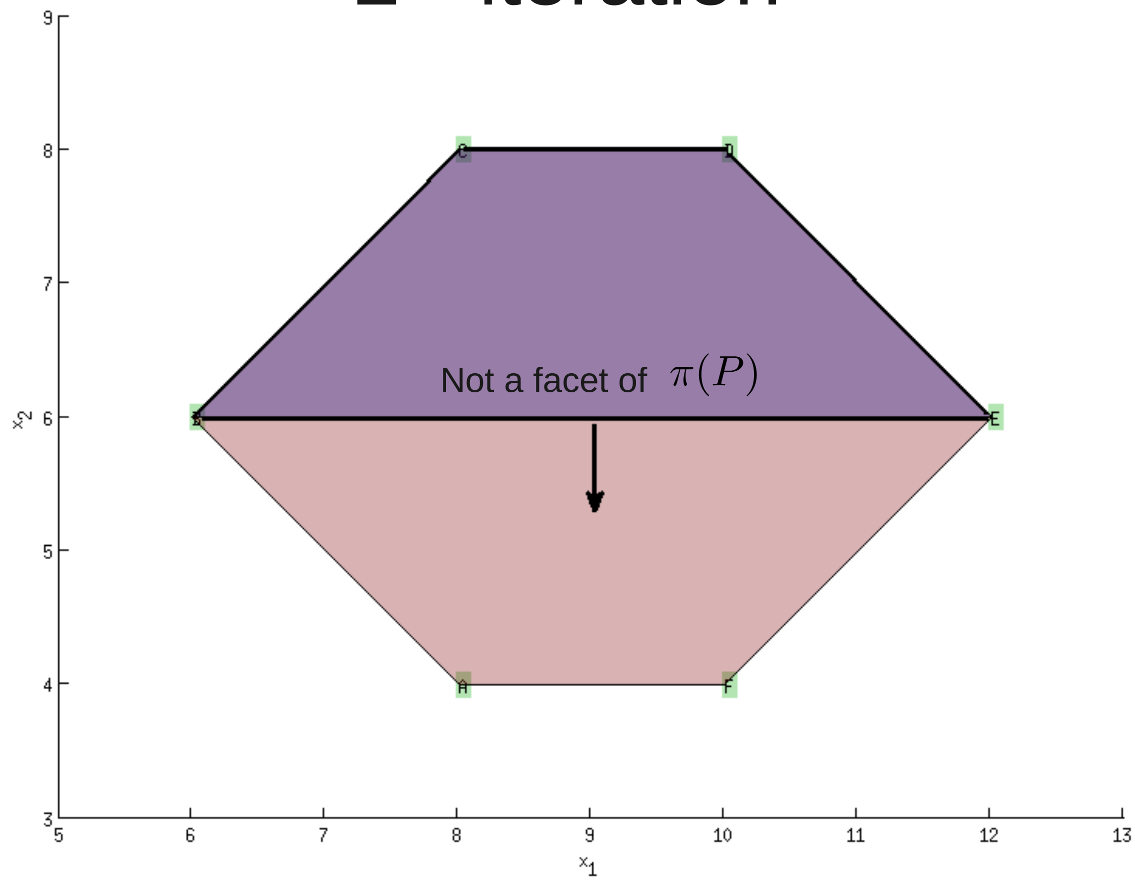
Example



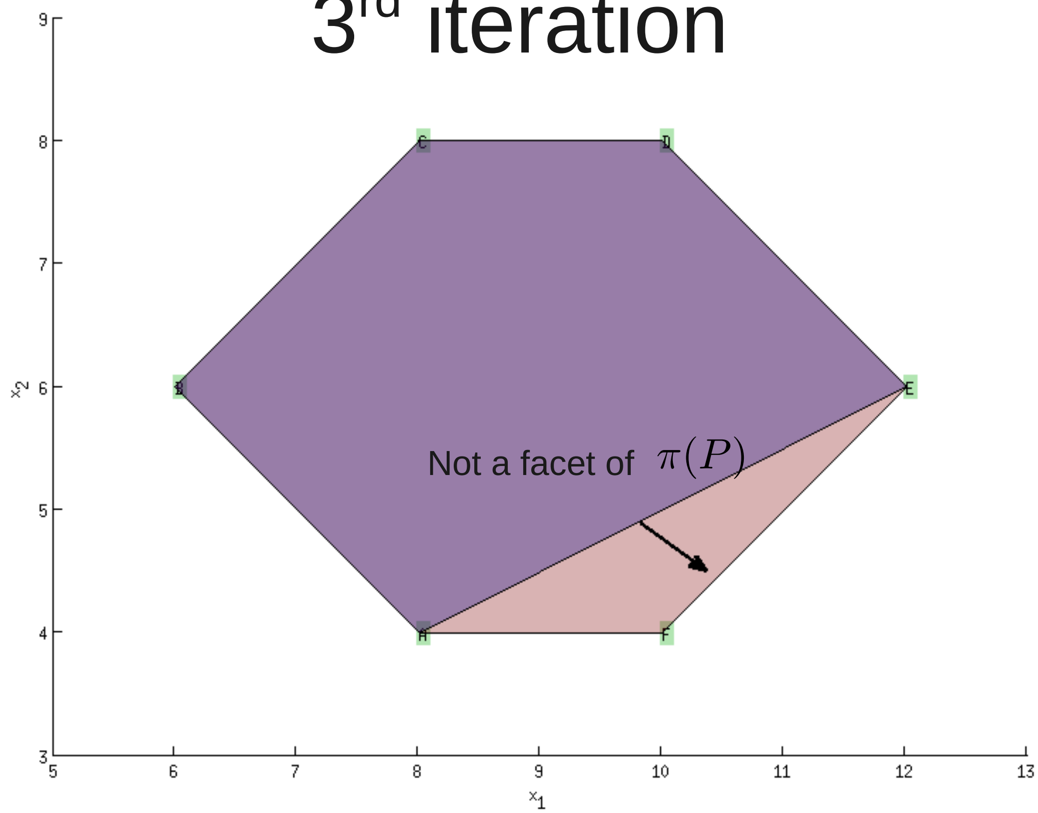
Example



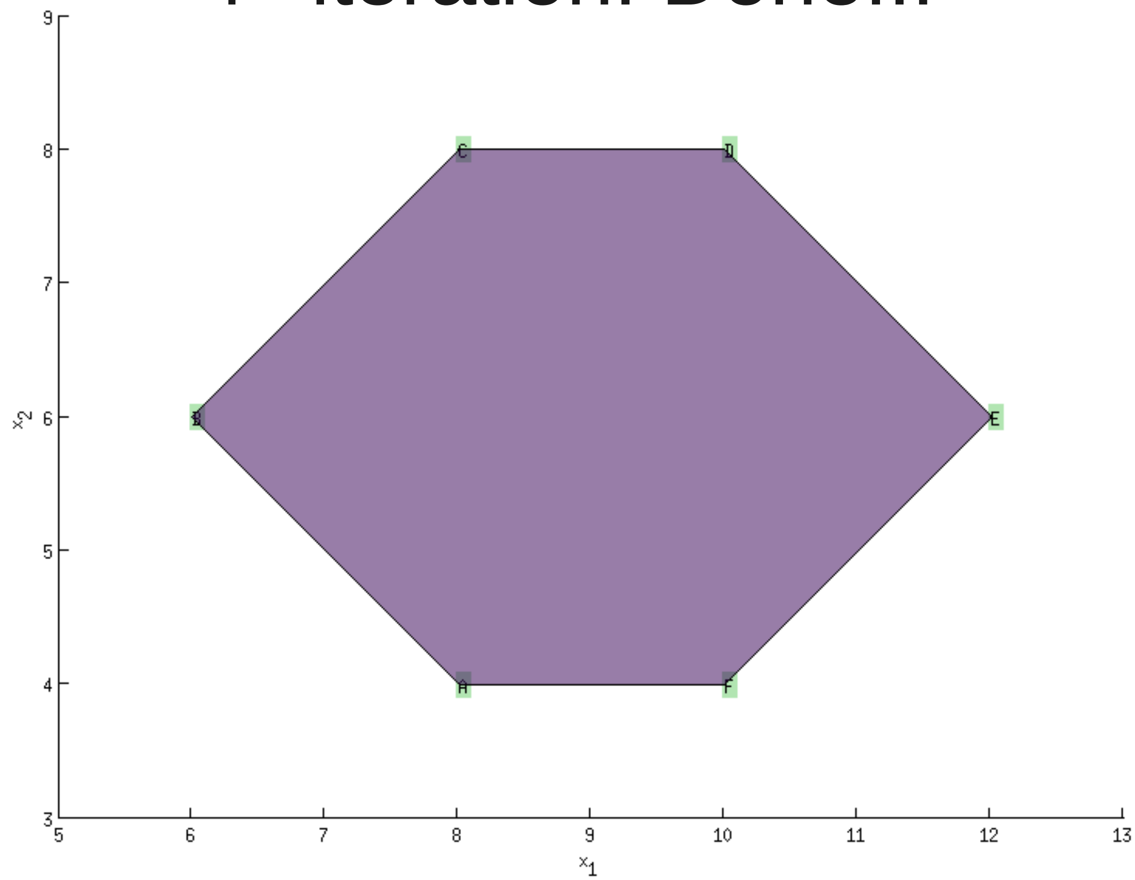
2nd iteration



3rd iteration



4th iteration: Done!!!



Comparison of Projection Algorithms

- Fourier Motzkin Elimination and Block Elimination doesn't work well when used on big problems
- CHM works very well when the dimension of projection is relatively small as compared to the dimention of original polyhedron.
- Weidong Xu, Jia Wang, Jun Sun have already used CHM to get non-Shannon inequalities.

Computation of minimal representation/Redundancy Removal

Definitions

Definition A linear inequality $A_i x \leq b_i$ (for some $i \in \{1, ..., m\}$) of a polyhedron P is *redundant* if it is implied by the other inequalities of P

Definition An extreme point v of polytope P is said t be redundant if it can be represented as convex combination of any other extreme points in the polytope

References

- K. Fukuda and A. Prodon. Double description method revisited. Technical report, Department of Mathematics, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1995
- G. Ziegler, Lectures on Polytopes, Springer, 1994, revised 1998. Graduate Texts in Mathematics.
- Tien Huynh, Catherine Lassez, and Jean-Louis Lassez. *Practical issues on the projection of polyhedral sets*. Annals of mathematics and artificial intelligence, November 1992
- Catherine Lassez and Jean-Louis Lassez. *Quantifier elimination for conjunctions of linear constraints via a convex hull algorithm*. In Bruce Donald, Deepak Kapur, and Joseph Mundy, editors, Symbolic and Numerical Computation for Artificial Intelligence. Academic Press, 1992.
- R. T. Rockafellar. *Convex Analysis* (Princeton Mathematical Series). Princeton Univ Pr.
- W. Xu, J. Wang, J. Sun. *A projection method for derivation of non-Shannon-type information inequalities*. In IEEE International Symposiumon Information Theory (ISIT), pp. 2116 – 2120, 2008.

Minkowski's Theorem for Polyhedral Cones

- For any $m \times d$ real matrix A, $\exists$ some $d \times n$ real matrix R s.t. (A, R) is a DD pair, or in other words, the cone $P(A)$ is generated by R.
- Emphasis on finiteness of columns of R

Weyl's Theorem for Polyhedral Cones

- For any $d \times n$ real matrix R, $\exists$ some $m \times d$ matrix A s.t. (A, R) is a DD pair, or inother words, the set generated by R is the *cone* $P(A)$
- It is the converse of Monkowski's Theorem
- Together they form the *Representation theorem* of Polyhedral Cones