Linear Programming
(4541.554 Introduction to Computer-Aided Design)

School of EECS
Seoul National University

Optimization Problems

- **General nonlinear programming problem**
  - **minimize $f(x)$**
    subject to $g_i(x) \geq 0 \quad i=1,\ldots,m$
    $h_j(x)=0 \quad j=1,\ldots,p$
    $x \in R^n$
- **Convex programming problem**
  - **f : convex**
    g_i: concave
    h_j: linear
  - **local optimum = global optimum**

continuous variables, continuous optimization

- **Linear programming problem**
  - **f, g_i, h_j: linear (can be considered as convex or concave)**
  - **select a solution from a finite set of possible solutions**
  - **Simplex algorithm (1947 by G. B. Dantzig)**

boundary

- **Integer linear programming problem**
  - **integer-valued coordinates**

discrete variables combinatorial optimization

Definitions

- **Instance of an optimization problem**
  - **given (F, c)**

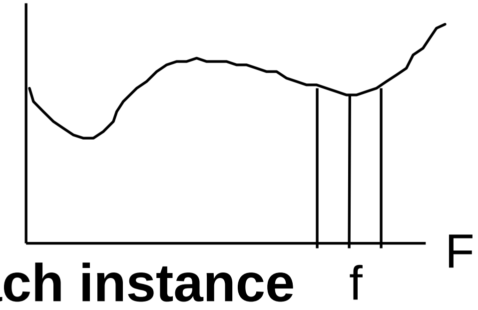
    where F: domain of feasible points

    c: F -> R^1 : cost function

    find f $\in$ F for which c(f) $\leq$ c(y) for all y $\in$ F

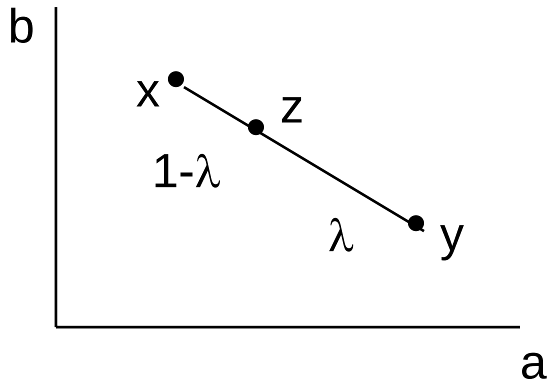
    -> f is a globally optimal solution
  - **e.g. instance of Traveling Salesman Problem (vertices and edges are given)**
- **Optimization problem**
  - **a set of instances of an optimization problem**
  - **e.g. Traveling Salesman Problem**
- **Locally optimal solution**
  - **c(f) $\leq$ c(g) for all g $\in$ N(f)**

    where N is a neighborhood defined for each instance
  - **e.g. N_ε(f) = {x: x $\in$ F and ||x-f|| $\leq$ ε}**

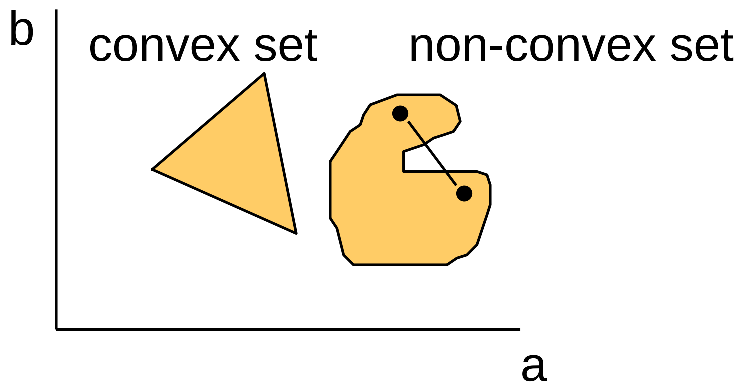


- **Convex combination of x, y $\in$ R^n is any point of the form:**

  $$z = \lambda x + (1-\lambda)y,\ \lambda \in R^1 \text{ and } 0 \le \lambda \le 1$$



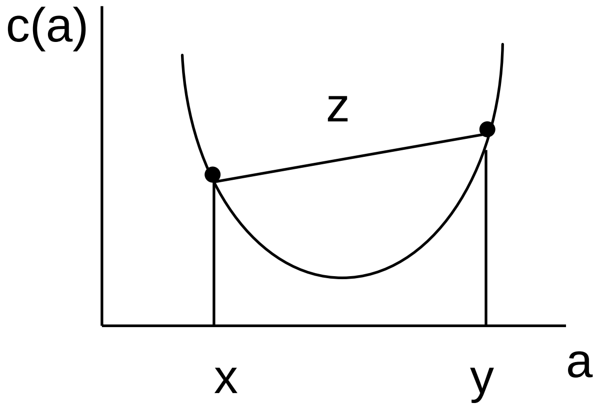
- **A set S $\subseteq$ R^n is convex if it contains all convex combinations of pairs of points x, y $\in$ S**

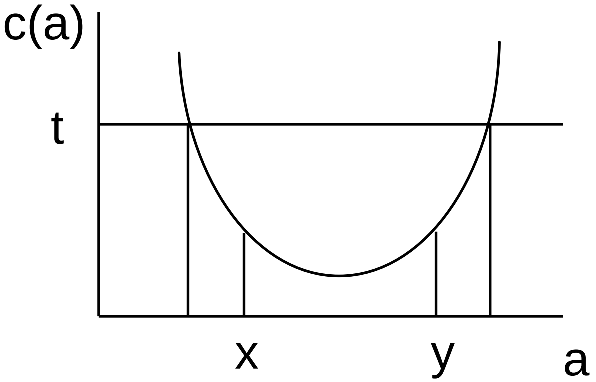


- **Lemma 1**

  The intersection of any number of convex sets is convex.

- **Convex Function**
  - **Let $S \subseteq R^n$ be a convex set. Function $c: S \rightarrow R^1$ is convex in S if $c(\lambda x + (1-\lambda)y) \leq \lambda c(x) + (1-\lambda)c(y)$, $\lambda \in R^1$ and $0 \leq \lambda \leq 1$ for all $x, y \in S$**





- **Lemma 2**

Let c(x) be a convex function on a convex set S. Then set $S_t = \{x : c(x) \leq t, x \in S\}$ is convex.

Proof

For any $x, y \in S_t$, $\lambda x + (1-\lambda)y$ is in S and

$c(\lambda x + (1-\lambda)y) \leq \lambda c(x) + (1-\lambda)c(y) \leq \lambda t + (1-\lambda)t \leq t$

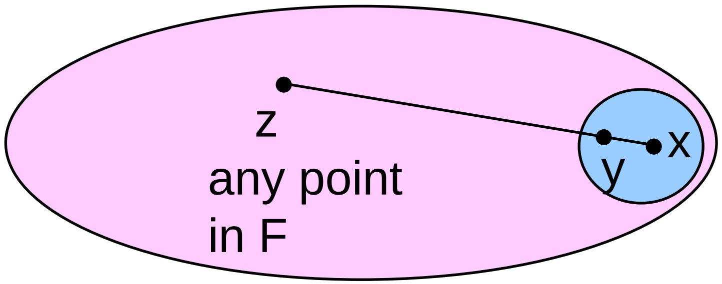
$\Rightarrow \lambda x + (1-\lambda)y$ is in S_t $\Rightarrow$ S_t is convex

Convex Programming Problem

- **Theorem 1**

  For an instance of an optimization problem (F, c) and neighborhood $N_\varepsilon(x) = \{y : y \in F \text{ and } ||x-y|| \le \varepsilon\}$, where $F \subseteq R^n$ is a convex set and c is a convex function, a locally optimal point with respect to N_ε is also a globally optimal point for any $\varepsilon > 0$.

Proof



Choose a λ such that

$y = \lambda x + (1-\lambda)z$ lies within $N_\varepsilon(x)$

$c(y) = c(\lambda x + (1-\lambda)z) \le \lambda c(x) + (1-\lambda)c(z)$

$\Rightarrow c(z) \ge (c(y) - \lambda\, c(x))/(1- \lambda)$

$\ge (c(x) - \lambda\, c(x))/(1- \lambda) = c(x)$

$\uparrow$

x is a local optimum point

- **Convex programming problem**

minimize f(x)

subject to $g_i(x) \geq 0 \quad i=1,\ldots,m$

$h_j(x)=0 \quad j=1,\ldots,p$

$x \in R^n$

where

f: convex

g_i**: concave**

h_j**: linear**

$F = \{x : g_i(x) \geq 0\}$

$= \{x : \underbrace{-g_i(x)}_{\text{convex function}} \leq 0\}$

=> F is convex by Lemma 1 and Lemma 2

=> For (F, f), local optimum = global optimum by theorem 1

Linear Programming Problem

- **General Form of LP**

$$\text{minimize } \mathbf{c}'\mathbf{x}$$
$$\text{subject to}$$
$$\mathbf{a_i}'\mathbf{x} = b_i \quad i \in M$$
$$\mathbf{a_i}'\mathbf{x} \geq b_i \quad i \in \overline{M}$$
$$x_j \geq 0 \quad j \in N$$
$$x_j \text{ unconstrained} \quad j \in \overline{N}$$

- **Canonical Form**

$$\text{minimize } \mathbf{c}'\mathbf{x}$$
$$\text{subject to}$$
$$\mathbf{a_i}'\mathbf{x} \geq b_i \quad \forall i$$
$$x_j \geq 0 \quad \forall j$$

- **Standard Form**

$$\text{minimize } \mathbf{c}'\mathbf{x}$$
$$\text{subject to}$$
$$\mathbf{a_i}'\mathbf{x} = b_i \quad \forall i$$
$$x_j \geq 0 \quad \forall j$$

- **Conversion**

$$\mathbf{Ax} = \mathbf{b} \Rightarrow \begin{cases} \mathbf{Ax} \geq \mathbf{b} \\ -\mathbf{Ax} \geq -\mathbf{b} \end{cases}$$

$$x_j \text{ unconstrained} \Rightarrow \begin{cases} x_j = x_j^+ - x_j^- \\ x_j^+ \geq 0 \\ x_j^- \geq 0 \end{cases}$$

$$\mathbf{Ax} \geq \mathbf{b} \Rightarrow \begin{cases} \mathbf{Ax} - \mathbf{s} = \mathbf{b} \\ \mathbf{s} \geq \mathbf{0} \end{cases} \qquad \mathbf{s} : \text{vector of surplus variables}$$

$$\mathbf{Ax} \leq \mathbf{b} \Rightarrow \begin{cases} \mathbf{Ax} + \mathbf{s} = \mathbf{b} \\ \mathbf{s} \geq \mathbf{0} \end{cases} \qquad \mathbf{s} : \text{vector of slack variables}$$

Examples

- **Diet Problem**

$$\text{minimize } \mathbf{c}'\mathbf{x}$$

$$\text{subject to}$$

$$\mathbf{Ax} \geq \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}$$

where

$\mathbf{c}$: unit cost of each food (c_j won/1gr of kimchi, ...)

$\mathbf{x}$: amount of each food (x_j gr of kimchi, ...)

$\mathbf{A}$: amount of each nutrient per one unit of each food

(a_{ij} mgr of nutrient i in 1gr of kimchi, ...)

$\mathbf{b}$: requirement for each nutrient

(at least b_i mgr of nutrient i is required per day)

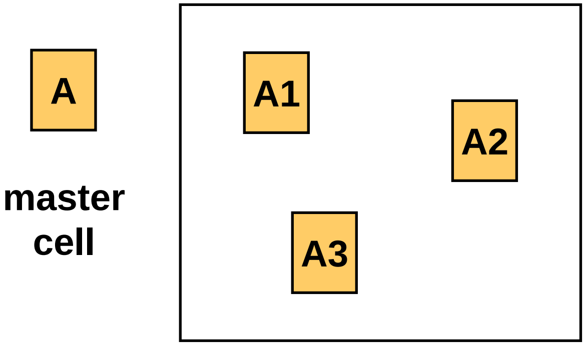
- **Hierarchical Compaction**
  - **Exploit design hierarchy to reduce computation time**
  - **Compact bottom-up**
  - **Fixed-cell**
    - **Cell abstraction with protection frame and terminal frame**
    - **Interconnections among Sub-cells require routing**
  - **Stretching and pitch matching**
    - **Connection by abutment**
  - **Limitations of the previous hierarchical compactor**
    - **Protection frame or fixed terminal location**

      --> Area is wasted.
    - **Stretching and pitch-matching**

      --> Sub-cells can be distorted.

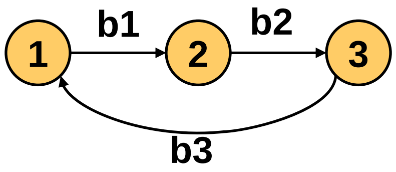
      --> New master cells are generated.

      --> Original layout hierarchy is lost.

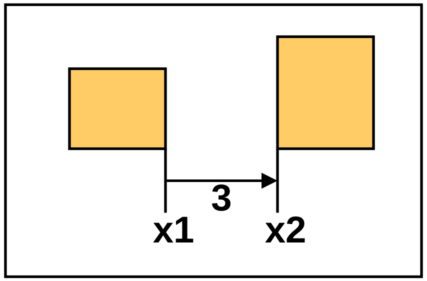


  - **David Marple, "A hierarchy preserving hierarchical compactor," *Proc. 27th Design Automation Conference*, 1990.**

- Constraints
  - **Flat compaction**
    - **$AX \geq B$, A: incidence matrix**
    - **Each constraint is related with two object locations. Each row of A has one '-1' and one '1'.**

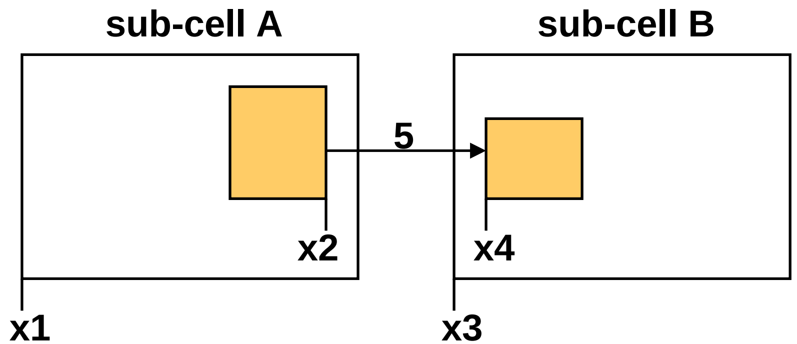


$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} \geq \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix}$$

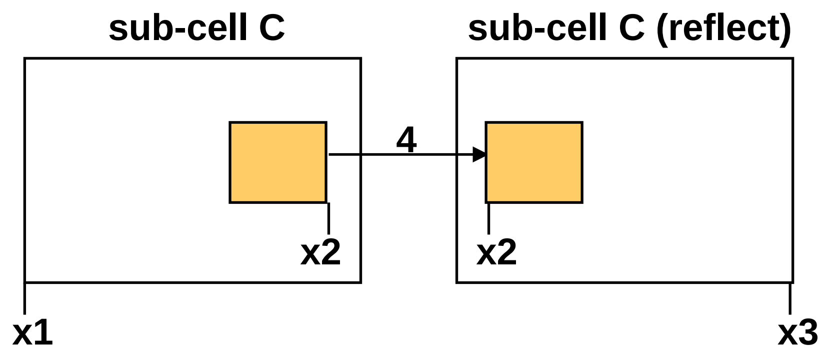


$x1 + 3 \leq x2$

- **Hierarchical compaction**
  - **Each constraint may be related with more than two object locations.**



$$x1 + x2 + 5 \leq x3 + x4$$



$$x1 + x2 + 4 \leq x3 - x2$$

or

$$x1 + 2x2 + 4 \leq x3$$

Examples

- Compaction algorithm
  - Linear program
    - minimize $x_t = C^T X = [000...1]X$
      subject to $AX \geq b$
      $X \geq 0$
      where x_t is the location of the sink vertex
    - Example:
      $x1 + 2 \leq x2$
      $x2 - 1 \leq x3$
      $x1 + 5 \leq x3$
      $x1 + 2x2 + 4 \leq x3$

$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \\ -1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} \geq \begin{bmatrix} 2 \\ -1 \\ 5 \\ 4 \end{bmatrix}$$

Basic Feasible Solution

- **Definitions**
  - **Given a standard form**

    minimize $\mathbf{c}'\mathbf{x}$

    subject to

    $\mathbf{A}\mathbf{x} = \mathbf{b}$

    $\mathbf{x} \geq \mathbf{0}$

    Assume $\mathbf{A}$ is an $m \times n$ matrix ($m < n$) and the rank is m.

  - **Basis of** A
    - **linearly independent collection of columns of** A

      $$B = \left\{\mathbf{A}_{j_1}, \ldots, \mathbf{A}_{j_m}\right\}$$

    - **can be represented by an m x m matrix**

      $$\mathbf{B} = \left[\mathbf{A}_{j_1}, \ldots, \mathbf{A}_{j_m}\right]$$

– **Basic solution**

$$\begin{cases} x_j = 0 \quad \text{for } \mathbf{A}_j \notin \boldsymbol{B} \\ x_{j_k} = k\text{th component of } \mathrm{B}^{-1}\mathbf{b}, k = 1, \ldots, m \end{cases}$$

x_{j_k} : basic variable

$$[\mathbf{A}_1 \ldots \mathbf{A}_n] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

$$\rightarrow [\mathbf{A}_{j_1} \ldots \mathbf{A}_{j_m}] \begin{bmatrix} x_{j_1} \\ \vdots \\ x_{j_m} \end{bmatrix} + \mathbf{0} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

– **Basic feasible solution (BFS)**

- **basic solution in F (domain of feasible points), i.e. x ≥ 0**

- **Lemma 1**
  - **Let x be a BFS of**

    Ax=b

    $x \geq 0$

    corresponding to the basis *B*. Then there exists a cost vector c such that x is the unique optimal solution of the LP

    min c'x

    Ax=b

    $x \geq 0$
- **Proof**
  - **Let c_j=0 if $A_j \in B$**

    1 otherwise

    Then c'x=0 which is optimum (c is non-negative).

    If there is another feasible solution y such that c'y=0, then y_j=0 for $A_j \notin B$. Other y_is are computed by B^{-1}b.

    --> y=x --> unique

- **Theorem 1**
  - **If F is not empty, then at least one BFS exists.**
- **Proof**
  - **WLOG, assume one solution is $[x_1,...,x_n]$, where $x_1,...,x_t>0$ and $x_{t+1},...x_n=0$. Then**

    $A_1x_1+...+A_tx_t=b$ **(1)**

    Let r=rank of $[A_1,...,A_t] \leq m$ (A is an m x n matrix).

    WLOG, assume first r columns are linearly independent.

    Rewrite equations (1) as follows

    $A_1x_1+...+A_rx_r=b-A_{r+1}x_{r+1}-...-A_tx_t$ **(2)**

    Solving (2) gives

    $[x_1,...,x_r]'=\beta-\alpha_{r+1}x_{r+1}-...-\alpha_tx_t$

    As x_t-->0, some of $x_1,...,x_r$ increase or decrease. If any of $x_1,...,x_r$ becomes 0, then stop decreasing x_t.

    --> feasible solution with more zero component

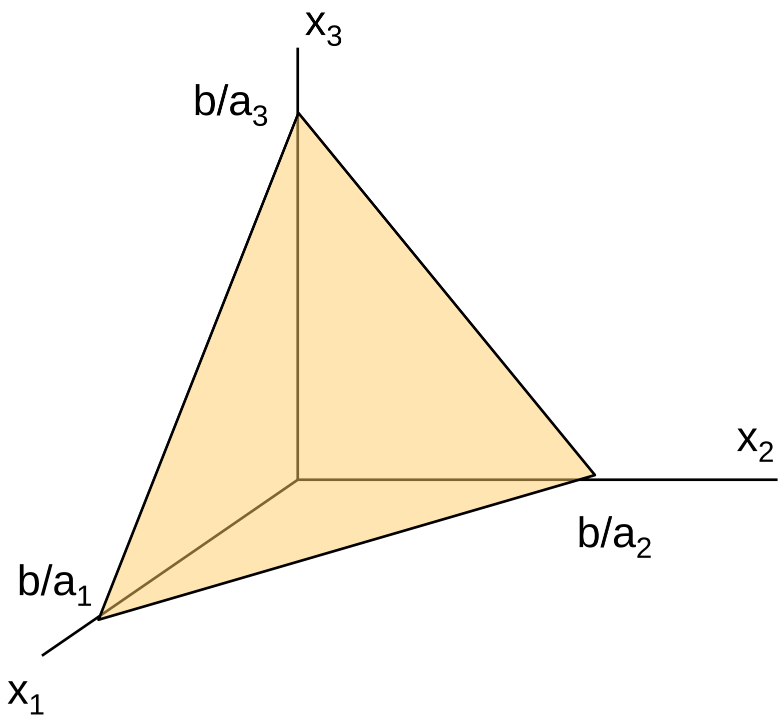
    Continue to obtain a feasible solution with $t \leq m$ nonzero components. The corresponding columns are independent. Otherwise, we can reduce t further until t=r.

Geometry of Linear Program

- **Definitions**
  - **Example**

    $a_1x_1+a_2x_2+a_3x_3=b$ --> dimension: 3 -> 2

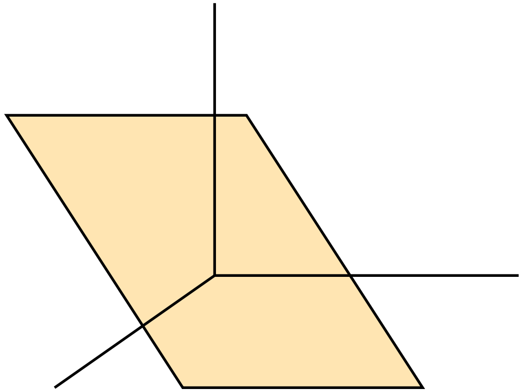
    $x \geq 0$



– **Linear subspace S of R^d**

$S = \{x \in R^d : a_{j1}x_1+...+a_{jd}x_d= 0, j=1,...m\}$

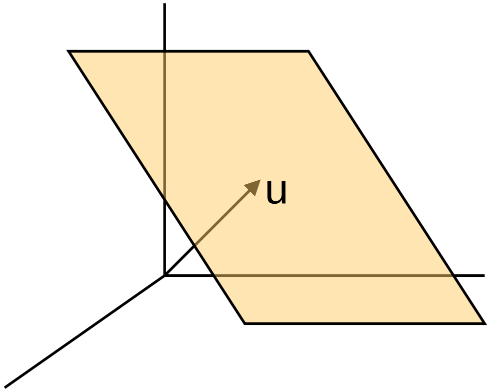
--> dimension=d-rank($[a_{ij}]$)=d-m



– **Affine subspace A of R^d**

$A = \{x \in R^d : a_{j1}x_1+...+a_{jd}x_d= b_j, j=1,...m\}$

$= \{u+x : x \in S\}$



– **Hyperplane**

An affine subspace of R^d of dimension d-1

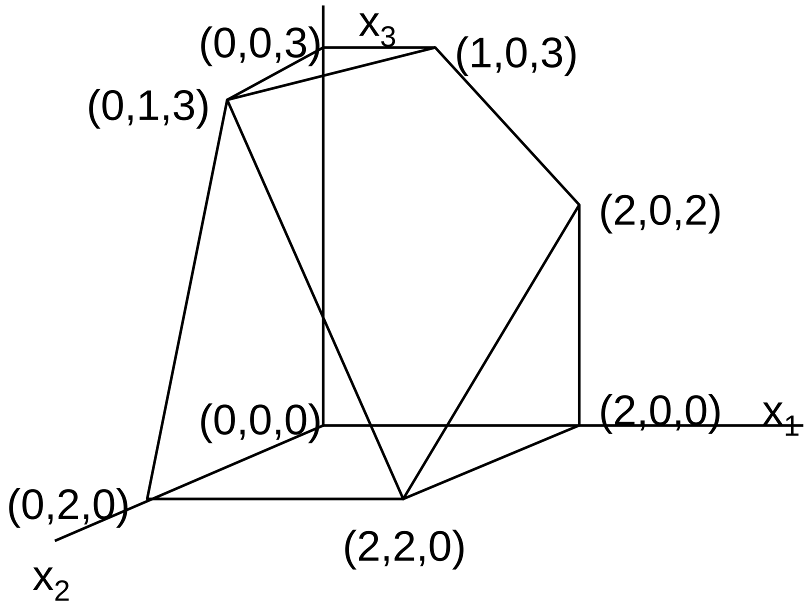
$\{x \in R^d : a_1x_1+...+a_dx_d= b\}$

Defines two halfspaces

$\{x \in R^d : a_1x_1+...+a_dx_d \geq b\}$

$\{x \in R^d : a_1x_1+...+a_dx_d \leq b\}$

– **Convex polytope**
  - **Bounded nonempty intersection of a finite number of halfspaces**
  - **Every point in a convex polytope is the convex combination of its vertices (convex hull)**



$$
\begin{aligned}
x_1 + x_2 + x_3 &\le 4 \\
x_1 &\le 2 \\
x_3 &\le 3 \\
3x_2 + x_3 &\le 6 \\
x_1 &\ge 0 \\
x_2 &\ge 0 \\
x_3 &\ge 0
\end{aligned}
$$

- **Polytope and LP**

$$\mathbf{A}\mathbf{x} = \mathbf{b} \rightarrow [m \times (n-m) \mid m \times m]\mathbf{x} = \mathbf{b}$$
$$\rightarrow [m \times (n-m) \mid \mathbf{B}]\mathbf{x} = \mathbf{b}$$

$$\mathbf{B}^{-1}\mathbf{A}\mathbf{x} = \mathbf{B}^{-1}\mathbf{b} \rightarrow [m \times (n-m) \mid \mathbf{I}]\mathbf{x} = \mathbf{B}^{-1}\mathbf{b}$$
$$\rightarrow [\hat{\mathbf{A}} \mid \mathbf{I}]\mathbf{x} = \hat{\mathbf{b}}$$
$$\rightarrow [[\hat{a}_{ij}] \mid \mathbf{I}]\mathbf{x} = [\hat{b}_i]$$

$$\sum_{j=1}^{n-m} \hat{a}_{ij} x_j + x_{n-m+i} = \hat{b}_i, i = 1, \ldots, m$$

$$x_{n-m+i} = \hat{b}_i - \sum_{j=1}^{n-m} \hat{a}_{ij} x_j, i = 1, \ldots, m$$

$$\mathbf{x} \geq \mathbf{0} \rightarrow \begin{cases} \hat{b}_i - \sum_{j=1}^{n-m} \hat{a}_{ij} x_j \geq 0, i = 1, \ldots, m \\ x_j \geq 0,\ j = 1, \ldots, (n-m) \end{cases}$$

$$\rightarrow \hat{x} = (x_1, \ldots, x_{n-m}) \in \text{polytope P}$$

$$\begin{cases} \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases} \leftrightarrow \begin{cases} \left[\hat{a}_{ij}\right]\hat{\mathbf{x}} \geq \left[\hat{b}_i\right] \\ \hat{\mathbf{x}} \geq \mathbf{0} \end{cases}$$

$$\mathbf{B}^{-1}\mathbf{Ax} = \mathbf{B}^{-1}\mathbf{b} \rightarrow \left[m \times (n-m) \mid \mathbf{I}\right]\mathbf{x} = \mathbf{B}^{-1}\mathbf{b}$$

$$\rightarrow \left[\left[\hat{a}_{ij}\right] \mid \mathbf{I}\right]\mathbf{x} = \left[\hat{b}_i\right]$$

After solving for $\hat{\mathbf{x}}$, $x_{n\text{-}m+i}$ can be obtained by

$$x_{n-m+i} = \hat{b}_i - \sum_{j=1}^{n-m} \hat{a}_{ij} x_j, i = 1, \ldots, m$$

- **Example 1**

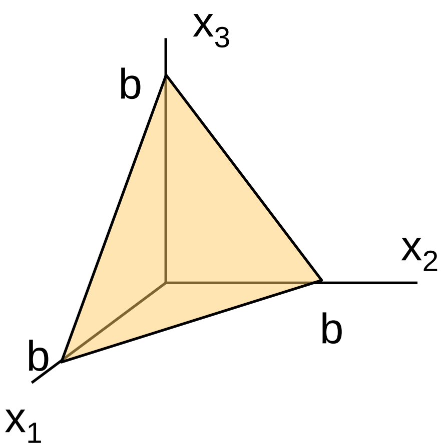
$$\begin{array}{l}
x_1 + x_2 + x_3 + x_4 = 4 \\
x_1 + x_5 = 2 \\
x_3 + x_6 = 3 \\
3x_2 + x_3 + x_7 = 6 \\
\\
x_1, \ldots, x_7 \geq 0
\end{array}$$

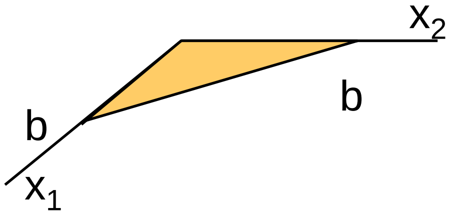
$$\begin{array}{l}
x_1 + x_2 + x_3 \leq 4 \\
x_1 \leq 2 \\
x_3 \leq 3 \\
3x_2 + x_3 \leq 6 \\
\\
x_1, \ldots, x_3 \geq 0
\end{array}$$

$$x_4 = 4 - (x_1 + x_2 + x_3) \geq 0$$

- **Example 2**

$$x_1 + x_2 + x_3 = b \longrightarrow x_1 + x_2 \leq b$$





- **Theorem 2**

$\mathbf{x}^*$ is a bfs of F defined by

$$\mathbf{Ax} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

$\Leftrightarrow$ The corresponding $\hat{\mathbf{x}}^*$ is a vertex of the convex polytope P

- **Proof**

$\Rightarrow$

There exists a cost vector $\mathbf{c}$ such that

$\mathbf{x} = \mathbf{x}^*$ is the unique vector satisfying

$$\mathbf{c}'\mathbf{x} \leq \mathbf{c}'\mathbf{x}^*$$
$$\mathbf{Ax} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

(Lemma1)

$$\mathbf{c}'\mathbf{x} = \sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{n-m} c_j x_j + \sum_{j=n-m+1}^{n} c_j x_j$$

$$i = j - (n - m)$$

$$\mathbf{c}'\mathbf{x} = \sum_{j=1}^{n-m} c_j x_j + \sum_{i=1}^{m} c_{n-m+i} x_{n-m+i}$$

$$= \sum_{j=1}^{n-m} c_j x_j + \sum_{i=1}^{m} c_{n-m+i} (\hat{b}_i - \sum_{j=1}^{n-m} \hat{a}_{ij} x_j)$$

$$= \sum_{j=1}^{n-m} (c_j - \sum_{i=1}^{m} c_{n-m+i} \hat{a}_{ij}) x_j + \sum_{i=1}^{m} c_{n-m+i} \hat{b}_i$$

$$= \mathbf{d}'\hat{\mathbf{x}} + k$$

$$\mathbf{c}'\mathbf{x}^* = \mathbf{d}'\hat{\mathbf{x}}^* + k$$

$$\mathbf{c}'\mathbf{x} \le \mathbf{c}'\mathbf{x}^* \to \mathbf{d}'\hat{\mathbf{x}} \le \mathbf{d}'\hat{\mathbf{x}}^*$$

There exists a cost vector $\mathbf{c}$ such that

$\mathbf{x} = \mathbf{x}^*$ is the **unique** vector satisfying

$$\mathbf{c}'\mathbf{x} \le \mathbf{c}'\mathbf{x}^*$$

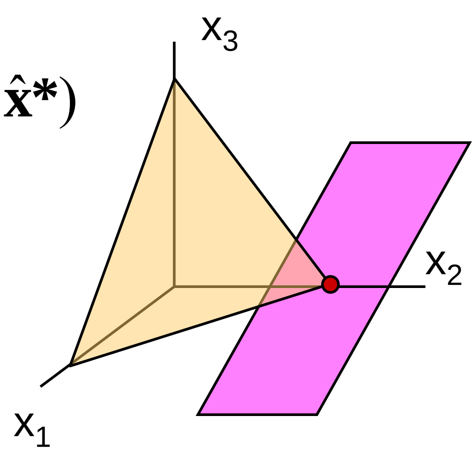
$$\left.\begin{array}{l}\mathbf{A}\mathbf{x} = \mathbf{b}\\ \mathbf{x} \ge \mathbf{0}\end{array}\right\} \rightarrow \hat{\mathbf{x}} \in \mathrm{P}$$

Then $\hat{\mathbf{x}}^* = (x_1, \ldots, x_{n-m})$ is the **unique** point in $\mathrm{R}^{\,\mathrm{n\text{-}m}}$ satisfying

$$\left.\begin{array}{l}\mathbf{d}'\hat{\mathbf{x}} \le \mathbf{d}'\hat{\mathbf{x}}^* \text{ (halfspace)}\\ \hat{\mathbf{x}} \in \mathrm{P}\end{array}\right\} \rightarrow \text{intersection}$$

intersection is a unique point $\rightarrow$ vertex

($\mathbf{d}'\hat{\mathbf{x}} = \mathbf{d}'\hat{\mathbf{x}}^*$ is a supporting hyperplane defining $\hat{\mathbf{x}}^*$)



$\mathbf{x}^*$ is a bfs $\Leftarrow \hat{\mathbf{x}}^*$ is a vertex

$$\hat{\mathbf{x}}^* \text{ is a vertex} \rightarrow \hat{\mathbf{x}}^* \in \mathrm{P} \rightarrow \begin{cases} \mathbf{A}\mathbf{x}^* = \mathbf{b} \\ \mathbf{x}^* \geq \mathbf{0} \end{cases} \rightarrow \sum_j \mathbf{A}_j x_j^* = \mathbf{b}, \forall j \text{ s.t. } x_j^* > 0$$

If $\mathbf{A}_j$'s are linearly independent, then $\mathbf{x}^*$ is a bfs.

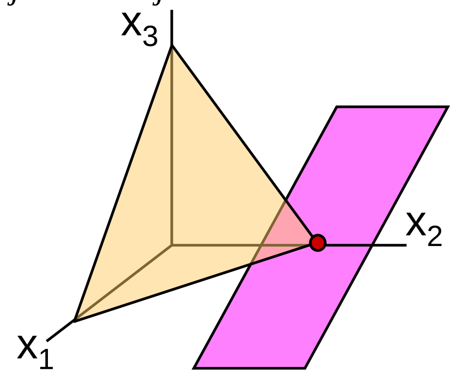
Assume $\mathbf{A}_j$'s are not linearly independent.

Then $\sum_j \mathbf{A}_j d_j = \mathbf{0}$ for some $d_j \neq 0$

$$\rightarrow \sum_j \mathbf{A}_j \left(x_j^* \pm \Theta d_j \right) = \mathbf{b} \text{ for sufficiently small } \Theta \text{ such that } x_j^* \pm \Theta d_j \geq 0$$

Define two points $\mathbf{x}'$ and $\mathbf{x}''$

$$x'_j = \begin{cases} x_j^* - \Theta d_j & x_j^* > 0 \\ 0 & x_j^* = 0 \end{cases} \qquad x''_j = \begin{cases} x_j^* + \Theta d_j & x_j^* > 0 \\ 0 & x_j^* = 0 \end{cases}$$



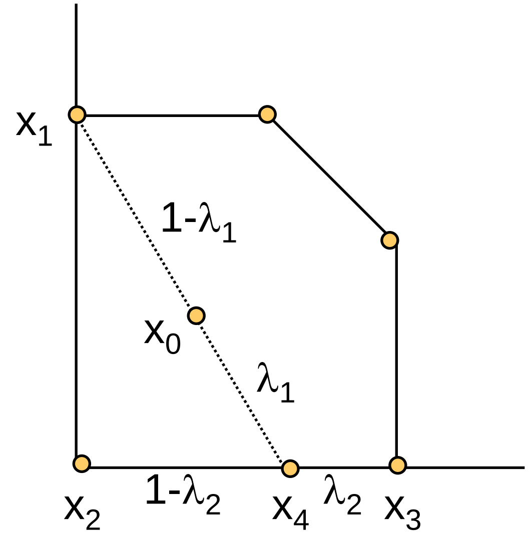
$$\text{then } \mathbf{x}', \mathbf{x}'' \in \mathrm{F} \text{ and } \mathbf{x}^* = \frac{1}{2}\mathbf{x}' + \frac{1}{2}\mathbf{x}'' \rightarrow \text{vertex } \hat{\mathbf{x}}^* = \frac{1}{2}\hat{\mathbf{x}}' + \frac{1}{2}\hat{\mathbf{x}}''$$

However, a vertex cannot be a strict convex combination of points in P.

- **Theorem 3**

  In any instance of LP with bounded F, there is an optimal vertex of P (optimal BFS).

- **Proof**



Assume $\mathbf{x_0}$ is an optimal solution.

$$\mathbf{x}_0 = \sum_{i=1}^{N} a_i \mathbf{x}_i, \ \mathbf{x}_i = \text{vertices of P}$$

where $\sum_{i=1}^{N} a_i = 1, a_i \geq 0$

Let $\mathbf{x}_j$ be the vertex with lowest cost.

Then

$$\mathbf{c}'\mathbf{x}_0 = \sum_{i=1}^{N} a_i \mathbf{c}'\mathbf{x}_i \geq \mathbf{c}'\mathbf{x}_j \sum_{i=1}^{N} a_i = \mathbf{c}'\mathbf{x}_j$$

$\therefore \mathbf{x}_j$ is an optimal solution.

$$\begin{aligned}
x_0 &= \lambda_1 x_1 + (1-\lambda_1)x_4 \\
&= \lambda_1 x_1 + (1-\lambda_1)(\lambda_2 x_2 + (1-\lambda_2)x_3) \\
&= \lambda_1 x_1 + (1-\lambda_1)\lambda_2 x_2 + (1-\lambda_1)(1-\lambda_2)x_3 \\
&= \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3
\end{aligned}$$

$$\begin{aligned}
&\alpha_1 + \alpha_2 + \alpha_3 \\
&= \lambda_1 + (1-\lambda_1)\lambda_2 + (1-\lambda_1)(1-\lambda_2) \\
&= 1
\end{aligned}$$

Moving from BFS to BFS

Let $\mathbf{x}^*$ be a BFS for a basis $\mathcal{B} = \{\mathbf{A}_{j_k} : k = 1,\dots,m\}$

$$\mathbf{A}\mathbf{x}^* = \mathbf{b} \rightarrow \sum_{k=1}^{m} \mathbf{A}_{j_k} x^*_{j_k} = \mathbf{b} \qquad (1)$$

$$\mathbf{A}_j = \sum_{k=1}^{m} e_{kj} \mathbf{A}_{j_k}, \quad \mathbf{A}_j \notin \mathcal{B} \rightarrow \sum_{k=1}^{m} e_{kj} \mathbf{A}_{j_k} - \mathbf{A}_j = \mathbf{0} \qquad (2)$$

$$(1) - \theta \cdot (2) \rightarrow \sum_{k=1}^{m} (x^*_{j_k} - \theta \cdot e_{kj}) \mathbf{A}_{j_k} + \theta \cdot \mathbf{A}_j = \mathbf{b}$$

Increase θ until some $(x^*_{j_k} - \theta \cdot e_{kj})$ becomes 0

$$\rightarrow \theta = \min_{k | e_{kj} > 0} \frac{x^*_{j_k}}{e_{kj}} \rightarrow k = l \qquad \text{If } e_{kj} \le 0, \forall k, \text{ then F is unbounded } (\theta \rightarrow \infty)$$

$$\text{BFS } x^*_i = \begin{cases} x^*_{j_k}, & i = j_k \\ 0, & \text{otherwise} \end{cases} \rightarrow \text{new BFS } x'_i = \begin{cases} x^*_{j_k} - \theta \cdot e_{kj}, & i = j_k \text{ and } k \neq l \\ 0, & i = j_l \\ \theta, & i = j \\ 0, & \text{otherwise} \end{cases}$$

- **Theorem 4**

$$B' = \{A_{j_1}, ..., A_{j_{l-1}}, A_j, A_{j_{l+1}}, ..., A_{j_m}\} \text{ is a new basis}$$

- **Proof**

$$\text{Let } B' = \lfloor A_{j_1}, ..., A_{j_{l-1}}, A_j, A_{j_{l+1}}, ..., A_{j_m} \rfloor$$

$$B'd = d_l A_j + \sum_{k=1, k\neq l}^{m} d_k A_{j_k} = d_l \sum_{k=1}^{m} e_{kj} A_{j_k} + \sum_{k=1, k\neq l}^{m} d_k A_{j_k}$$

$$= d_l e_{lj} A_{j_l} + \sum_{k=1, k\neq l}^{m} (d_l e_{kj} + d_k) A_{j_k} \qquad (3)$$

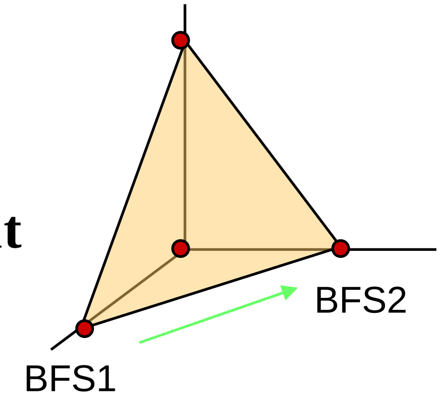
$$A_{j_k}, k = 1, ..., m \text{ are linearly independent}$$

$$\rightarrow \text{if we set } (3) = 0, \text{ then } d_l e_{lj} = 0 \rightarrow \text{since } e_{lj} > 0, d_l = 0$$

$$\rightarrow (3) = 0 \text{ becomes } \sum_{k=1, k\neq l}^{m} d_k A_{j_k} = 0 \rightarrow d_k = 0, k = 1, ..., m$$

$$\rightarrow d = 0 \rightarrow \text{All columns of } B' \text{ are linearly independent}$$

$$\rightarrow B' \text{ is a basis}$$



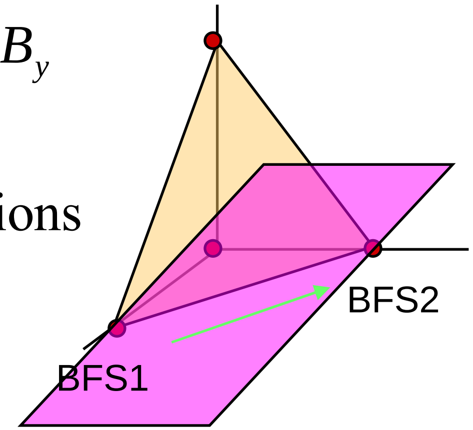
- **Theorem 5**

If x and y are adjacent, i.e. $B_y = (B_x - \{\mathbf{A}_j\}) \cup \{\mathbf{A}_k\}$,

then $\hat{x}$ and $\hat{y}$ are adjacent, i.e. $[\hat{x}, \hat{y}]$ is an edge of the polytope.

- **Proof**

Let us construct a cost vector $c_j = \begin{cases} 0 & \text{if } A_j \in B_x \cup B_y \\ 1 & \text{otherwise} \end{cases}$



Then all feasible solutions that are convex combinations of x and y are uniquely optimal.

To prove uniqueness, suppose z is optimal.

Then z is a convex combination of bfs's with bases subsets of $B_x \cup B_y$.

However, x and y are the only such bfs's.

$$\boxed{\sum_{k=1}^{m} (x^*_{j_k} - \theta \cdot e_{kj})\mathbf{A}_{j_k} + \theta \cdot \mathbf{A}_j = \mathbf{b}}$$

Therefore, only convex combinations w of x and y satisfy $Aw = b, w \geq 0$, and $c'w \leq c'x$.

Therefore, in P, only w on $[\hat{x}, \hat{y}]$ satisfy $d'\hat{w} \leq d'\hat{x}$.

Hence only $[\hat{x}, \hat{y}]$ is the intersection of a halfspace with P and is therefore an edge.

Tableau

- **Example**

3x1 + 2x2 + x3 = 1

5x1 + x2 + x3 + x4 = 3

2x1 + 5x2 + x3 + x5 = 4

	x1	*x2*	*x3*	*x4*	*x5*
1	3	2	1	0	0
3	5	1	1	1	0
4	2	5	1	0	1

Select B={A_3, A_4, A_5} --> make an identity matrix

	x1	*x2*	*x3*	*x4*	*x5*
1	3	2	1	0	0
2	2	−1	0	1	0
3	−1	3	0	0	1

x1=x2=0, x3=1, x4=2, x5=3 : BFS

x3, x4, x5 : basic variables

Tableau

	$x1$	$x2$	$x3$	$x4$	$x5$
1	3	2	1	0	0
2	2	−1	0	1	0
3	−1	3	0	0	1

$$\mathbf{A}_1 = \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix} = 3\mathbf{A}_3 + 2\mathbf{A}_4 - \mathbf{A}_5 = \sum_{k=1}^{m=3} e_{k1}\mathbf{A}_{j_k} \rightarrow j_1 = 3,\ j_2 = 4,\ j_3 = 5$$

To put $\mathbf{A}_1$ into a new basis,

$$\Theta = \min_{k|e_{k1}>0} \frac{x^*_{j_k}}{e_{k1}} = \min\left(\frac{1}{3}, \frac{2}{2}\right) = \frac{1}{3} \rightarrow l = 1 \rightarrow \mathbf{A}_{j_1} = \mathbf{A}_3 \text{ becomes non-basic}$$

$$\text{Make } \mathbf{A}_1 = [1\ 0\ 0]' \rightarrow \text{BFS} = \left[\frac{1}{3}, 0, 0, \frac{4}{3}, \frac{10}{3}\right], B = \{\mathbf{A}_1, \mathbf{A}_4, \mathbf{A}_5\}$$

	$x1$	$x2$	$x3$	$x4$	$x5$
1/3	1	2/3	1/3	0	0
4/3	0	−7/3	−2/3	1	0
10/3	0	11/3	1/3	0	1

Choosing a Profitable Column

$$x_j : 0 \rightarrow \Theta$$

$$x_{j_k} : x^*_{j_k} \rightarrow x^*_{j_k} - \Theta e_{kj}$$

$$\text{cost} : \sum_{k=1}^{m} c_{j_k} x^*_{j_k} \rightarrow \sum_{k=1}^{m} c_{j_k} x^*_{j_k} - \sum_{k=1}^{m} c_{j_k} \Theta e_{kj} + c_j \Theta$$

$$\Delta_j = \Theta \cdot \left(c_j - \sum_{k=1}^{m} c_{j_k} e_{kj} \right) = \Theta \cdot \left(c_j - z_j \right) = \Theta \cdot \bar{c}_j$$

Theorem

If $\bar{c}_j \geq 0$ for all j, then we are at an optimum

Proof

$$\bar{\mathbf{c}} = \left[\bar{c}_1 \ \bar{c}_2 \ \ldots \ \bar{c}_n \right]' \geq 0 \rightarrow \bar{\mathbf{c}} = \mathbf{c} - \mathbf{z} \geq \mathbf{0} \rightarrow \mathbf{c} \geq \mathbf{z}$$

$$\rightarrow \text{for any } \mathbf{y} \geq \mathbf{0}, \mathbf{c}'\mathbf{y} \geq \mathbf{z}'\mathbf{y}$$

$$\mathbf{A}_j = \sum_{k=1}^{m} e_{kj}\mathbf{A}_{j_k} = \mathbf{B}\begin{bmatrix} e_{1j} \\ \vdots \\ e_{mj} \end{bmatrix} \rightarrow \mathbf{A} = \mathbf{B}\begin{bmatrix} e_{11} & \cdots & e_{1n} \\ \vdots & \ddots & \vdots \\ e_{m1} & \cdots & e_{mn} \end{bmatrix} = \mathbf{BE} \rightarrow \mathbf{E} = \mathbf{B}^{-1}\mathbf{A}$$

For any $\mathbf{y} \geq \mathbf{0}$,

$$\mathbf{c}'\mathbf{y} \geq \mathbf{z}'\mathbf{y} = \begin{bmatrix} \sum_{k=1}^{m} c_{j_k} e_{k1} & \cdots & \sum_{k=1}^{m} c_{j_k} e_{kn} \end{bmatrix} \cdot \mathbf{y}$$

$$= \begin{bmatrix} c_{j_1} & \cdots & c_{j_m} \end{bmatrix} \cdot \begin{bmatrix} e_{11} & \cdots & e_{1n} \\ \vdots & \ddots & \vdots \\ e_{m1} & \cdots & e_{mn} \end{bmatrix} \cdot \mathbf{y}$$

$$= \begin{bmatrix} c_{j_1} & \cdots & c_{j_m} \end{bmatrix} \cdot \mathbf{E} \cdot \mathbf{y} = \begin{bmatrix} c_{j_1} & \cdots & c_{j_m} \end{bmatrix} \cdot \mathbf{B}^{-1}\mathbf{A} \cdot \mathbf{y}$$

$$= \begin{bmatrix} c_{j_1} & \cdots & c_{j_m} \end{bmatrix} \cdot \mathbf{B}^{-1}\mathbf{b}$$

$$= \begin{bmatrix} c_{j_1} & \cdots & c_{j_m} \end{bmatrix} \cdot \begin{bmatrix} x_{j_1}^{*} & \cdots & x_{j_m}^{*} \end{bmatrix}' = \mathbf{c}'\mathbf{x}^{*} \quad \text{(non - basic variables are 0)}$$

$$\rightarrow \mathbf{c}'\mathbf{y} \geq \mathbf{c}'\mathbf{x}^{*}$$

$$\therefore \mathbf{x}^{*} \text{ is a global optimum}$$

Simplex Algorithm

procedure simplex

begin

opt := 'no'; unbounded := 'no';

(when either becomes 'yes' the algorithm terminates)

while opt = 'no' and unbounded = 'no' do

if $\bar{c}_j \geq 0$ for all j then opt := 'yes'

else begin

choose any j such that $\bar{c}_j < 0$;

if $e_{kj} \leq 0$ for all k then unbounded := 'yes'

else

$$\sum_{k=1}^{m} (x^*_{j_k} - \theta \cdot e_{kj}) \mathbf{A}_{j_k} + \theta \cdot \mathbf{A}_j = \mathbf{b}$$

find $\theta_0 = \min_{k \mid e_{kj} > 0} \left[\frac{x_{j_k}}{e_{kj}} \right] = \frac{x_{j_l}}{e_{lj}}$

and pivot on e_{lj}

end

end

- **Example**

z=x1+x2+x3+x4+x5

0	1	1	1	1	1
1	3	2	1	0	0
3	5	1	1	1	0
4	2	5	1	0	1

4. $-\sum_{k=1}^{m} x_{j_k} c_{j_k}$

1. obtain basis x3 x4 x5

3. $\bar{c}_j = c_j - \sum_{k=1}^{m} e_{kj} c_{j_k}$

$$\bar{c}_1 = 1 - \left(3 \cdot 1 + 2 \cdot 1 + (-1) \cdot 1\right) = -3$$

$$\bar{c}_2 = 1 - \left(2 \cdot 1 + (-1) \cdot 1 + 3 \cdot 1\right) = -3$$

−6	−3	−3	0	0	0
1	3	2	1	0	0
2	2	−1	0	1	0
3	−1	3	0	0	1

x_{j_k} e_{k1} e_{k2}

2. make identity matrix

-6	-3	-3	0	0	0
1	3	2	1	0	0
2	2	-1	0	1	0
3	-1	3	0	0	1

5. select column j that gives

   most negative $\bar{c}_j$ or

   most negative $\Theta \cdot \bar{c}_j$ (more computation)

6. compute $\Theta = \min_{k \mid e_{kj} > 0} \dfrac{x_{j_k}}{e_{kj}}$

7. pivot

8. repeat steps 3 - 7

   until $\bar{c}_j \geq 0$ for all j

$-9/2$	$3/2$	0	$3/2$	0	0
$1/2$	$3/2$	1	$1/2$	0	0
$5/2$	$7/2$	0	$1/2$	1	0
$3/2$	$-11/2$	0	$-3/2$	0	1

Beginning the Simplex Algorithm

- **How to obtain an initial BFS?**
  - **Use slack variables**
    - **Ax ≤ b --> Ax + Ix^s = b, x^s are initial basic variables**
    - **What if b < 0? --> -Ax - Ix^s = -b, then use artificial variables**
  - **Use artificial variables, then two-phase method**
    - **Ax=b --> Ax + Ix^a = b, x^a are initial basic variables**
    - **All the artificial variables are driven out of the basis**

$$\rightarrow \begin{cases} \mathbf{x}^a = \mathbf{0} \\ x_{j_k} \geq 0, k = 1, \ldots, m \rightarrow \text{basic variables} \\ x_j = 0, \text{otherwise} \rightarrow \text{non} - \text{basic variables} \end{cases}$$

$$\rightarrow \text{original problem}$$

- **Two-phase method**
  - **In phase I, minimize the cost function**

$$\xi = \sum_{i=1}^{m} x_i^a$$

$$\begin{cases} \xi = 0 \text{ and all } x_i^a\text{'s are driven out of the basis : ok} \\ \xi > 0 : \text{no feasible solution to the original problem} \\ \xi = 0 \text{ but some } x_i^a\text{'s remain in the basis :} \\ \qquad\qquad \text{continue pivoting until we get a basis} \\ \qquad\qquad \text{with the original variables} \end{cases}$$

In the last case, $e_{kj} < 0$ can be a pivot because $\Theta = 0$

$$\text{(recall} \sum_{k=1}^{m} \left(x_{j_k} - \Theta \cdot e_{kj}\right) \mathbf{A}_{j_k} + \Theta \cdot \mathbf{A}_j = \mathbf{b}\text{)}$$

```
procedure two-phase
begin
    infeasible := 'no'; redundant := 'no';
    (Phase I may set these to 'yes')
Phase I:
    introduce an artificial basis, x_i^a;
    call simplex with cost ξ = Σ x_i^a;
    if ξ_opt > 0 in Phase I then infeasible := 'yes'
    else begin
        if an artificial variable is in the basis and cannot be driven out (e_kj = 0 ∀j)
            then redundant := 'yes' and omit the corresponding row;
Phase II:
        call simplex with original cost
    end
end
```

- **Example**

		$x1^a$	$x2^a$	$x3^a$	$x1$	$x2$	$x3$	$x4$	$x5$
$-z=$	0	0	0	0	1	1	1	1	1
$-\xi=$	0	1	1	1	0	0	0	0	0
	1	1	0	0	3	2	1	0	0
	3	0	1	0	5	1	1	1	0
	4	0	0	1	2	5	1	0	1

c_j

$$\xi = \sum_{i=1}^{m} x_i^a$$

		$x1^a$	$x2^a$	$x3^a$	$x1$	$x2$	$x3$	$x4$	$x5$
$-z=$	0	0	0	0	1	1	1	1	1
$-\xi=$	−8	0	0	0	−10	−8	−3	−1	−1
$x1^a=$	1	1	0	0	3	2	1	0	0
$x2^a=$	3	0	1	0	5	1	1	1	0
$x3^a=$	4	0	0	1	2	5	1	0	1

$$\bar{c}_j = c_j - \sum_{k=1}^{m} e_{kj} c_{j_k}$$

$$\xi = \sum_{i=1}^{m} x_i^a$$

$$\bar{c}_j = c_j - \sum_{k=1}^{m} e_{kj} c_{j_k}$$

		$x1^a$	$x2^a$	$x3^a$	x1	x2	x3	x4	x5
$-z=$	−1/3	−1/3	0	0	0	1/3	2/3	1	1
$-\xi=$	−14/3	10/3	0	0	0	−4/3	1/3	−1	−1
$x1=$	1/3	1/3	0	0	1	2/3	1/3	0	0
$x2^a=$	3/4	−5/3	1	0	0	−7/3	−2/3	1	0
$x3^a=$	10/3	−2/3	0	1	0	11/3	1/3	0	1

		$x1^a$	$x2^a$	$x3^a$	x1	x2	x3	x4	x5
$-z=$	−1/2	−1/2	0	0	−1/2	0	1/2	1	1
$-\xi=$	−4	4	0	0	2	0	1	−1	−1
$x2=$	1/2	1/2	0	0	3/2	1	1/2	0	0
$x2^a=$	5/2	−1/2	1	0	7/2	0	1/2	1	0
$x3^a=$	3/2	−15/6	0	1	−11/2	0	−3/2	0	1

		$x1^a$	$x2^a$	$x3^a$	$x1$	$x2$	$x3$	$x4$	$x5$
$-z=$	-3	0	-1	0	-4	0	0	0	1
$-\xi=$	$-3/2$	7/2	1	0	11/2	0	3/2	0	-1
$x2=$	1/2	1/2	0	0	3/2	1	1/2	0	0
$x4=$	5/2	$-1/2$	1	0	7/2	0	1/2	1	0
$x3^a=$	3/2	$-5/2$	0	1	$-11/2$	0	$-3/2$	0	1

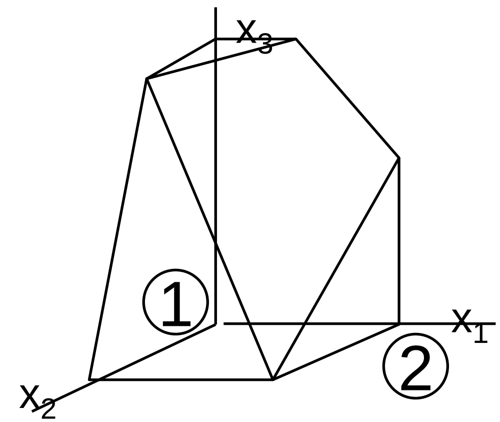
		$x1^a$	$x2^a$	$x3^a$	$x1$	$x2$	$x3$	$x4$	$x5$
$-z=$	$-9/2$	0	-1	-1	3/2	0	3/2	0	0
$-\xi=$	0	7/2	1	1	0	0	0	0	0
$x2=$	1/2	1/2	0	0	3/2	1	1/2	0	0
$x4=$	5/2	$-1/2$	1	0	7/2	0	1/2	1	0
$x5=$	3/2	$-5/2$	0	1	$-11/2$	0	$-3/2$	0	1

This example gives an optimum point when Phase I is finished $\left(\bar{c}_j \geq 0 \forall j\right)$

Geometric Aspects of Pivoting

		x1	x2	x3	x4	x5	x6	x7
	−34	−1	−14	−6	0	0	0	0
x4=	4	1	1	1	1	0	0	0
x5=	2	1	0	0	0	1	0	0
x6=	3	0	0	1	0	0	1	0
x7=	6	0	3	1	0	0	0	1

1



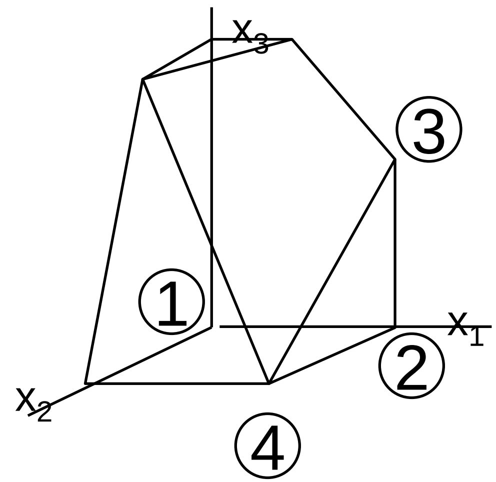
		x1	x2	x3	x4	x5	x6	x7
	−32	−1	−14	−6	0	0	0	0
x4=	2	0	1	1	1	−1	0	0
x1=	2	1	0	0	0	1	0	0
x6=	3	0	0	1	0	0	1	0
x7=	6	0	3	1	0	0	0	1

2

Geometric Aspects of Pivoting

		x1	x2	x3	x4	x5	x6	x7
	−20	0	−8	0	6	−5	0	0
x3=	2	0	1	1	1	−1	0	0
x1=	2	1	0	0	0	1	0	0
x6=	1	0	−1	0	−1	1	1	0
x7=	4	0	2	0	−1	1	0	1

3



		x1	x2	x3	x4	x5	x6	x7
	−4	0	0	8	14	−13	0	0
x2=	2	0	1	1	1	−1	0	0
x1=	2	1	0	0	0	1	0	0
x6=	3	0	0	1	0	0	1	0
x7=	0	0	0	−2	−3	3	0	1

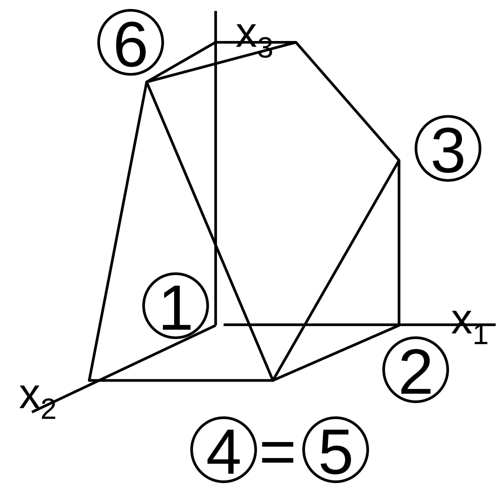
4

		x1	x2	x3	x4	x5	x6	x7
	−4	0	0	−2/3	1	0	0	13/3
x2=	2	0	1	1/3	0	0	0	1/3
x1=	2	1	0	2/3	1	0	0	−1/3
x6=	3	0	0	1	0	0	1	0
x5=	0	0	0	−2/3	−1	1	0	1/3

5

		x1	x2	x3	x4	x5	x6	x7
	−2	1	0	0	2	0	0	4
x2=	1	−1/2	1	0	−1/2	0	0	1/2
x3=	3	3/2	0	1	3/2	0	0	−1/2
x6=	0	−3/2	0	0	−3/2	0	1	1/2
x5=	2	1	0	0	0	1	0	0

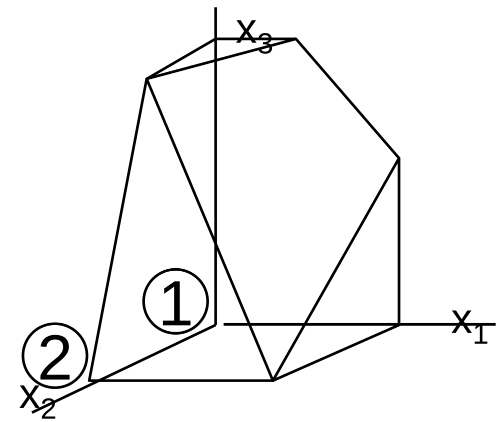
6



- **Taking an alternative path**

		x1	x2	x3	x4	x5	x6	x7
	−34	−1	−14	−6	0	0	0	0
x4=	4	1	1	1	1	0	0	0
x5=	2	1	0	0	0	1	0	0
x6=	3	0	0	1	0	0	1	0
x7=	6	0	3	1	0	0	0	1

1



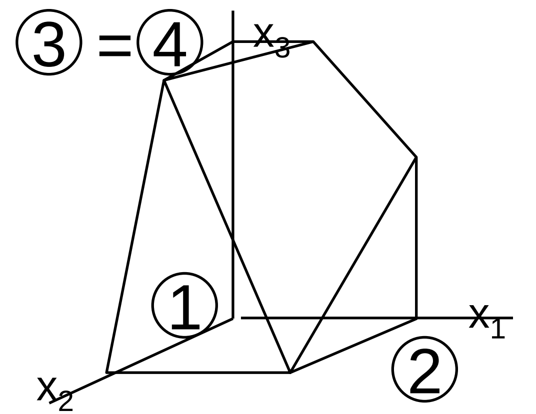
		x1	x2	x3	x4	x5	x6	x7
	−6	−1	0	−4/3	0	0	0	14/3
x4=	2	1	0	2/3	1	0	0	−1/3
x5=	2	1	0	0	0	1	0	0
x6=	3	0	0	1	0	0	1	0
x2=	2	0	1	1/3	0	0	0	1/3

2

Geometric Aspects of Pivoting

③

		x1	x2	x3	x4	x5	x6	x7
	−2	−1	0	0	0	0	4/3	14/3
x4=	0	1	0	0	1	0	−2/3	−1/3
x5=	2	1	0	0	0	1	0	0
x3=	3	0	0	1	0	0	1	0
x2=	1	0	1	0	0	0	−1/3	1/3



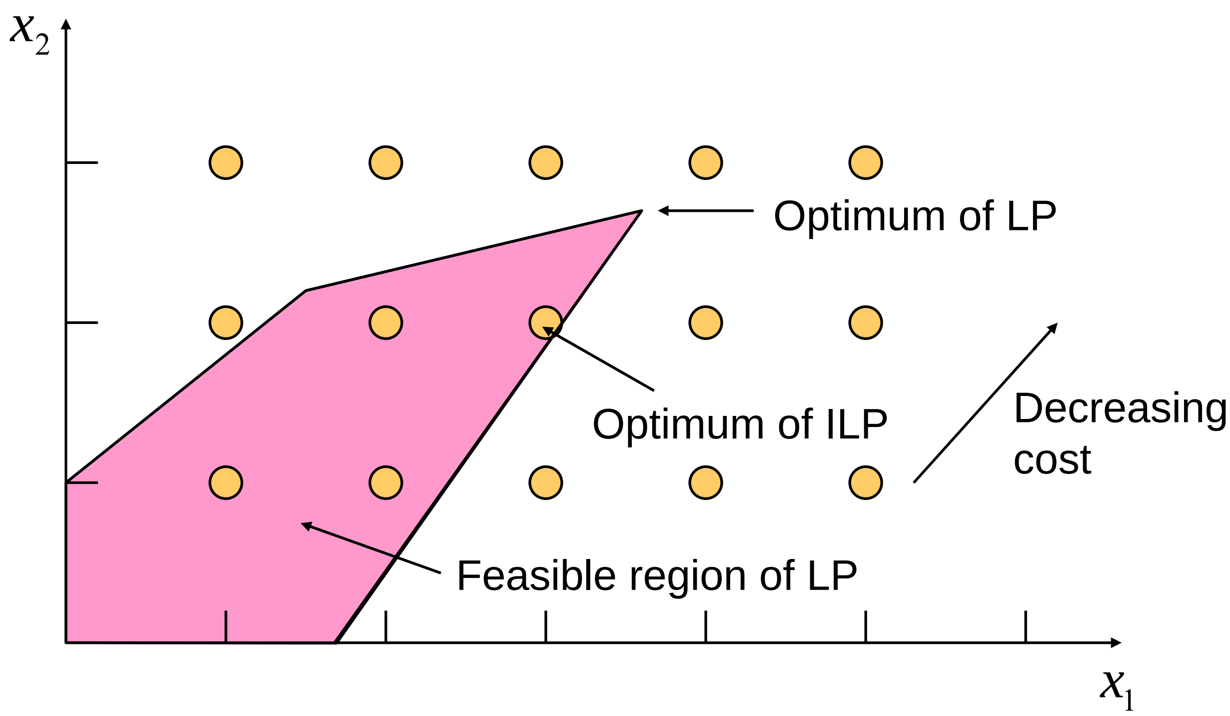
④

		x1	x2	x3	x4	x5	x6	x7
	−2	0	0	0	1	0	2/3	13/3
x1=	0	1	0	0	1	0	−2/3	−1/3
x5=	2	0	0	0	−1	1	2/3	1/3
x3=	3	0	0	1	0	0	1	0
x2=	1	0	1	0	0	0	−1/3	1/3

Integer Linear Programming

- **Problem**

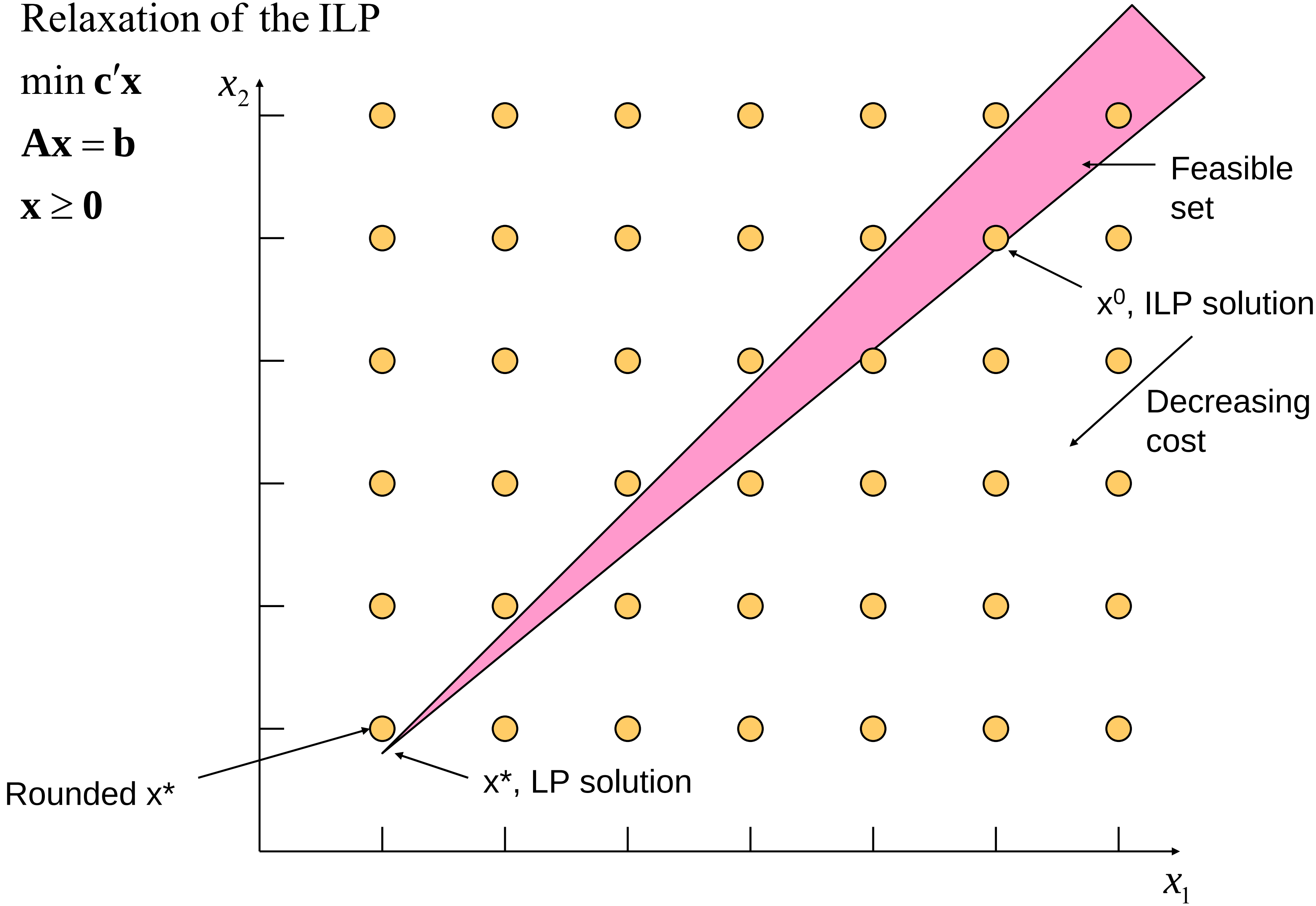
$$\min \mathbf{c}'\mathbf{x}$$
$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$
$$\mathbf{x} : \text{integer}$$



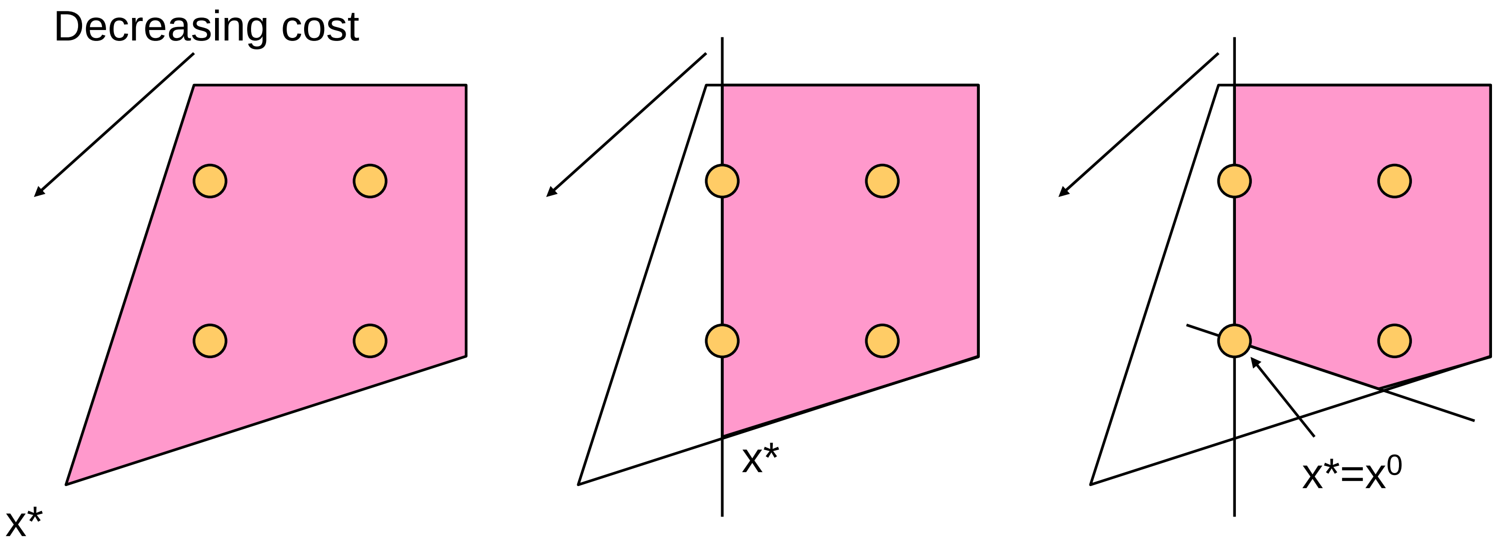
- **Cutting-Plane Algorithm**

Relaxation of the ILP

$$\min \mathbf{c}'\mathbf{x}$$
$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$



- Add constraints to an ILP that do not exclude integer feasible points until the solution to the LP relaxation is integer



$$\mathbf{Ax} = \mathbf{b} \to \mathbf{B}^{-1}\mathbf{Ax} = \mathbf{B}^{-1}\mathbf{b} \to \mathbf{Ex} = \widetilde{\mathbf{b}}$$

$$x_{j_k} + \sum_{j:\mathbf{A}_j \notin B} e_{ij} x_j = \widetilde{b}_i, A_{j_k} \in B$$

Let $e_{ij} = \lfloor e_{ij} \rfloor + f_{ij}$ **and** $\widetilde{b}_i = \lfloor \widetilde{b}_i \rfloor + f_i$

Then $x_{j_k} + \sum_{j:\mathbf{A}_j \notin B} \left(\lfloor e_{ij} \rfloor + f_{ij} \right) x_j = \lfloor \widetilde{b}_i \rfloor + f_i$

$$\to x_{j_k} + \sum_{j:\mathbf{A}_j \notin B} \lfloor e_{ij} \rfloor x_j - \lfloor \widetilde{b}_i \rfloor = f_i - \sum_{j:\mathbf{A}_j \notin B} f_{ij} x_j$$

For x_{j_k} **and** x_j **to be integers,** $f_i - \sum_{j:\mathbf{A}_j \notin B} f_{ij} x_j$ **must be an integer. (1)**

Because $\mathbf{0} \le f_i < \mathbf{1}$, $\mathbf{0} \le f_{ij} < \mathbf{1}$, **and** $x_j \ge \mathbf{0}$, $f_i - \sum_{j:\mathbf{A}_j \notin B} f_{ij} x_j < \mathbf{1}$. **(2)**

From (1) and (2), $f_i - \sum_{j:A_j \notin B} f_{ij} x_j \le \mathbf{0}$.

$\to$ **This excludes the solution (**$x_j = \mathbf{0}$, $f_i > \mathbf{0}$**) of the relaxation LP.**

But does not exclude any integer feasible point.

$\to$ **Add** $-\sum_{j:\mathbf{A}_j \notin B} f_{ij} x_j + x^s = -f_i$ **as a new constraint (Gomory cut).**