Advances and Trends of Linear Programming

Outline

- What is LP?
- What are the applications?
- History of LP
- What is new in LP?
- Where to go in the future?

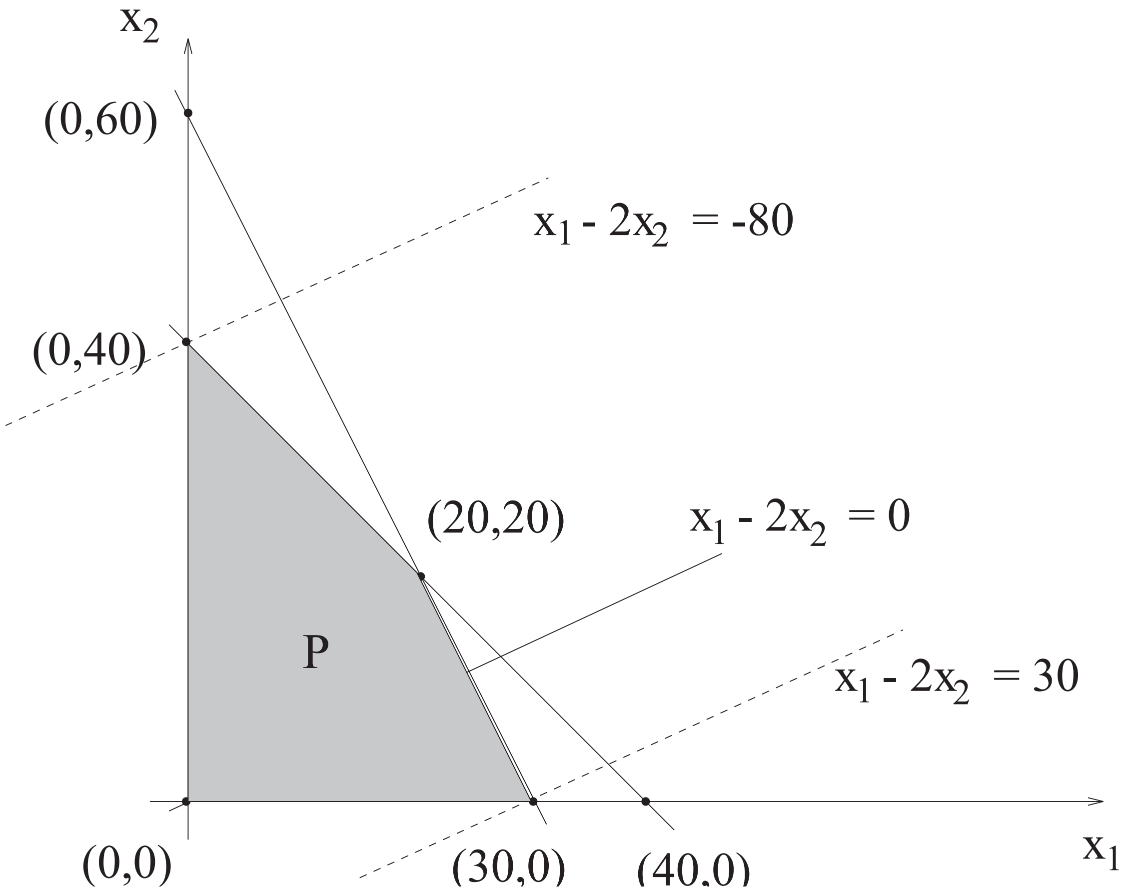
What is Linear Programming (LP)?

- Optimize a linear objective function of decision variables subject to a set of linear constraints.

Example :

$$\begin{aligned} \text{Minimize} \quad & x_1 - 2x_2 \\ \text{subject to} \quad & x_1 + x_2 \le 40 \\ & 2x_1 + x_2 \le 60 \\ & x_1, x_2 \ge 0. \end{aligned}$$

- **Graphic Representation**



$$\mathbf{P} = \{(x_1, x_2) \mid x_1 + x_2 \leq 40,\ 2x_1 + x_2 \leq 60,\ x_1, x_2 \geq 0\}$$

Feasible Domain

- **General Form**

$$\text{Minimize (Maximize)} \quad \underbrace{c_1x_1 + c_2x_2 + \cdots + c_nx_n}_{\text{linear objective function}}$$

s. t.

m linear constraints

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \overset{(<)}{\underset{(>)}{=}} b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \overset{(<)}{\underset{(>)}{=}} b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_1 + \cdots + a_{mn}x_n \overset{(<)}{\underset{(>)}{=}} b_m \end{cases}$$

$$\underbrace{x_1, x_2, \ldots, x_n}_{n \text{ decision variables}} \geq 0.$$

- **Matrix Form**

$$
\begin{array}{ll}
\text{Min} & \mathbf{c}^T\mathbf{x} \\
\text{(Max)} & \\
\text{s. t.} & \mathbf{A}\mathbf{x} \overset{(<)}{\underset{(>)}{=}} \mathbf{b} \\
& \mathbf{x} \geq \mathbf{0}.
\end{array}
$$

$x, c \in R^n$, $b \in R^m$, A: $m \times n$ matrix.

x → variable

c → cost coefficients

b → right-hand-side

A → constraint matrix

- **Applications**
  - One of the most widely applied methodologies.
  - 85 % of Fortune 500 companies had used LP.
  - Industrial Environments
    - Corporate Planning
    - Factory Planning
    - Product Distribution
    - Lease-Buy Decision
    - Production Scheduling
    - Inventory Control
    - Workman's Compensation

      ⋮
  - Agriculture Environments
    - Food Manufacturing
    - Deport Location
    - Irrigation System

      ⋮

- Other Environments
  - Manpower Planning
  - Activity Planning
  - Accounting & Finance
  - Administration, Education, & Politics
  - Advertising and Marketing

    ⋮
- Allocation of Financial Budgets
- Capital Investment

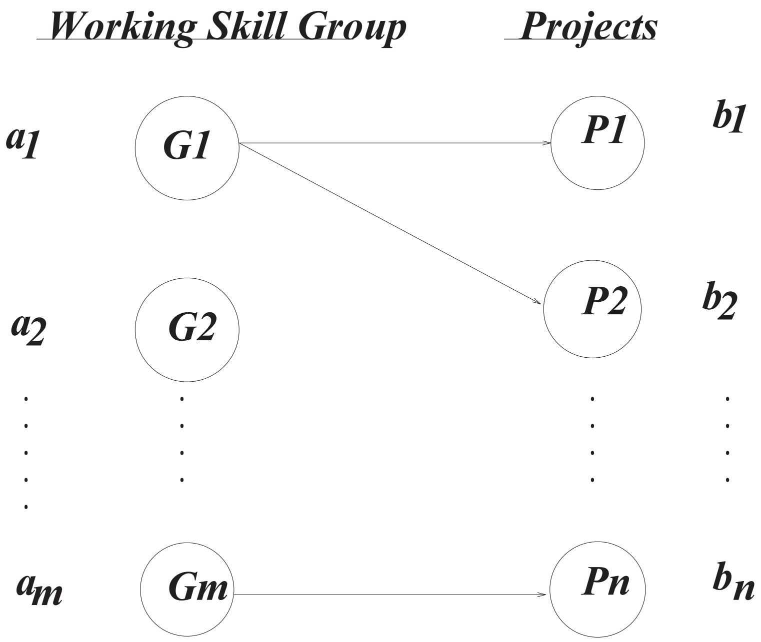
Reference:

"Operations Research," Volume 50, Number 1, January/February 2002.

□ How to apply LP?

1. Understand your problem and limits.
2. List possible actions as decision variables
   - $x_1, x_2, \ldots, x_n$
3. Figure out the cost/reward associated with each activity level
   - $c_1, c_2, \ldots, c_n$
4. Determine the restrictions of actions
   - $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \underset{(>)}{\overset{(<)}{=}} b_i, i = 1, \ldots, m$

- **Example** (Allocation of Manpower)

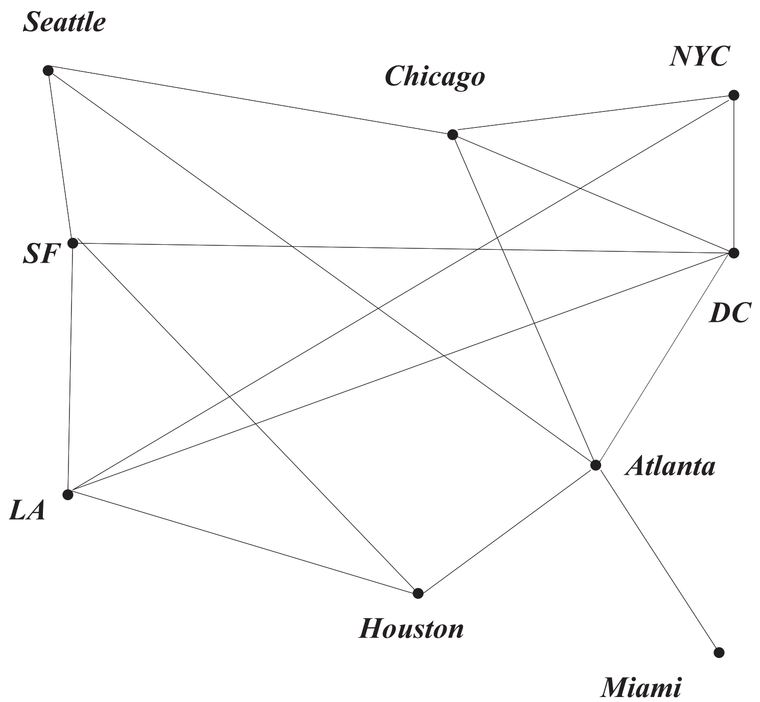


x_{ij} = # of people w/ skill i working on project j.

$$\begin{aligned} \text{Minimize} \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\ \text{s. t.} \quad & \textstyle\sum_{i=1}^{m} x_{ij} \geq b_j, \ j = 1, \ldots, n \\ & \textstyle\sum_{j=1}^{n} x_{ij} \leq a_i, \ i = 1, \ldots, m \\ & l_{ij} \leq x_{ij} \leq u_{ij}, \ \forall i, j. \end{aligned}$$

Telecommunication Network Planning

- AT&T Domestic Facility Network



- Node: Switching Machine
- Link: Optical Fiber, Coaxial Cable, Satellite, ...

Problem: Given a set of point-to-point traffic, load onto network with minimum cost. (LP with millions of variables and constraints!)

□ History of LP

- Conceived by G. B. Dantzig (1947).
  - mechanized planning tool for a deployment, training, and logistical supply program in USAF.
- Named by T. C. Koopmans & G. B. Dantzig (1948).
- Simplex Method proposed by G. B. Dantzig (1948).
  (L. V. Kantorovich worked on a restricted class in 1939).
- Kantorovich & Koopmans received the Nobel Prize in Economic Science (1975).
  - Theory of optimum allocation of resources.
- Ellipsoid Method proposed by L. G. Khachian (1979).
  - First polynomial-time algorithm for LP.

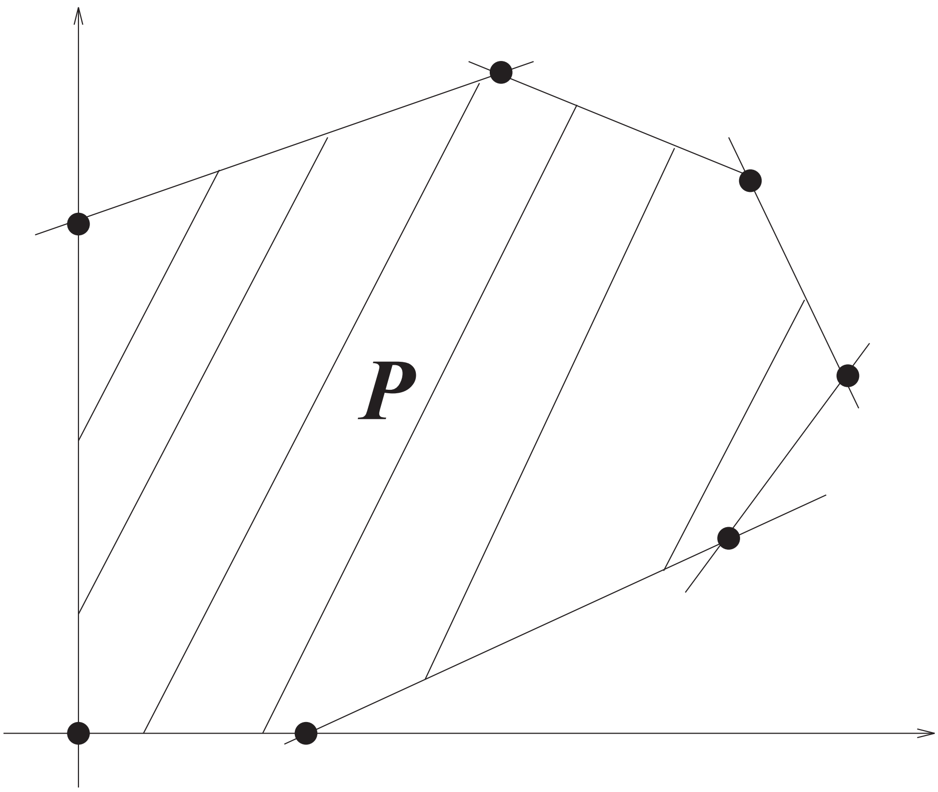
- Interior-Point Method proposed by N. Karmarkar (1984).
  - First “good” polynomial-time algorithm for LP.

□ What's New in Solving LP?

- What is special about LP?
  - Its feasible domain.

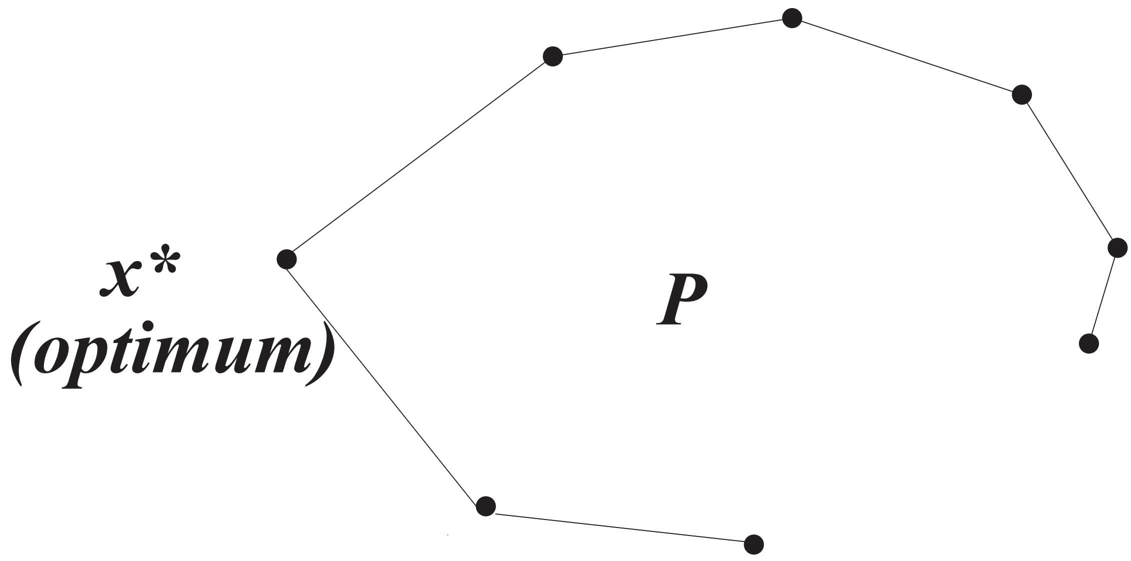
$$P = \{\mathbf{x} \in \mathbf{R^n} | \mathbf{AX} \overset{(<)}{\underset{(>)}{=}} \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

is a polyhedral set with vertices.



– (Fundamental theorem of LP)
For a consistent LP, its optimum is either unbounded or is attained at one vertex of $\mathbf{P}$.

- Fact:



- Question:
How to find $\mathbf{x}^*$?

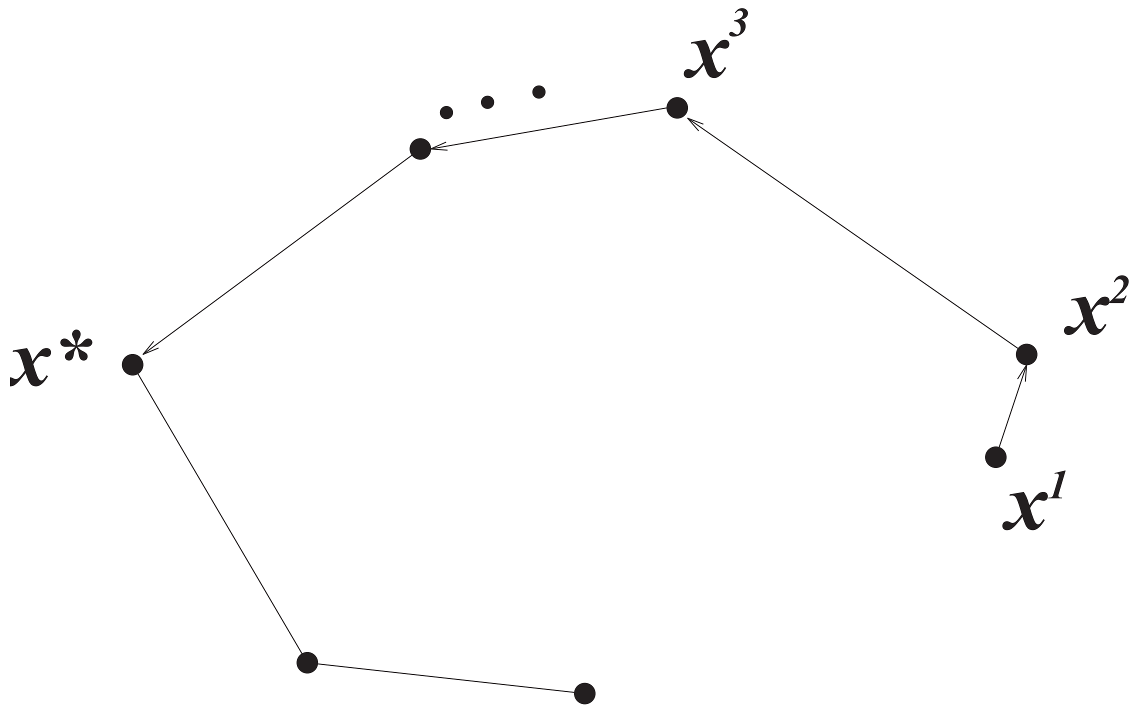
□ **How to Solve LP?**

- Enumeration Method
  - Find all vertices and choose the optimal one by comparison.
  - Impractical when $C(n,m) = \frac{n!}{m!(n-m)!}$ is large.

$$n = 1000,\ m = 500 \Longrightarrow C(n,m) > 10^{15}.$$

- Simplex Method
  - Basic Idea:

Step 1: Start at a vertex.
Step 2: If current vertex is optimal, STOP!
Otherwise,
Step 3: Move to a better neighboring vertex.
Go To Step 2.



- Performance:

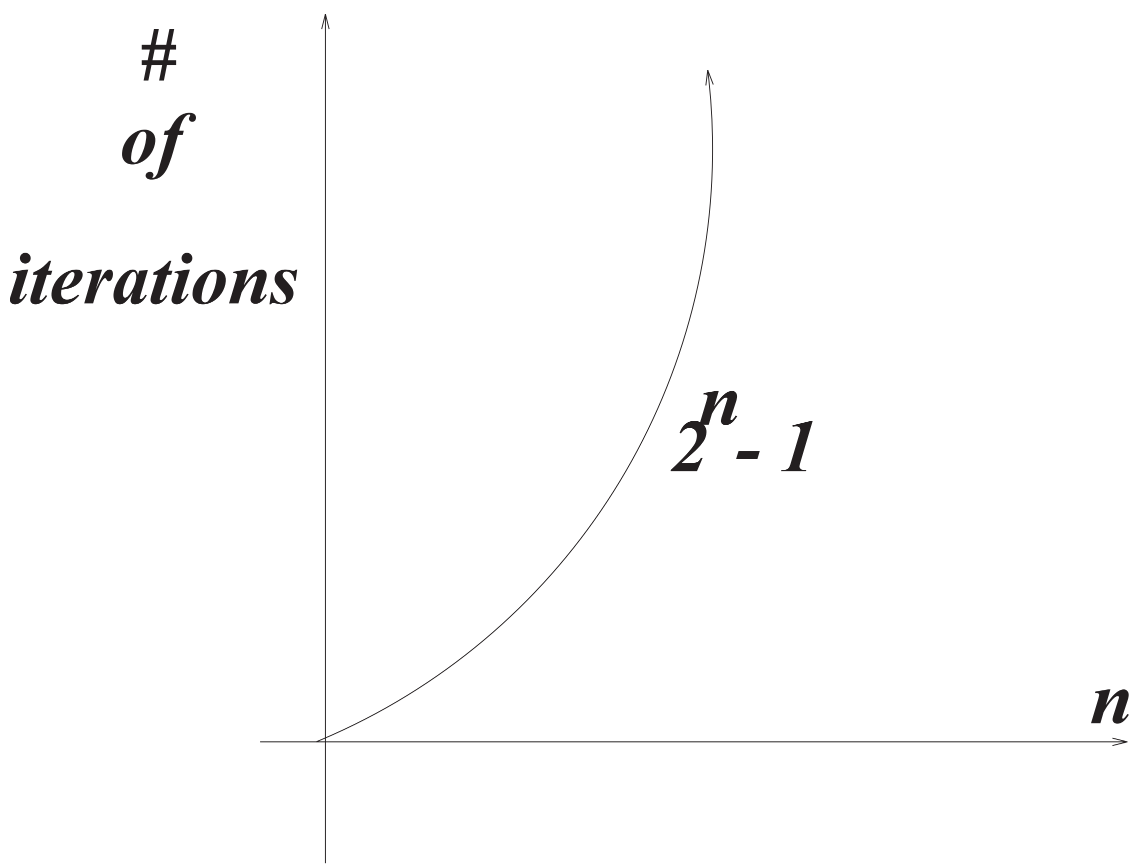
  (1) In general, the simplex method works well, it visits about

  $$0.7159\ m^{0.9522}\ n^{0.3109}$$

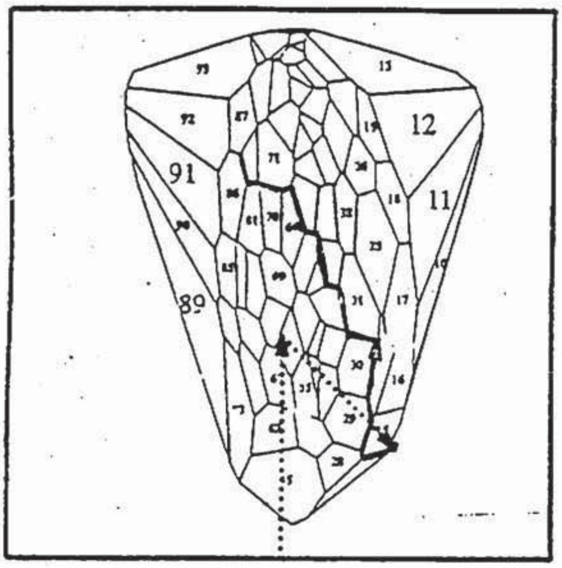
  vertices.

  (2) In the worst case, Klee & Minty (1971) showed that it requires to traverse $2^n - 1$ vertices.

  (3) Problem of exponential-time algorithm.



Picture of Simplex Path



CRAWLING ON THE BOUNDARIES!

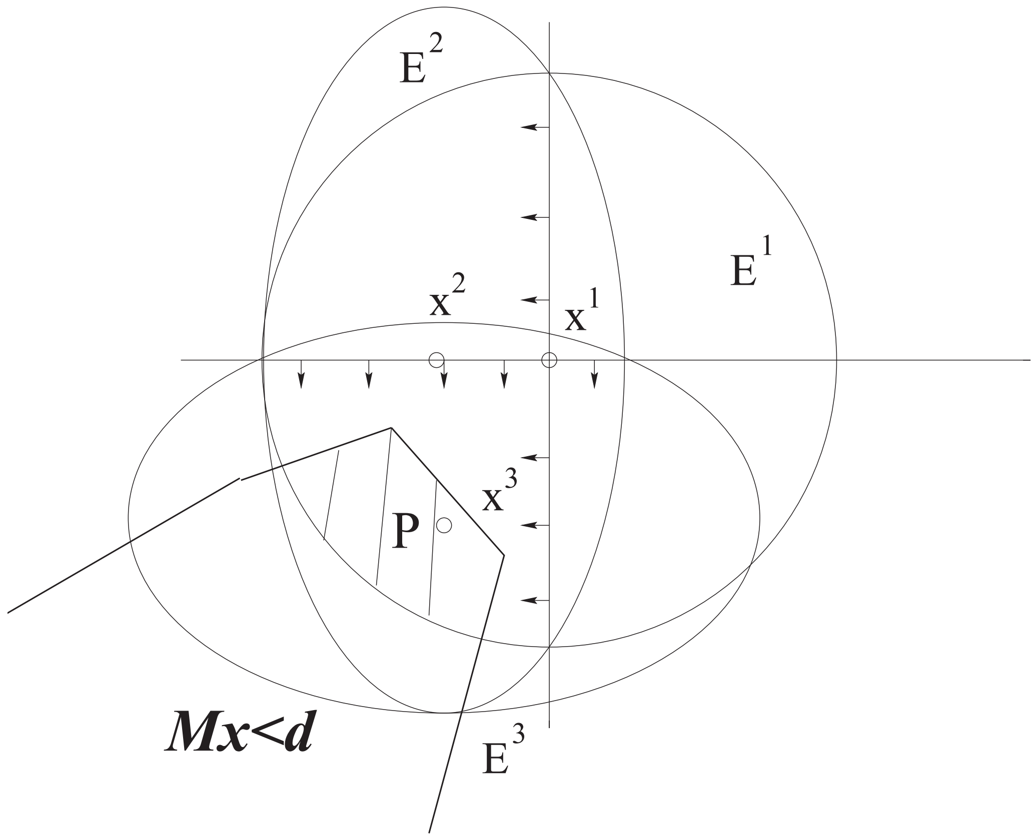
Question: Is there any polynomial-time algorithm for LP?

Answer: Yes!
(Ellipsoid Method by L. G. Khachian 1979)

Ellipsoid Method

Consider solving $\mathbf{Mx} < \mathbf{d}$

$\mathbf{M} : m \times n,\ \mathbf{x} \in R^n,\ \mathbf{d} \in R^m.$



□ Ellipsoid Method

- Basic Idea:

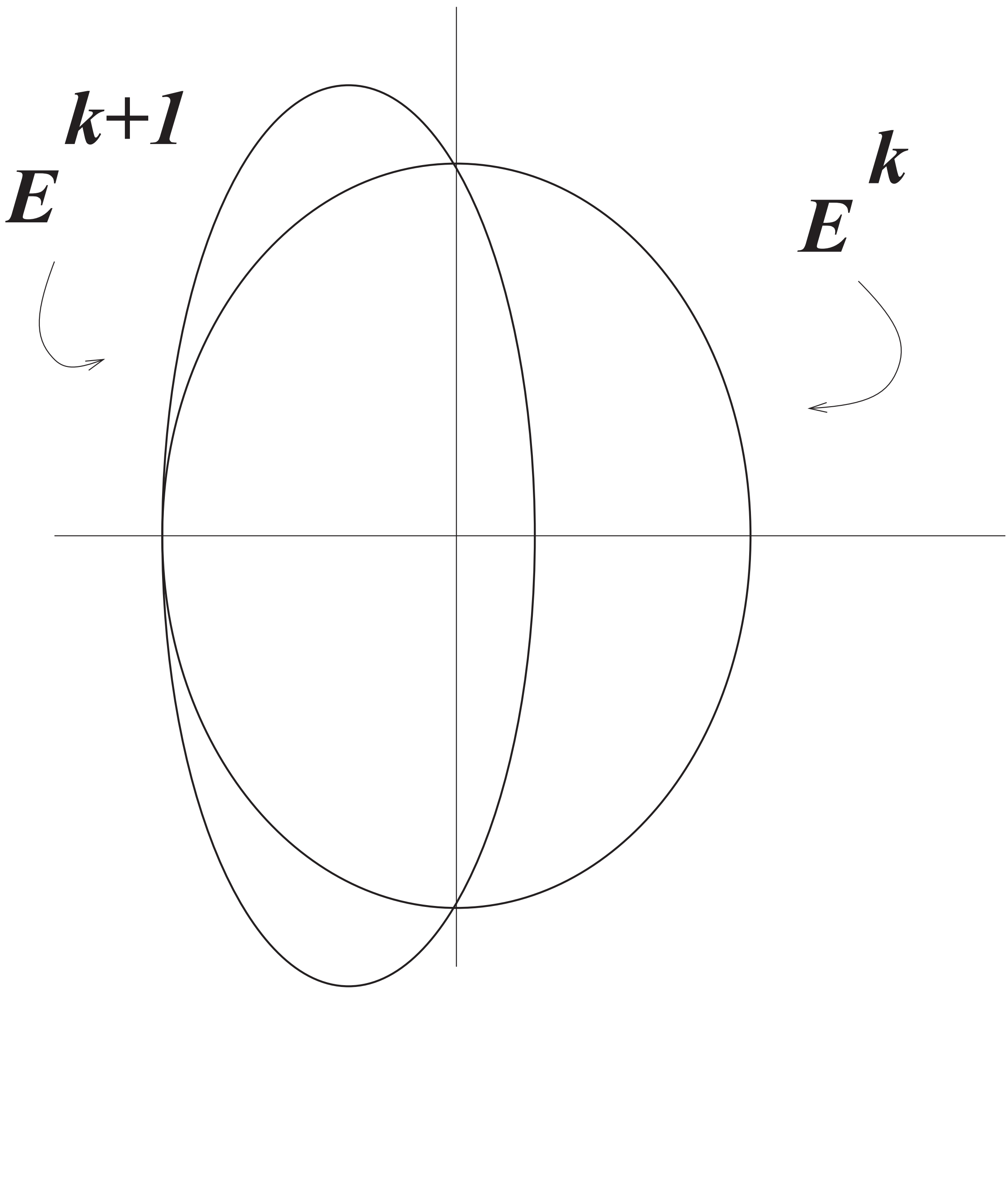
  LP solution is characterized by the optimality conditions

$$(*) \quad \begin{cases} \mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{w} = \mathbf{0} \\ \mathbf{A}\mathbf{x} \le \mathbf{b},\ \mathbf{x} \ge \mathbf{0} \\ \mathbf{A}^T\mathbf{w} \ge \mathbf{c},\ \mathbf{w} \ge \mathbf{0} \end{cases}$$

Step 1: In the $(\mathbf{x}, \mathbf{w})$ space, open a large ellipsoid $E_0 = S(0, 2^{2L})$.

Step 2: If the center of the current ellipsoid E^k solves $(*)$, then STOP. Otherwise, replace E^k by a smaller ellipsoid E^{k+1}.

Step 3: If $Vol(E^{k+1})$ is small enough, STOP! $(*)$
has no solution.



- Performance:

(1) In theory, since

$$\frac{Vol(E^{k+1})}{Vol(E^k)} < e^{-\frac{1}{2}(n+1)}$$

the Ellipsoid Method terminates in polynomial-time.

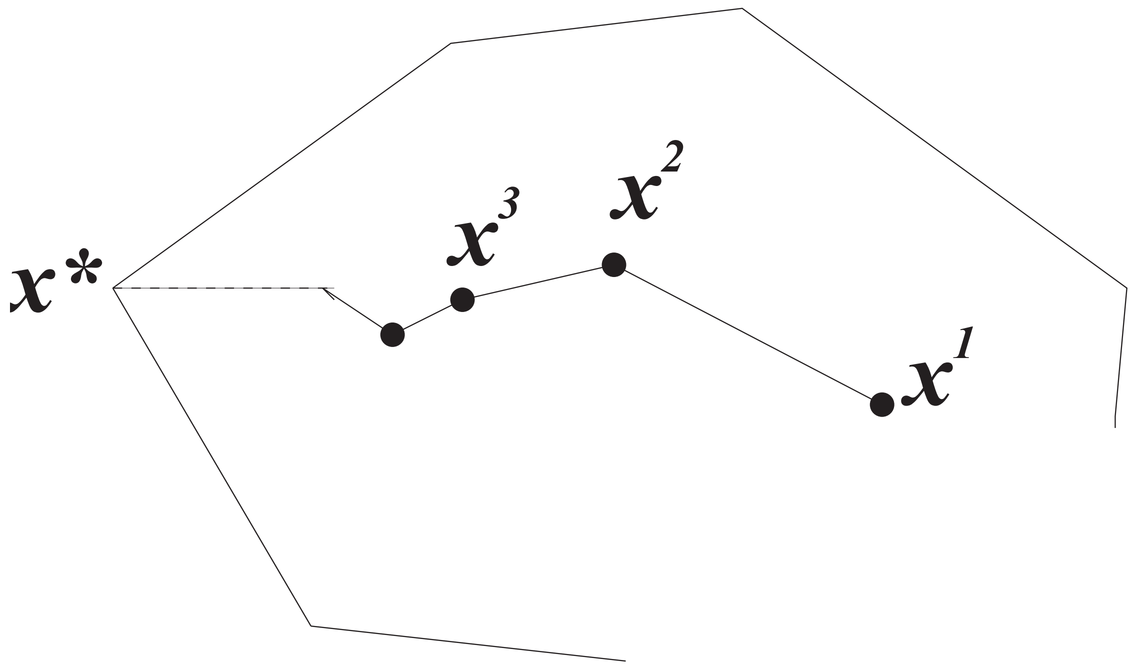
(2) In practice, it is NOT as good as the Simplex Method.
Complexity$= O(n^4L^2)$.

Question: Is there a “Good” polynomial-time algorithm for LP?

Answer: Yes!
(Interior Point Method by Karmarkar in 1984)

□ Interior Point Method

- Basic Idea:



Step 1: Start with an interior solution.

Step 2: If current solution is good enough, STOP.
Otherwise,

Step 3: Check all directions for improvement and
move to a better interior solution.
Go to Step 2.

- Performance:

  (1) It is a polynomial-time algorithm with complexity= $O(n^3L)$.

  (2) It outperformed the simplex method for large size LP.

  (3) 53 Netlib experiments.

□ Interior Movement

- Format:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha \mathbf{d}_{\mathbf{x}}^k$$

$$\begin{cases} \alpha \geq 0 : \text{ Step} - \text{length} \\ \mathbf{d}_{\mathbf{x}}^k \in R^n : \text{ moving direction} \end{cases}$$

- Questions:

1 How to find a “good” direction $\mathbf{d}_{\mathbf{x}}^k$?

2 How far should we move?

□ "Good" Directions

(A) Reduce the objective value

$$\underline{\mathbf{c}^T\mathbf{x}^{k+1}} \qquad \leq \mathbf{c}^T\mathbf{x}^k$$

$$\downarrow$$

$$\mathbf{c}^T\mathbf{x}^k + \alpha\mathbf{c}^T\mathbf{d}_{\mathbf{x}}^k$$

$$\Longrightarrow \mathbf{c}^T\mathbf{d}_{\mathbf{x}}^k \leq 0$$

Candidate: $\mathbf{d}_{\mathbf{x}}^k = -\mathbf{c}$
(negative gradient)
(steepest decent)

(B) Keep feasibility

$$
\begin{array}{ccc}
\underline{\mathbf{A}\mathbf{x}^{k+1}} & = & \mathbf{b} \\
\downarrow & & \\
\mathbf{A}\mathbf{x}^k + \alpha \mathbf{A}\mathbf{d}_{\mathbf{x}}^k & & \\
\| & & \\
\mathbf{b} & &
\end{array}
$$

$$\Longrightarrow \mathbf{A}\mathbf{d}_{\mathbf{x}}^k = 0$$

i.e. $\mathbf{d}_{\mathbf{x}}^k \in \mathcal{N}(\mathbf{A})$ null space of $\mathbf{A}$.

Candidate: Projected negative gradient

$$\mathbf{d}_{\mathbf{x}}^k = (\mathbf{I} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A})(-\mathbf{c}).$$

- Valid Step-length

Fact: As long as $\mathbf{d}_{\mathbf{x}}^k \in \mathcal{N}(\mathbf{A})$,

$\mathbf{A}\mathbf{x}^{k+1} = \mathbf{b}$ no matter the value of α.
However

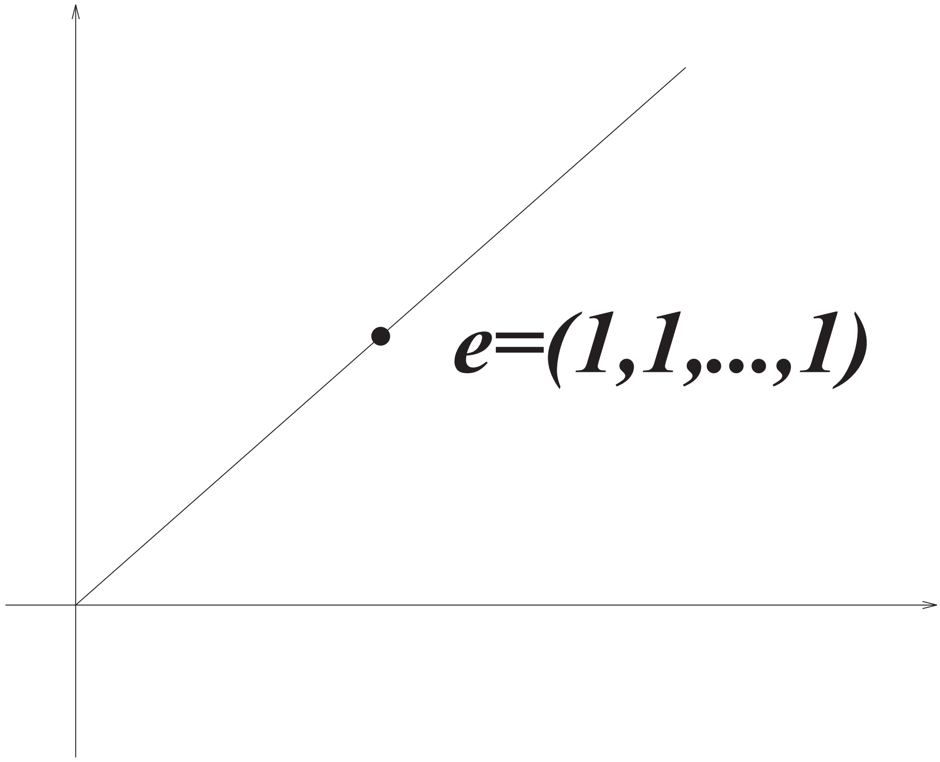
$$\mathbf{x}^{k+1} \geq 0$$

is required!

i.e., We have to know how far $\mathbf{x}^k$ is away from the boundary of non-negative orthant

$$\{\mathbf{x} \in R^n \mid \mathbf{x} \geq 0\}$$

(C) Scaling



If $\mathbf{x}^k = \mathbf{e}$, then

(1) $\mathbf{x}^k$ is one unit away from the boundary.

(2) As long as $\alpha < 1$, $\mathbf{x}^k > 0$.

- <u>Scale $\mathbf{x}^k$ to be $\mathbf{e}$</u>

Define $\mathbf{X}_k = \mathrm{diag}(\mathbf{x}^k) = \begin{pmatrix} x_1^k & & \mathbf{0} & \\ & x_2^k & & \\ & & \ddots & \\ & \mathbf{0} & & x_n^k \end{pmatrix}$

then

$$\mathbf{X}_k^{-1}\mathbf{x}^k = \mathbf{e}.$$

Moreover,

$$\begin{array}{ccc} & \mathbf{X}_k^{-1} & \\ \mathbf{x} & \longrightarrow & \mathbf{y} = \mathbf{X}_k^{-1}\mathbf{x} \\ \boxed{R_+^n} & & \boxed{R_+^n} \\ \mathbf{x} = \mathbf{X}_k\mathbf{y} & \longleftarrow & \mathbf{y} \\ & \mathbf{X}_k & \end{array}$$

$$\begin{cases} \text{one} - \text{one} \\ \text{onto} \\ \text{boundary to boundary} \\ \text{interior to interior} \end{cases}$$

$$\mathbf{x} = \mathbf{X}_k \mathbf{y}$$

Min $\mathbf{c}^T\mathbf{x}$ s. t. $\mathbf{A}\mathbf{x} = \mathbf{b}$ $\mathbf{x} \geq 0$	$\longrightarrow$	Min $\mathbf{c}^T\mathbf{X}_k\mathbf{y}$ s. t. $\mathbf{A}\mathbf{X}_k\mathbf{y} = \mathbf{b}$ $\mathbf{y} \geq 0$
$\mathbf{x}^k \geq 0$		$\mathbf{y}^k = \mathbf{e}$ $\mathbf{y}^{k+1} = \mathbf{y}^k + \alpha_k \frac{\mathbf{d}_\mathbf{y}^k}{\|\mathbf{d}_\mathbf{y}^k\|}$

$$\mathbf{d}_\mathbf{y}^k = [\mathbf{I} - \mathbf{X}_k\mathbf{A}^T(\mathbf{A}\mathbf{X}_k\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{X}_k] \cdot (-\mathbf{X}_k\mathbf{c})$$

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{X}_k\mathbf{y}^{k+1} \\ &= \mathbf{X}_k\mathbf{y}^k + \alpha_k\mathbf{X}_k \frac{\mathbf{d}_\mathbf{y}^k}{\|\mathbf{d}_\mathbf{y}^k\|} \\ &= \mathbf{x}^k + \frac{\alpha_k}{\|\mathbf{d}_\mathbf{y}^k\|}\mathbf{d}_\mathbf{x}^k \end{aligned}$$

$$\mathbf{d}_\mathbf{x}^k = -\mathbf{X}_k[\mathbf{I} - \mathbf{X}_k\mathbf{A}^T(\mathbf{A}\mathbf{X}_k\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{X}_k](\mathbf{X}_k\mathbf{c})$$

$\alpha_k = 0.99$ (say).

□ Variations

- Projective Scaling (N. Karmarkar 1984).
- Primal Affine Scaling
  (Barnes/Vanderkei-Mcketon-Freedman 1985).
- Dual Affine Scaling
  (Adler-Karmarkar-Rosende-Veiga 1986).
- Prima-Dual (Kojima-Mizuno-Yoshise 1987).
- Primal Affine with logarithmic barrier
  (Gonzaga 1989).
- Potential Reduction (Ye 1990).
- Unconstrained Convex Approach (Fang 1990).

  ⋮

□ Current Trends

1 Integration of Interior-point methods to develop hybrid algorithms for solving very large size LP for real-world applications by exploring:
   - special structure.
   - sparsity.
   - decomposition.
   - parallel computation.

2 Nonlinear optimization with linear constraints.

3 Conic Programming.

4 Semi-definite Programming.