# LECTURE 1: INTRODUCTION

Shu-Cherng Fang
North Carolina State University – Fall 2011

# Outline

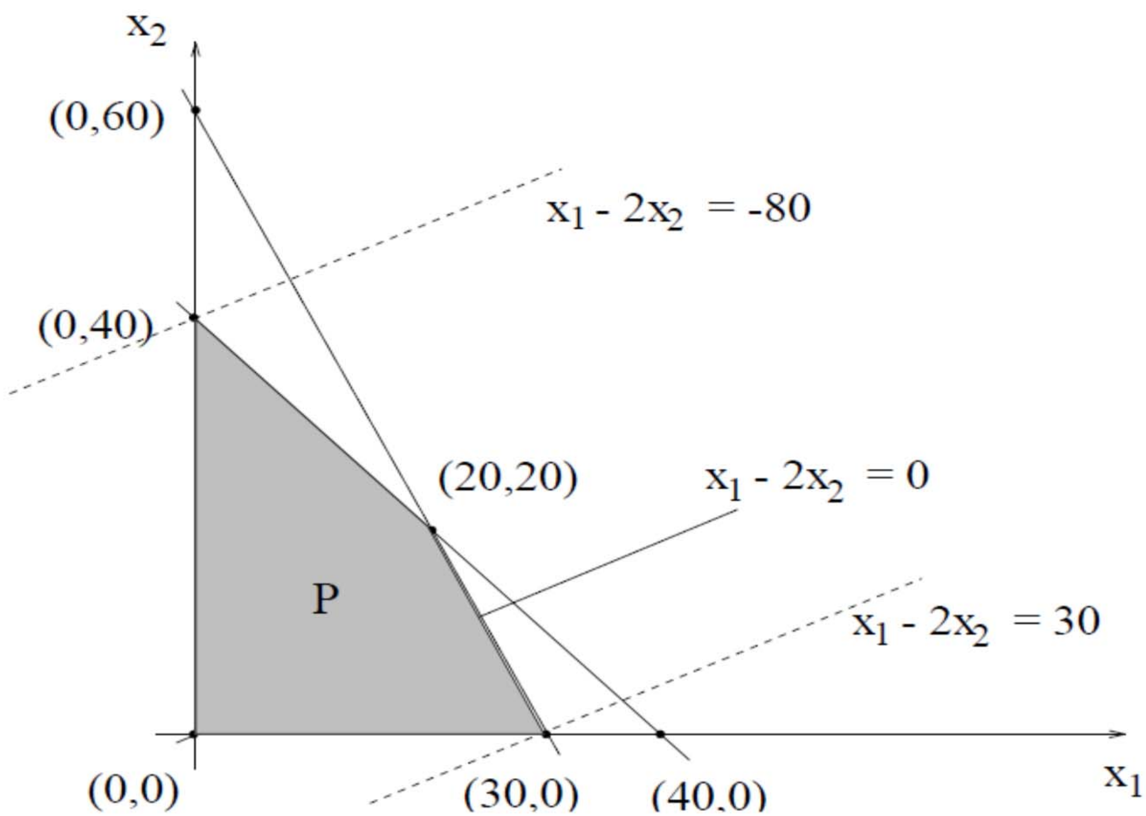
- What is Linear Programming?
- Why to study Linear programming?
- How to study Linear Programming?
- History of Linear Programming
- How to solve an LP problem?
- Where to go?

# What is Linear Programming (LP)?

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

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

# 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

linear objective function

$$\text{Minimize} \quad \underline{c_1x_1 + c_2x_2 + \cdots + c_nx_n}$$

(Maximize)

s. t.

$m$ linear constraints

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

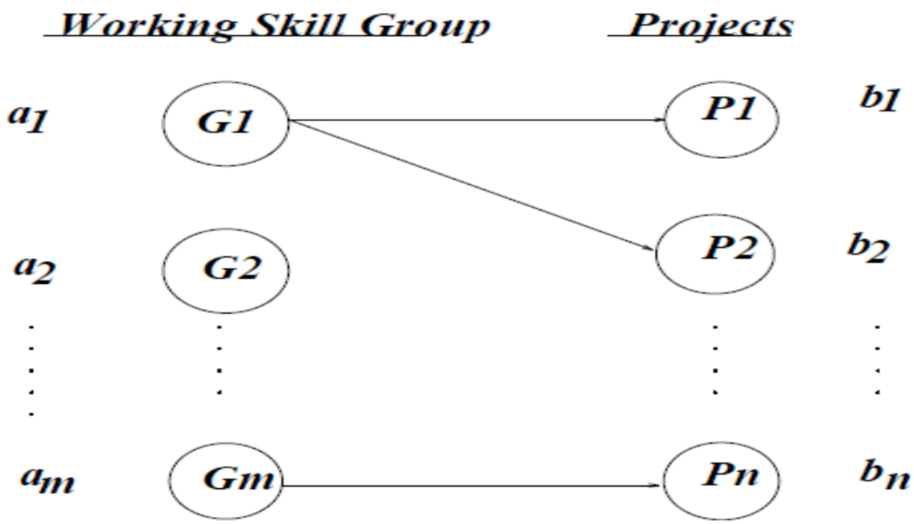
$$\underbrace{x_1, x_2, \ldots, x_n}_{n \text{ decision variables}} \geq 0.$$

# Why study LP ?

- Wide applications:
  - One of the most widely applied methodologies.
  - 85 % of Fortune 500 companies had used LP models.
- Passage to advanced subjects:
  - Nonlinear Programming
  - Network Flows
  - Integer Programming
  - Conic Programming
  - Semidefinite Programming
  - Robust Optimization

Reference: Operations Research, Vol. 50, No. 1, 2002

# Example – manpower allocation




- $x_{ij}$ = man-hours of skill *i* working on project *j*

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

# How to study LP ?

- Geometric intuition
- Algebraic manipulation
- Computer programming

# History of Linear Programming

- 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.

# History of Linear Programming

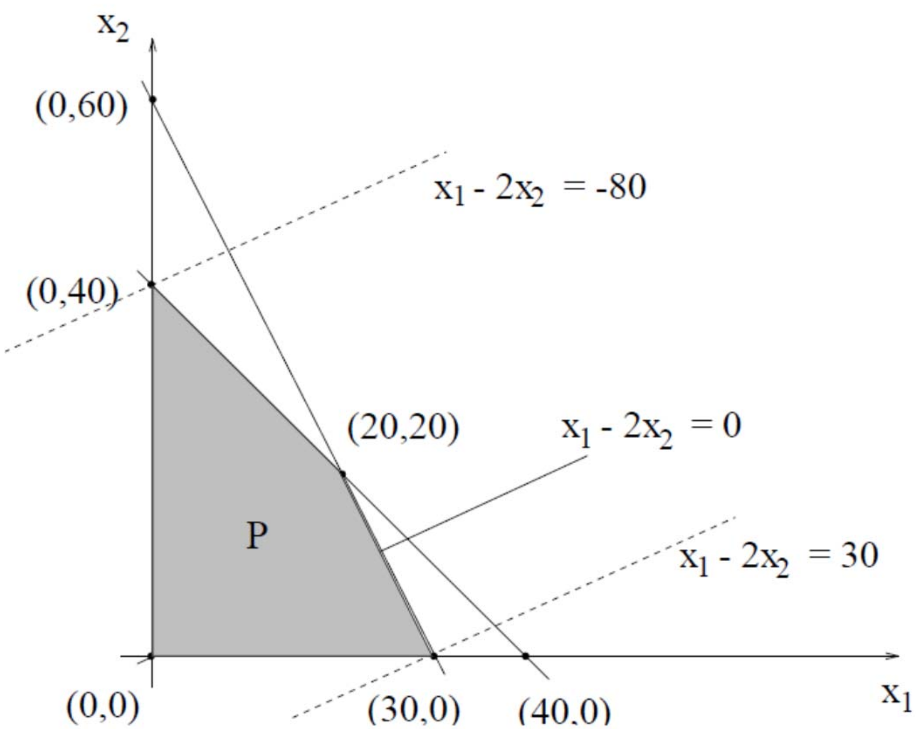
- Ellipsoid Method proposed by L. G. Khachian (1979).
  - First polynomial-time algorithm for LP.
- Interior-Point Method proposed by N. Karmarkar (1984).
  - First “good” polynomial-time algorithm for LP.

# How to solve an LP problem?

- What's special about LP?

  - linear constraints shape the feasible domain as a convex polyhedral set with a finite number of vertices.

  - linear objective function provides a linear contour of each fixed value.

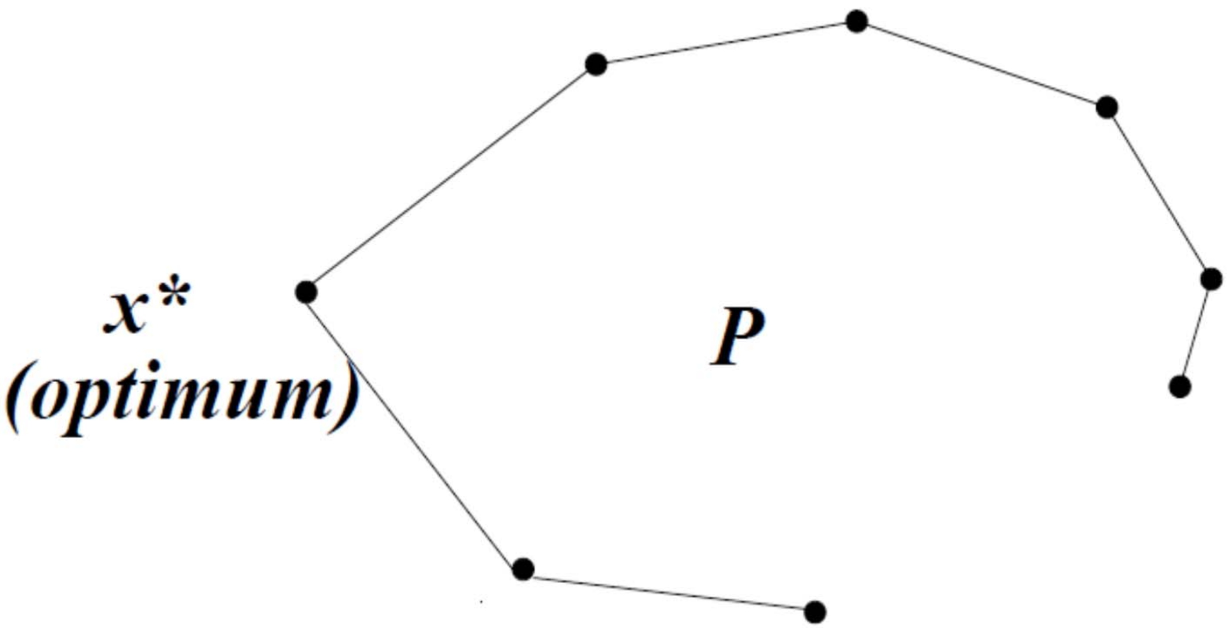


# Fundamental theorem of LP

- For a linear programming problem, if its feasible domain is not empty, then its optimum is either unbounded or is attained at least at one vertex of the feasible domain.

# How to find an optimal solution?

- Fact:




- Question: How to find $x^*$ ?

# Different solution methods

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

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

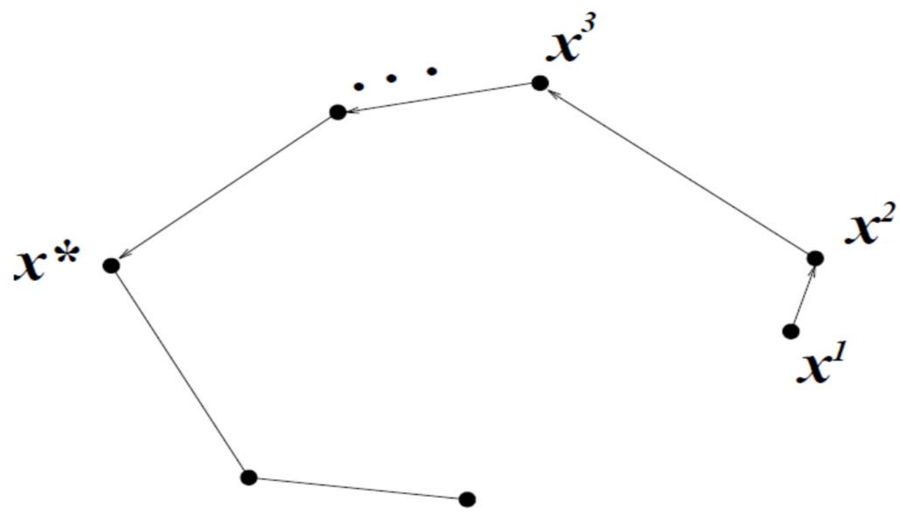
# Simplex Method

- Basic ideas: (walking on the boundary)

  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.



# Is Simplex Method Good?

(1) In general, the simplex method works well,

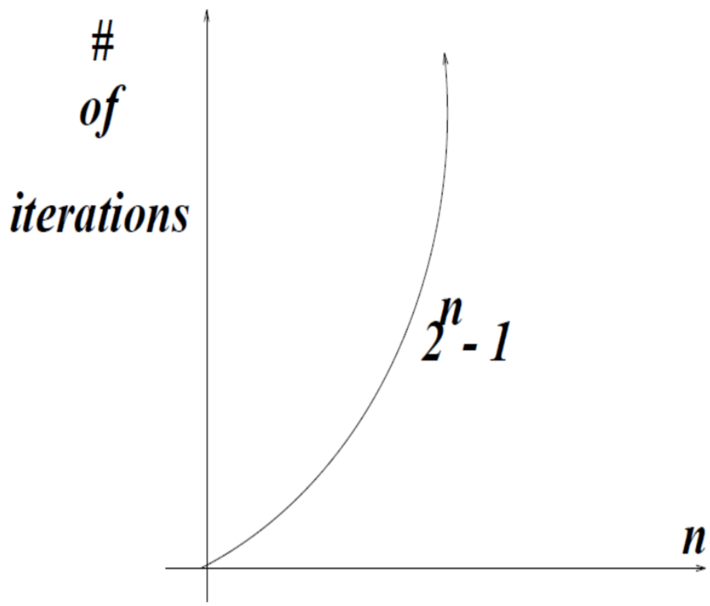
it visits about

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

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.



# Polynomial Time Algorithm

An algorithm runs in polynomial time if the number of steps taken by an algorithm on any instance *I* is bounded by a polynomial in the size of *I*.

An algorithm runs in exponential time if it does not run in polynomial time.

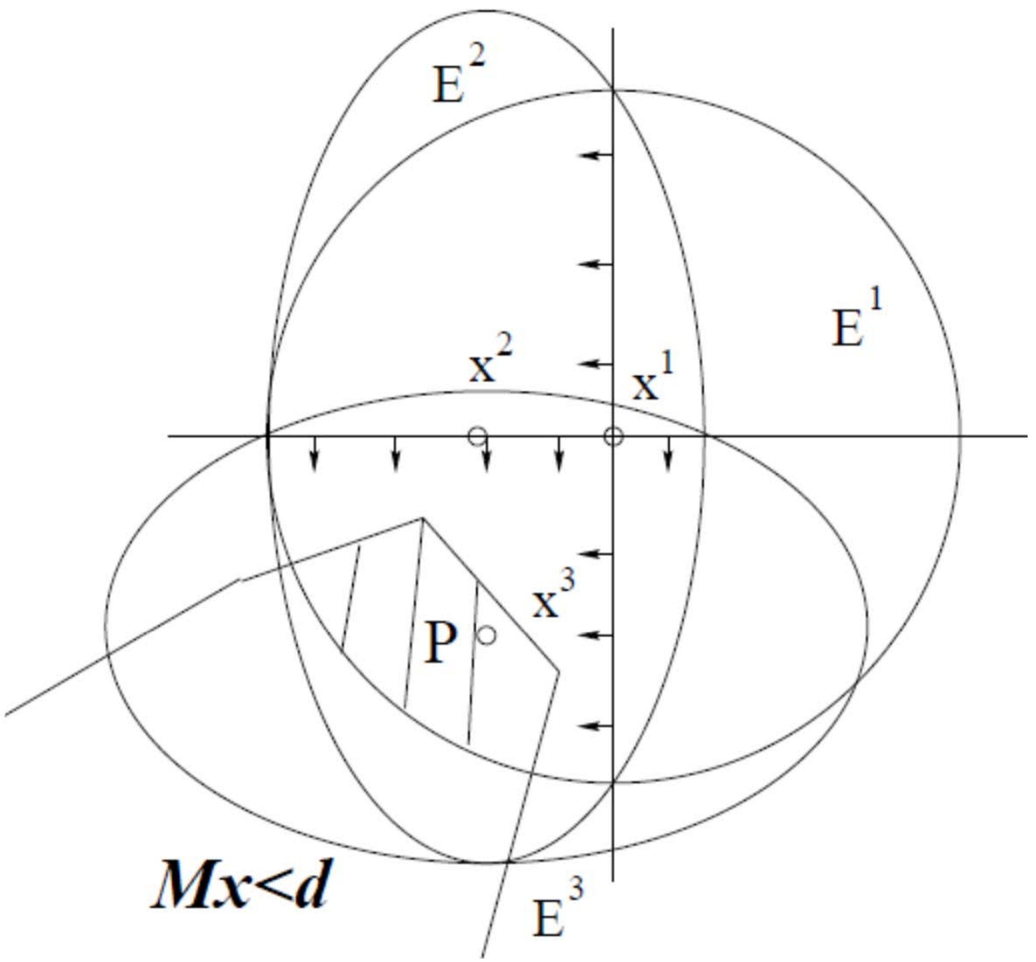
# Is there a polynomial-time algorithm for LP?

- Yes!  - Ellipsoid Method by L. G. Khachian 1979.

Basic ideas:

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

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



# Ellipsoid method

LP solution is characterized by the optimality conditions

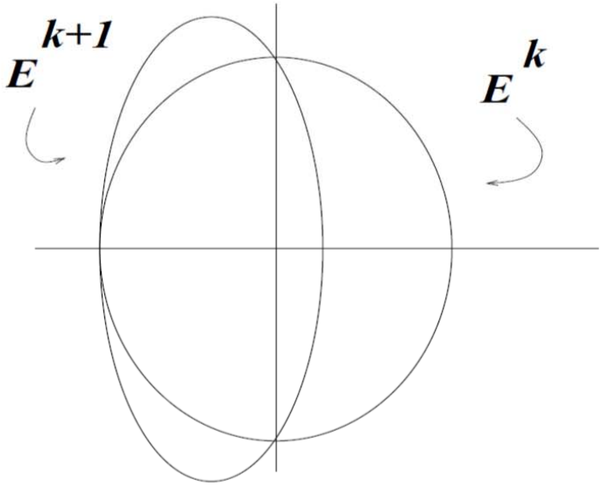
$$(*) \quad \begin{cases} \mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{w} = \mathbf{0} \\ \mathbf{A}\mathbf{x} \le \mathbf{b}, \ \ \mathbf{x} \ge \mathbf{0} \\ \mathbf{A}^T\mathbf{w} \ge \mathbf{c}, \ \ \mathbf{w} \ge \mathbf{0} \end{cases}$$

# Ellipsoid method

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.



# Is ellipsoid method good?

(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$= O(n^4 L^2)$.

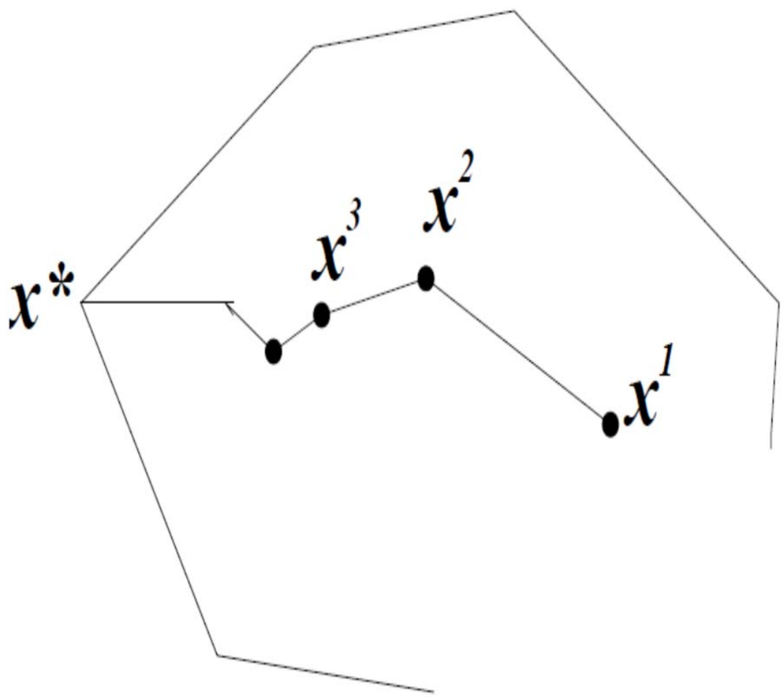
# Is there a good polynomial-time algorithm for LP?

- Yes! - Interior Point Method by Karmarkar in 1984.
- Basic ideas: (walk through interior)

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.



# Is interior point method good?

(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.

# Where to go?

- 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.
- Nonlinear optimization with linear constraints.
- Conic Programming.
- Semidefinite Programming.
- Robust optimization