COT 6936: Topics in Algorithms

Giri Narasimhan

ECS 254A / EC 2443; Phone: x3748

giri@cs.fiu.edu

https://moodle.cis.fiu.edu/v2.1/course/view.php?id=**612**

Gaussian Elimination

- Solving a system of simultaneous equations

$$
\begin{aligned}
x_1 \quad\quad - 2x_3 \quad\quad &= 2 \\
x_2 + x_3 \quad\quad &= 3 \\
x_1 + x_2 \quad\quad - x_4 &= 4 \\
x_2 + 3x_3 + x_4 &= 5
\end{aligned}
$$

O(n³) algorithm

$$
\begin{aligned}
x_1 \quad\quad - 2x_3 \quad\quad &= 2 \\
x_2 + x_3 \quad\quad &= 3 \\
x_2 + 2x_3 - x_4 &= 2 \\
x_2 + 3x_3 + x_4 &= 5
\end{aligned}
$$

Linear Programming

- Want more than solving simultaneous equations
- We have an objective function to optimize

Chocolate Shop [DPV book]

- 2 kinds of chocolate
  - milk [Profit: $1 per box] [Demand: 200]
  - Deluxe [Profit: $6 per box] [Demand: 300]
- Production capacity: 400 boxes
- Goal: maximize profit
  - Maximize $x_1 + 6x_2$ subject to constraints:
    - $x_1 \le 200$
    - $x_2 \le 300$
    - $x_1 + x_2 \le 400$
    - $x_1, x_2 \ge 0$

Diet Problem

- Food type: $F_1, \ldots, F_m$
- Nutrients: $N_1, \ldots, N_n$
- Min daily requirement of nutrients: $c_1, \ldots, c_n$
- Price per unit of food: $b_1, \ldots, b_m$
- Nutrient N_j in food F_i: a_{ij}
- Problem: Supply daily nutrients at minimum cost
    - Min $\Sigma_i\, b_i x_i$
    - $\Sigma_i\, a_{ij} x_i \geq c_j$ for $1 \leq j \leq n$
    - $x_i \geq 0$

Transportation Problem

- Ports or Production Units: $P_1, \ldots, P_m$
- Markets to be shipped to: $M_1, \ldots, M_n$
- Min daily market need: $r_1, \ldots, r_n$
- Port/production capacity: $s_1, \ldots, s_m$
- Cost of transporting to M_j from port P_i: a_{ij}
- Problem: Meet market need at minimum transportation cost

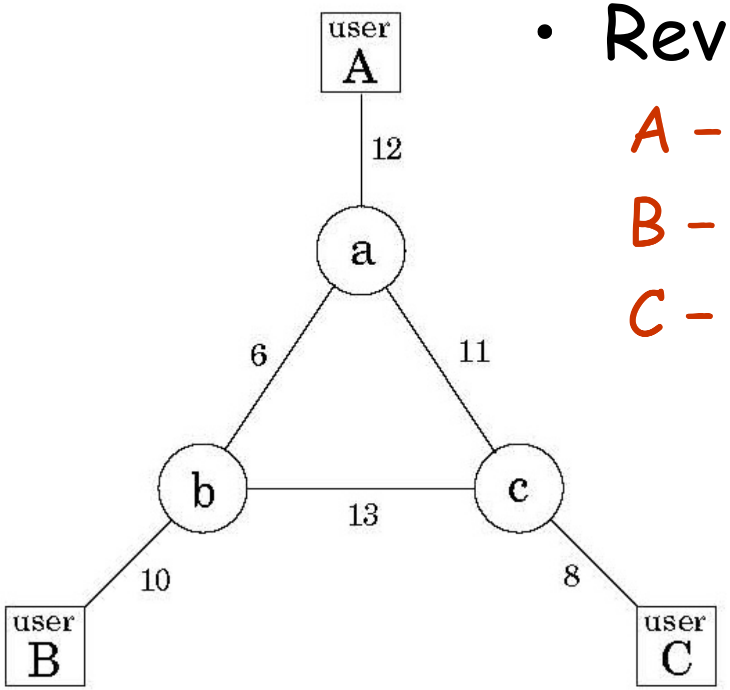
Assignment Problem

- Workers: $b_1,\ldots,b_n$
- Jobs: $g_1,\ldots,g_m$
- Value of assigning person b_i to job g_j: a_{ij}
- Problem: Choose job assignment to maximize value

Bandwidth Allocation Problem

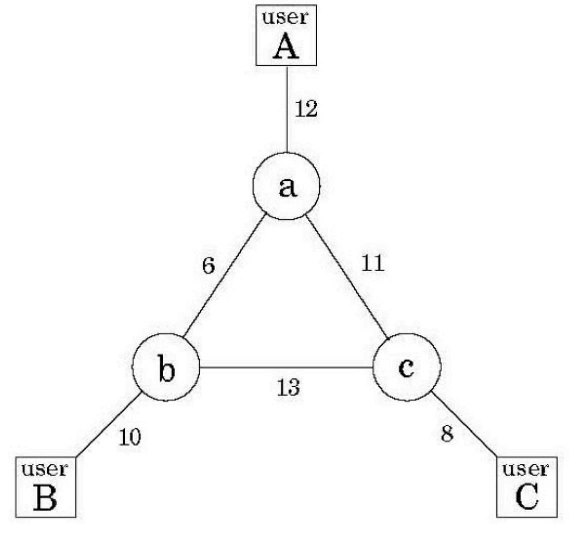
Figure 7.3 A communications network between three users A, B, and C. Bandwidths are shown.

- Need:
  - A – B ≥ 2 units
  - B – C ≥ 2 units
  - C – A ≥ 2 units
- Connections:
  - Short route
  - Long route



- Revenue:
  - A – B pays $3 per unit
  - B – C pays $2 per unit
  - C – A pays $4 per unit

Bandwidth Allocation Pr



- Maximize revenue by allocating connections along two routes wi exceeding bandwidth capacities
- Max $3(x_{AB}+x_{AB}') + 2(x_{BC}+x_{BC}') + 4(x_{AC}+x_{AC}')$ s.t.

$x_{AB} + x_{AB}' + x_{BC} + x_{BC}' \leq 10$

$x_{AB} + x_{AB}' + x_{AC} + x_{AC}' \leq 12$

$x_{BC} + x_{BC}' + x_{AC} + x_{AC}' \leq 8$

$x_{AB} + x_{BC}' + x_{AC}' \leq 6$; $x_{AB} + x_{AB}' \geq 2$; $x_{BC} + x_{BC}' \geq 2$

$x_{AB}' + x_{BC} + x_{AC}' \leq 13$; $x_{AC} + x_{AC}' \geq 2$

$x_{AB}' + x_{BC}' + x_{AC} \leq 11$; & all nonneg constraints

Standard LP

- Maximize $\Sigma c_j x_j$ [Objective Function]

  Subject to $\Sigma a_{ij} x_j \leq b_j$ [Constraints]

  and $x_j \geq 0$ [Nonnegativity Constraints]

- Matrix formulation of LP

  Maximize $c^T x$

  Subject to $Ax \leq b$

  and $x \geq 0$

Converting to standard form

- Min $-2x_1 + 3x_2$ Subject to

  $x_1 + x_2 = 7$

  $x_1 - 2x_2 \leq 4$

  $x_1 \geq 0$

- Max $2x_1 - 3x_2$ Subject to

  $x_1 + x_2 \leq 7$

  $-x_1 - x_2 \leq -7$

  $x_1 - 2x_2 \leq 4$

  $x_1 \geq 0$

Converting to standard form

- Max $2x_1 - 3x_2$ Subject to

  $x_1 + x_2 \leq 7$

  $-x_1 - x_2 \leq -7$

  $x_1 - 2x_2 \leq 4$

  $x_1 \geq 0$

x_2 is not constrained to be non-negative

- Max $2x_1 - 3(x_3 - x_4)$ Subject to

  $x_1 + x_3 - x_4 \leq 7$

  $-x_1 - (x_3 - x_4) \leq -7$

  $x_1 - 2(x_3 - x_4) \leq 4$

  $x_1, x_3, x_4 \geq 0$

Converting to Standard form

- Max $2x_1 - 3x_2 + 3x_3$ Subject to

  $x_1 + x_2 - x_3 \leq 7$

  $-x_1 - x_2 + x_3 \leq -7$

  $x_1 - 2x_2 - 2x_3 \leq 4$

  $x_1, x_2, x_3 \geq 0$

Slack Form

- Max $2x_1 - 3x_2 + 3x_3$ Subject to

  $x_1 + x_2 - x_3 \leq 7$

  $-x_1 - x_2 + x_3 \leq -7$

  $x_1 - 2x_2 - 2x_3 \leq 4$

  $x_1, x_2, x_3 \geq 0$

- Max $2x_1 - 3x_2 + 3x_3$ Subject to

  $x_1 + x_2 - x_3 + x_4 = 7$

  $-x_1 - x_2 + x_3 + x_5 = -7$

  $x_1 - 2x_2 - 2x_3 + x_6 = 4$

  $x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$

Duality

- Max $c^T x$ [Primal]
  Subject to $Ax \leq b$
  and $x \geq 0$

- Min $y^T b$ [Dual]
  Subject to $y^T A \geq c$
  and $y \geq 0$

Understanding Duality

- Maximize $x_1 + 6x_2$ subject to constraints:
  - $x_1 \leq 200$ (1)
  - $x_2 \leq 300$ (2)
  - $x_1 + x_2 \leq 400$ (3)
  - $x_1, x_2 \geq 0$

How were mutipliers determined?

- (100,300) is feasible; value = 1900. Optimum?
- Adding 1 times (1) + 6 times (2) gives us
  - $x_1 + 6x_2 \leq 2000$
- Adding 1 times (3) + 5 times (2) gives us
  - $x_1 + 6x_2 \leq 1900$
  - "Certificate of Optimality" for solution (100,300)

Understanding Duality

- Maximize $x_1 + 6x_2$ subject to:
  - $x_1 \le 200$ (y_1)
  - $x_2 \le 300$ (y_2) [(100,300)]
  - $x_1 + x_2 \le 400$ (y_3)
  - $x_1, x_2 \ge 0$
- Different choice of multipliers gives us different bounds. We want smallest bound.
- Minimize $200y_1 + 300y_2 + 400y_3$ subject to:
  - $y_1 + y_3 \ge 1$ (x_1)
  - $y_2 + y_3 \ge 6$ (x_2) [(0,5,1)]
  - $y_1, y_2 \ge 0$

Duality Principle

- Primal feasible values ≤ dual feasible values
- Max primal value = min dual value
- Duality Theorem: If a linear program has a bounded optimal value then so does its dual and the two optimal values are equal.

Shortest Path Problem as a LP

- Graph $G = (V,E)$;
  - Vertices: $v_1,\ldots,v_n$; Edges: $e_1,\ldots,e_m$;
  - Weight function on edges $w(e_i)$; Source s; Dest t;
- LP: min $w^T x$
  - s.t. $A\, x = b$ and $x \geq 0$
- Here A and b are defined as follows:
  - $A_{ij} = +1$ if e_j leaves v_i;
  - $= -1$ if e_j enters v_i;
  - $= 0$ otherwise;

  $b_s = +1$
  $b_t = -1$
  $b_i = 0$ else;
- We want integral solutions for x

Dual LP

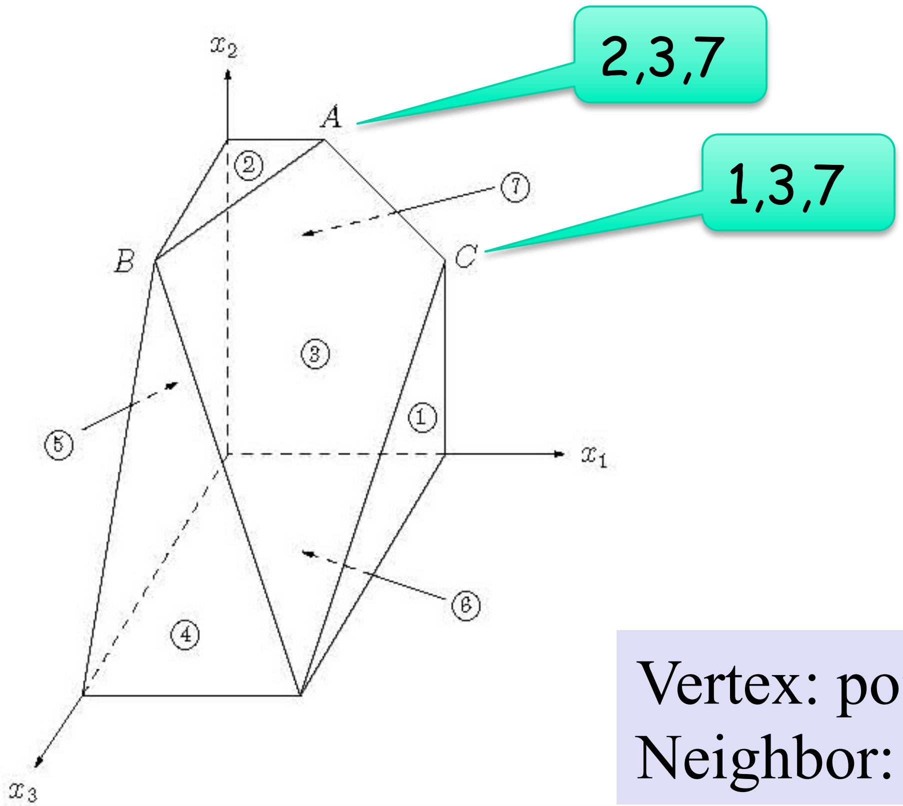
- LP: min $w^T x$
  - s.t. $A x = b$ and $x \geq 0$
- Dual LP: max $y_s - y_t$
  - s.t. $|y_u - y_v| \leq w(e)$ for every edge $e = (u,v)$

Visualizing Duality

- Shortest Path Problem
  - Build a physical model and between each pair of vertices attach a string of appropriate length
  - To find shortest path from s to t, hold the two vertices and pull them apart as much as possible without breaking the strings
  - This is exactly what a dual LP solves!
    - Max $x_s - x_t$
    - subject to $|x_u - x_v| \leq w_{uv}$ for every edge $(u.v)$
  - The taut strings correspond to the shortest path, i.e., they have no slack

Linear Constraints: Geometric View

Figure 7.12 A polyhedron defined by seven inequalities.



$$\max \quad x_1 + 6x_2 + 13x_3$$

$$\begin{aligned} x_1 &\le 200 && (1)\\ x_2 &\le 300 && (2)\\ x_1 + x_2 + x_3 &\le 400 && (3)\\ x_2 + 3x_3 &\le 600 && (4)\\ x_1 &\ge 0 && (5)\\ x_2 &\ge 0 && (6)\\ x_3 &\ge 0 && (7) \end{aligned}$$

i.e., some inequalities satisfied as equalities

Vertex: point where n hyperplanes meet;
Neighbor: vertices sharing n-1 hyperplanes

Simplex Algorithm

- Start at v, any vertex of feasible region
- while (there is neighbor v′ of v with better objective value) do

  set v = v′

- Report v as optimal point and its value as optimal value

- What is a
  - Vertex?, neighbor?
- Start vertex? How to pick next neighbor?

Steps of Simplex Algorithm

- In order to find next neighbor from arbitrary vertex, we do a change of origin (pivot)

Initial LP:

$$\max\ 2x_1 + 5x_2$$

$$\begin{array}{rcll} 2x_1 - x_2 & \leq & 4 & ① \\ x_1 + 2x_2 & \leq & 9 & ② \\ -x_1 + x_2 & \leq & 3 & ③ \\ x_1 & \geq & 0 & ④ \\ x_2 & \geq & 0 & ⑤ \end{array}$$

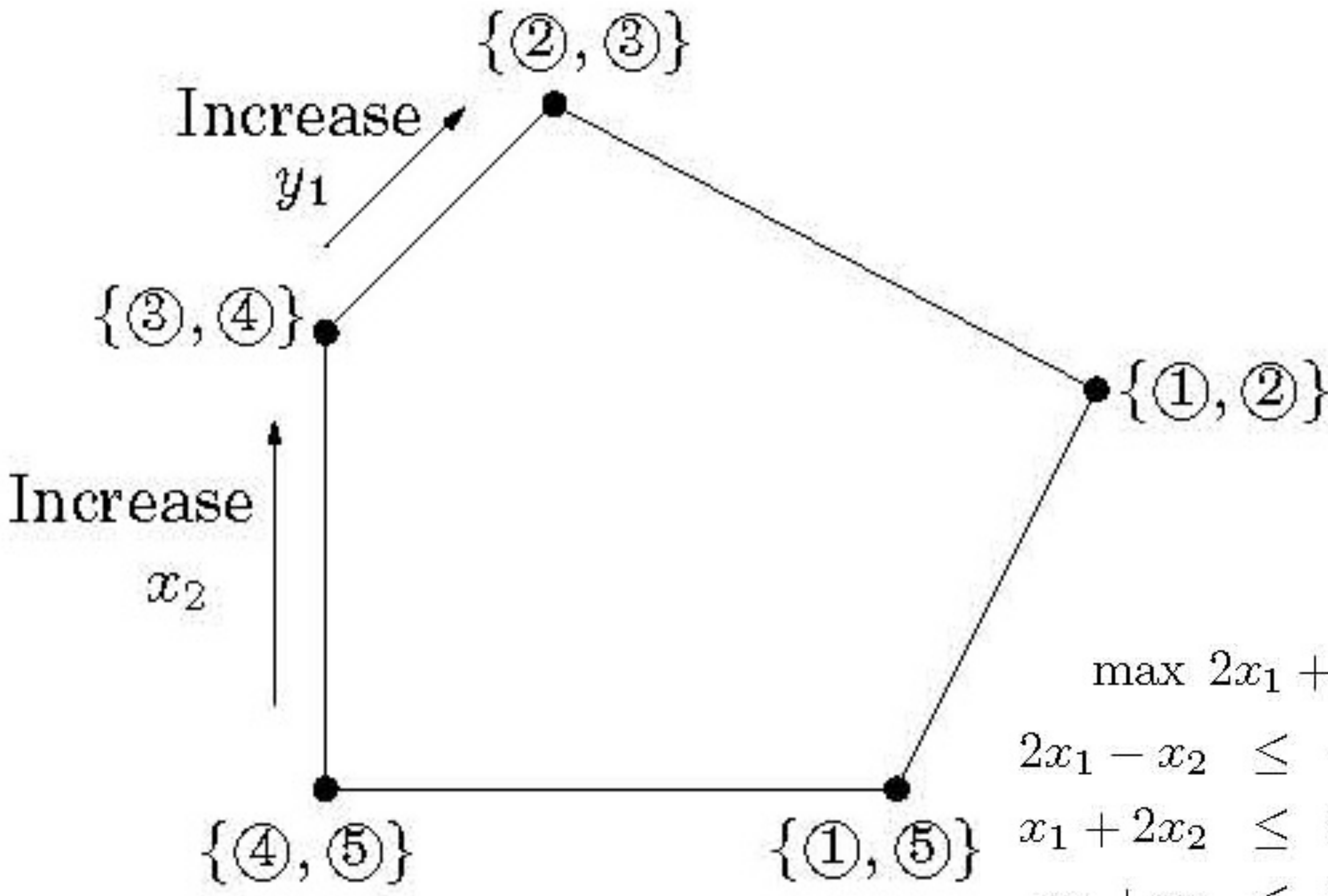
Current vertex: {④, ⑤} (origin).
Objective value: 0.

Move: increase x_2.
⑤ is released, ③ becomes tight. Stop at $x_2 = 3$.

New vertex {④, ③} has local coordinates (y_1, y_2):

$$y_1 = x_1, \quad y_2 = 3 + x_1 - x_2$$

Simplex Algorithm Example



$$
\begin{aligned}
&\max \ 2x_1 + 5x_2 & & \\
&2x_1 - x_2 \leq 4 & & ① \\
&x_1 + 2x_2 \leq 9 & & ② \\
&-x_1 + x_2 \leq 3 & & ③ \\
&x_1 \geq 0 & & ④ \\
&x_2 \geq 0 & & ⑤
\end{aligned}
$$

Simplex Algorithm Example

Initial LP:

$$\max\ 2x_1 + 5x_2$$

$$\begin{array}{rcll} 2x_1 - x_2 & \leq & 4 & \text{①} \\ x_1 + 2x_2 & \leq & 9 & \text{②} \\ -x_1 + x_2 & \leq & 3 & \text{③} \\ x_1 & \geq & 0 & \text{④} \\ x_2 & \geq & 0 & \text{⑤} \end{array}$$

Current vertex: {④, ⑤} (origin).
Objective value: 0.

Move: increase x_2.
⑤ is released, ③ becomes tight. Stop at $x_2 = 3$.

New vertex {④, ③} has local coordinates (y_1, y_2):

$$y_1 = x_1, \quad y_2 = 3 + x_1 - x_2$$

Rewritten LP:

$$\max\ 15 + 7y_1 - 5y_2$$

$$\begin{array}{rcll} y_1 + y_2 & \leq & 7 & \text{①} \\ 3y_1 - 2y_2 & \leq & 3 & \text{②} \\ y_2 & \geq & 0 & \text{③} \\ y_1 & \geq & 0 & \text{④} \\ -y_1 + y_2 & \leq & 3 & \text{⑤} \end{array}$$

Current vertex: {④, ③}.
Objective value: 15.

Move: increase y_1.
④ is released, ② becomes tight. Stop at $y_1 = 1$.

New vertex {②, ③} has local coordinates (z_1, z_2):

$$z_1 = 3 - 3y_1 + 2y_2, \quad z_2 = y_2$$

Simplex Algorithm Example

Rewritten LP:

$$\max\ 15 + 7y_1 - 5y_2$$

$$\begin{array}{rcll} y_1 + y_2 & \leq & 7 & ① \\ 3y_1 - 2y_2 & \leq & 3 & ② \\ y_2 & \geq & 0 & ③ \\ y_1 & \geq & 0 & ④ \\ -y_1 + y_2 & \leq & 3 & ⑤ \end{array}$$

Current vertex: {④, ③}.
Objective value: 15.

Move: increase y_1.
④ is released, ② becomes tight. Stop at $y_1 = 1$.

New vertex {②, ③} has local coordinates (z_1, z_2):

$$z_1 = 3 - 3y_1 + 2y_2, \quad z_2 = y_2$$

Rewritten LP:

$$\max\ 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$$

$$\begin{array}{rcll} -\frac{1}{3}z_1 + \frac{5}{3}z_2 & \leq & 6 & ① \\ z_1 & \geq & 0 & ② \\ z_2 & \geq & 0 & ③ \\ \frac{1}{3}z_1 - \frac{2}{3}z_2 & \leq & 1 & ④ \\ \frac{1}{3}z_1 + \frac{1}{3}z_2 & \leq & 4 & ⑤ \end{array}$$

Current vertex: {②, ③}.
Objective value: 22.

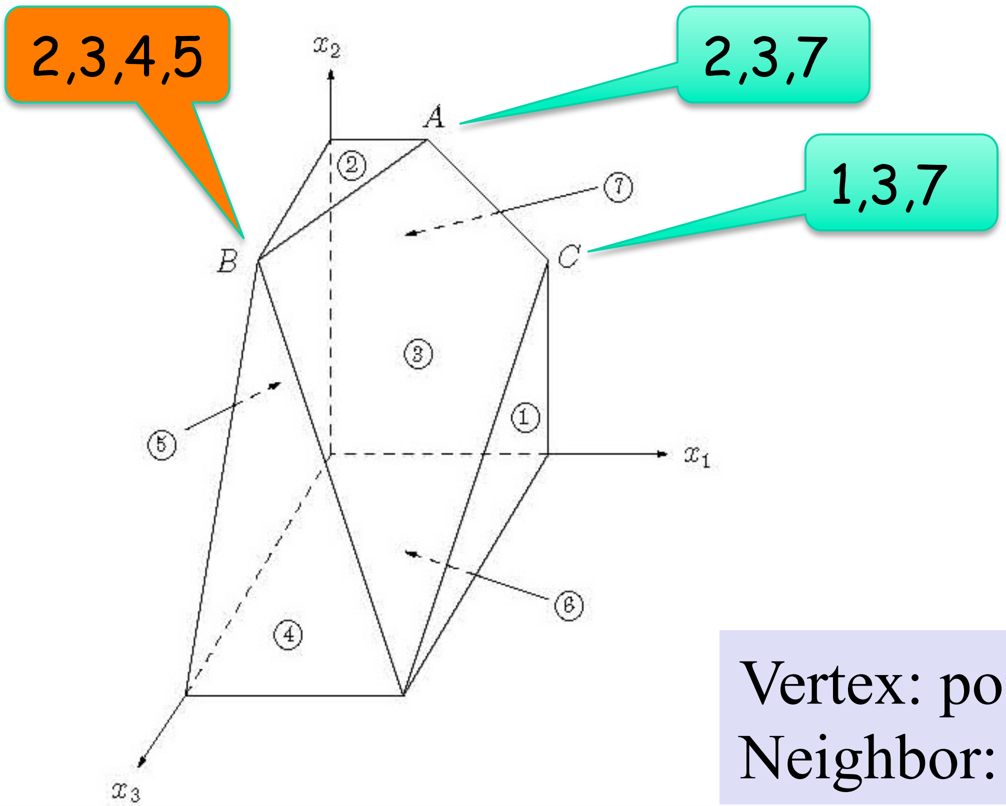
Optimal: all $c_i < 0$.

Solve ②, ③ (in original LP) to get optimal solution $(x_1, x_2) = (1, 4)$.

Simplex Algorithm: Degenerate vertices

i.e., some inequalities satisfied as equalities

Figure 7.12 A polyhedron defined by seven inequalities.



$$
\begin{aligned}
\max \quad & x_1 + 6x_2 + 13x_3 & \\
& x_1 \le 200 & ① \\
& x_2 \le 300 & ② \\
& x_1 + x_2 + x_3 \le 400 & ③ \\
& x_2 + 3x_3 \le 600 & ④ \\
& x_1 \ge 0 & ⑤ \\
& x_2 \ge 0 & ⑥ \\
& x_3 \ge 0 & ⑦
\end{aligned}
$$

Vertex: point where n hyperplanes meet;
Neighbor: vertices sharing n-1 hyperplanes

Polynomial-time algorithms for LP

- Simplex is not poly-time in the worst-case
- Khachiyan's ellipsoid algorithm: LP is in $\underline{\mathcal{P}}$
- Karmarkar's interior-point algorithm
- Good implementations for LP exist
  - Works very well in practice
  - More competitive than the poly-time methods for LP

Integer Linear Programming

- LP with integral solutions
- NP-hard
- If A is a totally unimodular matrix, then the LP solution is always integral.
  - A TUM is a matrix for which every nonsingular submatrix has determinant 0, +1 or -1.
  - A TUM is a matrix for which every nonsingular submatrix has integral inverse.

Vertex Cover as an LP?

- For vertex v, create variable x_v
- For edge (u,v), create constraint $x_u + x_v \geq 1$
- Objective function: Σx_v
- Additional constraints: $x_v \leq 1$

- Doesn't work because x_v needs to be from {0,1}

Integer Linear Programming

- LP with integral solutions
- NP-hard
- If A is a totally unimodular matrix, then the LP solution is always integral.
  - A TUM is a matrix for which every nonsingular submatrix has determinant 0, +1 or -1.
  - A TUM is a matrix for which every nonsingular submatrix has integral inverse.

Vertex Cover as an LP?

- For vertex v, create variable x_v
- For edge (u,v), create constraint $x_u + x_v \geq 1$
- Objective function: Σx_v
- Additional constraints: $x_v \leq 1$
- Doesn't work because x_v needs to be from {0,1}

Set Cover

- Given a universe of items $U = \{e_1, \ldots, e_n\}$ and a collection of subsets $S = \{S_1, \ldots, S_m\}$ such that each S_i is contained in U
- Find the minimum set of subsets from S that will cover all items in U (i.e., the union of these subsets must equal U)
- Weighted Set Cover: Given universe U and collection S, and a cost $c(S_i)$ for each subset S_i in S, find the minimum cost set cover

The Greedy Set Cover Algorithm

The Integer Linear Program (ILP)

$$\begin{aligned}
\min \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\
\text{subject to} \quad & \sum_{S : e \in S} x_S \geq 1, \quad e \in U \\
& x_S \in \{0,1\}, \quad S \in \mathcal{S}
\end{aligned}$$

The LP Relaxation

$$\begin{aligned}
\min \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\
\text{subject to} \quad & \sum_{S : e \in S} x_S \geq 1, \quad e \in U \\
& x_S \geq 0, \qquad S \in \mathcal{S}
\end{aligned}$$

The Dual LP

$$\begin{aligned}
\max \quad & \sum_{e \in U} y_e \\
\text{subject to} \quad & \sum_{e : e \in S} y_e \leq c(S), \quad S \in \mathcal{S} \\
& y_e \geq 0, \qquad e \in U
\end{aligned}$$

Fractional LP may have higher objective function value than integer LP

- $U = \{e, f, g\}$
- $S_1 = \{e, f\}$
- $S_2 = \{f, g\}$
- $S_3 = \{e, g\}$
- Optimal set cover = $\{S_1, S_2\}$
- Fractional optimal set cover assigns $\frac{1}{2}$ to each of three sets giving a total optimal value of 3/2.

The LP Relaxation

$$\begin{aligned} \min \quad & \sum_{S \in \mathcal{S}} c(S) x_S \\ \text{subject to} \quad & \sum_{S: e \in S} x_S \geq 1, \quad e \in U \\ & x_S \geq 0, \qquad S \in \mathcal{S} \end{aligned}$$

The Dual LP Relaxation

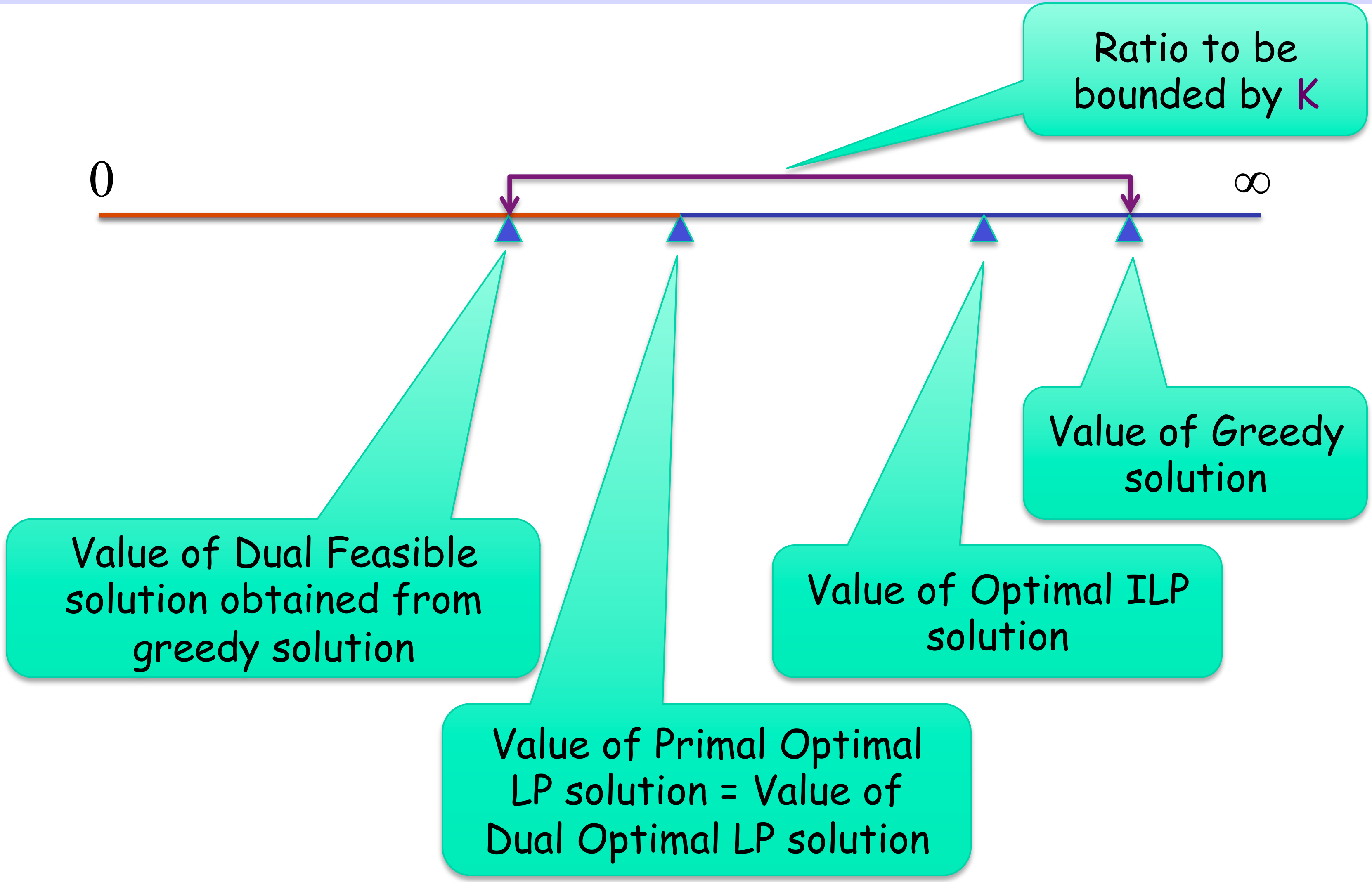
$$\begin{aligned} \max \quad & \sum_{e \in U} y_e \\ \text{subject to} \quad & \sum_{e: e \in S} y_e \leq c(S), \quad S \in \mathcal{S} \\ & y_e \geq 0, \qquad e \in U \end{aligned}$$

Weak Duality Principle

If $\bar{x}$ is primal feasible and $\bar{y}$ is dual feasible then

$$\min \sum_{S \in \mathcal{S}} c(S) x_S \geq \max \sum_{e \in U} y_e$$

K-Approximation Alg using Dual Fitting



Analysis of Greedy Set Cover

- In each iteration, greedy algorithm picks the set with the most uncovered items.
- In iteration j, let S_j be the set picked covering m previously uncovered items. Let

$$\text{price}(e) = c(S_j)/m$$

be the price of each item e covered in this iteration.

- If $S_1, \ldots, S_k$ are sets chosen by greedy algorithm,

$$\begin{aligned}\text{Total Cost of Greedy Solution} &= \sum_{j=1}^{k} c(S_j) \\ &= \sum_{e \in U} price(e)\end{aligned}$$

Analysis of Greedy Set Cover

Let price$(e) = \frac{c(S_j)}{m}$

Consider the following dual variables:

$$y_e = \frac{\text{price}(e)}{H_n}$$

Claim: All dual constraints are satisfied.

$$\sum_{i=1}^{k} y_{e_i} \leq \frac{c(S)}{H_n} \cdot \left(\frac{1}{k} + \frac{1}{k-1} + \ldots + \frac{1}{1}\right) = \frac{H_k}{H_n} c(S) \leq c(S)$$

Thus $(y_{e_1}, \ldots, y_{e_n})$ gives us a dual feasible point.

$$\sum_{e \in U} price(e) = H_n \left(\sum_{e \in U} y_e \right) \leq H_n \cdot \text{OPT}_f \leq H_n \cdot \text{OPT}$$

Rounding Algorithm for Set Cover

- Algorithm
  - Find an optimal solution to the LP Relaxation
  - Pick all sets S for which $x_S \geq 1/f$ in this solution
    - f = frequency of most frequent item

- Analysis
  - Is the resulting solution a valid set cover?
  - How good is the solution? How close is to the optimal set cover?

Analysis of Rounding Algorithm

Let $\mathcal{C}$ = sets picked by **Rounding Algorithm**.

Claim 1: $\mathcal{C}$ is a valid set cover. Arbitrary item e appears in at most f sets. At least one of these sets is assigned value $1/f$. Thus, e will get picked.

Claim 2: The rounding algorithm is f-approximate. Rounding increases the value of each set by a factor of at most f.