COT 6936: Topics in Algorithms

Giri Narasimhan

ECS 254A / EC 2443; Phone: x3748

giri@cs.fiu.edu

http://www.cs.fiu.edu/~giri/teach/COT6936_S10.html

https://online.cis.fiu.edu/portal/course/view.php?id=427

Gaussian Elimination

- Solving a system of simultaneous equations

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

O(n^3) algorithm

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

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 $\Sigma_i b_i x_i$
  - $\Sigma_i a_{ij} x_i \ge c_j$ for $1 \le j \le n$
  - $x_i \ge 0$

Transportation Problem

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

Multicommodity versions

Assignment Problem

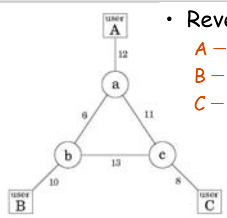
- Workers: $b_1,\ldots,b_n$
- Jobs: $g_1,\ldots,g_m$
- Value of assigning person b_i to job g_k: a_{ik}
- Problem: Choose job assignment with maximum value

The General Assignment Problem generalizes the Bipartite Matching Problem

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 Problem

- Maximize revenue by allocating bandwidth to connections along two routes without 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 $\Sigma c_k x_k$ [Objective Function]
  Subject to $\Sigma a_{ik} x_k \le b_i$ [Constraints]
  and $x_k \ge 0$ [Nonnegativity Constraints]
- Matrix formulation of LP
  Maximize $c^T x$
  Subject to $Ax \le b$
  and $x \ge 0$

2/11/10 COT 6936 10

Converting to standard form

- Min $-2x_1 + 3x_2$ Subject to
  $x_1 + x_2 = 7$
  $x_1 - 2x_2 \le 4$
  $x_1 \ge 0$
- Max $2x_1 - 3x_2$ Subject to
  $x_1 + x_2 \le 7$
  $-x_1 - x_2 \le -7$
  $- x_1 - 2x_2 \le 4$
  $- x_1 \ge 0$

2/11/10 COT 6936 11

Converting to standard form

- Max $2x_1 - 3x_2$ Subject to
  $x_1 + x_2 \le 7$
  $-x_1 - x_2 \le -7$
  $x_1 - 2x_2 \le 4$
  $x_1 \ge 0$
- Max $2x_1 - 3(x_3 - x_4)$ Subject to
  $x_1 + x_3 - x_4 \le 7$
  $-x_1 - (x_3 - x_4) \le -7$
  $x_1 - 2(x_3 - x_4) \le 4$
  $x_1, x_3, x_4 \ge 0$

2/11/10 COT 6936 12

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:
  - $x_1 \le 200$ (1)
  - $x_2 \le 300$ (2)
  - $x_1 + x_2 \le 400$ (3)
  - $x_1, x_2 \ge 0$
- (100,300) is feasible; value = 1900. Optimum?
- Adding 1 times (1) + 6 times (2) gives us
  - $x_1 + 6x_2 \le 2000$
- Adding 1 times (3) + 5 times (2) gives us
  - $x_1 + 6x_2 \le 1900$
  - "Certificate of Optimality" for solution (100,300)

How were mutipliers determined?

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.

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 $|x_u - x_v| \le w_{uv}$ for every edge (u.v)
  - The taut strings correspond to the shortest path, i.e., they have no slack

Simplex Algorithm

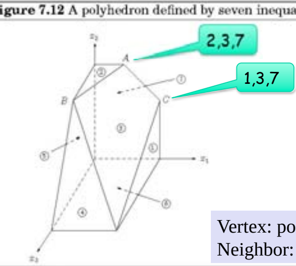
- Start at v, any vertex of feasible region
- while (there is neighbor v' of v with better objective value) do
  - set v = v'
- Report v as optimal point and its value as optimal value

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

Simplex Algorithm: Example

i.e., some inequalities satisfied as equalities

Figure 7.12 A polyhedron defined by seven inequalities.



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

$x_1 \le 200$	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

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

Steps of Simplex Algorithm

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

Initial LP:

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

$$\begin{aligned} 2x_1 - x_2 &\le 4 & ① \\ x_1 + 2x_2 &\le 9 & ② \\ -x_1 + x_2 &\le 3 & ③ \\ x_1 &\ge 0 & ④ \\ x_2 &\ge 0 & ⑤ \end{aligned}$$

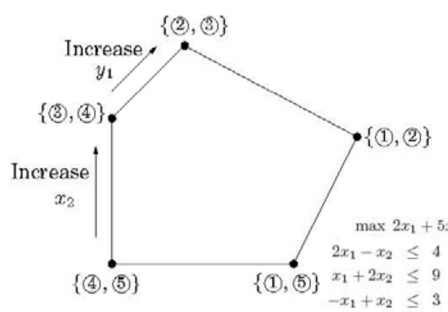
Current vertex: {④, ⑤} (origin).
Objective value: 0.

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



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

$$\begin{aligned} 2x_1 - x_2 &\le 4 & ① \\ x_1 + 2x_2 &\le 9 & ② \\ -x_1 + x_2 &\le 3 & ③ \\ x_1 &\ge 0 & ④ \\ x_2 &\ge 0 & ⑤ \end{aligned}$$

Simplex Algorithm Example

Initial LP:

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

$$\begin{aligned} 2x_1 - x_2 &\le 4 & ① \\ x_1 + 2x_2 &\le 9 & ② \\ -x_1 + x_2 &\le 3 & ③ \\ x_1 &\ge 0 & ④ \\ x_2 &\ge 0 & ⑤ \end{aligned}$$

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:

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

$$\begin{aligned} y_1 + y_2 &\le 7 & ① \\ 3y_1 - 2y_2 &\le 3 & ② \\ y_2 &\ge 0 & ③ \\ y_1 &\ge 0 & ④ \\ -y_1 + y_2 &\le 3 & ⑤ \end{aligned}$$

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:

$$\begin{aligned} \max\ 15 + 7y_1 - 5y_2 & \\ y_1 + y_2 \le 7 & \quad ① \\ 3y_1 - 2y_2 \le 3 & \quad ② \\ y_2 \ge 0 & \quad ③ \\ y_1 \ge 0 & \quad ④ \\ -y_1 + y_2 \le 3 & \quad ⑤ \end{aligned}$$

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:

$$\begin{aligned} \max\ 22 - \tfrac{7}{5}z_1 - \tfrac{1}{5}z_2 & \\ -\tfrac{1}{5}z_1 + \tfrac{5}{5}z_2 \le 6 & \quad ① \\ z_1 \ge 0 & \quad ② \\ z_2 \ge 0 & \quad ③ \\ \tfrac{1}{5}z_1 - \tfrac{2}{5}z_2 \le 1 & \quad ④ \\ \tfrac{1}{5}z_1 + \tfrac{1}{5}z_2 \le 4 & \quad ⑤ \end{aligned}$$

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

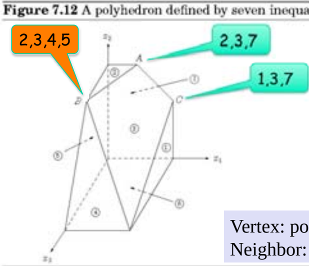
Optimal: all $c_i < 0$.

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

Simplex Algorithm: Degenerate vertices

Figure 7.12 A polyhedron defined by seven inequalities.

i.e., some inequalities satisfied as equalities



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

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 is a polynomial-time algorithm
  - "LP is in *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

Network Flow Problem

- Max $\Sigma_v f(s,v)$ Subject to

$f(e) \le c(e)$	for each edge e
$f(u,v) = -f(v,u)$	for each u,v in set of vertices
$\Sigma_v f(u,v) = 0$	for each u in V – {s,t}
$f(e) \ge 0$	for each edge e

Min-Cost Network Flow Problem

- Min $\Sigma_e a(e)f(e)$ Subject to

$f(e) \le c(e)$	for each edge e
$f(u,v) = -f(v,u)$	for each u,v in set of vertices
$\Sigma_v f(u,v) = 0$	for each u in V – {s,t}
$\Sigma_v f(s,v) = F$	
$f(e) \ge 0$	for each edge e

Vertex Cover as an LP?

- For vertex v, create variable x_v
  - Takes value 0 if it is not in vertex cover
  - Takes value 1 if it is in vertex cover
- For edge (u,v), create constraint $x_u + x_v \ge 1$
- Objective function: Σx_v
- Additional constraints: $x_v \le 1$
- DOES THIS WORK?
- 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 (TUM), 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.