Summary of MA 421, Fall 2021

(LP) (P) $\max \zeta = c^T x$ (D) $\min \xi = b^T y$

s.t. $Ax \le b \longleftrightarrow$ s.t. $A^T y \ge c$

$x \ge 0$ $y \ge 0$

(1) Simplex Method, dictionary, basic vs non-basic vars. various pivoting rules

(2) Duality, Complementary slackness

(3) Primal vs Dual Simplex, applications, Phase I, II.

(4) Sensitivity Analysis

(See also note on Tuesday, Dec 3, 2024)

7. A Dual-Based Phase I Algorithm

The dual simplex method described in the previous section provides us with a new Phase I algorithm, which if nothing else is at least more elegant than the one we gave in Chapter 2. Let us illustrate it using an example:

$$
\begin{array}{lrcr}
\text{maximize} & -x_1 + 4x_2 & & \\
\text{subject to} & -2x_1 - x_2 & \le & 4 \\
& -2x_1 + 4x_2 & \le & -8 \\
& -x_1 + 3x_2 & \le & -7 \\
& x_1,\ x_2 & \ge & 0.
\end{array}
$$

(P)

$$
\begin{array}{rcl}
\zeta & = & - x_1 + 4x_2 \\
\hline
w_1 & = & 4 + 2x_1 + x_2 \\
w_2 & = & -8 + 2x_1 - 4x_2 \\
w_3 & = & -7 + x_1 - 3x_2
\end{array}
$$

(D)

$$
\begin{array}{rcl}
-\xi & = & - 4y_1 + 8y_2 + 7y_3 \\
\hline
z_1 & = & 1 - 2y_1 - 2y_2 - y_3 \\
z_2 & = & -4 - y_1 + 4y_2 + 3y_3
\end{array}
$$

Neither primal or dual dictionaries are feasible.

7. A Dual-Based Phase I Algorithm

The dual simplex method described in the previous section provides us with a new Phase I algorithm, which if nothing else is at least more elegant than the one we gave in Chapter 2. Let us illustrate it using an example:

$$
\begin{array}{lrcr}
\text{maximize} & -x_1 + 4x_2 & & \\
\text{subject to} & -2x_1 - x_2 & \leq & 4 \\
 & -2x_1 + 4x_2 & \leq & -8 \\
 & -x_1 + 3x_2 & \leq & -7 \\
 & x_1,\ x_2 & \geq & 0.
\end{array}
$$

change to $\zeta = -x_1 - x_2$ and then consider its dual

(P)

$$
\begin{array}{rcrrr}
\zeta & = & & -\ 1\,x_1 & +\ 4\,x_2 \\
\hline
w_1 & = & 4 & +\ 2\,x_1 & +\ x_2 \\
w_2 & = & -8 & +\ 2\,x_1 & -\ 4\,x_2 \\
w_3 & = & -7 & +\ x_1 & -\ 3\,x_2
\end{array}
$$

(D)

$$
\begin{array}{rcrrrr}
-\xi & = & & -\ 4\,y_1 & +\ 8\,y_2 & +\ 7\,y_3 \\
\hline
z_1 & = & 1 & -\ 2\,y_1 & -\ 2\,y_2 & -\ y_3 \\
z_2 & = & -4 & -\ y_1 & +\ 4\,y_2 & +\ 3\,y_3
\end{array}
$$

Neither primal or dual dictionaries are feasible.

(D)

$$
\begin{array}{rcrrlrlrl}
-\xi & = & & -4 & y_1 & +8 & y_2 & +7 & y_3 \\
\hline
z_1 & = & 1 & -2 & y_1 & -2 & y_2 & - & y_3 \\
z_2 & = & \underset{1}{\cancel{-4}} & - & y_1 & +4 & y_2 & +3 & y_3
\end{array}
$$

① y_2 enters, z_1 leaves:

$$y_2 = \frac{1}{2} - y_1 - \frac{1}{2} z_1 - \frac{1}{2} y_3$$

$$z_2 = 1 - y_1 + 2 - 4y_1 - 2z_1 - 2y_3 + 3y_3$$

$$= 3 - 5y_1 - 2z_1 + y_3$$

$$-\xi = -4y_1 + 4 - 8y_1 - 4z_1 - 4y_3 + 7y_3$$

$$= 4 - 12y_1 - 4z_1 + 7y_3$$

$$
\begin{array}{lrrrrrrrr}
\text{(D)} \qquad -\xi & = & & - & 4\,y_1 & + & 8\,y_2 & + & 7\,y_3 \\
\hline
z_1 & = & 1 & - & 2\,y_1 & - & 2\,y_2 & - & y_3 \\
z_2 & = & \underset{1}{\cancel{-4}} & - & y_1 & + & 4\,y_2 & + & 3\,y_3
\end{array}
$$

② y_3 enters, y_2 leaves:

$$y_3 = 1 - 2y_1 - z_1 - 2y_2$$

$$z_2 = 3 - 5y_1 - 2z_1 + 1 - 2y_1 - z_1 - 2y_2$$

$$= 4 - 7y_1 - 3z_1 - 2y_2$$

$$-\xi = 4 - 12y_1 - 4z_1 + 7 - 14y_1 - 7z_1 - 14y_2$$

$$= 11 - 26y_1 - 11z_1 - 14y_2 \qquad \text{Opt!}$$

$$
\text{(D)} \qquad
\begin{array}{rcrrrr}
-\xi & = & & -\,4\,y_1 & +\,8\,y_2 & +\,7\,y_3 \\
\hline
z_1 & = & 1 & -\,2\,y_1 & -\,2\,y_2 & -\,y_3 \\
z_2 & = & \underset{1}{\cancel{-4}} & -\,y_1 & +\,4\,y_2 & +\,3\,y_3
\end{array}
$$

Opt. Dict for Dual Problem:

$$
\begin{aligned}
-\xi &= 11 - 26y_1 - 11z_1 - 14y_2 \\
y_3 &= 1 - 2y_1 - z_1 - 2y_2 \\
z_2 &= 4 - 7y_1 - 3z_1 - 2y_2
\end{aligned}
$$

(D)

$$
\begin{array}{rcrrrrrrr}
-\xi & = & & - & 4\,y_1 & + & 8\,y_2 & + & 7\,y_3 \\
\hline
z_1 & = & 1 & - & 2\,y_1 & - & 2\,y_2 & - & y_3 \\
z_2 & = & \underset{1}{\cancel{-4}} & - & y_1 & + & 4\,y_2 & + & 3\,y_3
\end{array}
$$

Opt. Dict for Primal Problem:

$$
\begin{aligned}
w_1 &= 26 + 2w_3 + 7x_2 \\
x_1 &= 11 + w_3 + 3x_2 \quad \Leftarrow \; -\begin{pmatrix} -26 & -11 & -14 \\ -2 & -1 & -2 \\ -7 & -3 & -2 \end{pmatrix}^T \\
w_2 &= 14 + 2w_3 + 2x_2
\end{aligned}
$$

$$
\begin{aligned}
\zeta &= -x_1 + 4x_2 \quad \text{(Original obj fct)} \\
&= 11 + w_3 + 2x_2 + 4x_2 \\
&= 11 + w_3 + 6x_2
\end{aligned}
$$

Ready for Phase II

Convex Analysis

(1) Definition of convex sets, convex comb.
convex hull

(2) Caratheodory Theorem

(3) Separation Theorem

(4) Farkas Lemma (and its variants)

LEMMA 10.5. *The system $Ax \leq b$ has no solutions if and only if there is a y such that*

how to find y

$$
\begin{aligned}
A^T y &= 0 \\
y &\geq 0 \\
b^T y &< 0.
\end{aligned}
\tag{10.8}
$$

Convex Analysis

(1) Definition of convex sets, convex comb.
convex hull

(2) Caratheodory Theorem

(3) Separation Theorem

(4) Farkas Lemma (and its variants)

$$\begin{array}{ll} \text{maximize} & 0 \\ \text{subject to} & Ax \leq b \end{array} \quad \text{vs} \quad \begin{array}{lrcl} \text{minimize} & b^T y & & (<0) \\ \text{subject to} & A^T y & = & 0 \\ & y & \geq & 0. \end{array}$$

Applications of LP

(1) Regression, L^1, L^2, L^∞

(2) Binary Classification

(3) Maximum inscribed circle

(4) .... Formulations of problems in terms of LP...

(5) Integer Programming (Branch-and-Bound and Gomory cuts)

Networks

(1) Graph, nodes, arcs, spanning tree

(2) Network flow problem: (Transshipment)

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

A - incidence matrix

(3) Dual Network flow problem:

$$\begin{aligned} \max \quad & -b^T y \\ \text{s.t.} \quad & A^T y + z = c \\ & z \geq 0 \end{aligned}$$

Given a spanning tree $\Rightarrow$ find x, y, z.

x_{ij}, y_i, z_{ij}

Networks

(4) Primal vs Dual Network Simplex method

(i) Spanning tree as basic variables

(ii) Determine entering and leaving arcs

(iii) Updating x_{ij}, y_i, z_{ij}

(5) Variants of Network Flow problems

(i) Transportation (Bipartite graph)

(ii) Assignment

(iii) Inequality Constraints

(iv) Upper bounded flows

Algorithms Specific to Networks/Graphs

(1) Shortest Path (Bellman vs Dijkstra)

Dynamics programming : Bellman Eqn

Let v_i = shortest distance from i to r

Then $\boxed{v_i = \min_j \{ c_{ij} + v_j \}}$

Algorithms Specific to Networks/Graphs

(1) Shortest Path (Bellman vs Dijkstra)

Dynamics programming: Bellman Eqn

$$v_i^{0} = \begin{cases} +\infty, & i \neq r \\ 0 & i = r \end{cases}$$

$$v_i^{(k)} = \min_j \{ c_{ij} + v_j^{(k)} \}, \quad V_r^{(k)} = 0$$

Algorithms Specific to Networks/Graphs

(1) Shortest Path (Bellman vs Dijkstra)

Initialize:

$\mathcal{F} = \emptyset$

$$v_j = \begin{cases} 0 & j = r, \\ \infty & j \neq r. \end{cases}$$

while $(|\mathcal{F}^c| > 0)\{$

$j = \text{argmin}\{v_k : k \notin \mathcal{F}\}$

$\mathcal{F} \leftarrow \mathcal{F} \cup \{j\}$

for each i for which $(i, j) \in \mathcal{A}$ and $i \notin \mathcal{F}$ $\{$

if $(c_{ij} + v_j < v_i)$ $\{$

$v_i = c_{ij} + v_j$

$h_i = j$

$\}$

$\}$

$\}$

Algorithms Specific to Networks/Graphs

(2) Max-Flow-Min-Cut

Ford-Fulkerson (Augmented Path)

Table 22.1 The Travelers' Example: Seat Availability

From	To	Number of seats
San Francisco	Denver	5
San Francisco	Houston	6
Denver	Atlanta	4
Denver	Chicago	2
Houston	Atlanta	5
Atlanta	New York	7
Chicago	New York	4

[C] p. 368

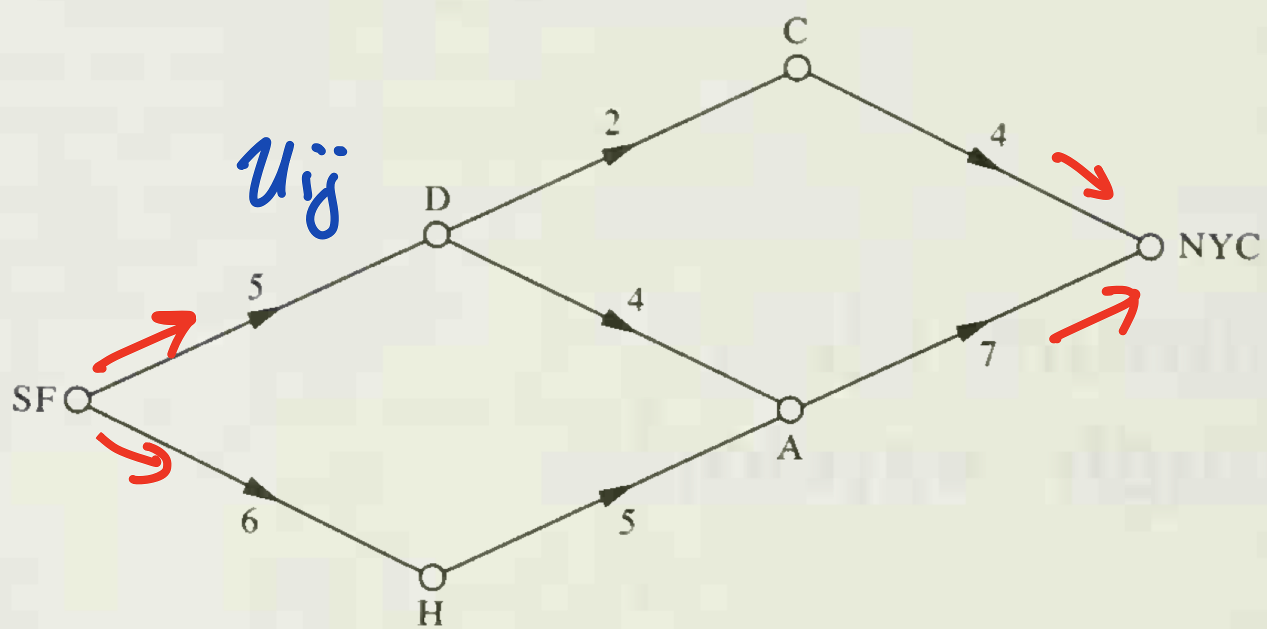


Figure 22.1

Algorithms Specific to Networks/Graphs

(3) Minimum Spanning Tree (MST)

Properties of a (Spanning) tree

Theorem 7.1

(a) *Every tree with more than one node has at least one leaf.*

(b) *An undirected graph is a tree if and only if it is connected and has $|\mathcal{N}| - 1$ arcs.*

(c) *For any two distinct nodes i and j in a tree, there exists a unique path from i to j.*

(d) *If we start with a tree and add a new arc, the resulting graph contains exactly one cycle (as long as we do not distinguish between cycles involving the same set of nodes).*

Algorithms Specific to Networks/Graphs

(3) Minimum Spanning Tree (MST)

(Greedy Algorithm)

Greedy algorithm for the minimum spanning tree problem

1. The input to the algorithm is a connected undirected graph $G = (\mathcal{N}, \mathcal{E})$ and a coefficient c_e for each edge $e \in \mathcal{E}$. The algorithm is initialized with a tree $(\mathcal{N}_1, \mathcal{E}_1)$ that has a single node and no edges ($\mathcal{E}_1$ is empty).

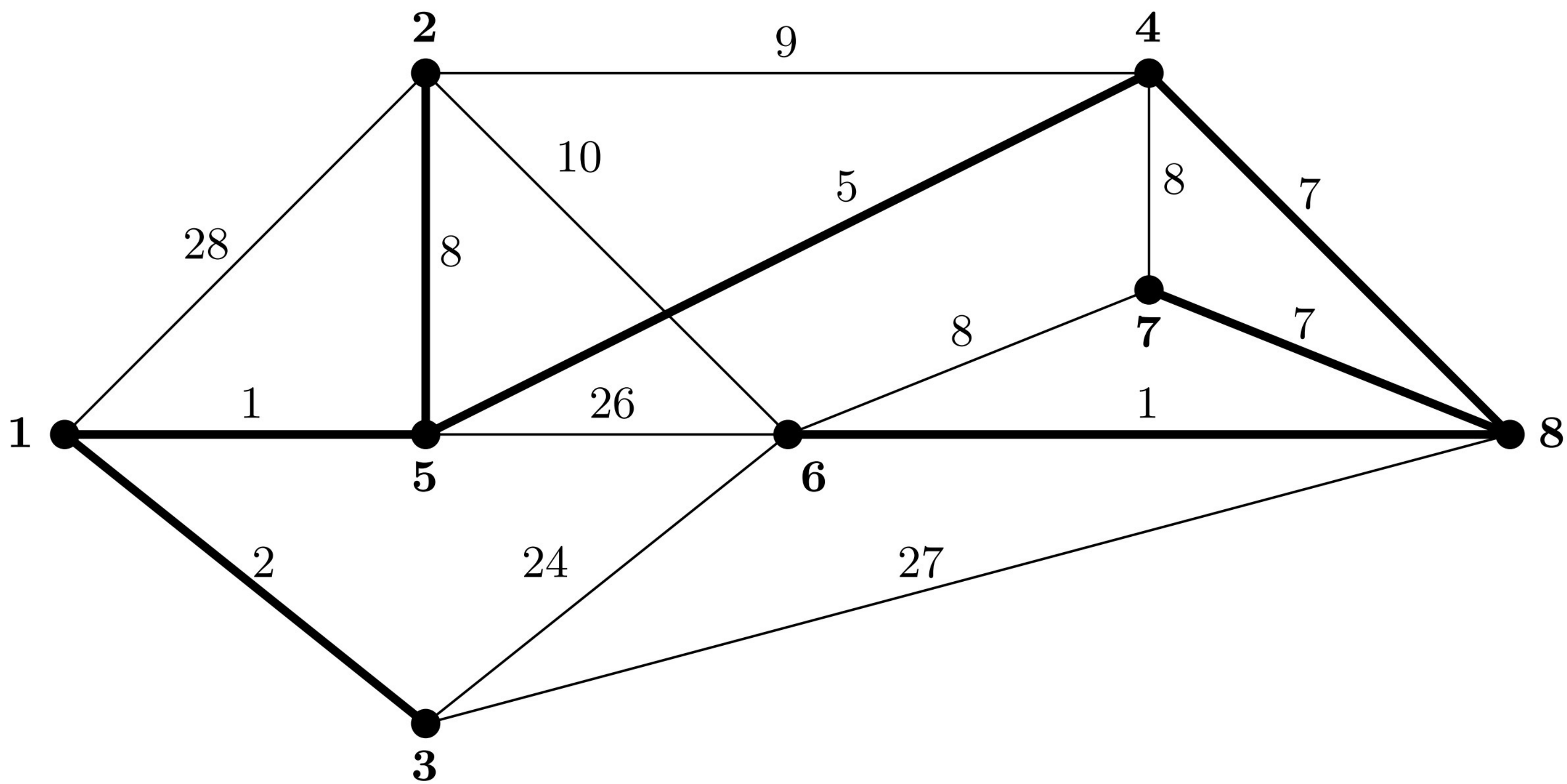
2. Once $(\mathcal{N}_k, \mathcal{E}_k)$ is available, and if $k < n$, we consider all edges $\{i, j\} \in \mathcal{E}$ such that $i \in \mathcal{N}_k$ and $j \notin \mathcal{N}_k$. Choose an edge $e^* = \{i, j\}$ of this type whose cost is smallest. Let

$$\mathcal{N}_{k+1} = \mathcal{N}_k \cup \{j\}, \qquad \mathcal{E}_{k+1} = \mathcal{E}_k \cup \{e^*\}.$$

Algorithms Specific to Networks/Graphs

(3) Minimum Spanning Tree (MST)

(Greedy Algorithm)



Algorithms Specific to Networks/Graphs

(3) Minimum Spanning Tree (MST)

(Greedy Algorithm)