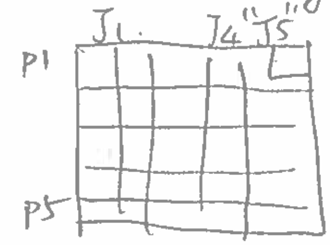
Note on Homework 8

Section 7.5

1. ① Create a dummy job.



② Set the cost for J5 to be 0/M for P1, . , P5

---

6 (a) Use Hungarian method for the max problem
(Change all # to their negative.)

(b) Use the transportation table set up. Argue the initial bfs consists of 0, 1 as the x_{ij} values. In each iteration, Δ can be . . .

Section 7.6

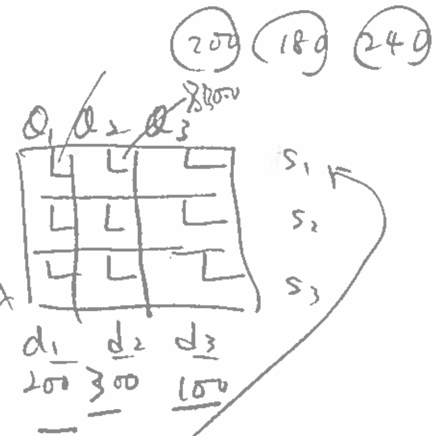
1. Adjust the supply / demand values, add the dummy demand point

* Make sure the optimal solution of your problem will give the min cost for the original problem.

Chapter 7

#10
#11 } sensitivity analysis

#18

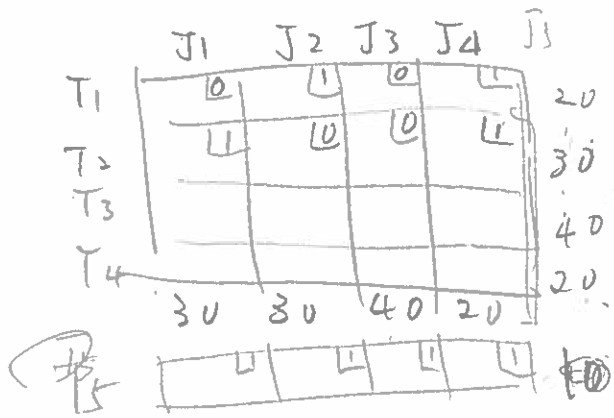


Carefully adjust the cost

e.g. c_{31} = production cost in 3rd quarter + 60+60 for the backlog.

c_{13} = production cost in 1st quarter + 100+100 holding cost.

#19



A special case - Matching problem. Try to match n boys $b_1, \dots, b_n$ with n girls $g_1, \dots, g_n$ with $c_{i,j} = 1$ if b_i know g_j. Set a source vertex so and a sink vertex si such that capacity constraint from so to b_i equals 1, and the capacity constraint from g_i to si equals 1, for $i = 1, \dots, n$.

Example

Compatibilities for Matching

	Leni Anderson	Meryl Streep	Katharine Hepburn	Linda Evans	Victoria Principal
Kevin Costner	—	C	—	—	—
Burt Reynolds	C	—	—	—	—
Tom Selleck	C	C	—	—	—
Michael Jackson	C	C	—	—	C
Tom Cruise	—	—	C	C	C

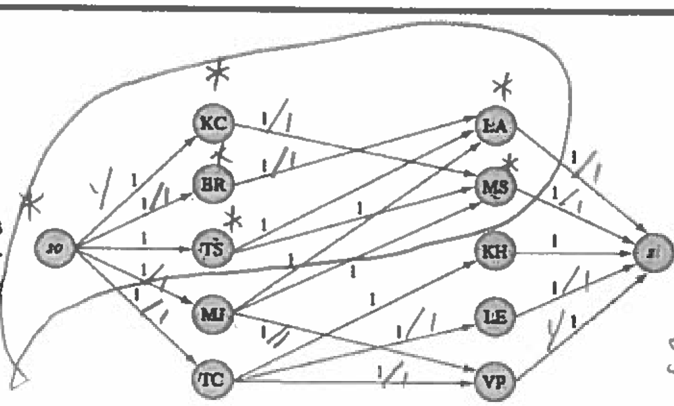


FIGURE 8
Network for Matchmaker

$\boxed{f = 4}$

$S = \{so, KC, BR, TS, LA, MS\}$

$\bar{S} = V - S$

$K(S, \bar{S}) = 4 = C(so, MJ) + C(so, TC) + C(LA, si) + C(MS, si) = 1 + 1 + 1 + 1.$

Hall's Theorem There is a complete matching if and only if every group of k boys know at least k girls for $k = 1, \dots, n$. (X -10 pt Homework credits :)

Theorem Given a capacitated network with source vertex so and sink vertex si. Then there is a flow with value x_0 if and only if

$$\sum_{i,j \in S \times \overline{S}} c_{ij} - \sum_{i,j \in \overline{S} \times S} c_{ij} \geq x_0.$$

Remark We can always set up the maximum flow problem as a transportation problem:

$$\boxed{\max Z = x_0}$$

subject to: $\sum_i f_{ij} - \sum_k f_{jk} = 0, \quad j = 1, \ldots, n.$

$$\boxed{\sum f_{so,i} = x_0,} \quad \sum f_{j,si} = x_0, \quad \boxed{0 \le f_{ij} \le c_{ij}.}$$

Ford-Fulkerson Algorithm

Consider a capacitated network with source vertex so and sink vertex si. Partition $V = S \cup \overline{S}$ with $so \in S, si \in \overline{S}$. Define the cut associated with $(S, \overline{S})$ as $K(S, \overline{S}) = \sum_{(i,j) \in (S \times \overline{S}} c_{ij}$. Then maximum flow in the network equals the minimum cut.

1. Set initial flow to be 0.

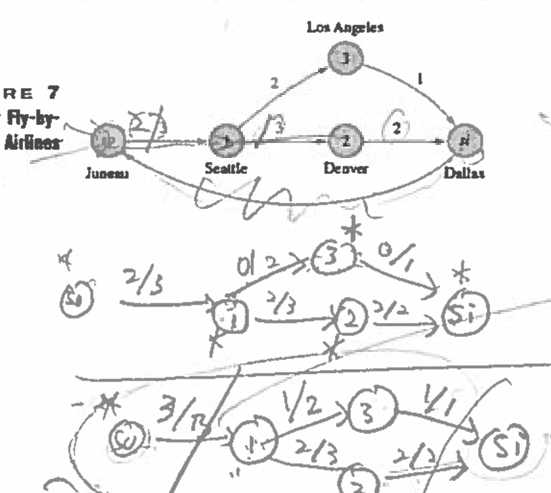
2. Find a chain from so to si consisting of non-saturated forward arcs $c_{ij} - f_{ij} > 0$, and backward arcs with non-zero flows f_{rs}; increase the flow by the minimum of the values $c_{ij} - f_{ij}$ and f_{rs}.

   This can be done by labeling the vertices starting from so; after adding a new round of newly labeled vertices, move on to the next round by labeling those vertices connected with those labeled in the last round by forward non-saturated forward arcs or and backward arcs with positive flow until we reach si, or find it impossible.

3. If no such chain exists, then we have an optimal flow. (Letting S be the vertices reachable from so with a positive chain. Then $K(S, \overline{S})$ is a minimum cut.)

Example

FIGURE 7
Network for Fly-by-Night Airlines



Step 1. $(so) \xrightarrow{2} (1) \xrightarrow{2} (2) \xrightarrow{2} (si)$

Step 2 $(so) \xrightarrow{1} (1) \xrightarrow{1} (3) \xrightarrow{1} (si)$

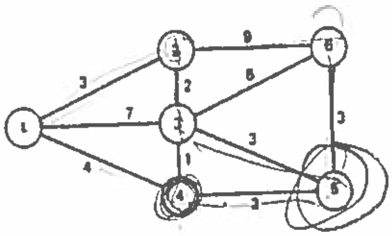
$f = 3$

$K(S, \bar{S})$ $\quad V = S \cup \bar{S}$

Solution The appropriate network is given in Figure 7. Here the capacity of arc (i, j) is the maximum number of daily flights between city i and city j. The optimal solution to this maximum flow problem is $z = x_0 = 3$, $x_{J,S} = 3$, $x_{S,L} = 1$, $x_{S,De} = 2$, $x_{L,D} = 1$, $x_{De,D} = 2$. Thus, Fly-by-Night can send three flights daily connecting Juneau and Dallas. One flight connects via Juneau–Seattle–L.A.–Dallas, and two flights connect via Juneau–Seattle–Denver–Dallas.

Additional Examples and Algorithms

Example 1. Shortest Path



0. $[0^*, 3, 7, 4, \infty, \infty]$
1. $[0^*, 3^*, 7, 4, \infty, \infty]$
2. $[0^*, 3^*, 5, 4^*, \infty, 12]$
3. $[0^*, 3^*, 5^*, 4^*, 7, 12]$
4. $[0^*, 3^*, 5^*, 4^*, 7^*, 11]$
5. $\boxed{[0^*, 3^*, 5^*, 4^*, 7^*, 10^*]}$

Example 2. A maximal flow problem