CS 170

1. Study Group

None

2. Flow vs LP

(a) Create a directed weighted Graph G as follows:
Use s_i to represent the *i-th* supplier, use b_j to represent the *j-th* buyer. From hyper source s derive n edges to each s_i with weight $s[i]$, and from each b_j derive an edge to hyper sink t with weight $b[j]$. For each pair$(i, j) \in L$, create an edge s_i to b_j with infinity weight. Then the maximum flow of the Graph G is the answer.

(b) Create a linear programming problem as follows:
Define x_{ij} as the number of products the supplier i supplies to buyer j, $(i, j) \in L$.

$$\begin{aligned} &max \quad \textstyle\sum_{i,j} x_{ij} \quad (i,j) \in L \\ &\quad 0 \leq \textstyle\sum_i x_{ij} \leq s[i] \\ &\quad 0 \leq \textstyle\sum_j x_{ij} \leq b[j] \end{aligned}$$

(c) The max-flow algorithm will be better!

3. Feasible Routing

(a) No. Since $f^{in}(v) - f^{out}(v) = d_v$, we can use this characteristic for each vertex and add them together. The conservation of flow tells us the LHS must be 0, which proves our statements.

(b) Create a source node s and a sink node t. Create edges from s to each supply node v with weight $-d_v$, and create edges from each demand node v to sink node t with weight d_v. Run maxflow algorithm on the new Graph, and there is a $feasible\ routing$ on the original Graph if and only if the maxflow of the new Graph equals $\sum_{u \in T} d_u$.

4. Applications of Max-Flow Min-Cut

The proof is too obvious to write.

5. Restoring the Balance!

Run augmenting-path from u to v on the residual graph, and if there exists an augmenting-path between u and v, then the max-flow will not change. Otherwise the max-flow will minus one.

6. Zero-Sum Games

(a) Alice:

$$\begin{aligned}
&\mathit{maximize}\ \ p\\
&4x_1 + 2x_2 \le p\\
&x_1 + 5x_2 \le p\\
&x_1 + x_2 = 1\\
&x_1, x_2 \ge 0
\end{aligned}$$

(b) Bob:

$$\begin{aligned}
&\mathit{minimize}\ \ p\\
&4y_1 + y_2 \ge p\\
&2y_2 + 5y_1 \ge p\\
&y_1 + y_2 = 1\\
&y_1, y_2 \ge 0
\end{aligned}$$