Algorithm

18-7-19

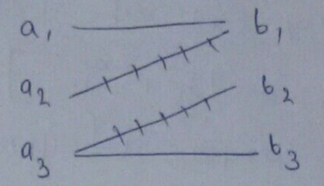
1. Start with dual feasible.
2. Find maximum matching (unweighted) in tight subgraph H
3. Dual update

till we get perfect M.

Step 1: $\forall a \in X$, $y_a = \max_b \{w_{ab}\}$

$\forall b \in Y$, $y_b = 0$

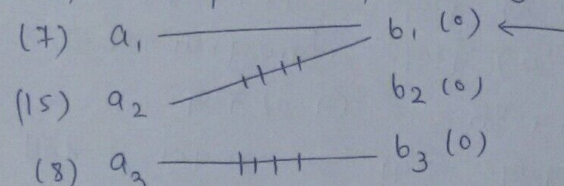
Suppose Max. M in H is as follows:



Max. M in (tight subgraph) H is not perfect

So, we need step 3: Dual update, to get more edges tight, without violating dual feasibility

For yesterday's example, tight subgraph H =



← dual variable settings

One possible dual update that makes $a_2 b_2$ tight: $a_1: 7 \to 4$, $a_2: 15 \to 12$, $b_1: 0 \to 3$

Q: How to find $A' \subseteq X$, $B' \subseteq Y$ & k s.t.

if $\forall a' \in A'$, $y_{a'} \leftarrow y_{a'} - k$

$\forall b' \in B'$, $y_{b'} \leftarrow y_{b'} + k$

& we don't violate dual feasibility.

& hopefully get more edges tight.

Let S = set of vertices reachable via alternating path from unmatched X-vertices w.r.t. max. M in H.

In example, $S = \{a_1, b_1, a_2\}$

$(X \setminus S) \cup (S \cap Y)$ is v.c. for H.

$\Rightarrow \exists$ no edge in H that goes from $S \cap X$ to $Y \setminus S$.

* these are "interesting" edges for us.
  ~ candidates to make tight.

Since, S is a Hall's set for H,

$|S| > |N(S)|$

So, if all y_a s.t. $a \in S$ is decreased by $k > 0$ & y_b s.t $b \in N(S)$ is increased by same k,
dual objective is guaranteed to decrease.

Q : How to find k ?

Look at "slack" edges s.t

$w_{ab} < y_a + y_b$

Ask : what is the slack ?

Slack (a, b) = $y_a + y_b - w_{ab}$

e.g. initially, [for dual feasible initial setting]
slack $(a_1, b_2) = 3$
slack $(a_1, b_3) = 4$
slack $(a_2, b_2) = 3$
slack $(a_2, b_3) = 6$

Take the minimum among these & let it be k.

<u>Step 3</u> : Dual update procedure.

- We found out H, Max Matching M, set S (Hall's set) & "interesting" edges ie those from $S \cap X$ to $Y \setminus S$.

- $k = \min\limits_{\substack{(a,b): \\ a \in S \cap X, \\ b \in Y \setminus S}} \{ y_a + y_b - w_{ab} \}$

- For $\forall a \in S \cap X$, $y_a \leftarrow y_a - k$

- For $\forall b \in S \cap Y$, $y_b \leftarrow y_b + k$

Q : Is this update violating dual feasibility ?

4 types of edges :

$S \cap X \rightarrow Y \cap S$: $+k - k$: feasible ✓ tight ✓

$S \cap X \rightarrow Y \setminus S$: $\because k$ was picked as min. : feasible ✓ : at least 1 edge becomes tight

$X \setminus S \rightarrow Y \cap S$: $+k$ @ Y-end, so feasible ✓ : can become non-tight ⊛

$X \setminus S \rightarrow Y \setminus S$: no change, feasible ✓

(*) : There is a loss in tight ~~ae~~ edges.

Q: Is that okay ?

No matched edges cross $X \setminus S \rightarrow Y \cap S$

so, we don't lose any matched edges.

Q : Can we always increase the size of matching by at least 1 in every iteration ?

→ Need not : if tight edge is $a_1 - b_3$, then maximum M size remains same.

Q : So, what progress is guaranteed ?

→ $|N(S)|$ increases, after every iteration.

→ at most n iterations needed for at least 1 increase in maximum matching.

→ Running time = $O(n * n * m)$

where $O(m)$ = per iteration work needed

∵ G is complete bipartite graph, $m = \Theta(n^2)$

So, $O(n^4)$ time.