Mathematical Methods for Computer Science I

Fall 2016

Algorithm *for finding an augmenting path P of length ≥ 1 in a bipartite graph.*

Idea: Start from an unmatched vertex and construct an alternating path iteratively.

Input: A bipartite graph $G = (V \cup W, E)$ and a matching $M \subseteq E$.
Output: An augmenting path in G w.r.t. M, if one exists.
Step 1 (initialize): Label the vertices

$$v \in V \text{ is labelled } \begin{cases} Y(es) \text{ if unmatched} \\ N(o) \text{ if matched} \end{cases}$$

$w \in W$ get $N(o)$

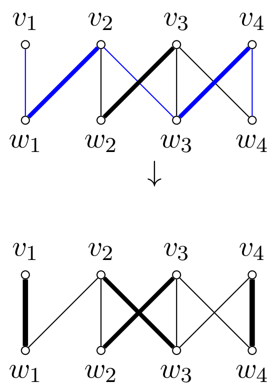
Set S = set of Y vertices from V
T = set of Y vertices from W (=$\emptyset$ here)

Step 2: Connect 1 vertex from V to a vertex of W:
All vertices $w \in W$ with label N that are connected to a $v \in S$ through an edge not in M get label Y and are added to T. Set $S = \emptyset$.

Step 3: Augmenting path found or continue the path via an edge of M:
If any $w \in T$ is unmatched, then an augmenting path has been found. Otherwise, all $v \in V$ with label N that are connected to a $w \in T$ through an edge in M get Y and are added to S. Set $T = \emptyset$.

Step 4: If an augmenting path has been found ($\rightarrow$ augment M) or $S = \emptyset$, stop.
Otherwise, go to step 2.

Example:



Step 1: $S = \{v_1\}, T = \emptyset$
Step 2: $S = \emptyset, T = \{w_1\}$
Step 3: $S = \{v_2\}, T = \emptyset$
Step 2: $S = \emptyset, T = \{w_2, w_3\}$
Step 3: $S = \{v_3, v_4\}, T = \emptyset$
Step 2: $S = \emptyset, T = \{w_3, w_4\}$
Step 3: w_4 unmatched $\rightsquigarrow$ augmenting path found!