5 Matching

5.1 The Matching Problem

A *matching* M of a graph $G = (V, E)$ is a subset of the edges with the property that no two edges of M share the same node. [M is a piecewise disjoint edge set].

Given a graph $G = (V, E)$, the *matching problem* is to find a maximum matching M of G, i.e.,

$$\max\{|M| : \quad M \text{ is a matching}\}.$$

When the cardinality of a matching is $\lfloor |V|/2 \rfloor$, the largest possible in a graph with $|V|$ nodes, we say the matching is *complete*, or *perfect*.

Let M be a fixed matching of graph G. Edges in M are called *matched* edges; the other edges are *free*. If $[v, u]$ is a matched edge, then u is the mate of v. Nodes that are not incident to any matched edge are called *exposed*; the remaining nodes are *matched*.

A path $p = [u_1, u_2, \ldots, u_k]$ is called *alternating* if the edges $[u_1, u_2], [u_3, u_4], \ldots, [u_{2j-1}, u_{2j}], \ldots$ are free, whereas $[u_2, u_3], [u_4, u_5], \ldots, [u_{2j}, u_{2j+1}], \ldots$ are matched. Vertices that lie on an alternating path starting with an exposed vertex and have odd rank on this path are called *outer*. Those with even rank are called *inner*.

An alternating path $p = [u_1, u_2, \ldots, u_k]$ is called *augmenting* if both u_1 and u_k are exposed vertices.

$$\text{(exposed)} \quad u_1 \underset{\text{free}}{\overline{\quad\quad}} u_2 \underset{\text{match}}{\overline{\quad\quad\quad}} u_3 \underset{\text{free}}{\overline{\quad\quad}} u_4 \underset{\text{match}}{\overline{\quad\quad\quad}} \ldots u_{k-1} \underset{\text{free}}{\overline{\quad\quad}} u_k \quad \text{(exposed)}.$$

If S, T are sets, then $S \bigoplus T$ denotes the *symmetric difference* of S and T, defined by

$$S \bigoplus T = (S - T) \cup (T - S).$$

For example, if $S = \{1, 2, 3, 4, 5\}$ and $T = \{2, 4, 5, 6, 7\}$, then

$$S \bigoplus T = \{1, 3\} \cup \{6, 7\} = \{1, 3, 6, 7\}.$$

Lemma 1. If M is a matching and P is the set of edges on an augmenting path with respect to M, then $M' = M \bigoplus P$ is also a matching and $|M'| = |M| + 1$.
Proof. Suppose by the contradiction that M' is not a match of G. Then there are two edges e, e' in M' incident to the same node. Since

$$M' = (M - P) \cup (P - M),$$

we have three cases:

1. $e, e' \in M - P$;

2. $e, e' \in P - M$;

3. $e \in M - P, \quad e' \in P - M$.

In Case 1, we have two edges in M sharing the same node, which is a contradiction because M is a match and thus cannot have two edges sharing the same node.

In Case 2, we note that the edges in $P - M$ are the edges of the form $[u_{2j-1}, u_{2j}]$ and hence two of them cannot be incident to the same node.

For Case 3, suppose that an edge $e' = [u_{2j-1}, u_{2j}]$ in $P - M$ has a node in common, say u_{2j}, with another edge $e \in M - P$. But, u_{2j} is a node of the edge $e'' = [u_{2j}, u_{2j+1}] \in M$, and hence e and e'' of M share a common node, which is a contradiction because M is a match. It follows that M' is match.

Next, we show $|M'| = |M| + 1$. Now, P contains $2k - 1$ edges; k of them are free ($[u_1, u_2], [u_3, u_4], \ldots, [u_{2k-1}, u_{2k}]$) and $k - 1$ of them belong to M. Hence, M' has $|M| + 1$ edges. $\square$

Theorem 1. A matching M is maximal $\boxed{\text{IF AND ONLY IF}}$ there is no augmenting path with respect to M.

Proof. "Only if " by Lemma 1.

"If" part. Suppose that M is not a maximum matching. Let M' be a maximum matching. Consider $M \bigoplus M'$. For the subgraph $G' = (V, M \bigoplus M')$, we have $d(i) \leq 2$ for all $i \in V$. If the degree of a vertex is 2, one of the two edges is in M and the other is in M'. Thus all connected components of G' will be either paths or cycles of even length. In all cycles, we have the same number of edges in M as in M'. Because $|M'| > |M|$, it must be the case that in one of the paths we have more edges from M' than from M, and hence this path is an augmenting path. This is a contradiction. $\square$

5.2 Bipartite Matching

$G = (S, T, E)$ is a bipartite graph.

Node Cover. Suppose we have a graph $G(N, E)$. $X \subset N$ is called a *node cover* if any edge $e \in E$ has at least one node in X.

Then the *minimum node cover problem* is

$$\min\{|X| : \; X \text{ is a node cover}\}.$$

Lemma. The number of edges in a matching $\leq$ the number of nodes in a node cover.
Proof. Suppose $M \subseteq E$ is a matching and X is a node cover. Then for any $e \in M$, there is an end node of E in X. Hence, $|M| \leq |E|$. $\square$

Theorem (König - Egervary). For a **BIPARTITE GRAPH**, the maximum number of edges in a matching is equal to the minimum number of nodes in a node cover. [max-match=min-cover]

For a given matching $M \in E$, we define an *alternating tree* relative to the matching to be a tree which satisfies the following two conditions. First, the tree contains exactly one exposed node from S, which we call its *root*. Second, all paths between the root and any other node in the tree are alternating paths.

BIPARTITE CARDINALITY MATCHING ALGORITHM

Step 0 (*Start*) The bipartite graph $G = (S, T, E)$ is gievn. Let M be any matching, possibly the empty matching. No nodes are labeld.

Step 1. (*Labeling*)

(1.0) Give the label "$\emptyset$" to each exposed node in S.

(1.1) If there are no unscanned labels, go to Step 3. Otherwise, find a node i with an unscanned label. If $i \in S$, go to Step 1.2; if $i \in T$, go to Step 1.3.

(1.2) Scan the label on node i ($i \in S$) as follows. For each edge $(i, j) \notin M$ incident to i, give node j the label "i" unless node j is already labeled. Return to Step 1.1.

(1.3) Scan the label on node i ($i \in T$) as follows. If node i is exposed, go to Step 2. Otherwise, identify the unique edge $(i, j) \in M$ incident to node i and give j the label "i". Return to Step 1.1.

Step 2 (*Augmentation*) An augmenting path has been found, terminating at node i (identified in Step 1.3). The nodes preceding node i in the path are identified by "back-tracing". The initial node in the path has the label "$\emptyset$". Augment M by adding to M all edges in the augmenting path that are not in M and removing from M those which are. Remove all labels from nodes. Return to Step 1.0.

Step 3 (*Hungarian Labeling*) The labeling is Hungarian (it is not possible to add more nodes and edges), no augmenting path exists, and the matching M is of maximum cardinality. Let $L \subseteq S \cup T$ denote the set of labeled nodes. Then $C = (S - L) \cup (T \cap L)$ is a minimum cardinality node cover, dual to M. $\square$

- Complexity $O(|S|^2|T|)$.

The cardinality matching problem is particularly easy to solve for a special graph – "convex graph". A bipartite graph $G = (S, T, E)$ is said to be *convex* if it has the property that if (i, j) and (j, k) are edges, where $i < k$, then

$$(i + 1, j), \;\; (i + 2, j), \;\; \ldots, \;\; (k - 1, j)$$

are also edges.

The cardinality matching problem can be solved by the following procedure. For each node $j \in T$, let

$$\pi_j = \max\{i| \; (i, j) \in E\}.$$

Start with empty matching and iterate over $i = 1, 2, \ldots, m$. If there are any edge (i, j), where j is an exposed node, add to the matching the edge (i, j) for which π_j is as small as possible.

Max-Min or "bottleneck" matching ...

5.3 Bipartite Matching and Network Flows

Given any bipartite graph $G = (S, T, E)$, we define the following network with unit arc capacities $--$ $N(G) = (s, t, W, A)$, where s and t are two new nodes; W is the union of $\{s, t\}$, S, and T; and A is a set of arcs consisting of three categories:

1. The arcs (s, v) for all $v \in S$;
2. The arcs (u, t) for all $u \in T$;
3. The arcs (v, u) for all $v \in S$ and $u \in T$ such that $[v, u] \in E$.

Then $N(G)$ is *simple network* (unit arc capacities and each node has indegree either 0 or 1, or outdegree either 0 or 1).

Lemma. The cardinality of the maximum matching in a bipartite graph G equals the value of the maximum $s - t$ flow in $N(G)$.

- Any max-flow algorithm can be applied to maximum matching problem.
- The maximum matching problem can be solved in $O(|V|^{1/2}|E|)$ time.

5.4 Bipartite Weighted Matching Algorithm

The weighted bipartite matching (WBM) problem is

$$\max\{\sum_{(i,j)\in M} w_{ij} : \ M \text{ is a matching}\},$$

where w_{ij} is the weight on edge (i,j). The WBM is equivalent to the *assignment problem.*

For simplicity, assume a complete bipartite graph $G = (S, T, S \times T)$ with $|S| = m$ and $|T| = n$, $m \leq n$. A linear programming formulation of the WBM problem is:

$$\begin{aligned} \max \quad & \sum_{i,j} w_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j} x_{ij} \leq 1, \\ & \sum_{i} x_{ij} \leq 1, \\ & x_{ij} \geq 0, \end{aligned}$$

with the understanding that

$$x_{ij} = 1 \Longrightarrow (i,j) \in M, \quad x_{ij} = 0 \Longrightarrow (i,j) \notin M.$$

The dual linear programming problem is:

$$\begin{aligned} \min \quad & \sum_{i} u_i + \sum_{j} v_j \\ \text{s.t.} \quad & u_i + v_j \geq w_{ij}, \\ & u_i \geq 0, \\ & v_j \geq 0. \end{aligned}$$

Orthogonality (complementary) conditions which are necessary and sufficient for optimality of primal and dual solutions are:

$$x_{ij} > 0 \Longrightarrow u_i + v_j = w_{ij}, \tag{5.1}$$

$$u_i > 0 \Longrightarrow \sum_{j} x_{ij} = 1, \tag{5.2}$$

$$v_j > 0 \Longrightarrow \sum_{i} x_{ij} = 1. \tag{5.3}$$

The Hungarian method maintains primal and dual feasibility at all times, and in addition maintains satisfaction of all orthogonality conditions, except condition (5.2). The number of such unsatisfied conditions is decreased monotonically during the course of computation.

The procedure is begun with the feasible matching $M = \emptyset$ and with the feasible dual solution $u_i = W$, where $W \geq \max\{w_{ij}\}$, and $v_j = 0$, for all i, j. These initial primal and dual solutions satisfy (5.1) and (5.3), but not (5.2).

At the general step, M is feasible, u_i and v_j are dual feasible, (5.1) and (5.3) are satisfied, but some of (5.2) are not. One then tries, by means of a labeling procedure, to find an augmenting path within the subgraph containing only edges (i, j) for which $u_i + v_j = w_{ij}$. In particular, an augmenting path is sought from an exposed node i in S for which $u_i > 0$. If such a path can be found, the new matching will be feasible, (5.1) and (5.3) continue to be satisfied, and one more of the conditions (5.2) will be satisfied than before. If augmentation is not possible, then a change of δ is made in the dual variables, by subtracting $\delta > 0$ from u_i for each labeled S-node i and adding δ to v_j to each labeled T-node j.

BIPARTITE WEIGHTED MATCHING ALGORITHM

Step 0 (*Start*) The bipartite graph $G = (S, T, E)$ and a weight w_{ij} for each $(i, j) \in E$ are given. Set $M = \emptyset$, $u_i = \max\{w_{ij}\}$ for each node $i \in S$, $v_j = 0$ and $\pi_j = +\infty$ for each node $j \in T$. No nodes are labeled.

Step 1. (*Labeling*)

(1.0) Give the label "$\emptyset$" to each exposed node in S.

(1.1) If there are no unscanned labels, or if there are unscanned labels, but each unscanned label is on a node i in T for which $\pi_i > 0$, then go to Step 3.

(1.2) Find a node i with an unscanned label, where either $i \in S$ or else $i \in T$ and $\pi_i = 0$. If $i \in S$, go to Step 1.3; if $i \in T$, go to Step 1.4.

(1.3) Scan the label on node i ($i \in S$) as follows. For each edge $(i, j) \notin M$ incident to i, if $u_i + v_j - w_{ij} < \pi_j$, then give node j the label "i" (replacing any existing label) and set $\pi_j = u_i + v_j - w_{ij}$. Return to Step 1.1.

(1.4) Scan the label on node i ($i \in T$) as follows. If node i is exposed, go to Step 2. Otherwise, identify the unique edge $(i, j) \in M$ incident to node i and give j the label "i". Return to Step 1.1.

Step 2 (*Augmentation*) An augmenting path has been found, terminating at node i (identified in Step 1.4). The nodes preceding node i in the path are identified by "backtracing". Augment M by adding to M all edges in the augmenting path that are not in M and removing from M those which are. Remove all labels from nodes. Let $\pi_j = \infty$, for each node $j \in T$. Return to Step 1.0.

Step 3 (*Change in Dual Variables*) Find

$$\delta_1 = \min\{u_i | \ i \in S\},$$

$$\delta_2 = \min\{\pi_j | \pi_j > 0, \ j \in T\},$$

$$\delta = \min\{\delta_1, \delta_2\}.$$

Subtract δ from u_i, for each labeled node $i \in S$. Add δ to v_j for each node $j \in T$ with $\pi_j = 0$. Subtract δ from π_j for each labeled node $j \in T$ with $\pi_j > 0$. If $\delta < \delta_1$ go to Step 1.1. Otherwise, M is a maximum weight matching and the u_i and v_j variables are an optimal dual solution. $\square$

- Complexity $O(m^2 n)$.

5.5 Nonbipartite Matching: Blossoms

1. Troubles

The task of finding augmenting paths far more difficult than the bipartite case ...

2. Blossom

3. Shrinking.

Suppose that G is a graph, M is matching, and b is a blossom of G with respect to M. Then M/B is the matching that results from M if we omit all matched edges of b and change any matched edge of the form $[v, u]$ with u a node of b to $[v, v_b]$.

Theorem. Suppose that, while searching for an augmenting path from a node u of a graph G with respect to a matching M, we discover a blossom b. Then there is an augmenting path from u in G with respect to M iff there is one from u (v_b if u is the basis of b) in G/b with respect to M/b. $\square$

4. Even and Kariv (1975) $(O(|V|^{2.5}))$
Micali and Vazichi (1980) $(O(|V|^{0.5}|E|))$

5. Weighted nonbipartite matching
Lawler (1976) $(O(|V|^3))$

5.6 Duality of Nonbipartite Matching

$G = (V, E)$.

$V_k \subset V$ is called an *odd* set of V if $|V|$ is odd.

An edge (i, j) is said to be *covered* by an odd set V_k if $|V_k| = 1$, i or j is in V_k and if $|V_k| \geq 3$, then both i and j are in V_k.

An odd *set cover* of G is $N := \{V_k \subseteq V\}$, where V_k are odd sets, $k = 1, \dots p$ and all the edges in E are covered by them.

Capacity

$$C(V_k) = \begin{cases} 1 & \text{if } |V_k| = 1, \\ \dfrac{|V_k| - 1}{2} & \text{if } |V_k| \geq 3. \end{cases}$$

$$C(N) = \sum_{V_k \in N} C(V_k).$$

Theorem. (Edmonds (1965)) For any graph, the maximum number of edges in a matching is equal to the minimum capacity of an odd-set cover.

5.7 The Chinese Postman Problem

Chinese postman problem (Gwan, 1962; Edmonds, 1965)

Euler path – a closed path crossing each edge exactly once.

Theorem (Euler, 1736) A graph G is Eulerian iff G is connected and each node of G has an even degree.

Proof. "Only if" is obvious. "If" part. Prove it by induction. We will find a cycle C of G. Consider $G \backslash C$. We will have a number of component subgraphs. Nodes of all these subgraphs have even degrees ... $\square$

Chinese postman problem: $G = (V, E)$, length c_{ij}, walk on each edge at least once such that the cost is minimized.

If G is Eulerian, an Euler path is the optimal postman tour. Suppose G is not Eulerian, some edges may be walked on twice.

Kwan's method: anti-cycle method. No efficient way to identify negative cycles.

Edmonds' Method

Step 1. Identify odd nodes as V^*. If no, go to Step 4.

Step 2. Compute the shortest paths between all odd nodes. ($O(|V|^3)$.

Step 3. Let $G = (V^*, E^*)$ with $u_{ij} = M - c_{ij}$, where M is a big number. Find a maximum weighted matching M^* in G^*. Duplicate edges in the shortest paths in M^* (An Euler Graph).

Step 4. Find the Euler path in G.

- Complexity $O(|V|3)$.