Notes for Operations Research & More

Lan Peng, PhD Student

Department of Industrial and Systems Engineering
University at Buffalo, SUNY
lanpeng@buffalo.edu

December 4, 2019

December 4, 2019

Contents

Chapter 1

Matchings

1.1 Maximum Matching

Definition 1.1.1 (Matching). Let $G = (V, E)$ be a graph, a **matching** is a subset of edges $M \subseteq E$ such that no two elements of M are adjacent. The two ends of an edge in M are said to be **matched under** M. A matching M saturates a vertex v, and v is said to be **M-saturated** or **M-covered**, if some edge of M is incident with v. Otherwise, v is **M-unsaturated** or **M-exposed**.

Definition 1.1.2 (Perfect matching, Maximum matching). If every vertex of G is M-saturated, then the matching is said to be **perfect matching**. M is a **maximum matching** if G has no matching M' with $|M'| > |M|$. Every perfect matching is maximum. The maximum matching does not necessarily to be perfect. Perfect matching and maximum matching may not be unique.

Definition 1.1.3 (M-alternating). An **M-alternating** path in G is a path whose edges are alternately in $E \setminus M$ and M.

Definition 1.1.4 (M-augmenting). An **M-augmenting** path in G is an M-alternating path whose origin and terminus are M-unsaturated.

Lemma 1.1. *Every augmenting path P has property that let $M' = P \Delta M = (M \cup P) \setminus (M \cap P)$ then M' contains one more edge then M*

The following path is an M-augmenting path
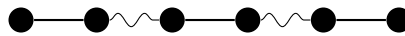
The following path is $M' = P \Delta M (M \cup P) \setminus (M \cap P)$ and all the vertices are M-saturated.

Theorem 1.2 (Berge, 1957). *A matching M in a graph G is maximum iff G has no M-augmenting path.*

Proof. ($\Rightarrow$) It is clear that if M is maximum, it has no augmenting paths since otherwise by problem claim we can increase by one.
($\Leftarrow$) Suppose M is not maximum and let M' be a bigger matching. Let $A = M \Delta M'$ now no vertex of G is incident to more than two members of A. For otherwise either two members of M or two members of M' would be adjacent. Contradict the definition of matching. It follows that every component of the edges incident subgraph $G[A]$ is either an even cycle with edge augmenting in $M \Delta M'$ or else A path with edges alternating between M and M'.
Since $|M'| \geq |M|$ then the even cycle cannot help because exchanging M and M' will have same cardinality.
The path case implies that p is alternating in M and since $|M'| > |M|$ the end arc exposed so that p is augmenting. $\square$

Definition 1.1.5 (Vertex-cover). The **vertex-cover** is a subset of vertices X such that every edge of G is incident to some member of X.

Lemma 1.3. *The cardinality of any matching is less than or equal to the cardinality of any vertex cover.*

Proof. Consider any matching. Any vertex cover must have nodes that at least incident to the edges in the matching. Since all the edges in the matching are disjointed, so for a single node can at most cover one edge in the matching. If the matching is not perfect, for the edges that not in the tree, they may or may not be possible to be covered by the nodes incident to the edges in the matching, with an easy triangle graph example, we can prove this lemma. □

Theorem 1.4 (König Theorem). *If G is bipartite, the cardinality of the maximum matching is equal to the cardinality of the minimum vertex cover.*

Proof. Let G be a bipartite graph, $G = (V, E)$ where $V = X \cup Y$ as X and Y are two disjointed sets of vertices. Let M be a maximum matching on G. For each edge in M, denoted by $e_i = a_i b_i$ where $e_i \in M$, $a_i \in A$ and $b_i \in B$ and $A = \{a_i : e_i \in M\} \subseteq X$, and $B = \{b_i : e_i \in M\} \subseteq Y$. Therefore, we can partition X by A and $U = X \setminus A$, partition Y by B and $W = Y \setminus B$.
We can further partition the matching M into M_1 and M_2. For all the edges in M_1 can be included into an M-alternating path starts from a vertex in U (which includes the edges directly linked to vertices in U), and $M_2 = M \setminus M_1$. For edges in M_1, we take the ends of edges in B in the vertex cover, denoted by B_1, take the ends of edges in A as a subset denoted by $A_1 \subseteq A$. For the edges in M_2, we take the ends of edges in A in the vertex cover, denoted by A_2, and the ends of edges in B as a subset denoted by $B_2 \subseteq B$.
We claim that all the vertices in U can only be connected to vertices in B_1 and vertices in W can only be connected to vertices in A_2.
$U \subset X$ connects to vertices in B_1 by definition. If vertices in $W \subset Y$ is connected to vertices in A_1, then we will have M-augmenting path which is contradicted to the assumption that M is maximum matching. □

The following is an example. Where the edge in the matching that accessible from members of $U = \{1, 2\}$ in an M-alternating path is edge $3a, 4b, 5c, 6d$.
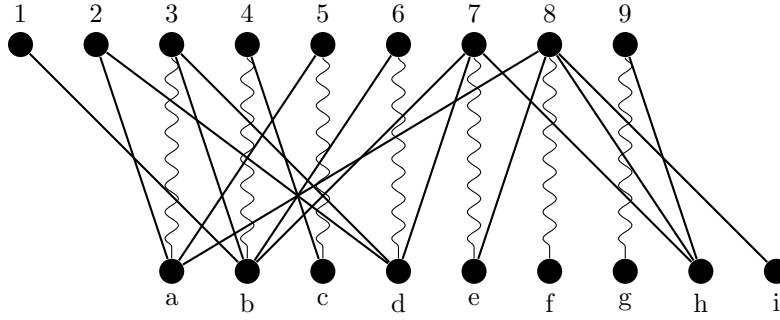
In which $U = \{1, 2\}$, $M_1 = \{3a, 4b, 5c, 6d\}$. $U = \{1, 2, \}, A_1 = \{3, 4, 5, 6\}, A_2 = \{7, 8, 9\}, W = \{h, i\}, B_1 = \{a, b, c, d\}, B_2 = \{e, f, g\}$. The vertex cover is $\{a, b, c, d, 7, 8, 9\}$.
The above theorem does not apply to non-bipartite graph. The following is an example
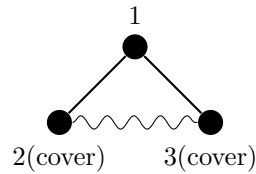
1.2   Maximum Matching Algorithm

Definition 1.2.1 (M-alternating tree). An **M-alternating tree** T is a rooted tree satisfied the following condition:

- The root r is M-unsaturated

- The unique path from r to any vertex T is M-alternating

- Every vertex in T, except r is incident to a matching edge of T

A vertex x of T is called **inner** if (r, s)-path in T has an odd number of edges. Otherwise x is called **outer**.

Lemma 1.5. *Let M be a matching in G and let T be an M-alternating tree with root r, then the following conclusion hold*

- *If $v \neq r$ is an outer vertex and p is the unique (r, v)-path in T then the edge of p incident to v is in M*

- *The number of inner vertices in T equals the number of matching edges in T.*

Definition 1.2.2 (Alternating forest)**.** An **alternating forest** is a forest of G where every components is an alternating tree. An alternating forest (F, e) is an alternating forest to be then with an edge $e = M, V$ where M and V are outer vertices contained in two distinct components of F.

Example (Hungarian forest)**.** A **Hungarian forest** F is an alternating forest containing all exposed vertices of G and such that the outer vertices of F are adjacent in G only to inner vertices of F

Definition 1.2.3 (Augmenting forest)**.** An **augmenting forest** (F, e) where F is an augmenting forest and $e = uv$ connects two outer vertices in distinct components of F.

The plan is to grow an alternating forest that eventually become augmenting or Hungarian. Augmenting forest will increase the cardinality of the match, Hungarian implies that you have found optimal maximum cardinality matching.

Theorem 1.6. *Let (F, e) be an augmenting forest. And let T_1 and T_2 be the two components of F containing an end of e. Let p_i be the unique path in T_i from the root to the end of e, then $p_1 e p_2^{-1}$ is an augmenting path.*

Theorem 1.7. *If F is a Hungarian forest for some matching M then M is a maximum match.*

The above theorems suggest a method for computing maximum matching. Let M be a matching of G and let F be an alternating forest in G made up of all M-exposing vertices. If F happens to be Hungarian, stop with the max matching M. If (F, e) for some e is augmenting then we increase our matching by 1 and start process.
Suppose F is neither Hungarian nor augmenting, by definition, there must be an edge e incident to an outer vertex of F to no inner vertex of F. But e cannot be incident to two outer vertices of F in distinct components, since (F, e) is not augmenting. Hence there are only two cases:
Case 1: $e = uv$ where u is outer in F and v is not outer in F. The only way its possible if v is covered. Augmenting F to M will increase the matching.
Case 2: $e = uv$ where u and v are outer vertices in F. Let r be the root of the component of F containing u and v, let p_u and p_v be (r, u)-path and (r, v)-path in F. Let b be the last vertex these two paths have in common and let p be the (u, v)-path in F, let p'_u be $(b - u)$-path and p'_v be the (b, v)-path respectively. Let n be the length of p_u, let m be the length of p_v, let k be the length of (r, b)-path. Let c be the cycle $(p'_u e p'_v{}^{-1})$. Number of vertices in c is $n + m - 2k + 1$, n, m are even, so c is always an odd length cycle. If G has no odd cycles, we call those graph bipartite. To this case can not happen in bipartite graph so algorithm without case 2 will solve bipartite matching. If a graph has no odd cycles, i.e., bipartite, then we have an algorithm using augmenting forest and Hungarian forest and case 1.
We now have to deal with odd cycles. The idea is to "shrink" add cycles to a super node
Let $S \subseteq E(G)$, denote by $G : S$ the subgraph with edge set S

$$G : S = G \setminus (E(G) - S) \tag{1.1}$$

The contraction of S to be the $G \setminus S$ with $E(G/S) = E(G) - S$. $V(G/S)$ to be the components of $G : S$ and if $e \in E(G/S)$ then the ends of e in G/S are in components of $G : S$ containing both ends in G
Let network to general case 2. b is outer vertex. Let B be the set of edges of cycle C, we call B a **blossom**. We propose to replace M by $M - B$, G by G/B and F by F/B

1.3 Edmonds's Blossom Algorithm

$O(|V|^4)$ - Non-bipartite matching is one of very few problems in P, for which LP relaxation will not provide optimal solution.

1.4   Hall's Marriage Theorem

1.5   Transversal Theory

1.6   Menger's Theorem

1.7   The Hungarian Algorithm