Lecture notes for Mar 20, 2023
Weighted bipartite matching

Chun-Hung Liu

March 20, 2023

A *weak fractional vertex-cover* of a graph G is a function $f : V(G) \to \mathbb{R}$ such that for every $e = uv \in E(G)$, $f(u) + f(v) \geq w(e)$.

Proposition 1 *Let (G, w) be a weighted graph. If M is a perfect matching of G and f is a weak fractional vertex-cover of G, then $\sum_{e \in M} w(e) \leq \sum_{v \in V(G)} f(v)$.*

Proof. For each edge $e \in M$, let u_e, v_e be the ends of e. Since M is a perfect matching, $\{u_e, v_e : e \in M\} = V(G)$. Then $\sum_{e \in M} w(e) \leq \sum_{e \in M}(f(u_e) + f(v_e)) = \sum_{v \in V(G)} f(v)$. ■

The *Assignment Problem* is: given a nonnegative weighted balanced complete bipartite graph $(K_{n,n}, w)$, find a maximum weighted matching. It is a special case for the Weighted Bipartite Matching Problem, but we will show that solving the Assignment Problem is equivalent to solving the Weighted Bipartite Matching Problem.

==================================

Hungarian method for the Assignment Problem
Input: A complete bipartite graph G with a bipartition $\{A, B\}$ with $|A| = |B|$, and a function $w : E(G) \to \mathbb{R}_{\geq 0}$.
Output: A matching M in G and a weak fractional vertex-cover $f : V(G) \to \mathbb{R}$ for (G, w) such that $\sum_{e \in M} w(e) = \sum_{v \in V(G)} f(v)$.
Procedure:

Step 1: For every $v \in A$, define $f(v) = \max_{e \in \delta(v)} w(e)$. For every $v \in B$, define $f(v) = 0$.

Step 2: Let G_f be the graph with $V(G_f) = V(G)$ and $E(G_f) = \{e = uv \in E(G) : f(u) + f(v) = w(e)\}$. Find a maximum (unweighted) matching M in G_f and a minimum vertex-cover S of G_f. If M is a perfect matching in G_f, then output M and f and stop. Otherwise, do Step 3.

Step 3: Let $\epsilon = \min_{e=uv\in\delta(A-S)-(E(G_f)\cup\delta(B\cap S))}(f(u) + f(v) - w(e))$. Redefine f as follows: for every $v \in A - S$, define $f(v) = f(v) - \epsilon$; for every $v \in B \cap S$, define $f(v) = f(v) + \epsilon$. Repeat Step 2.

=================================

Lemma 2 *During the entire process, f is always a weak fractional vertex-cover for (G, w). In particular, $\epsilon \geq 0$ during the process.*

Proof. Let f_1 be the function f at some moment. Assume that f_1 is a weak fractional vertex-cover for (G, w). Let f_2 be the new function f when it is updated. It suffices to show that $f_2(u) + f_2(v) \geq w(e)$, where $e = uv$ is an edge of G, and $u \in A$ and $v \in B$. Since f_1 is a weak fractional vertex-cover for (G, w), the ϵ used for defining f_2 is nonnegative. So $f_2|_B \geq f_1|_B$.

If $u \in A \cap S$, then $f_2(u) = f_1(u)$, so $f_2(u) + f_2(v) \geq f_1(u) + f_1(v) \geq w(e)$. So we may assume $u \in A - S$. If $v \in B \cap S$, then $f_2(u) + f_2(v) = (f_1(u) - \epsilon) + (f_1(v) + \epsilon) = f_1(u) + f_1(v) \geq w(e)$. So we may assume $v \in B - S$. Since S is a vertex-cover of G_f, and $u, v \notin S$, we know $e \notin E(G_f)$. So e is an edge in $\delta(A - S) - (E(G_f) \cup \delta(B \cap S))$. Hence $\epsilon \leq f_1(u) + f_1(v) - w(e)$. So $f_2(u)+f_2(v) \geq f_1(u)-\epsilon+f_1(v) \geq f_1(u)-(f_1(u)+f_1(v)-w(e))+f_1(v) = w(e)$. ■

Lemma 3 *Let f_1, M_1, S_1 be the functions f, M, S, respectively, in a round for Step 2. Let f_2, M_2, S_2 be the functions f, M, S, respectively, in the next round for Step 2. If we use the augmenting path argument for the corresponding network to find M and S, then either $|M_2| > |M_1|$, or $|M_2| = |M_1|$ and $|S_2 \cap B| > |S_1 \cap B|$.*

Proof. Since $\epsilon \geq 0$ by Lemma 2, every edge in $E(G_{f_1}) - E(G_{f_2})$ is between $A \cap S_1$ and $B \cap S_1$. Since M_1 is a matching for G_{f_1} and S_1 is a vertex-cover for G_{f_1} with $|M_1| = |S_1|$, we know every edge in M_1 is either between $A \cap S_1$ and $B - S_1$ or between $A - S_1$ and $B \cap S_1$. So $M_1 \subseteq E(G_{f_2})$ is a matching in G_{f_2}. Since M_2 is a maximum (unweighted) matching in G_{f_2}, $|M_2| \geq |M_1|$. So we may assume $|M_2| = |M_1|$, for otherwise we are done. As we use the augmenting path argument and M_1 is a matching in $G_{f_1} \cap G_{f_2}$, $M_1 = M_2$.

Note that $|M_1| < |A|$, for otherwise the algorithm stops and should not produce M_2. By König's theorem, $|S_1| = |M_1| < |A|$. Since G is a complete bipartite graph, there exists an edge $e \in E(G_{f_2}) - E(G_{f_1})$ between $A - S_1$ and $B - S_1$. Let $u \in A - S_1$ and $v \in B - S_1$ be the ends of e.

Recall the network $(D_f, s, t, 1)$ we construct is the one obtained from G_f by directing the edges from A to B, adding new vertices s, t and adding edges from s to A and edges from B to t. Note that the matching M comes from deleting s and t from the internally disjoint path from s to t given from the maximum (integral) flow, and $S = (A - C) \cup S' \cup (B \cap C)$, where C is a minimum cut for $(D_f, s, t, 1)$ and S' is the subset of $A \cap C$ incident with an edge in M. In the residue graph, C is the set of vertices reachable from s. So every vertex in $S \cap B$ are exactly the vertices in B reachable from s (i.e. $B \cap S = B \cap C$). And every vertex in $A - S$ is contained in $A \cap C$ and hence is reachable from s. Moreover, every vertex in $A \cap S$ is matched by M to a vertex in $B - S$, and every vertex in $B \cap S$ is matched by M to a vertex in $A - S \subseteq A \cap C$.

Note that D_{f_2} is obtained from D_{f_1} by deleting edges in $E(G_{f_1}) - E(G_{f_2})$ and adding edges in $E(G_{f_2}) - E(G_{f_1})$, and $M_1 = M_2$.

Now we consider the case $f = f_1$, and we call the cut in the network C_1. For every directed path P in the residue graph from s to a vertex of C_1, $P - \{s, t\}$ is an M_1-alternating path from an M_1-unsaturated vertex in A, and $V(P) \cap V(B)$ are reachable and hence contained in $S_1 \cap B$, so P only uses edges between $A - S_1$ and $B \cap S_1$, and hence P is still in the residue graph for f_2. So every vertex in B reachable for f_1 is also reachable for f_2. That is, $C_1 \cap B \subseteq C_2 \cap B$.

Recall that $e = uv \in E(G_{f_2})$ is an edge with $u \in A - S_1$ and $v \in B - S_1$. So $u \in C_1$, but $v \notin C_1$. Since $M_1 = M_2$, if u is saturated by $M_1 = M_2$, say $uu' \in M_1 = M_2$, then (u', u) is an edge in the residue graph for both f_1 and f_2 and $u' \in C_1 \cap B \subseteq C_2 \cap B$, so $u \in C_1 \cap C_2$; if u is unsaturated by $M_1 = M_2$, then u is reachable from s in the residue graph for both f_1 and f_2 by an edge. So u is reachable for both f_1 and f_2 in either case. But $uv \in E(G_{f_2}) - M_1 = E(G_{f_2}) - M_2$, v is in the reachable set for f_2. Therefore, the reachable set C_2 for f_2 is strictly bigger than the one C_1 for f_1. That is, $C_2 \cap B \supset C_1 \cap B$. Note that $S_1 \cap B = C_1 \cap B \subset C_2 \cap B = S_2 \cap B$. ■

Theorem 4 *If we use the augmenting path argument in Step 2, then Hungarian method for Assignment Problem correctly outputs a matching M in G and a weak fractional vertex-cover $f : V(G) \to \mathbb{R}$ for (G, w) such that*

$\sum_{e\in M} w(e) = \sum_{v\in V(G)} f(v)$ *in time* $O(|V(G)|^4)$. *In particular,* M *is a matching in* (G, w) *with maximum weight.*

Proof. By Lemma 3, the matching size must increase by executing Step 2 at most $|B|$ times, so the algorithm will stop by executing Step 2 at most $|B|^2 \leq |V(G)|^2$ times. Note that we only have to find augmenting path $O(|V(G)|^2)$ times in total for Step 2, each taking time $O(|E(G)|) = O(|V(G)|^2)$. And we can update the residue graph for the augmenting path argument in each Step 3 in time $O(|E(G)|) = O(|V(G)|^2)$, and we do Step 3 at most $O(|V(G)|^2)$ times. So the total running time is $O(|V(G)|^4)$.

Since M is a perfect matching in G_f by the definition of the algorithm and f is a weak fractional vertex-cover of (G, w) by Lemma 2, we know $\sum_{e\in M} w(e) = \sum_{u,v\in V(M)}(f(u) + f(v)) = \sum_{x\in V(G)} f(x)$. And by Proposition 1, since $w \geq 0$, for every matching M^* of (G, w), $\sum_{e\in M^*} w(e) \leq \sum_{x\in V(G)} f(x) = \sum_{e\in M} w(e)$. So M is a maximum weighted matching in (G, w). ■

Note that the time complexity can be improved, as we will see in the next section.

Now we can solve the maximum weighted matching problem for general bipartite graphs.

==============================

Algorithm for Maximum Weighted Bipartite Matching
Input: A simple bipartite graph G with a bipartition $\{A, B\}$, and a function $w : E(G) \to \mathbb{R}$.
Output: A matching M in G with maximum $\sum_{e\in M} w(e)$.
Procedure:

Step 0: Let G_0 be the graph obtained from G by deleting all edges e of G with $w(e) < 0$.

Step 1: By symmetry, we may assume $|A| \leq |B|$. Let G' be the complete bipartite graph obtained from G_0 by adding $|B| - |A|$ vertex into A to form a new set A' and add edges such that $\{A', B'\}$ is the bipartition of G' with $A' \supseteq A$ and $B' = B$ with $|A'| = |B'|$. Assign $w(e) = 0$ for every edge in $E(G') - E(G_0)$.

Step 2: Use an algorithm for the Assignment Problem to find a maximum weighted matching M' of (G', w').

Step 3: Let $M = M' \cap E(G_0)$ and output M.

================================

Theorem 5 *Given a simple bipartite graph G and a weight function $w : E(G) \to \mathbb{R}$, a matching in (G, w) with maximum weight can be found in time $O(|V(G)|^4)$.*

Proof. Since every matching with maximum weight in (G, w) does not use any edge with negative weight, we know maximum weighted matching in (G, w) are exactly the maximum weighted matching in (G_0, w). Since every edge in $E(G') - E(G_0)$ has weigh 0, the weight of a maximum weighted matching in (G_0, w) equals the weight of a maximum weighted matching in (G', w'). And clearly the matching M' mentioned in the algorithm is a maximum weighted matching in (G', w'). Since every edge in $E(G') - E(G)$ has weight 0, $\sum_{e \in M} w(e) = \sum_{e \in M'} w(e)$. Hence M is a maximum weighted matching (G, w).

The time complexity follows from the time complexity for the Assignment problem. ■