Algorithm to compute Fair Matching

Saptarshi Sadhukhan

April 2024

1 Introduction

- Let $G = (A \cup B, E)$ be a bipartite graph, where every vertex ranks its neighbors in an order of preference (with ties allowed) and let r be the largest rank used.
- A matching M is fair in G if it has maximum cardinality, subject to this, M matches the minimum number of vertices to rank r neighbors, subject to that, M matches the minimum number of vertices to rank $(r-1)$ neighbors, and so on.

Here is our algorithm to compute a fair matching in bipartite graph G.

2 Main Algorithm

Input : $G = (A \cup B, E)$ and r is the worst (largest) rank used in any preference list.
Recall : r is the worst rank in the problem instance, and r^* is the worst rank in a fair matching.
Proposition 1 : M_j and y^j are the optimal solutions to the primal and dual programs of the j-th iteration, iff the following holds:

1. if u is unmatched in M_j (thus u has to be outside K_j), then $y_u^j = 0$;
2. if $e = (u, v) \in M_j$, then $y_u^j + y_v^j = w_j(e)$.

We present an algorithm that runs for r iterations and we show how our algorithm terminates in r^* iterations.

Algorithm:

1. Initialization. Let $G_0 = G$ and $K_{-1} = \emptyset$.
2. For $j = 0$ to $r - 1$ do

a. Find the optimal solution $\left\{y_u^j\right\}_{u \in A \cup B}$ to the dual program of the $j+1$-st iteration.

b. Delete from G_j every edge (a, b) such that $y_a^j + y_b^j > w_j(e)$. Call this subgraph G_{j+1}.

c. Add all vertices with positive dual values to the critical set, i.e., $K_j = K_{j-1} \cup \{u\}_{y_u^{j-1} > 0}$.

3. Return the optimal solution to the primal program of the last iteration.

- The solution given by above algorithm is a maximum weight matching in the graph G_{r-1} under the weight function w_{r-1} such that this matching matches all vertices in K_{r-2}.
- By Proposition 1, this is a matching in subgraph G_r that matches all vertices in K_{r-1}.

The following lemma guarantees that the algorithm is never stuck in any iteration (due to the infeasibility of the primal/dual.

Lemma 1 : The primal and dual programs of the $j+1$-th iteration are feasible, for $0 \leq j \leq r-1$.

Following proves the correctness of our algorithm.

Lemma 2 : *For every $0 \leq j \leq r-1$, the following hold:*

1. *any matching M in G_j that matches all $v \in K_{j-1}$ is j-optimal;*
2. *conversely, a j-optimal matching in G is a matching in G_j that matches all $v \in K_{j-1}$.*

Proof : By induction.
Base case : $j = 0$. We have that $G_0 = G$ and $K_{-1} = \emptyset$. As all matchings are by default 0-optimal, the lemma holds directly.

For the induction step, $j \geq 1$, suppose that the lemma holds up to $j-1$. As $K_{j-1} \supseteq K_{j-2}$ and G_j is a subgraph of G_{j-1}, M is a matching in G_{j-1} that matches all vertices of K_{j-2}.
By induction hypothesis, M is $(j-1)$-optimal. For each edge $e = (a,b) \in M$, e must be a tight edge in the j-th iteration, to be present in G_j

$$y_a^{j-1} + y_b^{j-1} = w_{j-1}(e)$$

Also, $K_{j-1} \supseteq \{u\}_{y_u^{j-1} > 0}$,

$$w_{j-1}(M) = \sum_{e=(a,b)\in M} w_{j-1}(e) = \sum_{e=(a,b)\in M} y_a^{j-1} + y_b^{j-1} \geq \sum_{u \in A\cup B} y_u^{j-1}$$

where the final inequality holds because all vertices v with positive y_v^{j-1} are matched in M.
By LP duality, M must be optimal in the primal program of the j-th iteration. So the j-th primal program has optimal solution of value $w_{j-1}(M)$.

By definition, OPT is also $(j-1)$-optimal.
By (2) of IH, OPT is a matching in G_{j-1} and OPT matches all vertices in K_{j-2}.
$\therefore$ OPT is a feasible solution of the primal program in the j-th iteration.
Thus, $w_{j-1}(\mathrm{OPT}) \leq w_{j-1}(M)$. but this is not possible,
(else signature $(M) \succ$ signature(OPT), as both signatures have the same first $j-1$ coordinates).
$\therefore w_{j-1}(\mathrm{OPT}) = w_{j-1}(M) \Rightarrow M$ is j-optimal as well. Proved (1).

To show (2), let M' be a j-optimal matching in G. Hence, it is also $(j-1)$-optimal and by (2) of the IH, it is a matching in G_{j-1} that matches all vertices in K_{j-2}.
$\Rightarrow$ M' is a feasible solution to the primal program of the j-th iteration.
Since signature (M') has j-th coordinate $= w_{j-1}(\mathrm{OPT})$,

M' has to be an optimal solution to the primal program of the j-th iteration;
(else theres j-optimal matching with a larger value than $w_{j-1}(\text{OPT})$ in the j-th coordinate of its signature, which contradicts the optimality of OPT.)
By Proposition 1.2, all edges of M' are present in G_j.
By Proposition 1.1, all vertices $u \notin K_{j-2}$ with $y_u^{j-1} > 0$, (i.e. all $v \in K_{j-1} \setminus K_{j-2}$) have to be matched by M'. Proved (2).

Our algorithm returns a matching in G_r that matches all vertices in K_{r-1}.
$\therefore$ From (2) of above Lemma that this matching is r-optimal.
Thus the matching returned by our algorithm is fair.

Bounding the running time of the algorithm :
We showed how to solve the dual program in $O(m\sqrt{n})$ time once we have the solution to the primal program and we have seen that the primal program can be solved in $O(m\sqrt{n}\log n)$ time.

Improving the running time :
The algorithm can be modified so that it terminates in r^* iterations, where r^* is the largest rank used in OPT. The value of r^* can be computed at the start of our algorithm as follows.

- Let M^* be a maximum cardinality matching in G. The value r^* is the smallest index j such that the subgraph $\bar{G}_j$ admits a matching of size $|M^*|$, where $\bar{G}_j$ is obtained by deleting all edges $e = (a, b)$ from G where either a or b (or both) ranks the other as a rank $> j$ neighbor.
- We compute r^* by first computing M^* and then computing a maximum cardinality matching in $\bar{G}_1, \bar{G}_2, \ldots$ and so on till we see a subgraph $\bar{G}_j$ that admits a matching of size $|M^*|$. This index $j = r^*$ and it can be found in $O\left(r^* m\sqrt{n}\right)$ time.

We showed how to solve the dual program in $O\left(m\sqrt{n}\right)$ time after we solve the primal program and we have seen that the primal program can be solved in $O\left(m\sqrt{n}logn\right)$ time.

Theorem : A fair matching M in $G = (A \cup B, E)$ can be computed in $O\left(r^* m\sqrt{n}\right)$ time, where r^* is the largest rank incident on an edge in M, $n = |A \cup B|$, $|m = |E|$