Topics in Algorithms - Assignment 1

Kishlaya Jaiswal

February 15, 2020

Exercise 1

Proof. Consider the following preference lists:

$$\begin{aligned} m_1 &: w_1, w_2, w_3 \\ m_2 &: w_2, w_3, w_1 \\ m_3 &: w_3, w_1, w_2 \end{aligned} \qquad \begin{aligned} w_1 &: m_2, m_3, m_1 \\ w_2 &: m_3, m_1, m_2 \\ w_3 &: m_1, m_2, m_3 \end{aligned}$$

Then the *man-optimal* matching is:

$$\begin{aligned} m_1 &: w_1, w_2, w_3 \\ m_2 &: w_2, w_3, w_1 \\ m_3 &: w_3, w_1, w_2 \end{aligned} \qquad \begin{aligned} w_1 &: m_2, m_3, m_1 \\ w_2 &: m_3, m_1, m_2 \\ w_3 &: m_1, m_2, m_3 \end{aligned}$$

The *woman-optimal* matching is:

$$\begin{aligned} m_1 &: w_1, w_2, w_3 \\ m_2 &: w_2, w_3, w_1 \\ m_3 &: w_3, w_1, w_2 \end{aligned} \qquad \begin{aligned} w_1 &: m_2, m_3, m_1 \\ w_2 &: m_3, m_1, m_2 \\ w_3 &: m_1, m_2, m_3 \end{aligned}$$

But we have another stable matching:

$$\begin{aligned} m_1 &: w_1, w_2, w_3 \\ m_2 &: w_2, w_3, w_1 \\ m_3 &: w_3, w_1, w_2 \end{aligned} \qquad \begin{aligned} w_1 &: m_2, m_3, m_1 \\ w_2 &: m_3, m_1, m_2 \\ w_3 &: m_1, m_2, m_3 \end{aligned}$$

□

Exercise 2

Proof. Consider the following preference lists (with the circled people as *man-optimal* stable matching):

$$m_1 : w_1, w_2, w_3 \qquad w_1 : m_2, m_1, m_3$$
$$m_2 : w_2, w_1, w_3 \qquad w_2 : m_1, m_3, m_2$$
$$m_3 : w_2, w_3, w_1 \qquad w_3 : m_3, m_1, m_2$$

But clearly, we can re-pair (m_1, w_1) and (m_2, w_2) to get an unstable matching which is not *pareto-optimal*:

$$m_1 : w_1, w_2, w_3 \qquad w_1 : m_2, m_1, m_3$$
$$m_2 : w_2, w_1, w_3 \qquad w_2 : m_1, m_3, m_2$$
$$m_3 : w_2, w_3, w_1 \qquad w_3 : m_3, m_1, m_2$$

□

Exercise 3

Proof. Consider the following preference lists (with the circled people as *man-optimal* stable matching), then no man gets matched to their first choice.

$$m_1 : w_1, w_2, w_3 \qquad w_1 : m_2, m_1, m_3$$
$$m_2 : w_2, w_1, w_3 \qquad w_2 : m_1, m_3, m_2$$
$$m_3 : w_2, w_3, w_1 \qquad w_3 : m_3, m_1, m_2$$

The following preference list gives a counter-example for: there doesn't always exist a stable matching that matches some person to their first choice (because both the *man-optimal* and *woman-optimal* matchings are the same) as follows:

$$m_1 : w_1, w_2, w_3, w_4$$
$$m_2 : w_2, w_1, w_3, w_4$$
$$m_3 : w_2, w_1, w_3, w_4$$
$$m_4 : w_3, w_4, w_1, w_2$$

$$w_1 : m_4, m_3, m_2, m_1$$
$$w_2 : m_4, m_1, m_3, m_2$$
$$w_3 : m_1, m_2, m_3, m_4$$
$$w_4 : m_1, m_4, m_2, m_3$$

□

Exercise 4

Proof. **(a)** Suppose a pair (m, w) is deleted during a run of the GS algorithm (with men-proposing), that is m was deleted from $w's$ list and w was deleted from $m's$ list. This means some man m' must have proposed w who was more preferable (to w) than m.
But we know that during a run of GS algo, if a woman gets engaged to someone during the run, then she can only get a better partner later during the run. Therefore, w is engaged to m' (or better) in the *woman-pessimal* matching (GS outpust *man-optimal*, *women-pessimal* matching). Thus, m can't be a partner of w in any stable matching. Hence, (m, w) can't form a stable pair.
Furthermore, since in any matching w is engaged to someone strictly better m, (m, w) can't block any stable matching.
Hence the stable matchings are not affected by this operation.

(b) No. Consider the following preference lists (with the circled people as *man-optimal* stable matching):

$$m_1 : w_1, w_2, w_3$$
$$m_2 : \cancel{w_1}, w_2, w_3$$
$$m_3 : \cancel{w_1}, \cancel{w_2}, w_3$$

$$w_1 : m_1, \cancel{m_2}, \cancel{m_3}$$
$$w_2 : m_1, m_2, \cancel{m_3}$$
$$w_3 : m_1, m_2, m_3$$

We can check that *woman-optimal* matching is also the same and hence there is exactly one stable matching. This tells us that the "not-crossed-out" pair (m_1, w_2) cannot appear in any of the stable matchings.

(c) No. Consider the following preference lists:

$$m_1 : w_3, w_2, w_1 \qquad w_1 : m_1, m_3, m_2$$
$$m_2 : w_3, w_1, w_2 \qquad w_2 : m_2, m_1, m_3$$
$$m_3 : w_2, w_1, w_3 \qquad w_3 : m_3, m_1, m_2$$

The reduced list after first run of GS (man-proposing) will be (with the circled people as *man-optimal* stable matching):

$$m_1 : w_3, w_2, w_1 \qquad w_1 : m_1, m_3, m_2$$
$$m_2 : w_1, w_2 \qquad w_2 : m_2, m_1, m_3$$
$$m_3 : w_2, w_1, w_3 \qquad w_3 : m_3, m_1$$

Further reducing these lists using GS (woman-proposing) we get (with the circled people as *woman-optimal* stable matching):

$$m_1 : w_3, w_2, w_1 \qquad w_1 : m_1, m_3, m_2$$
$$m_2 : w_1, w_2 \qquad w_2 : m_2, m_1, m_3$$
$$m_3 : w_2, w_1, w_3 \qquad w_3 : m_3, m_1$$

But now note that, (m_1, w_2) cannot be a stable pair. For the sake of contradiction, suppose it is. Then the only option is to pair m_2 with w_1 and m_3 with w_3. But then (m_3, w_1) is a blocking pair.

□

Exercise 5

(i) Let R_1, R_2 and R_3 be any three stable matchings.
Claim: If $(x, y) \in R_1$ but $(x, y) \notin R_2$, then one of $\{x, y\}$ prefers R_1 and the other one prefers R_2.
Proof: Let S_1 be the set of people who prefer R_1 to R_2 and S_2 be the set of people who prefer R_2 to R_1. Then for any $x \in S_1$, $R(x) \in S_2$ otherwise (x, y) blocks R_2 and so $|S_1| \leq |S_2|$. Similarly, for any $x \in S_2$, $R_2(x) \in S_1$ and so $|S_1| \leq |S_2|$. Hence $|S_1| = |S_2|$ and it follows that one of $\{x, y\}$ prefers R_1 and the other one prefers R_2.

To conclude that taking medians results in a matching, let y be the median choice of x. Then note that if x prefers R_i to R_j to R_k, then y prefers R_k to R_j to R_i and hence x is the median choice of y.

Thus, taking the median partners w.r.t R_1, R_2 and R_3 results in a matching. Now we shall show that it is indeed a stable matching.
Suppose not, then there is a blocking pair (a, b) w.r.t to the median matching. Let's say $R_i(a)$ was the median partner of a and $R_j(b)$ was the median partner of b. Then

$$a : \ldots b \ldots R_i(a) \ldots$$

$$b : \ldots a \ldots R_j(b) \ldots$$

Now if $(i = j)$ or $(b : \ldots a \ldots R_j(b) \ldots R_i(b) \ldots)$ then (a, b) blocks R_i which is a contradiction. So, we must have

$$b : \ldots R_i(b) \ldots a \ldots R_j(b) \ldots$$

Similarly,

$$a : \ldots R_j(a) \ldots b \ldots R_i(a) \ldots$$

Consider third matching R_k. Since $R_i(a)$ was median partner of a and $R_j(b)$ was median partner of b, we must have

$$a : \ldots R_j(a) \ldots b \ldots R_i(a) \ldots R_k(a) \ldots$$

$$b : \ldots R_i(b) \ldots a \ldots R_j(b) \ldots R_k(b) \ldots$$

And hence (a, b) blocks R_k, which is a contradiction to the stability of R_k.

(ii) Consider the set $\mathscr{S} = \{P(R_0) \oplus P(R) |$ stable matching $R\}$. We will first show that $\mathscr{S}$ is closed under intersection.
For that let us setup some notation first. Let $\mathscr{R} = \{R_i \mid 0 \leq i \leq m\}$ be the set of all stable matchings, and let $P_i = P(R_0) \oplus P(R_i)$ then $\mathscr{S} = \{P_i \mid 0 \leq i \leq m\}$.
Now, let $P_i, P_j \in S$. Let R be median (stable) matching of R_0, R_i, R_j. We claim that $P_i \cap P_j = P(R_0) \oplus P(R)$.
To show $P_i \cap P_j \subseteq P(R_0) \oplus P(R)$, let $(x, y) \in P_i \cap P_j$ then:
Case 1: x prefers R_0 over $R_i \implies x$ prefers R_0 over R_j as $(x, y) \in P_j$. WLOG assume x prefers R_i over R_j

$$x : \ldots R_0(x) \ldots y \ldots R_i(x) \ldots R_j(x) \ldots$$

Thus, median partner of x is $R_i(x)$ and hence $(x, y) \in P(R_0) \oplus P(R)$.
Case 2: x prefers R_i over $R_0 \implies x$ prefers R_j over R_0 as $(x, y) \in P_j$. WLOG assume x prefers R_i over R_j

$$x : \ldots R_i(x) \ldots R_j(x) \ldots y \ldots R_0(x) \ldots$$

Thus, median partner of x is $R_j(x)$ and hence $(x, y) \in P(R_0) \oplus P(R)$.
Case 3: x is indifferent between R_0 and R_i. This cannot happen because otherwise x's list would be empty in P_i.

For the other direction, let $(x, y) \in P(R_0) \oplus P(R)$. Since R is the median of R_i, R_j, R_0, we have the following three cases:

Case 1: $R(x) = R_i(x)$ then we have either

$$x : \ldots R_j(x) \ldots R_i(x) \ldots y \ldots R_0(x) \ldots$$

OR

$$x : \ldots R_0(x) \ldots y \ldots R_i(x) \ldots R_j(x) \ldots$$

But in both the cases, it is easy to see that $(x, y) \in P_i \cap P_j$
Case 2: $R(x) = R_j(x)$ Similar to the above case.
Case 3: $R(x) = R_0(x)$ This cannot happen as x's list will be empty in $P(R_0) \oplus P(R)$.

Therefore, $(\mathscr{S}, \subseteq)$ is a poset (under subset inclusion) closed under intersection and hence forms a meet-semi-lattice, in which empty set $= P(R_0) \oplus P(R_0)$ is the minimal element. Consider the map $\varphi : \mathscr{R} \to \mathscr{S}$ defined as $R_i \mapsto P_i$. Then φ is clearly a bijection as $P_i \neq P_j \forall i \neq j$. Hence, φ^{-1} induces a meet-semi-lattice structure on set of all stable matchings $\mathscr{R}$.

Exercise 6

Proof. .
(a) Notice that regret of T is defined as the maximum regret edge, say (x, y) where $y = l_T(x)$. If in phase 2, the rotation that y leads to is never eliminated, then the pair $\{x, y\}$ will never be deleted and hence (x, y) will be partners in the stable matching generated at the end of phase 2 algorithm. Thus, the regret of T remains unaltered and hence we obtain the maximum regret stable matching among all the matchings embedded in T.

(b) Instead of choosing an arbitrary rotation, we first find the maximum regret edge of T, say (x, y) where $y = l_T(x)$. If y is the only entry in x's list, then in any stable matching (embedded in T), (x, y) will be partners and so in particular any minimum regret stable matching has this edge; otherwise, we find the rotation that y leads to, that is, consider y's list: $x, x', \ldots$, then we can find y' such that $y' : x', x'', \ldots$. Repeating this process we can find a rotation. We eliminate *this particular rotation* in the next iteration. Either (x, y) was a part of this rotation and hence gets deleted or we can repeat this process until (x, y) gets deleted in the subsequent iterations. Having removed the maximum regret edge, we choose the next maximum regret edge and repeat the above process.
The only amendment we made was in phase 2 of algorithm where we find a maximum regret edge before finding/deleting an arbitrary rotation. This new sub-routine to find a maximum regret edge takes time $O(n^2)$ and it doesn't interfere with find/delete of rotation which takes time $O(n^2)$ already. Hence the total time taken by the algorithm is $O(n^2) + O(n^2) = O(n^2)$.

(c) We argue by contradiction. Let M_0 be any minimum regret matching and M be the output of our modified algorithm such that regret of M is strictly more than M_0 ($regret(M_0) < regret(M)$). Consider the sequence of table reductions we followed to get M: $T_0 \to T_1 \to T_2 \to \ldots \to M$. Let T_i be the last table in this sequence which contains M_0. We can find such a table because we started with the phase 1 table T_0 which contains both M_0 and M. Therefore, T_{i+1} doesn't contain M_0. This means, when going from T_i to T_{i+1}, we must have deleted a rotation which contains an edge of M_0. Let's say, when at T_i, we picked a maximum regret edge (x, y) w.r.t to T_i and the rotation ρ leading to it, which we deleted to obtain T_{i+1}. By the above observation, ρ contains an M_0 edge say (x', y'). But then $regret(M_0) \geq regret((x, y)) \geq regret(x', y') \geq regret(M)$, which leads us to a contradiction. $\square$