Stable matching algorithm: summary

Warning: no claims about real life, just a terminology

Problem setting

Consider n men and n women; each person has an preference list in decreasing order of all the participants of the opposite sex (no ties).

Notation: $m <_w m'$ means that woman w prefers m' to m as a partner; we use also a symmetric notation for women.

Matching: one-to-one correspondence between (all) men and women

Unstable pair for a matching: man m and woman w not in the matching, each of them prefers the other to the current partner in the matching (so they are both inclined to leave their current partners and become a pair)

Stable matching: a matching such that no unstable pair exists.

Theorem: a stable matching exists

The proof is of algorithmic nature, a process is described below that gives a stable matching (Gale–Shapley algorithm).

Algorithm and correctness proof

Data structures:

(a) for every man some prefix of his preference list (in the decreasing order) is marked as "rejected proposals"; if a women w is in this prefix for a man m, we say that "w have rejected m";

(b) the next woman in the m's list (after all that have rejected m, assuming that not everybody has rejected m) may be marked as "current partner" of m or not.

Invariants:

(a) current partners form a one-to-one correspondence between some set of men and some set of women;

(b) if a man m is rejected by woman w, then w has a current partner m' that is preferable ($m' >_w m$).

Initialization:

The prefixes are empty; no current partners for anybody (empty one-to-one correspondence). The invariant is obviously true.

Algorithm

while there exists at least one man without a current partner:
    choose some m without a current partner
    assertion: his rejection prefix does not include all women
    let w be the first candidate in his list after the rejection prefix
    let m propose to w: this means that
        if w has no current partner, then
            she becomes marked as m's current partner
            no change in the rejection lists
        if w has current partner m' and $m' >_w m$:
            m is rejected by w (rejection list for m increases)
            no change in the current partners
        if w has current partner m' and $m' <_w m$:
            w rejects m' (rejection list for m' increases,
                no current partner for m')
            w accepts m (marked as a current partner for m)

Assertion check: if m has no current partner but all women rejected m, then the invariant guarantees that they all have some current partners (in fact, better than m, according to their preferences), so the matching involves n women but less than n men — a contradiction.

Invariant preservation involves three cases:

First case: rejection lists do not change, and m and w were free, so the current partners relation remains a one-to-one correspondence, and old partners for women remain valid.

Second case: rejection list for m increases but the assumption $m' >_w m$ guarantees that the invariant relation is still true.

Third case: w changes her partner replacing m' by m (who was free), so the correspondence is still one-to-one. The new rejection satisfies the invariant, since w has now a better partner. The other cases where w rejected somebody also satisfy the invariant, since the new partner is better that the previous one.

Termination: at each step either some rejection list increases, or the size of current matching increases (while the other parameters remain the same). So the algorithm terminates, and at that stage all men have current partners, so we have a one-to-one correspondence between men and women.

Stability. Why this correspondence is stable? If (m, w) is an unstable pair for the final correspondence, the current partner w' for m is worse than w (i.e., $w' <_m w$) and the current partner m' for w is worse than m (i.e., $m' <_w m$). Then w is in the rejection list for m (since w preceeds w' in m's list) and the invariant guarantees that the current partner m' for w is better than m (i.e., $m' >_w m$, a contradiction).

(End of proof)

Algorithm is unfair

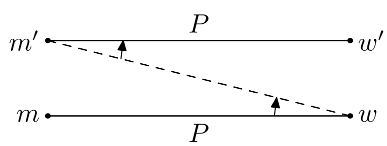
Theorem. Let P be any stable matching. Then the stable matching Q provided by the algorithm is not worse for every man m than P (his Q-partner is the same or better for him than his P-partner, for every m).

To prove this theorem, we fix P and prove the other invariant: *during the algorithm no man m can be rejected by his P-partner*. Equivalent statement: the rejection list for m will never reach his Q-partner. Another reformulation: if some m is rejected by some woman w, than w is worse for m than his Q-partner.

This statement implies that the at every step the current partner for m (if exists) and, therefore, his final partner in the algorithm is not worse than his P-partner (the statement of the theorem).

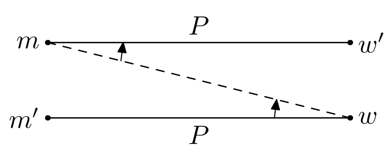
We need to check that the new invariant remains true. If the rejection lists do not increase (the first case), we have nothing to worry about.

In the second case m proposes to w and is rejected since w has a current partner m' that is better for w than m. The rejection list increases only for m, and w is added to it. To prove that our new invariant will remain true, we need to prove that m and w cannot be partners in P. Assume that it happens: (m, w) is in P.



Consider the pair (m', w). It is not an unstable pair for P (since P is assumed to be stable), but $m' >_w m$. Therefore, P-partner for m' (we denote her by w') is better than w for m'. But w is the current partner for m' in the algorithm, so w' is in the rejection list of m'. At the same time the pair (m', w') is in P, and we get a contradiction with our invariant.

Now consider the third case. Now m proposes to w and is accepted because m is better for w than her current partner m'. The rejection list increases for m'. The invariant will be violated only if (m', w) is in P. Assume this is the case.



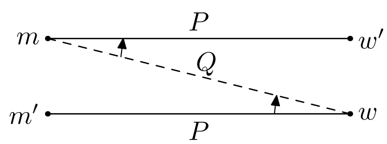
Since (m, w) cannot be an unstable pair for P, we know that P-partner w' for m is better than w for m. But m proposes to w, therefore w' is already in the rejection list for m. On the other hand, (m, w') is in P, so our invariant has to be already false.

(End of proof)

Algorithm is doubly unfair

Theorem. Let P be any stable matching. Then the stable matching Q provided by the algorithm is not better than P for every woman w (her Q-partner is the same or worse for her than her P-partner).

This result is an easy corollary of the previous one. Assume that (m, w) is in Q, and (m', w) is in P, and $m >_w m'$.



The pair (m, w) cannot be an unstable pair for P, so the P-partner of m (let us denote her by w') is better for m that w who is the Q-partner of m. Therefore, the algorithm provides a worse partner for m than his P-partner, and this contradicts the previous result.

(End of proof)