Felix Céard-Falkenberg
7174020

1 Question 3

1.1 *Prove that Till's Stable Matching algorithm leads to a perfect matching and terminates, or give a counter example.*

Theorem 1. *Till's Stable Matching algorithm leads to a perfect matching and terminates.*

Proof.

We can do so by using by running the Gale-Shapley algorithm with the following input:

- $X = P$
- $Y = S$

This results in a perfect matching between the individuals of P and S, with a benefit for the individuals in P.

By including individuals in P from both W and M, i.e. $P \cap W \neq \emptyset \wedge P \cap M \neq \emptyset$, we can make the algorithm more fair between the individuals from W and M.

The conclusion that a perfect matching between P and S exists, i.e., that each individual in P is matched to an individual in S, follows from the fact that the matching returned by the Gale-Shapley algorithm is stable and perfect.

Furthermore, since we run the Gale-Shapley algorithm on the instance (P, S), we know that it will not only terminate in $O(|P \cup S|^2) = O(|W \cup M|^2)$ time, but also that a stable matching between P and S is being found.

□

1.2 *Prove that Till's algorithm leads to a stable perfect matching, or give a counter example.*

To answer this question, we can consider two cases: the case of a stable matching between P and S, and the case of a stable matching between W and M.

Theorem 2. *Till's algorithm is* (P, S)*-stable.*

Proof. As Till's algorithm is a variant of the Gale-Shapley algorithm, we can guarantee that the algorithm returns a (P, S)-stable perfect matching from the properties of the Gale-Shapley algorithm.

□

However, in the latter, we cannot guarantee that the algorithm returns a (W, M)-stable perfect matching. To this end, we construct an instance where Till's algorithm does not return a (W, M)-stable perfect matching to prove Theorem 3.

Theorem 3. *Till's algorithm is not (W, M)-stable.*

Proof. Let us assume that $|W| = |M| = n$ with $n > 0$. We can chose W and M to be arbitrary sets, as the Gale-Shapley algorithm is garantees to return a (W, M)-stable perfect matching when running on the stance (W, M).

Naturally, if we were to chose $P = W$ and $S = M$, we would obtain a (W, M)-stable perfect matching.

However, let us consider two individuals $w \in W$ and $m \in M$ such that they are paired together in the (W, M)-stable perfect matching returned by the Gale-Shapley algorithm and w' and m' the next individuals in the preference list of w and m, respectively. Note that we constrain $w' \in W$ and $m' \in M$.

We can now construct our adversary instance by setting $P' = W \setminus \{w\} \cup \{m\}$ and $S' = M \setminus \{m\} \cup \{w\}$. We assume, for the case of the adversary instance, that the pairing between (P', S') is the same as the pairing between (W, M), with the exception of w being paired with w' and m being paired with m', and the original partners of w and m in the original pairing are now paired together in the adversary instance.

Therefore, the returned pairing between P' and S' is therefore (P', S')-stable and perfect, however, the pair (w, m) is now a blocking pair in the adversary matching.

Therefore, due to the existence of a blocking pair in the adversary matching, and the existence of such an adversary instance, we can conclude that Till's algorithm is not (W, M)-stable in general. This concludes the proof. $\square$

Note that Theorem 3 does not hold for all instances. For example, the instance given in the lecture notes (see Figure 1) does not result in a blocking pair when switching one or more pairs between P and S, and therefore serves as a counter example that Theorem 3 holds in all cases.

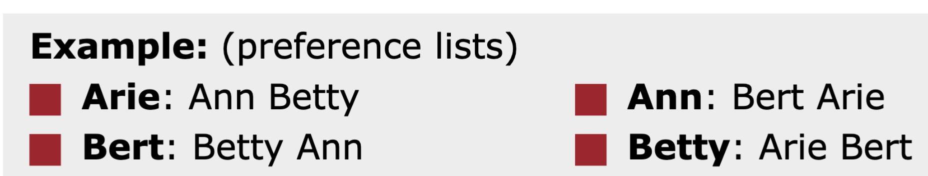

Figure 1: Example from the lecture notes.