Lecture 3

David Zynda

August 28, 2018

Theorem: Gale-Shapley (1962):

In any one-to-one matching environment there exists a stable matching μ.

The proof is constructive such that it makes a stable matching μ such that it employs the T-proposal Deferred Acceptance Algorithm.

Sketch of Algorithm

1. Each T agent proposes to her most preferred B agent if there is one that is acceptable to her. Each B agent tentatively holds on to the most preferred proposal provided that there is one acceptable to her.
2. Denote this step as $k \geq 2$. Each T agent that was rejected at Step $k-1$ makes the proposal to her next highest acceptable choice if there is one. And each B agent tentatively holds on to her most preferred option amongst new proposals plus $k-1$ held proposals if there is one acceptable. All other proposals are rejected.

Algorithm will terminate when: $k = k + 1$

Fix $X \subset J$ where J is a set of agents. Write $\max_i X = \{j \in X : j \succsim_i j' \forall j' \in X\}$

If $X = \emptyset$ then $\max_i X = \emptyset$

Formally, the Algorithm can be written as follows:

Round 1:

$A^1(t) = A(t)$ T ask what agents are acceptable to me in round 1?

$\hat{P}^1 : T \to B \cup \{\emptyset\} : \hat{P}^1 \in \max_t A^1(t)$ I want to make proposal to best B agent. T is matched with best B agent or keeps to himself (empty set).

If $A^1(t) \neq \emptyset$ choose the best B agent. If $A^1(t) = \emptyset$, choose $\emptyset$

What are B agents going to do?

$P^1(b) = (\hat{P}^1)^{-1}(\{b\}) \cap A(b) = \{t \in T : \hat{P}^1(t) = b : t \in A(b)\}$

The above means t has made a proposal to me and that t is acceptable to me.

Now, $\hat{\mu}^1 : B \to T \cup \{\emptyset\}$ such that $\hat{\mu}^1 \in \max_b P^1(b)$

If $P^1(b) = \emptyset$ then $\hat{\mu}^1(b) = \emptyset$. That is if none are acceptable, I hold on to mysel

Round k+1:

Inductively, I have defined:

- Sets $(A^k(t) : t \in T)$
- maps $\hat{P}^k : T \to B \cup \{\emptyset\}$
- sets $(P^k(b) : b \in B)$ and $P^k \subseteq T$
- maps $\hat{\mu}^k : B \to T \cup \{\emptyset\}$

$A^{k+1}(t) =$

(1) $A^k(t)$ if $\hat{\mu}^k(\hat{P}^k) = t$ meaning t proposed at k to $b = \hat{P}^k(t)$ and b accepted t at k.

But, he if rejected the offer, I throw him out of the running:

(2) $A^k(t) \setminus \{\hat{P}^k(t)\}$

if

$$\hat{\mu}^k(\hat{P}^k(t)) \neq t$$

If I was accepted on round k, then I offer the same offer to the same person on round $k+1$.

$\hat{P}^{k+1}(t) = \hat{P}^k$ if it is the case that $\hat{P}^k(t) \in A^{k+1}(t)$

$\hat{P}^{k+1}(t) \in \max_t A^{k+1}(t)$ if it is the case $\hat{P}^k(t) \notin A^{k+1}(t)$

The set of proposals that b has on round $k+1$:

$$P^{k+1}(b) = (\hat{P}^{k+1})^{-1}(\{b\}) \cap A(b)$$

Which is equivalent to:

$$\{t \in T : \hat{P}^{k+1}(t) = b : t \in A(b)$$

I hold on to exactly the same offer if it is the case that b is still a maximizer for me.

$$\hat{\mu}^{k+1} : B \to T \cup \{B\}$$

$\hat{\mu}^{k+1}(b) = \hat{\mu}^k(b)$ if it is the case that $\hat{\mu}^{k+1}(b) \in \max_b P^{k+1}(b)$

And

$\hat{\mu}^{k+1}(b) = \max_b P^{k+1}(b)$ if it is the case that $\hat{\mu}^{k+1}(b) \notin \max_b P^{k+1}(b)$

Lemma 1: the T-proposal in the algorithm terminates:

$$\exists K < \infty : \forall k \geq K : \hat{P}^k = \hat{P}^K \wedge \hat{\mu}^k = \hat{\mu}^K$$

Proof

k is when the acceptable offer stops for every t agent.

For each $t \in T$, $(A^k(t) : k = 1, 2, ...)$ is decreasing.

$..., \subseteq A^3(t) \subseteq A^2(t) \subseteq A^1(t)$

$\exists K : \forall t \in T, A^k(t) = A^K(t) \forall k \geq K$

By definition, $\hat{P}^k(t) = \hat{P}^k(t)$ for all $t \in T$ and $k \geq K$

This implies that $\hat{P}^k(b) = \hat{P}^k(b)$ for all $b \in B$ and $k \geq K$

Which implies that $\hat{\mu}^k(b) = \hat{\mu}^K(b)$ for all $b \in B$ and $k \geq K$.

Show that the algorithm terminates. But does it give us a stable match?

Example

Let $T = \{t_1, t_3, t_3\}$ and also $B = \{b_1, b_2, b_3\}$

$$t_1 : b_2 \succ_{t_1} b_1 \succ_{t_1} b_3 \succ_{t_1} t_1$$
$$t_2 : b_1 \succ_{t_2} b_2 \succ_{t_2} b_3 \succ_{t_2} t_2$$
$$t_3 : b_1 \succ_{t_3} b_2 \succ_{t_3} b_3 \succ_{t_3} t_3$$

Notice t_2 and t_3 have the same preferences.

B's preferences:

$$b_1 : t_1 \succ_{b_1} t_3 \succ_{b_1} t_2 \succ_{b_1} b_1$$
$$b_2 : t_2 \succ_{b_2} t_1 \succ_{b_2} t_3 \succ_{b_2} b_2$$
$$b_3 : t_1 \succ_{b_3} t_3 \succ_{b_3} t_2 \succ_{b_3} b_3$$

Notice b_1 and b_3 have the same preference.

Step 1:

- for all $t \in T : A^1(t) = B$
- $\hat{P}^1(t_1) = b_2, \hat{P}^1(t_2) = b_1, \hat{P}^1(t_3) = b_1$
- $P^1(b_1) = \{t_2, t_3\}$ and $P^1(b_2) = \{t_1\}$ and $P^1(b_3) = \emptyset$
- $\hat{\mu}^1(b_1) = t_3$ and $\hat{\mu}^1(b_2) = t_1$ and $\hat{\mu}^1(b_3) = \emptyset$

Step 2:

- $A^2(t_1) = B$ and $A^2(t_2) = \{b_2, b_3\}$ and $A^2(t_3) = B$
- $\hat{P}^2(t_1) = b_2$ and $\hat{P}^2(t_2) = b_2$ and $\hat{P}^2(t_3) = b_1$
- $P^2(b_1) = \{t_3\}$ and $P^2(b_2) = \{t_1, t_2\}$ and $P^3(b_3) = \emptyset$
- $\hat{\mu}^2(b_1) = t_3$ and $\hat{\mu}^2(b_2) = t_2$ and $\hat{\mu}^2(b_3) = \emptyset$

Step 3:

- $A^3(t_1) = \{b_1, b_3\}$ and $A^3(t_2) = \{b_2, b_3\}$ and $A^3(t_3) = B$
- $\hat{P}^3(t_1) = b_1$ and $\hat{P}^3(t_2) = b_2$ and $\hat{P}^3(t_3) = b_1$
- $P^3(b_1) = \{t_1, t_3\}$ and $P^3(b_2) = \{t_2\}$ and $P^3(b_3) = \emptyset$
- $\hat{\mu}^3(b_1) = t_3$ and $\hat{\mu}^3(b_2) = t_2$ and $\hat{\mu}^3(b_3) = \emptyset$

Step 4:

- $A^4(t_1) = \{b_1, b_3\}$ and $A^4(t_2) = \{b_2, b_3\}$ and $A^4(t_3) = \{b_2, b_3\}$
- $\hat{P}^4(t_1) = b_1$ and $\hat{P}^4(t_2) = b_2$ and $\hat{P}^4(t_3) = b_2$
- $P^4(b_1) = \{t_1\}$ and $P^4(b_2) = \{t_2, t_3\}$ and $P^4(b_3) = \emptyset$
- $\hat{\mu}^4(b_1) = t_1$ and $\hat{\mu}^4(b_2) = t_2$ and $\hat{\mu}^4(b_3) = \emptyset$

Step 5:

- $A^5(t_1) = \{b_1, b_3\}$ and $A^5(t_2) = \{b_2, b_3\}$ and $A^5(t_3) = \{b_3\}$
- $\hat{P}^5(t_1) = b_1$ and $\hat{P}^5(t_2) = b_2$ and $\hat{P}^5(t_3) = b_3$
- $P^5(b_1) = \{t_1\}$ and $P^5(b_2) = \{t_2\}$ and $P^5(b_3) = \{t_3\}$
- $\hat{\mu}^5(b_1) = t_1$ and $\hat{\mu}^5(b_2) = t_2$ and $\hat{\mu}^5(b_3) = t_3$.

You can check this is exactly what was completed last class without the Differed Acceptance Algorithm.

Since it terminates at $k + 1$ technically there is a **Step 6** where everything in **5** is repeated.