Stable Matching

We have a set H of M hospitals $h_1, h_2, \ldots, h_M$ looking for residents and a set S of M medical students $s_1, \ldots, s_M$ that are looking to hire residents. We would like to match each hospital to a student so that each student has exactly one hospital matched to it.

For example if $M = 4$ a possible matching might be

$$\{(h_1, s_3), (h_2, s_1), (h_3, s_1), (h_4, s_4)\}$$

i.e., the matching

$$h_1 \mapsto s_3$$

$$h_2 \mapsto s_2$$

$$h_3 \mapsto s_1$$

$$h_4 \mapsto s_4$$

Mathematically, a *matching* is a function $F : H \to S$ that is one-to-one, i.e., if $i \neq j$, then $F(h_i) \neq F(h_j)$. The intuition is that h_i is matched to $F(h_i)$.

It is easy to see that there are $M!$ possible matchings. But some matchings will be better that others.

Each hospital h_i has an ordering of which students they prefer. For example, if $M = 4$, then hospital h_1 might prefer

$$s_1 > s_3 > s_4 > s_2.$$

Also each student s_j has an ordering of which hospitals they prefer. For example, s_1 might prefer

$$h_4 > h_2 > h_1 > h_3.$$

We need to do this for each hospital and each student.

If $M = 4$ one possible set of preferences would be:

hospital preferences

$h_1 : s_1 > s_3 > s_4 > s_2$
$h_2 : s_1 > s_2 > s_4 > s_3$
$h_3 : s_2 > s_1 > s_3 > s_4$
$h_4 : s_2 > s_3 > s_1 > s_4$

student preferences

$s_1 : h_4 > h_1 > h_2 > h_3$
$s_2 : h_1 > h_2 > h_3 > h_4$
$s_3 : h_4 > h_2 > h_1 > h_3$
$s_4 : h_2 > h_4 > h_3 > h_1$

We would like to find a good matching. Here is a reason why a matching might not be good.

Suppose we matched $h_1 \mapsto s_1$ and $h_2 \mapsto s_2$, but hospital h_1 prefers student s_2 and s_2 prefers h_1. Then both hospital h_1 and student s_2 would prefer a different matching where $h_1 \mapsto s_2$. If this happens we say that matching is *unstable*.

More precisely, suppose $h, h' \in H$ and $s, s' \in S$ such that $h \mapsto s$ and $h' \mapsto s'$, but h prefers s' to s and s' prefers h to h'. We call this an *unstable configuration*. We say that a matching is *stable* if there are is no unstable configuration.

Can we always find a stable matching? Yes, and there is a simple algorithm to find one!

Gale–Shapley Deferred Acceptance Algorithm

• Let $n = 1$.

• In round n each hospital makes and offer to their preferred student among those students that have not previously rejected them.

• Each student tentatively accepts the hospital that made them an offer that they most prefer and rejects all others who made them an offer.

• If everyone is matched we halt and this is the final match. Otherwise, we go on to round $n + 1$.

Here is what happens if we applied the algorithm to the preferences given above. Here are the offers made round by round with the rejected offers marked in red.

	round 1	round 2	round 3	round 4	round 5
h_1	s_1	s_1	s_1	s_1	s_1
h_2	s_1	s_2	s_2	s_2	s_2
h_3	s_2	s_2	s_1	s_3	s_4
h_4	s_2	s_3	s_3	s_3	s_3

Theorem 1 *The Gale–Shapley algorithm will always halt and produce a stable matching.*

Proof We need to show two things.

claim 1 The algorithm will always halt producing a matching.

First note, that some hospital attempts to match with student s in round n. Then student s will have tentative match in every later round. They may reject matches, but they will always keep one tentative match. In particular, if hospital h is rejected in round n, there is always at least one student left that they have not proposed to.

In each round except the last there is a pair (h, s) such that s proposed to h in that round and was rejected. hospital h will never again propose to s, so this will only happen once with (h, s). Moreover, each hospital is rejected at most $M - 1$ times. Thus there are at most $M(M - 1) + 1$ rounds (the $+1$ is for the last round where no one is rejected). Thus the algorithm always halts.

No one is rejected in the last round, thus each student only has one hospital that wants to match to it. Thus the algorithm has produced a matching.

claim 2 When the algorithm halts we have reached a stable matching .

First note the following. A student only rejects a hospital if they have a tentative match with a hospital they prefer. Thus as the rounds go only the hospital matched to a given student will be more desirable than any hospital they rejected.

We now show there are no unstable configurations. Suppose, for contradiction, that there is an unstable configuration h, h', s, s', where $h \mapsto s$, $h' \mapsto s'$, but h prefers s' to s and s' prefers to h to h'. At some stage h attempted to match with s. Since h prefers s' to s, h would only do this if already rejected by s'. But we just argued that if s' rejects h and then ultimately ends up with h', then s' prefers h' to h, a contradiction.

We have shown there are no unstable configurations thus the Gale–Shapley algorithm produces a stable matching. □

The stable matching we produced may or may not be the only possible stable matching.

For a simple example, consider the case where $M = 2$ and we have preferences

hospital preferences

$h_1 : s_1 > s_2$

$h_2 : s_2 > s_1$

student preferences

$s_1 : h_2 > h_1$

$s_2 : h_1 > h_2$

The Gale–Shapley algorithm will give the matching

$$h_1 \mapsto s_1, h_2 \mapsto s_2.$$

One way to try to get another stable matching would be to run a variant of the Gale–Shapley algorithm where the students make the offers and the hospitals decide to accept or reject. If we did that in this case, we would get the stable matching

$$h_1 \mapsto s_2, s_2 \mapsto h_1.$$

Is there a significant difference between these matches. Yes! We will see that the Gale–Shapley produces the best possible stable match for the hospital.

We say that student s is an *eligible match* for hospital h if there is some stable matching where $h \mapsto s$. In that case we also say that h is an eligible match for s.

Proposition 2 *In the Gale–Shapley algorithm each hospital is matched the the most desirable eligible student. In other words, for each hospital the Gale–Shapley algorithm produces at least as good a result as any other stable matching.*

Proof Suppose $h \mapsto s$ but h prefers s' to s. We need to show there is no stable matching where h is matched to s'. In the algorithm h would make an offer to s' before making an offer to s. Since h is not matched to s', s' must at some point reject h. The proposition then follows from the following lemma. □

Lemma 3 *If at some stage of the algorithm s rejects h, then there is no stable matching where h is matched with s.*

Proof Let $R_0 = \emptyset$ and let R_n be all of the pairs (h', s') that s' rejects h' at or before stage n. We will show by induction on n that if $(h, s) \in R_n$, then there is no stable matching where $h \mapsto s$.

$\underline{n = 0.}$ Because $R_0 = \emptyset$, there is nothing to prove.

$\underline{n \Rightarrow n = 1}$ Suppose this is true for R_n, we need to show that it is true for R_{n+1}.

Suppose (h, s) is rejected in round $n + 1$. Then some other hospital h' also made an offer to s in round $n+1$. Hospital h' has already made an offer to every student it prefers to s, and has been rejected by them. Thus, by induction, no stable matching pairs h' to a student that it prefers to s. Since s prefers h' to h and h' prefers h' prefers s' to s. There can not be a stable matching where $h \mapsto s$ as in this mapping we would have an unstable configuration. Thus no stable matching pairs h with s.

By induction if s rejects h at any stage, then there is no stable matching where h is matched with s. □

On the other hand, from the point of view of the students this is the worst possible stable matching.

Proposition 4 *In the Gale–Shapley algorithm, each student is matched with the least desirable eligible hospital.*

Proof For purposes of contradiction, suppose not. Suppose the algorithm matches $h \mapsto s$, and there is a different stable matching F where $h' \mapsto s$ and s prefers h to h'. Suppose $h \mapsto s'$ in F. By the Proposition 2, h prefers s to s'. This gives an unstable configuration in F, contradicting that F is stable. □

Exercise 5 Suppose we run the Gale–Shapley algorithm twice once where the hospitals make offers and once where the students make offers. Show that we get the same matching both times if and only if there is only one stable matching.

Is honesty the best policy?

To run the Gale–Shapley algorithm we need to obtain the preferences for each hospital and each student. Should they honestly tell them to us?

For the hospital the answer is yes. Proposition 1 tells us that the algorithm will produce the best possible stable matching for them, so they have no incentive to lie.

For the students, there may be incentives to lie. Consider the following example with $M = 3$.

hospital preference

$h_1 : s_1 > s_2 > s_3$
$h_2 : s_2 > s_1 > s_3$
$h_3 : s_1 > s_2 > s_3$

hospital preference

$s_1 : h_2 > h_1 > h_3$
$s_2 : h_1 > h_2 > h_3$
$s_3 : h_1 > h_2 > h_3$

If everyone honestly reports preferences, we would have:

	round 1	round 2	round 3
h_1	s_1	s_1	s_1
h_2	s_2	s_2	s_2
h_3	s_1	s_2	s_3

But suppose student s_1 reported the preference $h_2 > h_3 > h_1$. Now the algorithm would proceed

	round 1	round 2	round 3	round 4
h_1	s_1	s_2	s_2	s_2
h_2	s_2	s_2	s_1	s_1
h_3	s_1	s_1	s_1	s_3

By lying about preferences, student s_1 has obtained a preferable match. Thus honestly reporting your preferences may not be a Nash equilibrium.

It's not hard to make this more realistic. For example, we could have different numbers of hospitals and students. Each student could have a list of acceptable hospitals where they would not accept an offer from an unacceptable hospital.. Similarly, each hospital could have a list of unacceptable students they would never accept. Now a matching has to allow the possibility that some things are unmatched. Formulate what a stable matching should be in this setting and show that you can modify the Gale–Shapley algorithm to find one.