Compsci 590T Homework 6

Student: Brian Sun, Collaborator: Thai On

1 Problem 1.a

After the algorithm terminate to show every student assign to $best(s)$ we need to show there exist no student $s_i \in S_{sad},\ S_{sad} \subseteq S$ such that $M(s_i) \not\simeq best(s_i)$ which means there cannot exist a list of available hospital $H' \subseteq H \backslash M(s_i),\ h \in H'$ such that $h \succ_{s_i} M(s_i)$ and $s_i \succ_h M^{-1}(h)$

There will have four cases we need to consider if it can satisfy:
set $h_j \in H'$
1: $M^{-1}(h_j) \succ_{h_j} s_i \wedge M(s_i) \succeq_{s_i} h_j$
2: $M^{-1}(h_j) \succ_{h_j} s_i \wedge M(s_i) \prec_{s_i} h_j$
3: $M^{-1}(h_j) \prec_{h_j} s_i \wedge M(s_i) \succeq_{s_i} h_j$
4: $M^{-1}(h_j) \prec_{h_j} s_i \wedge M(s_i) \prec_{s_i} h_j$

$(\star)best(s)$ is the hospital that s ranks highest amongst all of its valid hospitals

For both case 1 and 2:
if h_j rank it's current student more than s_i that means h_j is not valid, so no matter how s_i rank h_j, h_j will stick with $M^{-1}(h_j)$. Since there are no valid hospital other $M(s_i)$ therefore base on $(\star)$, $M(s_i) \simeq best(s_i)$
For case 3:
if there exist hospital that rank s_i higher than it's current student but s_i rank those hospital lower or equally to $M(s_i)$ which means $M(s_i)$ rank the highest amongst all the valid hospitals since there are no hospital has a higher ranking, so base on $(\star)$, $M(s_i)$ is $best(s_i)$
For case 4:
if there exist hospital that rank s_i higher than it's current student and s_i also rank those hospital higher than $M(s_i)$ which is a blocking pair, but this cannot happen since Gale-Shapley algorithm always return stable matching.

Overall there are no cases shows s_i can get assign a hospital that is not the highest rank amongst all of it's valid hospitals. Therefore s is match to $best(s)$

2 Problem 1.b

In the stable match M hospital h_i pair with student $s^* = M^{-1}(h_i)$ if $M^{-1}(h_i) \not\simeq worst(h_i)$ that means there exist student s, $s^* \succ_{h_i} s$ and (s, h_i) is in another stable match M'.

This cannot happen since in M we know s^*, h_i is the valid pair and base on Problem 1.a we know h_i is $best(s^*)$.
Therefore in M' if h got match with s that means s^* will match with some other hospital h_j and since h_j got match with s^* so h_j is a valid hospital for s^*, since $h_j \not\simeq best(s^*)$, so $h_i \succeq_{s*} h_j$.

So we will have two cases:
1. $s^* \succ_{h_i} s$ and $h_i \succ_{s*} h_j$
2. $s^* \succ_{h_i} s$ and $h_i \simeq_{s*} h_j$

Case (1): Since we know: $s^* \succ_{h_i} s$ and $h_i \succ_{s*} h_j$ so there will have a blocking pair.
Case (2): Gale-Shapley algorithm don't have randomize picking, therefore s^* will always pair with h_i under Gale-Shapley algorithm.
Therefore there are no student exist that is worse than s^*.

3 Problem 2.a

Assume it is not unique stable matching which:
There are at least 2 students have different hospital between match M and M'
Define S_{change} be the set of student who has different assign hospital between match M and M'
let s be the highest ranking student in S_{change}
In stable matching M student s will pair with h
In stable matching M' student s will pair with h'

Assume $h \succ_s h'$ then we know:
If M' happen there will have two possible reasons:
1. s want to pair with h, but h pair with the other student s' and h like s' more than s such that $s' \succ_h s$. So s has to pair with some other hospital h'.
2. h want to pair with s, but s pair with the other hospital h' and s like h' more than h such that $h' \succeq_s h$.
Reason 1:
This cannot happen. Since for all students $s' \in S$, $s' \succ_h s$ has the same assign hospital between match M and M' therefore we know h cannot match with s'.

Reason 2:
this cannot happen if $h' \succ_s h$. This will have 2 cases which are:
a. $h' \succ_s h$ and $s \succ_{h'} s*$
b. $h' \succ_s h$ and $s* \succ_{h'} s$
Case (a): if this is true then in M, since student s pair with hospital h: (s, h) and h' pair with another student $s*$: $(s*, h')$. This will create a blocking pair since $h' \succ_s h$, and since $s*$ is in S_{change} so $s \succ_{h'} s*$. Therefore case (a) cannot happen.
Case (b): we know $s* \succ_{h'} s$ cannot happen base on reason 1. Which s must have higher ranking than $s*$ and bring back to case (a).

Therefore this instance is a unique stable matching since all the possible reasons to make multiple stable matching are not valid.

4 Problem 2.b

Base on question 1 we know:
($\star$) $best(s)$ be the hospital that s ranks highest amongst all of its valid hospitals
($\blacklozenge$) $worst(s)$ be the least preferred hospital that could be assigned to s under some stable matching
A: if student propose to hospital under G-S algorithm then student will get $best(s)$ and hospital get $worst(h)$.
B: if hospital propose to student under G-S algorithm then hospital will get $best(h)$ and student will get $worst(s)$.

1. Set $M_{student}$ be the stable match for student propose to hospital under G-S algorithm and base on A and ($\star$) we know $s \in S$, $M_{student}(s) \succeq_s M'(s)$ where $M_{student}(s)$ is the assign hospital to student s by $M_{student}$ and $M'(s)$ is the assign hospital to student s by any stable match M'.

2. Set $M_{hospital}$ be the stable match for hospital propose to student under G-S algorithm and base on B and ($\blacklozenge$) we know $s \in S$, $M_{hospital}(s) \preceq_s M'(s)$ where $M_{hospital}(s)$ is the assign hospital to student s by $M_{hospital}$ and $M'(s)$ is the assign hospital to student s by any stable match M'.

So base on 1. and 2. We know $M_{student}(s) \succeq_s M'(s) \succeq_s M_{hospital}$ where $M'(s)$ is the assign hospital to student s by any stable match M'. Given $M_{student} = M_{hospital}$ then we got $M_{student}(s) \succeq_s M'(s) \succeq_s M_{hospital}(s) = M_{student}(s)$. Therefore $M_{student} = M'$ so the stable matching is unique.

5 Problem 3

The least number should be 2 student and 2 hospital. Because if you have 2 student and 1 hospital or 1 student and 2 hospital then one student or hospital will left out. As well as 1 student and 1 hospital both student and hospital only have one choice.

An example for 2 student and 2 hospital will be:

$$\begin{array}{cc} s_a : h_a \succ h_b & h_a : s_b \succ s_a \\ s_b : h_b \succ h_a & h_b : s_a \succ s_b \end{array}$$

Run G-S algorithm with student propose to hospital we got:

$$M: \ (s_a, h_a), (s_b, h_b)$$

Run G-S algorithm with hospital propose to student we got:

$$M': \ (s_a, h_b), (s_b, h_a)$$

Since G-S algorithm return stable matching and M and M' are different therefore this instance gives more than one stable matching.

6 Problem 4

Stable matching is not always exist in this case for example:

$$s_a : s_b \succ s_c \succ s_d$$
$$s_b : s_c \succ s_a \succ s_d$$
$$s_c : s_a \succ s_b \succ s_d$$
$$s_d : s_a \succ s_a \succ s_c$$

We can have 3 possible matching:

1. $(s_a, s_b), (s_c, s_d)$
2. $(s_a, s_c), (s_b, s_d)$
3. $(s_a, s_d), (s_b, s_c)$

Matching (1.) is not a stable matching since $s_c \succ_{s_b} s_a$ and $s_b \succ_{s_c} s_d$ therefore s_b and s_c prefer each other more then their current partner so it's a blocking pair.

Matching (2.) is not a stable matching since $s_b \succ_{s_a} s_c$ and $s_a \succ_{s_b} s_d$ therefore s_a and s_b prefer each other more then their current partner so it's a blocking pair.

Matching (3.) is not a stable matching since $s_c \succ_{s_a} s_d$ and $s_a \succ_{s_c} s_b$ therefore s_a and s_c prefer each other more then their current partner so it's a blocking pair.

7 Problem 5.a

Base on the question we know both M and M' are valid stable matching and every student $s \in S$ has to pick between $M(s)$ and $M'(s)$ therefore we know every student will have a hospital to pair with.

To make sure this the out put is a stable matching we have to make sure: 1. there are no student causing a blocking pair. 2. There exist no student pair with two hospitals or a hospital pair with two students.

1.
Assume the match from $M \vee M'$ has (s, h) and there exist a student s_i that he prefer $h \succ_{s_i} M \vee M'(s_i)$ and h prefer s_i more than s as well. Since (s, h) happen therefore we know it's either in M or M' and since s_i want h more than the option in M and M' which will create a blocking pair in the match where (s, h) at. Since both M and M' are stable matching, therefore this situation cannot happen.

2.
For example $M \vee M'(s) = h$ and $M \vee M'(h) = s'$ cannot happen, because this situation create a blocking pair.
Base on the example we know there will have two possible pair: (s, h) and (s', h). Since we know M and M' both are stable matching therefore (s, h) is happen in either M or M' and (s', h) is happen in the other one.

Since we know $M \vee M'(h)$ is h less preferred student between M and M' therefore $s \succ_h s'$.
We also know $M \vee M'(s)$ is s more preferred hospital between M and M' therefore $h \succ_s h'$
Since $s \succ_h s'$ and $h \succ_s h'$ therefore $M \vee M'(s) = h$ and $M \vee M'(h) = s'$ cannot happen.

8 Problem 5.b

$M \geqslant M'$ or $M' \geqslant M$ is not true. For example:

$$
\begin{array}{ll}
s_1 : h_2 \succ h_1 \succ h_3 \succ h_4 & \quad h_1 : s_1 \succ s_2 \succ s_3 \succ s_4 \\
s_2 : h_1 \succ h_2 \succ h_3 \succ h_4 & \quad h_2 : s_2 \succ s_1 \succ s_3 \succ s_4 \\
s_3 : h_3 \succ h_4 \succ h_1 \succ h_2 & \quad h_3 : s_4 \succ s_3 \succ s_1 \succ s_2 \\
s_4 : h_4 \succ h_3 \succ h_1 \succ h_2 & \quad h_4 : s_3 \succ s_4 \succ s_1 \succ s_2
\end{array}
$$

Base on this instance we define:

$$
\begin{aligned}
M &= (s_1, h_2)(s_2, h_1)(s_3, h_4)(s_4, h_3) \\
M' &= (s_1, h_1)(s_2, h_2)(s_3, h_3)(s_4, h_4)
\end{aligned}
$$

M is a stable matching since s_1, s_2, h_3, h_4 got their first choice. And both s_3, s_4 prefer h_3, h_4 more than h_1, h_2 therefore M is a stable matching since there are no blocking pair.

M' is a stable matching since s_3, s_4, h_1, h_2 got their first choice. And both s_1, s_2 prefer h_1, h_2 more than h_3, h_4 therefore M is a stable matching since there are no blocking pair.

So base on M and M' we know s_1, s_2 prefer M and s_3, s_4 prefer M' therefore $M \geqslant M'$ or $M' \geqslant M$ is not true.

9 Problem 5.c

$M_{student}$ be the stable match of student propose to hospital under G-S algorithm.
$M_{hospital}$ be the stable match of hospital propose to student under G-S algorithm.

Define C be the set of all stable matching.
set $M' \in C \backslash M_{hospital}$ be the lowest one from student perspective such that $\forall M \in C \backslash (M', M_{hospital}),\ M \succeq_s M'$.

Base on lattice structure $M \vee M'$ student will always pick the better one.
Therefore the best case scenario will be $M^* \vee M_{student},\ M^* \in C$. Since $M_{student}$ return $best(s)$, so there are no stable match that is better than $M_{student}$. And base on the definition of $M \vee M'$ the outcome of $M^* \vee M_{student}$ will be $M_{student}$.

The worst case scenario will be $M' \vee M_{hospital}$. Since $M_{hospital}$ return the $worst(s)$, so there are no stable match that is worst than $M_{hospital}$. And base on the definition of $M \vee M'$ the outcome of $M' \vee M_{hospital}$ will be M'.