Stable Marriage II

Stable marriage problem

The setting:

- There are n boys $b_1, b_2, \dots, b_n$ and n girls $g_1, g_2, \dots, g_n$. We assume the number of boys and girls is the same.
- Each boy has his own ranked preference list of girls and each girl has her own ranked preference list of boys.
- The lists are complete and have no ties. Each boy ranks every girl and vice versa.

The goal:

Pair each boy with a unique girl so that there do not exist boys b_i, b_j, and girls g_k, g_l where b_i and g_k are paired up, but boy b_i prefers girl g_l to g_k, and girl g_k prefers boy b_j to b_i.

Gale-Shapley algorithm

Each Day

Morning:

Each girl stands on her balcony. Each boy stands under the balcony of his favorite girl whom he has not yet crossed off his list and serenades. If there are no girls left on his list, he stays home and does graph algorithms homework.

Afternoon:

Girls who have at least one suitor say to their favorite from among the suitors that day: „Maybe, come back tomorrow." To the others, they say „No, I will never marry you!"

Evening:

Any boy who hears „No" crosses that girl off his list.

Gale-Shapley algorithm

Termination Condition:

If there is a day when every girl has at most one suitor, we stop and each girl marries her current suitor (if any).

Truthfulness

Basic question:

Can a boy or a girl end up better off by lying about his or her preferences?

Consider for example a girl g. Suppose g prefers boy b to b'. Can it be the case that by falsely claiming that she prefers b' to b at some iteration of the Gale-Shapley algorithm, g will end up with a boy b'' that she truly prefers to both b and b'?

Truthfulness

$$b_1 \rightarrow (g_3, g_1, g_2)$$
$$b_2 \rightarrow (g_1, g_3, g_2)$$
$$b_3 \rightarrow (g_3, g_1, g_2)$$

$$g_1 \rightarrow (b_1, b_2, b_3)$$
$$g_2 \rightarrow (b_1, b_2, b_3)$$
$$g_3 \rightarrow (b_2, b_1, b_3)$$

Forbidden pairs

We have a set B of n boys, a set G of n girls, and a set $F \subseteq B \times G$ of pairs who are simply not allowed to get married.

Each boy b ranks all the girls g for which $(b, g) \notin F$, and each girl g ranks all the boys b for which $(b, g) \notin F$.

In this setting, we say that a matching M is stable if it does not exhibit any of the following types of instability.

Forbidden pairs

1. There are two pairs (b, g) and (b', g') in M with the property that $(b, g') \notin F$, b prefers g' to g, and g' prefers b to b'. (The usual kind of instability.)
2. There is a pair $(b, g) \in M$, and a boy b', so that b' is not part of any pair in the matching, $(b', g) \notin F$, and g prefers b' to b. (A single boy is more desirable and not forbidden.)

Forbidden pairs

3. There are two pairs (b, g) and (b', g') in M with the property that $(b, g') \notin F$, b prefers g' to g, and g' prefers b to b'. (The usual kind of instability.)
4. There is a pair $(b, g) \in M$, and a boy b', so that b' is not part of any pair in the matching, $(b', g) \notin F$, and g prefers b' to b. (A single boy is more desirable and not forbidden.)

Forbidden pairs

Note that under these more general definitions, a stable matching need not be a perfect matching.

For every set of preference lists and every set of forbidden pairs, is there always a stable matching?

College admission

There are n students $s_1, s_2, \dots, s_n$ and m universities $u_1, u_2, \dots, u_m$.

University u_i has n_i slots for students, and we're guaranteed that

$$n_1 + n_2 + \cdots + n_m = n.$$

Each student ranks all universities (no ties) and each university ranks all students (no ties).

Design an algorithm to assign students to universities with the following properties

College admission

- Every student is assigned to one university.
- University u_i gets assigned n_i students.
- There do not exist students s_i, s_j, and universities u_k, u_l where student s_i is assigned to university u_k, student s_j is assigned to university u_l, student s_j prefers university u_k to university u_l, and university u_k prefers student s_j to student s_i.
- It is student-optimal. This means that of all possible assignments satisfying the first three properties, every student gets his/her top choice of university amongst these assignments.

College admission revised

There are n students $s_1, s_2, \dots, s_n$ and m universities $u_1, u_2, \dots, u_m$.

University u_i has n_i slots for students, but now we're guaranteed only that

$$n_1 + n_2 + \cdots + n_m \leq n.$$

Each student ranks all universities (no ties) and each university ranks all students (no ties).

College admission revised

The interest is in finding a way of assigning each student to at most one university, in such a way that all available positions in all universities are filled.

We say that an assignment of students to universities is stable if neither of the following situations arises.

College admission revised

First type of instability:

There are students s_i and s_j, and a university u_k, so that

- s_i is assigned to u_k,
- s_j is assigned to no university,
- u_k prefers s_j to s_i.

College admission revised

Second type of instability:

There are students s_i and s_j, and universities u_k and u_l, so that

- s_i is assigned to u_k,
- s_j is assigned to u_l,
- u_k prefers s_j to s_i,
- s_j prefers u_k to u_l.

College admission revised

The following algorithm always find a (university optimal) stable assignment of students to universities.

College admission revised

```
While there is a university u_i which has available slots and hasn't
offered a position to every student
   u_i offers a position to the next student s_j on its preference list
   if s_j is free
      then
         s_j accepts the offer
      else /* s_j is already committed to a university u_k */
         if s_j prefers u_k to u_i
            then
               s_j remains committed to u_k
            else
               s_j becomes committed to u_i
               the number of available slots at u_k increases by one
               the number of available slots at u_i decreases by one
```