Stable Marriage Problem

This Algorithm has several applications for example

- college admissions {Joint seat allocation (JOSA) INDIA}
- National Residency matching Problem (NRMP) USA. (matching doctors to hospitals)
- Managing Internet traffic etc.

Problem:

1. There are n boys & n girls.
2. Each boy has his own ranked preference list of girls.
3. Each girl has his own ranked preference list of boys.
4. The lists are complete & no ties.

Each boy ranks every girl & vice versa.

Example:

Boys	Girls		
1	A	B	C
2	B	A	C
3	A	B	C

Girls	Boys		
A	2	1	3
B	1	2	3
C	1	2	3

Q How to pair them up?

Unstable Pair: If a boy $\ddot{b}$ and a girl $\ddot{g}$ are paired with other partners, but $\ddot{b}$ prefers $\ddot{g}$ to his current partner and $\ddot{g}$ prefers $\ddot{b}$ to her current partner. We call such a pair (b,g) is an unstable pair or ROGUE couple.

Problem: Pair each boy with a unique girl so that there are no ROGUE couples (Unstable Pairs).

Example :-

Boy	Girl		
1	A	B	C
2	B	A	C
3	A	B	C

Girl	Boy		
A	2	1	3
B	1	2	3
C	1	2	3

Stable matching (without unstable / Rogue couple)

{(1, A), (2, B), (3, C)} , {(1, B), (2, A), (3, C)}

unstable matching {(1, C), (2, B), (3, A)}

1 and B form a rogue couple,

Since 1 would rather be with B than C &

since B would rather be with 1 than 2

History

- This algorithm won Noble Prize for Economics in 2012
  [Alvin Roth, Lloyd Shapley]
  " For the theory of stable allocations and the Practice of market design"

- The algorithm was Proposed by David Gale and Lloyd Shapley (1962)

Stable marriage Problem

Warm up question

Q1 True/False:

For any given ranked preference lists of boys and girls, there is a unique stable matching.

If false give an example with two boys and two girls in which there is more than one stable matching.

Algorithm (Gale-Shapley) [West's text book]

(GS-Algorithm in Short)

① Each boy proposes to the highest girl on his preference list who has not previously rejected him.

② If each girl receives exactly one proposal, STOP and use the resulting matching.

③ Otherwise, every girl receiving more than one proposal rejects all of them except the one that is highest on her preference list.

④ Every girl receiving a proposal says "maybe" to the most attractive proposal received.

Remark: I use "day" to represent a step of the algorithm.

Example :-

$$b_1 \to (g_3, g_2, g_5, g_1, g_4)$$
$$b_2 \to (g_1, g_2, g_5, g_3, g_4)$$
$$b_3 \to (g_4, g_3, g_2, g_1, g_5)$$
$$b_4 \to (g_1, g_3, g_4, g_2, g_5)$$
$$b_5 \to (g_1, g_2, g_4, g_5, g_3)$$

$$g_1 \to (b_3, b_5, b_2, b_1, b_4)$$
$$g_2 \to (b_5, b_2, b_1, b_4, b_3)$$
$$g_3 \to (b_4, b_3, b_5, b_1, b_2)$$
$$g_4 \to (b_1, b_2, b_3, b_4, b_5)$$
$$g_5 \to (b_2, b_3, b_4, b_1, b_5)$$

If we run the Gale-Shapley algorithm on the above example.

We get the matching

$M = \{ (g_1, b_5), (g_2, b_2), (g_3, b_4), (g_4, b_3), (g_5, b_1) \}$

We have to show the following

1. Algorithm terminates
2. Perfect Matching (every one is matched)
3. Produces Stable Matching (No unstable pairs)

Theorem 1: The Gale - Shapley algorithm terminates in atmost n^2 days.

Proof:- Proof by Contradiction.

Suppose the algorithm does not terminate in n^2 days.

On a day in which GS algo doesn't terminate it must be because some girl must have atleast 2 suitors. i.e., some boy crosses a girl off his list on the same day.

So if GS algo doesn't terminate in n^2 days, there are atleast n^2+1 names crossed off in total. But at the start each list has size n, so the total size of all lists together is n^2,

So we couldn't have crossed off n^2+1 names, which is a contradiction.

Some Observations Pursue

Observation 1: If a boy marries, then he courted every girl he liked better

Observation 2: If a boy never marries, then he courted every girl.

Observation 3: A girl marries her favorite among her suitors.

Observation 4: If a girl is ever courted, she gets married.

Theorem: Everyone is married in GS algorithm.

Proof:- Suppose a boy b is not married,

by observation-2, he courted every girl.

then by observation-4, every girl is married.

But # of boys = # of girls = n, ie, b is

also married which is a contradiction.

Theorem: The GS Algorithm produces stable marriages.

Proof. By Contradiction

Suppose there is rogue couple (b, g)

Suppose b is married to g'

g is " " b' in GS Algo

if b married g', but likes g better,

then b visited g first and g said "NO" to b.

But then g must have married someone that she likes better than b. So g likes b' better than b which means (b, g) is not a rogue couple.

Q) Who do you think better off in GS algorithm?

Proposers or acceptors?

In our case: boys girls

Let S be the set of all Stable matchings.

$S \neq \phi$ as GS-Algorithm gives a Stable matching.

For each Person P,

Possible Partners of $P = \{ q \mid \exists M \in S, (P,q) \in M \}$

"That is, q is a Possible Partner of P iff there is a Stable matching where P marries q"

Also observe that Some mates just might be out of the question, Since no Stable Pairings are Possible if you marry them.

Eg:

$b_1 \to g_1 \; g_2 \; g_3$ $\qquad$ $g_1 \to b_3 \; b_2 \; b_1$

$b_2 \to g_1 \; g_2 \; g_3$ $\qquad$ $g_2 \to b_3 \; b_2 \; b_1$

$b_3 \to g_1 \; g_2 \; g_3$ $\qquad$ $g_3 \to b_3 \; b_2 \; b_1$

g_1 is a never a Possible mate of b_1, o.w (g_1, b_3) is a Rogue Couple.

Def: A Person's optimal mate is his/her favorite from the Possible Partners set.

Def: A Person's Worst mate is his/her LEAST favorite from the Possible Partners set.

Example:-

Boy	Girl		
1	A	B	C
2	B	A	C
3	A	B	C

Girl	Boy		
A	2	1	3
B	1	2	3
C	1	2	3

Stable Pairs

$M_1 = \{(1,A), (2,B), (3,C)\}$, $M_2 = \{(1,B), (2,A), (3,C)\}$

Q Are there any other stable matchings?

Boy 1's optimal mate is A
Boy 1's Worst mate is B
Boy 3's optimal mate is C
Boy 3's Worst mate is C

Girl A's optimal mate 2
Girl A's worst mate 1
Girl B's optimal mate 1
Girl B's worst mate 2

Theorem:

ⓐ The GS-Algorithm Pairs every boy with his optimal mate.

ⓑ " " " " " girl " her worst mate.

Proof: ⓐ Proof by contradiction.

Assume that some boy doesn't get his optimal girl.

Let b be the first (in time) boy who gets rejected by his optimal girl g.

Let b' be the boy who caused g to reject b in the GS-algorithm. Then g prefers b' to b. — ①

Since b is the first person to be rejected by the optimal mate in GS-algorithm, b' is not (yet) been rejected by the optimal mate when he proposed g.

So b' likes g at least as much as he likes his optimal mate g^* (g & g^* might be the same person). → Ⓐ

Let M be a stable matching where b marries g.

M exists because g is in the possible partners of b.

Also M is not produced by GS-Algorithm by assumption.

Let g' be the spouse of b' in M.

By definition, b' likes g^* at least as much as g' → B

(again g^* & g' might be same)

From A and B,

b' prefers g over g'. — 2

So (b', g) is a rogue couple.

↳ follows from 1 and 2

Which contradicts the fact that M is a stable matching.

Proof of Part b is similar to Part a

College Admission :- (generalization of Stable marriage Problem)

There are n students $S_1, S_2, \ldots S_n$

m Universities $U_1, U_2, \ldots U_m$

University U_i has n_i slots for student &

$$\sum_{i=1}^{m} n_i = n$$

Each student ranks all Universities & each University ranks all students.

Goal is to assign students to universities with the following properties.

1. Every student is assigned one University
2. University U_i gets n_i students
3. There does not exist S_i, S_j, U_k, U_l where student S_i assigned to U_k, S_j assigned to U_l student S_j prefers University U_k to U_l & University U_k prefers student S_j to S_i.

(4) It is student optimal.

Every student gets his/her top choice of University amongest these assignments satisfying first three rules.

Algorithm for College admission

The algorithm will be a minor modification of the Stable marriage algorithm.

① Each ~~boy~~ Student ~~Proposes~~ applies to the ~~highest~~ favorite ~~girl~~ University on his/her Preference list who has not Previously rejected him.

② If each ~~girl~~ University receives exactly ~~one Proposal~~ n_i applicants, STOP and use the resulting matching.

③ Otherwise, every ~~girl~~ University receiving more than ~~one Proposal~~ n_i applicants rejects all of them except the ~~one that is highest on her~~ top n_i applicants in their Preference list.

④ Every ~~girl~~ University receiving a Proposal says "maybe" to the ~~most attractive Proposal received~~ top n_i applicants.

Again, We have to show the following

1. Algorithm terminates
2. Every Student is assigned to one University
3. Each student is assigned to his/her optimal University.

College Admission generalized :

n students $(S_1 .. S_n)$

m Universities $(U_1 .. U_m)$

University U_i has n_i slots for students

$$\boxed{\sum_{i=1}^{m} n_i \leq n}$$

Here we have surplus students, some students may not be assigned to any university.

Q How to define instability in this case ?

Two types of instability

- 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.
- 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.

Truthfulness :-

Q :- Can a boy or a girl end up better off by lying about his/her Preferences?

Example :-

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

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

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

By Running GS- Algo on above example we get

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

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

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

$b_1 \rightarrow g_3$ $b_2 \rightarrow g_1$ $b_3 \rightarrow g_2$

Now, suppose g_3 pretends she prefers b_3 to b_1

i.e, $g_3 \rightarrow (b_2, b_3, b_1)$

Again run the algorithm on this modified input.

$b_1 \rightarrow (\cancel{g_3}, g_1, g_2)$ $\qquad$ $g_1 \rightarrow (b_1, b_2, b_3)$

$b_2 \rightarrow (\cancel{g_1}, g_3, g_2)$ $\qquad$ $g_2 \rightarrow (b_1, b_2, b_3)$

$b_3 \rightarrow (\cancel{g_3}, \cancel{g_1}, g_2)$ $\qquad$ $g_3 \rightarrow (b_2, b_3, b_1)$

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

As we see girl g_3 ends up with the boy b_2 who is her true favorite.

So falsely switching order of preferences may be able to give a more desirable partner in the GS-Algorithm.

Eg

$b_1 \to g_1\ g_2\ g_3$ $\qquad$ $g_1 \to b_2\ b_1\ b_3$

$b_2 \to g_2\ g_1\ g_3$ $\qquad$ $g_2 \to b_1\ b_3\ b_2$

$b_3 \to g_2\ g_3\ g_1$ $\qquad$ $g_3 \to b_2\ b_1\ b_3$

GS-Algo o/p: $(b_1, g_2)\ (b_2, g_1)\ (b_3, g_3)$

spse b_3 pretends he prefers g_3 to g_2.

$b_1 \to g_1\ g_2\ g_3$ $\qquad$ $g_1 \to b_2\ b_1\ b_3$

$b_2 \to g_2\ g_1\ g_3$ $\qquad$ $g_2 \to b_1\ b_3\ b_2$

$b_3 \to \cancel{g_2}\ \cancel{g_3}\ g_1$ $\qquad$ $g_3 \to b_2\ b_1\ b_3$

$\qquad\ \ g_3\ \ g_2$

GS-Algo o/p: $(b_1, g_1)\ (b_2, g_2)\ (b_3, g_3)$

The GS algorithm is a truthful mechanism from the point of view of the proposing side.

That is "No proposer can get a better matching by misrepresenting their preferences"

Moreover,

No coalition of proposers can coordinate a misrepresentation of their preferences such that all proposers in the coalition are strictly better-off.

However, it is possible for some coalition to misrepresent their preferences such that some proposers are better-off and other retain the same partner.

The GS-Algorithm is non-truthful for the non-proposing participants.

Each may be able to misrepresent their preferences and get a better match.

Theorem. Suppose several boys collude in a Gale-Shapley algorithm, each using a true or false preference list. Then they cannot all end up better off, relative to each boy's true preference list.

Variants of Stable marriage Problem:

① Forbidden Pairs: In this case some boy-girl Pairs are forbidden.

② Indifferences: In this case we allow ties in the ranking.

Example: $g_1 \rightarrow (b_1, b_2 \text{ or } b_3, b_4)$