15-251: Great Theoretical Ideas in Computer Science Anupam Gupta
Notes on Stable Matchings (draft!!) January 28, 2011

There are n boys and n girls. The boys are called $b_1, b_2, \ldots, b_n$, and the girls are called $g_1, g_2, \ldots, g_n$. Each boy has a rank-ordered list of the n girls (with the highest-ranked girl listed first, etc), and similarly each girl has a rank-ordered list of the n boys. There are no ties.

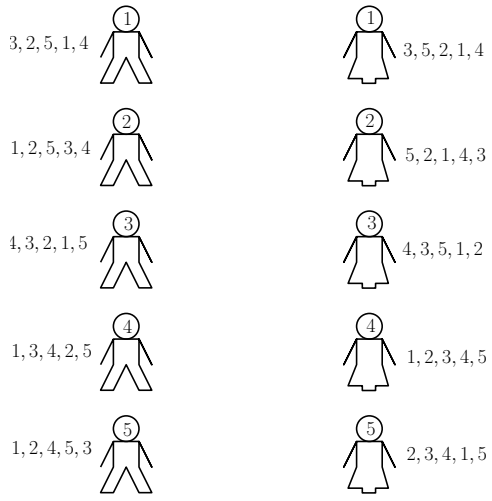

Figure 1: An instance of stable matching

Suppose we want to match the boys and girls to each other. (Any boy can be matched to any girl, it's just that the boy's happiness depends on how highly the girl places in his ranking, and the girl's happiness depends on how highly the boy places in her ranking.)

What criteria might we use to match the men and girls?

- We could try to minimize how far down we have to go in any list. Say, if every person was matched to someone who is the top person on their list, that'd be amazing. But maybe if every person was matched to someone who is within the top 3 of their list, that'd be pretty good too.
- We could do the same, but just for the boys, or the girls, and ignore the happiness of the other size. In fact, in the example above, it turns out that each of the girls has a different boy ranked as #1, so we could just output that matching. The girls would be very happy, but boy #1 is not that happy (he gets his last place girl).
- We could imagine an "unhappiness cost" of i for matching some person to a person i^{th} on their list. And try to minimize the unhappiness cost.

  Or the cost could be i^2, or 2^i, or something growing more rapidly.
- Or maybe some combination of these? Or some other criterion altogether?

For today, we will not consider these issues at all. We will consider a orthogonal but important issue: that of *stability*. In a situation like the one we are modeling, where there is no overseeing authority, and where the people involved selfishly want to maximize their own self interest (that is, they want the best partner they can get), a reasonable requirement from any matching we output is that it be stable: people should have no incentive to break the proposed matching and choose something else.

In this lecture, we will give algorithms to find stable matchings.

1 Stable Matchings

So how do we find a stable matching of the boys to the girls? But before that, let us carefully define what this "stability" is.

Well, a matching would be unstable if a boy b and a girl g who are not matched to each other, preferred each other to their current partners. They would want to dump their partners and match up. E.g., in Figure 1, consider the matching that matches boy i to girl i—now boy 5 and girl 1 prefer each other to their current partners, as do boy 5 and girl 2, or boy 3 and girl 4, etc.

Given a matching M of the boys and girls, let us say that (b, g) is a *rogue couple* in M if

1. b, g are not matched by M, but
2. b prefers g to his partner in M, and g prefers b to her partner in M.

Now we say that a matching M is *stable* if there are no rogue couples in M.

Here are a couple of stable matchings in the example from Figure 1:

- The matching $(b_3, g_1), (b_5, g_2), (b_4, g_3), (b_1, g_4), (b_2, g_5)$ is one. Why? Each girl is matched to her very top choice, so she cannot be part of a rogue couple!
- the matching $(b_1, g_5), (b_2, g_2), (b_3, g_4), (b_4, g_3), (b_5, g_1)$ is another one.

You should check that these are indeed stable matchings. Which other stable matchings can you find?

This leads us to the natural question:

> *Given boys and girls with rank-ordered lists, how do we find a stable matching?*

But before that, we need to answer an even more basic question

> *Given boys and girls with rank-ordered lists, does there always exist a stable matching?*

In the rest of the lecture, we will show that there always exists a stable matching, no matter what the preference lists! In fact, the proof will also give a very simple algorithm to find one. Coming up, right after a word from our sponsors.

2 Does There Always Exist a Stable Matching?

How do we show there always exists a stable matching?

A natural approach is: start with any old matching M_1 of boys to girls. Suppose there is no rogue couple in M_1, we've found one. Otherwise, pick a rogue couple and have them dump their partners and match them up, match the two dumpees together, and repeat. Maybe this "evolutionary" approach will eventually converge to a stable matching?

Maybe. How would we prove this converges?

Well, the newly-formed happy couple are now matched to people higher in their lists. But the couple formed by the dumpees, they may really hate each other. Hmm.

At this point, it is instructive to consider the closely related "roommates" problem. Again, there are $2n$ people, but each of them rank-orders *all* the others, not just the people of the opposite gender. And again, we can define stability of a matching the same way: the absence of *rogue roommates* (those who are not matched to each other, but prefer each other to their current mates).

Now, look at this example (think of poor person 4 as the roommate no one wants):

In this case of the roommates problem, there is no stable matching! Say you match person 4 to person 1 (and 2 and 3 together). Then 3 prefers 1 to his current partner 2, and 1 prefers 3 to 4, so they form a rogue pair. If we let nature take its course, now the matching will be $(1, 3)$ and $(2, 4)$. But now $1, 2$ form a rogue pair. And so on. So a evolutionary approach would never converge.

This says that if we are to show that an evolutionary approach eventually gives us a stable matching, we need a proof idea that must fail for the roommates problem.

Hmm again.

In fact, Knuth showed in 1990 that just doing evolutionary approach by choosing an arbitrary rogue couple may result in the process cycling indefinitely without reaching a stable matching. (See the comments at the end of the notes.)

So now let's see what does work.

2.1 The Traditional Marriage Algorithm (TMA)

So here's a simple algorithm that always succeeds in finding a stable matching. We'll call it the *traditional marriage algorithm*, since it uses some stereotypes as visual aids.

> Every day, each boy goes to highest ranked girl on his list who hasn't rejected him yet.
>
> Each girl now has some number of boys who've come to her. (Maybe none.) She **rejects** all these boys, except the one who's highest among them on her list. (She updates her status to say she is "in a relationship with" this boy.)
>
> The rejected boys cross the girl's name off their lists.

If no boys are rejected on some day, the process ends, and the girls are matched to the unique boy who has come to her on this day.

That's it. Let's run this algorithm on the instance we gave in Figure 1.

Day	Girl 1	Girl 2	Girl 3	Girl 4	Girl 5
1	2,4,5		1	3	
2	5	2	1,4	3	
3	5	1,2	4	3	
4	5	2	4	3	1

We'll now prove that this algorithm is correct. The first question is: is the algorithm well-defined? Will there always be a girl on each boy's list who hasn't rejected him yet, or can it be the case that some boy crosses off every name on his list? To answer this, we first need the following lemma.

Lemma 2.1 (Improvement Lemma) *If a girl is in a relationship with someone, then she remains in a relationship with someone for the rest of the algorithm. Moreover, the boy she is in a relationship with cannot get worse over time (according to her ranked list).*

Proof: If g is in a relationship with b, then she's the highest ranked girl on b's list who hasn't rejected her. So b will keep coming back, until either she rejects him for someone higher on her list, or the process ends. Hence, she always has a mate from then on, and also her mate at any time cannot ever get worse. ■

Corollary 2.2 *Each girl is eventually matched with her favorite among the boys who visit her during the TMA.*

Lemma 2.3 *No boy will be rejected by all the girls in the TMA.*

Proof: If a boy b is rejected by some girl g, she must be in a relationship with someone from that point onwards. Hence, once b is rejected by all girls, they must all be in a relationship with others. But there are n girls and $n-1$ other boys, this is not possible. ■

Good, so the algorithm is well-defined. But does it always terminate?

Lemma 2.4 *The TMA terminates in at most $n^2 - n + 1$ days.*

Proof: Consider all the boys lists, and look at the total number of names which haven't been crossed off. Each day the algorithm does not end, some name gets crossed off. Eventually, there will be at least 1 name not struck off on each list for a total of n names, and originally this total number was n^2. So the number of days before the algorithm terminates is n^2-n+1.

Exercise: can you show a set of rankings for boys/girls that will make TMA indeed take $n^2 - n + 1$ days to terminate? ■

Good, good. Now, to show that the matching produced is stable.

Lemma 2.5 (Stability) *The TMA outputs a stable matching M.*

Proof: Suppose not, and let (g^*, b^*) be a rogue couple in the TMA's matching: they prefer each other to their partners in this matching. Let TMA match g^* to b and b^* to g. Since b^* prefers g^* to g, he must have gone to her on some previous day, but was rejected.

g^* must have rejected b^* for a person she prefers to him. And by the Improvement Lemma, she prefers her eventual partner at least as much. So she cannot form a rogue couple with b^*. ■

Simple and elegant!

Note that this is just way to get a stable matching. If we switch the roles of the boys and girls, that gives us another stable matching, possibly a different one. And in general, there may be other stable matchings that are not produced by the TMA.

3 Optimal and Pessimal Matchings

A natural next question is: does the TMA give a result that is better for the boys, or for the girls, or neither? To answer this, we come back to the sticky question of how we should define "better".

One thing we can indeed show is this: if the boys and girls switch roles, with the girls now proposing and the boys rejecting/aceepting, then each girl will get a match that is no worse than their matches in the TMA, and the boys mates will be no better. So, in this sense, the girls will definitely benefit from swapping roles with the boys.

But in fact, we will show something much stronger: we'll show that the matching produced by the TMA is optimal for the boys and pessimal for the girls! (The preceding paragraph's claims will follow from what show.)

3.1 Defining Optimality

Given a set of preference lists, how should we define a person's optimal partner? It may not make sense to define this as the person at the top of their list—there may be no way that the two can be matched in any stable matching.

So let's define a boy's *optimal partner* as the highest ranked girl (according to his own rankings, of course) to whom this boy can be matched in *some* stable matching. And his *pessimal partner* as the lowest ranked girl to whom he can be matched in some stable matching. Similarly, one can define girls' optimal and pessimal partners.

Exercise: can you find a set of preference lists where some person's optimal girl is, in fact, the girl ranked last on his list? What about a case where person's pessimal girl is, in fact, the girl ranked highest on his list?

An immediate question arises: can two boys have the same optimal girl?

Lemma 3.1 *No two boys have the same optimal girl.*

Proof: Suppose two boys b_1 and b_2 did have the same optimal girl g. So there must be two stable matchings M_1 and M_2 such that b_1 is matched to g in M_1, and b_2 is matched to g in M_2. And say g prefers b_1 to b_2.

Now consider M_2. g is matched to b_2 but she prefers b_1. Also, b_1 is matched to someone but he prefers g to her (since g was his optimal girl). So (b_1, g) would be a rogue couple in M_2, a contradiction. ■

Similarly, no two boys have the same pessimal girl, and similarly, all girls have distinct optimal boys and distinct pessimal boys.

3.2 How Does the TMA Do?

We call a matching *male-optimal* if each boy is matched to his optimal girl. A matching *female-pessimal* if each girl is matched to her pessimal boy. By Lemma 3.1 above, at least these are both valid matchings. But are these stable matchings? Surprisingly, this is indeed the case. Even more surprising are the following two theorems.

Theorem 3.2 *The stable matching M produced by the TMA is male-optimal.*

Proof: Suppose not. So consider the first moment in time when some boy b is rejected by his optimal girl g. Say she is rejecting him for b^*, whom she likes more.

Note that boy b^* has not yet been rejected by his optimal girl—indeed, no boy has been rejected by his optimal girl before this moment in time. So b^* likes g at least as much as his optimal girl.

Since g is b's optimal girl, there must be some stable matching S where (b, g) are matched. We just showed that g prefers b^* to b. And we just showed that b^* likes g at least as much as his optimal girl, and hence at least as much as his partner in S. So (b, g^*) are a rogue couple in S, a contradiction!

Hence there is no boy who's rejected by his optimal girl in the TMA. So every boy is matched to his optimal girl in the TMA. ■

Theorem 3.3 *Any male-optimal matching M must also be female pessimal.*

Proof: Suppose not. Then there exists a girl g matched to b in M, and another stable matching S matching (g, b^*) and b^* is even worse for g. Then, g prefers b to her partner b^* in S. And g is b's optimal girl, so b prefers g to his partner in S. This means (g, b) is rogue in S, a contradiction! ■

In retrospect, it makes sense that the TMA gives the boys an advantage, since they are in a position to proactively seek out their favorites, but it is still surprising that the TMA produces the simultaneously best possible stable matching for the boys and the worst possible stable matching for the girls!

4 The TMA in the Real World

A variant of the TMA is actually used in the *National Residency Matching Program*, where medical school students are matched to residency programs. In fact, the NMRP algorithm had been developed ten years before David Gale and Lloyd Shapley published their paper proposing the TMA algorithm, and was essentially the same.

The students rank colleges, and the colleges rank students. For some time the process was run to be college-optimal, but since 1995, it has been changed to a student-optimal one. Moreover, the NMRP process is a bit more involved: the ranked lists are not complete, and also couples can apply together. Moreover, there is the game-theoretic questions to consider: is it in the students/colleges best interest to truthfully rank the other side?

5 Future Directions

As we mentioned, Knuth showed that just choosing an arbitrary rogue couple to resolve at each step can cause the process to cycle without reaching a stable matching. Roth and Vandevate showed that if we could choose the rogue couple carefully at each step, there is a "short" sequence of $O(n^2)$ break-ups and rematchings that leads to a stable matching. However, their proof already uses the fact that a stable matching exists, they just show how to get there — it does not give a different proof of the existence of stable matchings. Note that the result of Roth and Vandevate implies that the state space is connected, hence the random dynamics (where we choose a rogue pair at random) will eventually reach stability. However, recently, Ackermann et al. showed that there are instances where this random process can take exponentially long to converge to a satble matching.

Other things we could discuss: What about the asynchronous version, where on each day, only some of the boys go to their best current girls, and a girl says "maybe" to the better of her current boy, and the best of the boys who've come today. (Note: her current boy may be taking the day off.) Does this work?

What about collusion: what if the matching could be changed by a rogue $2k$-tet: this is a group of people who jointly decide to change partners so that everyone in this group is better off?

Another set of interesting questions are a *game-theoretic* ones: *can people lie about their preferences to get better results for themselves*? Think of this as being a situation where everyone submits their list to a central authority, who them runs TMA on these lists. Currently we assumed that each person truthfully reports their preferences. But could there be a situation where if a person submits a fake list, they get a mate who's better for them (higher on their real list) than if they'd submitted their real list? As an example: Gale and Shapley had shown that the non-proposing side (i.e., the girls in the TMA) could lie about their preferences to get a better result, but there is no advantage gained by lying if you are on the proposer's side. (It is an interesting exercise to show these facts.)

There are many other questions, some mathematical, some algorithmic, some economic and game-theoretic: see the Wikipedia article, books by Knuth, and by Gusfield and Irving, and many papers and articles.