CS 70 Spring 2008 — Discrete Mathematics for CS — David Wagner — Note 7

Stable Marriage — An Application of Proof Techniques to Algorithmic Analysis

Consider a dating agency that must match up n men and n women. Each man has an ordered *preference list* of the n women, and each woman has a similar list of the n men (ties are not allowed). Is there a good algorithm that the agency can use to determine a good pairing?

Example

Consider for example $n = 3$ men (represented by numbers 1, 2, and 3) and 3 women (A, B, and C), and the following preference lists:

Men	Women		
1	A	B	C
2	B	A	C
3	A	B	C

Women	Men		
A	2	1	3
B	1	2	3
C	1	2	3

For instance, the preference lists above mean that woman A is man 1's top choice; woman B is his second choice; and so on.

What properties should a good pairing have? One possible criterion for a "good" pairing is one in which the number of first ranked choices is maximized. Another possibility is to minimize the number of last ranked choices. Or perhaps minimizing the sum of the ranks of the choices, which may be thought of as maximizing the average happiness. In this lecture we will focus on a very basic criterion: *stability*. A pairing is unstable if there is a man and a woman who prefer each other to their current partners. We will call such a pair a *rogue couple*. So a pairing of n men and n women is stable if it has no rogue couples.

An unstable pairing from the above example is: $\{(1,C),(2,B),(3,A)\}$. The reason is that 1 and B form a rogue couple, since 1 would rather be with B than C (his current partner), and since B would rather be with 1 than 2 (her current partner). This is trouble: Before long, 1 and B are going to spending many late nights doing CS70 problem sets together. Obviously, the existence of rogue couples is not a good thing if you are a matchmaker, since they will lead to instability or customer dissatisfaction. That is why we focus on stable pairings.

An example of a stable pairing is: $\{(2,A),(1,B),(3,C)\}$. Note that $(1,A)$ is not a rogue couple. It is true that man 1 would rather be with woman A than his current partner. Unfortunately for him, she would rather be with her current partner than with him. Note also that both 3 and C are paired with their least favorite choice in this matching. Nonetheless, it is a stable pairing, since there are no rogue couples.

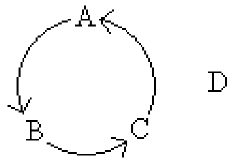
The problem facing us is to find a stable pairing, given the preference lists for all n men and all n women.

Does a Stable Pairing Always Exist?

Before we discuss how to find a stable pairing, let us ask a more basic question: do stable pairings always exist? Surely the answer is yes, since we could start with any pairing and make it more and more stable as follows: if there is a rogue couple, modify the current pairing so that they are together. Repeat. Surely this procedure must result in a stable pairing! Unfortunately this reasoning is not sound. To demonstrate this, let us consider a slightly different scenario, the roommates problem. Here we have $2n$ people who must be paired up to be roommates (the difference being that unlike the dating scenario, a person can be paired with any of the remaining $2n-1$). The point is that nothing about the above reasoning relied on the fact that men can only be paired with women in the dating scenario, so by the same reasoning we would expect that there would be a stable pairing for the roommates problem as well. The following counter-example illustrates the fallacy in the reasoning:

Roommates			
A	B	C	D
B	C	A	D
C	A	B	D
D	–	–	–

Visually, we have the following situation:



What is interesting about this problem is that there is no stable pairing (i.e., there is always a rogue couple). For example, the pairing $\{(A,B),(C,D)\}$ contains the rogue couple B and C. Using the reasoning above, we might decide to pair B and C together, giving us the pairing: $\{(B,C),(A,D)\}$. But this pairing is also unstable because now A and C are a rogue couple. In fact, in this example there is no stable pairing. Thus any proof that there must be a stable pairing in the dating problem must use the fact that there are two genders in an essential way.

In fact, we shall show that, when we have n men and n women, a stable pairing always exists. This fact is somewhat surprising, but true.

The Traditional Marriage Algorithm

We will study what is sometimes called the “Traditional Marriage Algorithm” (TMA), so-called because it is based on a 1950’s model of dating where the men propose to the women, and the women accept or reject these proposals. We will prove that the Traditional Marriage Algorithm always finds a stable pairing and study its many interesting properties.

The Traditional Marriage Algorithm works like this:

Every Morning: Each man goes to the first woman on his list not yet crossed off and proposes to her.

Every Afternoon: Each woman says “Maybe, come back tomorrow” to the man she likes best among the men who have proposed to her (she now has him on a string) and “No, I will never marry you!” to all the rest.

Every Evening: Each rejected suitor crosses off the woman who rejected him from his list.

The above loop is repeated each successive day until there are no more rejected suitors. On this day, each woman marries the man she has on a string.

We wish to show that this algorithm always outputs a stable pairing. But how do we even know that it must terminate? Let us prove something stronger: the algorithm is guaranteed to terminate in at most n^2 days.

Well, for each day (except the last), at least one woman is crossed off some man's list. Since there are n men, each starting with a list of size n, it follows that the algorithm must terminate after n^2 days.

To establish that the pairing output is stable, we need the following crucial lemma:

Improvement Lemma: If woman W has man M on a string on the kth day, then on every subsequent day she has someone on the string whom she likes at least as much as M.

Proof: Suppose that on day ℓ (for some $\ell \geq k$) W has some man M' on a string, where she likes M' at least as much as M. (Possibly $M = M'$.) Since she has M' on a string, that means that she told M' "maybe" on day ℓ. On day $\ell + 1$, M' will come back and propose to W again, since he was told "maybe" the previous day. So W has the choice of at least one man on day $\ell + 1$. Let M'' be her favorite choice among all of those who propose to her on day $\ell + 1$. She will tell M'' "maybe" on day $\ell + 1$, so on day $\ell + 1$ she will have M'' on a string. (Possibly $M'' = M'$.) Note that since M' was one of the proposers on day $\ell + 1$, this means that she must like M'' at least as much as she likes M', and as we stated before, she likes M' at least as much as she does M. Therefore on day $\ell + 1$ she has a man (namely, M'') on a string who she likes at least as much as M. We have proven that if the property is true on day n, then it is true on day $\ell + 1$, so by induction on n, it must be true for all $n \geq k$. $\square$

Let us now proceed to prove that at the end of the algorithm all $2n$ people are paired up. Before reading the proof, see if you can convince yourself that this is true. The proof is remarkably short and elegant and is based on the Improvement Lemma:

Lemma: The Traditional Marriage Algorithm terminates with a pairing.

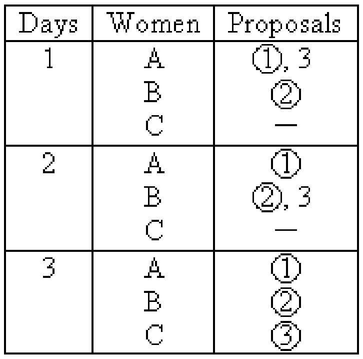
Proof: Suppose for contradiction that there is a man M who is left unpaired at the end of the algorithm. He must have proposed to every single woman on his list. By the improvement lemma, each of these women thereafter has someone on her string. Thus when the algorithm terminates, n women have n men on the string not including M. So there must be at least $n + 1$ men. Contradiction. So each man is paired up with his own partner, which means that all n women have a partner as well. $\square$

Now, before we prove that the output of the algorithm is a stable pairing, let us first do a sample run-through of the stable marriage algorithm. We will use the preference lists given earlier, which are duplicated here for convenience:

Men	Women		
1	A	B	C
2	B	A	C
3	A	B	C

Women	Men		
A	2	1	3
B	1	2	3
C	1	2	3

The following table shows which men propose to which women on the given day (the circled men are the ones who were told "maybe"):



Days	Women	Proposals
1	A	①, 3
	B	②
	C	—
2	A	①
	B	②, 3
	C	—
3	A	①
	B	②
	C	③

Thus, the pairing which the algorithm outputs is $\{(1,A),(2,B),(3,C)\}$, and this is a stable pairing.

Theorem: The pairing produced by the Traditional Marriage Algorithm is always stable.

Proof: We will show that no man M can be involved in a rogue couple. Consider any couple (M,W) in the pairing and suppose that M prefers some woman W^* to W. We will argue that W^* prefers her partner to M, so that (M,W^*) cannot be a rogue couple. Since W^* occurs before W in M's list, he must have proposed to her before he proposed to W. Therefore, according to the algorithm, W^* must have rejected him for somebody she prefers. By the improvement lemma, W^* likes her final partner at least as much, and therefore prefers her final partner to M. Thus no man M can be involved in a rogue couple, and the pairing is stable. $\square$

Optimality

Consider the situation in which there are 4 men and 4 women with the following preference lists:

Men	Women			
1	A	B	C	D
2	A	D	C	B
3	A	C	B	D
4	A	B	C	D

Women	Men			
A	1	3	2	4
B	4	3	2	1
C	2	3	1	4
D	3	4	2	1

For these preference lists, there are exactly two stable pairings: $S = \{(1,A),(2,D),(3,C),(4,B)\}$ and $T = \{(1,A),(2,C),(3,D),(4,B)\}$. The fact that there is more than one stable pairing brings up an interesting question. What is the best possible partner for each person? For instance, who is the best possible partner who man 2 could hope to end up with? The trivial answer is his first choice (i.e., woman A), but that is just not a realistic possibility for him. Pairing man 2 with woman A would simply not be stable, since he is so low on her preference list. And indeed there is no stable pairing in which 2 is paired with A. Examining the two stable pairings, we can see that the best possible realistic outcome for man 2 is to be matched to woman D.

Let us make some definitions to better express these ideas: we say the *optimal* woman for a man is the highest woman on his list whom he is paired with in some *stable* pairing. In other words, the optimal woman is the best that a man can do under the condition of stability. In the above example, woman D is the optimal woman for man 2. So the best that each man can hope for is to be paired with his optimal woman. But can all the men achieve this optimality *simultaneously*? In other words, is there a stable pairing such that each man is paired with his optimal woman? If such a pairing exists, we will call it a *male optimal* pairing. Turning to the example above, S is a male optimal pairing since A is 1's optimal woman, D is 2's optimal woman, C is 3's optimal woman, and B is 4's optimal woman. Similarly, we can define a female optimal pairing, which is the pairing in which each woman is paired with her optimal man. [Exercise: Check that T is a female optimal pairing.] We can also go in the opposite direction and define the *pessimal* woman for a man to be the lowest ranked woman whom he is ever paired with in some stable pairing. This leads naturally to the notion of a *male pessimal* pairing — can you define it, and also a female pessimal pairing?

Now, a natural question to ask is: Who is better off in the Traditional Marriage Algorithm: men or women? Think about this before you read on...

Theorem: The pairing output by the Traditional Marriage Algorithm is male optimal.

Proof: Suppose for the sake of contradiction that the pairing output by the TMA is *not* male optimal. This means there must exist an earliest day, let's say day k, in which some man was rejected by his optimal wife. Call that man M. (There might be multiple men who were rejected by their optimal wives on day k; in that case, we choose one of those men arbitrarily.) Let W be M's optimal wife. Since M was rejected by W on day k, on that day W must have accepted another man, let's call him M^*, who she likes better than M. Because no man was rejected by his optimal wife before day k, and because M^* was not rejected on day k, we conclude that M^* has not yet been rejected by his optimal wife on day k. Therefore, there are only two possibilities: (1) W is M^*'s optimal wife; or, (2) on day k, M^* has not yet proposed to his optimal wife and hence likes W better than his optimal wife.

Since W is M's optimal wife, this means there must exist some stable pairing, call it T, where they are paired together. In T, M^* must have been paired off with someone else, call her W^*. In other words, T looks like this: $T = \{\ldots,(M,W),\ldots,(M^*,W^*),\ldots\}$. Since T is stable, there are only two possibilities: (a) W^* is M^*'s optimal wife; or, (b) M^* likes his optimal wife better than W^*.

All in all, we have four cases. We will show that each of these cases is impossible or leads to a contradiction.

- Case (1)(a) is impossible: W and W^* are two different women, so they can't both be M^*'s optimal wife.
- In case (1)(b), M^* likes W, his optimal wife, better than W^*.
- In case (2)(a), M^* likes W better than W^*, his optimal wife.
- In case (2)(b), M^* likes W better than his optimal wife and his optimal wife better than W^*, so M^* likes W better than W^*.

In every possible case, M^* likes W better than W^*. And, as we mentioned earlier, W likes M^* better than M. But this means that (M^*,W) form a rogue couple in T: M^* likes W better than his wife in T, and W likes M^* better than her husband in T. The existence of a rogue couple in T contradicts the stability of T. Contradiction. Thus our original assumption—that the TMA-pairing is not male-optimal—must have been incorrect. $\square$

What proof techniques did we use to prove this theorem? We used the well-ordering principle. How do we see it as a regular induction proof? This is a bit subtle to figure out. See if you can do so before reading on... the proof is really showing by induction on k the following statement: for every k, no man gets rejected by his optimal woman on the kth day. [Exercise: Can you complete the induction?]

So men appear to fare very well by following the Traditional Marriage Algorithm. How about the women? The following theorem reveals the sad truth:

Theorem: If a pairing is male optimal, then it is also female pessimal.

Proof: Let $T = \{\ldots,(M,W),\ldots\}$ be the male optimal pairing. Consider any stable pairing where W is paired with someone else, say, $S = \{\ldots,(M^*,W),\ldots,(M,W'),\ldots\}$. Since T is male optimal, we know that M prefers W (his mate in T) to his mate in S. Since S is stable, we know that (M,W) is not a rogue couple, so W must prefer her mate in S (namely, M^*) to M. This means that every other stable pairing must match W up to some other male who she likes at least as much as her mate in T. In other words, T pairs W up with her pessimal man. Since W was arbitrary, we see that T pairs every female with her pessimal man. Therefore, the male optimal pairing is female pessimal. $\square$

All this seems a bit unfair to the women! Are there any lessons to be learnt from this? Make the first move!

The Residency Match

Perhaps the most well-known application of the stable marriage algorithm is the residency match program, which pairs medical school graduates and residency slots (internships) at teaching hospitals. Graduates and hospitals submit their ordered preference lists, and the stable pairing produced by a computer matches students with residency programs.

The road to the residency match program was long and twisted. Medical residency programs were first introduced about a century ago. Since interns offered a source of cheap labor for hospitals, soon the number of residency slots exceeded the number of medical graduates, resulting in fierce competition. Hospitals tried to outdo each other by making their residency offers earlier and earlier. By the mid-40s, offers for residency were being made by the beginning of junior year of medical school, and some hospitals were contemplating even earlier offers — to sophomores! The American Medical Association finally stepped in and prohibited medical schools from releasing student transcripts and reference letters until their senior year. This sparked a new problem, with hospitals now making "short fuse" offers to make sure that if their offer was rejected they could still find alternate interns to fill the slot. Once again the competition between hospitals led to an unacceptable situation, with students being given only a few hours to decide whether they would accept an offer.

Finally, in the early 50s, this unsustainable situation led to the centralized system called the National Residency Matching Program (N.R.M.P.) in which the hospitals ranked the residents and the residents ranked the hospitals. The N.R.M.P. then produced a pairing between the applicants and the hospitals, though at first this pairing was not stable. It was not until 1952 that the N.R.M.P. switched to the Traditional Marriage Algorithm, resulting in a stable pairing. Until recently the algorithm was run with the hospitals doing the proposing, and so the pairings produced were hospital optimal. Only recently were the roles reversed such that the medical students were proposing to the hospitals. In the 1990's, there were other improvements made to the algorithm which the N.R.M.P. used. For example, the pairing takes into account preferences for married students for positions at the same or nearby hospitals.

Unsurprisingly, the Traditional Marriage Algorithm is also (reportedly) used by a large dating agency. But it is apparently also used by Akamai, a large web hosting company. Akamai receives a great many web requests, and needs to direct each web request to one of a pool of Akamai web servers. Web requests are best served by nearby servers, and servers that aren't currently busy are better suited to handle web requests than servers that are. Apparently, Akamai uses the Traditional Marriage Algorithm to match web requests to servers efficiently, where the web requests play the role of the men and the web servers are the girls.

Further reading (optional!)

Though it was in use 10 years earlier, the Traditional Marriage Algorithm was first properly analyzed by Gale and Shapley, in a famous paper dating back to 1962 that still stands as one of the great achievements in the analysis of algorithms. The full reference is:

D. Gale and L.S. Shapley, "College Admissions and the Stability of Marriage," *American Mathematical Monthly* **69** (1962), pp. 9–14.

Stable marriage and its numerous variants remains an active topic of research in computer science. Although it is by now almost twenty years old, the following very readable book covers many of the interesting developments since Gale and Shapley's algorithm:

D. Gusfield and R.W. Irving, *The Stable Marriage Problem: Structure and Algorithms*, MIT Press, 1989.