SWIM 2009 – Matching algorithms

The "marriage theorem" and several variants

The marriage theorem

Example.
The example concerns a group of 8 childhood friends, 4 boys and 4 girls. They are used to always hanging out together, and everyone is always somewhat in love with someone else in the group, but there is nothing serious going on. Their names are [1]

girls: Rebecca, Paula, Johanna and Marie,
boys: Anton, Ivan, Boris and David.

At some point they have to decide on how to pair up for a costume party where people are expected in pairwise matched costumes (e.g. Papageno & Papagena, Liz Taylor & Richard Burton, Charlie Brown & Lucy, Kermit the Frog & Miss Piggy, ...)
Their preference rankings are rather fluid, but at this precise point they are as follows:

A	I	B	D
P	J	J	M
R	M	M	R
M	R	P	P
J	P	R	J

R	P	J	M
B	I	A	A
I	B	D	B
A	A	B	I
D	D	I	D

Let's use these preference rankings to provide a matching, according to the following "deferred acceptance" algorithm.

Deferred acceptance:

- Each boy invites the girls he prefers;
- Girls refuse invitations in they have more than one, in which case they retain the invitation of the boy they like best;
- Boys whose invitation was turned down invite the next girl on their preference list; the other boys reiterate their invitation of the previous round;
- Again, girls refuse invitations in they have more than one, retaining only that of their highest-ranked would-be partner;
- These rounds continue until there are no refusals in a "girl round". At that point all the invitation are accepted, and the matching is made.

Let's see how this works:

[1] For those of you who know and/or are interested in "golden oldie" songs in French: the names (and situation) are a "clin d'oeuil" (a.k.a. a wink) to the 1960s song "Anton, Ivan, Boris et moi" by Marie Laforêt.

1st "boy" round	1st "girl" round	2nd "boy" round	2nd "girl" round	3rd "boy" round	3rd "girl" round
Anton asks Paula Ivan and Boris both ask Johanna David asks Marie	Johanna refuses Ivan	Anton asks Paula Boris asks Johanna Ivan and David both ask Marie	Marie refuses David	Anton asks Paula Boris asks Johanna Ivan asks Marie David asks Rebecca	no refusals $\longrightarrow$ match!

In the end: Anton and Boris end up with their first choice
Ivan and David with their second choices.

Paula, Marie and Johanna end up with their third choice
Rebecca ends up with her fourth choice.

The boys are clearly better "served" by this algorithm than the girls. Although it may seem surprising, given the results, we can nevertheless not identify any boy/girl pair who are *not* matched together and who would each prefer the other one over their current partner.

For instance:
Rebecca likes Boris better than David, but Boris prefers his present partner Johanna to Rebecca.
Rebecca also likes Ivan better than David, but Ivan prefers Marie to Rebecca.
Rebecca likes Anton better than David, but even though Rebecca is his second choice, Anton still prefers his first choice Paula to Rebecca.

In fact, the deferred acceptance algorithm *always* leads to such a *stable matching*:

Definition. Given two (finite) sets $\mathcal{B}$ and $\mathcal{G}$ with an equal number of elements, a matching of $\mathcal{B}$ and $\mathcal{G}$ is a one-to-one correspondence between $\mathcal{B}$ and $\mathcal{G}$, i.e. a function f from $\mathcal{B}$ to $\mathcal{G}$ so that, for any $b_1, b_2 \in \mathcal{B}$,

$$b_1 \neq b_2 \iff f(b_1) \neq f(b_2) \ .$$

To " formalize" the preference ranking tables, we assume the following:

$$\forall b \in \mathcal{B}, \quad \text{we have } r_b : \{1, \dots N\} \longrightarrow \mathcal{G} \quad \text{one-to-one}$$

(with $r_b(1)$ standing for the first-place girl in the preference ranking of b, $r_b(2)$ for the second-place girl , ...)

$$\forall g \in \mathcal{G}, \quad \text{we have } \rho_g : \{1, \dots N\} \longrightarrow \mathcal{B} \quad \text{one-to-one} \ .$$

Equipped with these maps, we call $(\mathcal{B}, \mathcal{G}, r, \rho)$ a *fully ranked marriage game*. [2] We then have the following definition:

Definition. Suppose $(\mathcal{B}, \mathcal{G}, r, \rho)$ is a fully ranked marriage game. A matching f of $\mathcal{B}$ and $\mathcal{G}$ is called *stable* (in this game) if there exists no pair (b, g) in $\mathcal{B} \times \mathcal{G}$ for which

$$f(b) = g' \neq g \ , \quad f^{-1}(g) = b' \neq b \ ,$$

$$\text{where } \ g = r_b(i) \ \ g' = r_b(j) \quad \text{with } i < j \ , \quad \text{and } \ b = \rho_g(k) \ \ b' = \rho_g(\ell) \quad \text{with } k < \ell \ .$$

[2] This particular notation and definition is made here for convenience, and is probably not standard.

A matching in a fully ranked marriage game is thus stable if there is no pair (b, g) that is *not* matched by f, but where each would prefer the other over their f−match.

Using this notation, we observe that if the deferred acceptance algorithm matches $b \in \mathcal{B}$ with $g \in \mathcal{B}$, then we can conclude that

- b must have been refused, in the rounds preceding the end result, by all girls g' for which $r_b^{-1}(g') < r_b^{-1}(g)$;
- g never received an invitation from any boy b' for which $\rho_g^{-1}(b') < \rho_g^{-1}(b)$.

This observation will be crucial in the proof of

Theorem.
The deferred acceptance algorithm described above always leads to a stable matching of $\mathcal{B}$ and $\mathcal{G}$.

Proof: Let's call β the matching function from $\mathcal{B}$ to $\mathcal{G}$ that is the end product of the algorithm. Suppose β were not stable. Then there would be $\widetilde{b}$, $\widetilde{g}$ so that

$$\beta(\widetilde{b}) = g \neq \widetilde{g} = \beta(b) \;\text{ with }\; b \neq b' \;\; \rho_g^{-1}(b) < \rho_g^{-1}(\widetilde{b}) \;\text{ and }\; r_b^{-1}(g) < r_b^{-1}(\widetilde{g}) \; .$$

Because of his preference ranking, b must have invited g before he invited $\widetilde{g}$, since he prefers g. On the other hand, since $\beta(\widetilde{b}) = g$, g never received an invitation from any b' for which $\rho_g^{-1}(b') < \rho_g^{-1}(\widetilde{b})$, implying that she never received an invitation from b. This contradiction shows that our original assumption, that β was not stable, is mistaken, and completes the proof. □

Let's return to our example. Although β is stable, we can surely do "better" from the girl's point of view. Let's see what we get if we let the girls do the inviting:

1st "girl" round	1st "boy" round	2nd "girl" round	2nd "boy" round
Rebecca asks Boris Paula asks Ivan Johanna and Marie both ask Anton	Anton refuses Johanna	Rebecca asks Boris Paula asks Ivan Marie asks Anton Johanna asks David	no refusals $\longrightarrow$ match!

In the end: Rebecca, Paula and Marie end up with their first choice, Johanna with her second choice.

Anton gets his third choice,
Boris, Ivan and David all get their fourth choice.

It clearly matters <u>a lot</u> who does the "inviting" in these deferred acceptance algorithms! Let's label the two versions of the deferred acceptance algorithm described above by who does the inviting; with this convention, we call the first the $\mathcal{B}$-proposes version, the second the $\mathcal{G}$-proposes version of the deferred acceptance algorithm. Then we have the following theorem:

Theorem.
The $\mathcal{B}$-proposes version of the deferred acceptance algorithm is the optimal stable algorithm from the point of view of the members of $\mathcal{B}$, in the sense that there exists no stable matching f of $\mathcal{B}$ and $\mathcal{G}$ such that $f(b) = g'$, $\beta(b) = g$ and $r_b^{-1}(g') < r_b^{-1}(g)$.

(In other words, there is no stable matching in which *any* member of $\mathcal{B}$ would prefer another partner over his $\beta-$match.)

Proof. The crux of the Theorem is that we have to prove there is no *stable* matching f in which b would have another partner than his $\beta-$match. In fact, what we set out to prove is that if there is such an f, then it must be unstable.
So let's suppose that $\beta(b) = g$, $f(b) = g'$, and $r_b^{-1}(g') < r_b^{-1}(g)$.
Since $r_b^{-1}(g') < r_b^{-1}(g)$ and $\beta(b) = g$, there must have been a moment, in the procedure for the $\mathcal{B}$-proposes deferred acceptance algorithm (which we shall call β-algorithm in the remainder of this proof), where b invited g', but she turned him down. This shows that the set $\mathcal{T} = \{b \in \mathcal{B}$; b was turned down by his $f-$ match $\neq \emptyset$.
For each $b"$ in $\mathcal{T}$, we can determine the number $n"$ of the round in which he was turned down by his $f-$match. Pick a $\widetilde{b}$ in $\mathcal{T}$ for which this number, $\widetilde{n}$, is the lowest possible; that is, there is no element in $\mathcal{T}$ who was turned down by his $f-$match in an earlier round. (Note that $\widetilde{b}$ need not be unique; it suffices to pick one, however.)
Consider now $\widehat{g}$, the girl who, in round $\widetilde{n}$, turned down $\widetilde{b}$ in the β-algorithm, but retained, in that round, the invitation of boy $b"$; clearly she must have preferred $b"$ to $\widetilde{b}$.
The $f-$partner of the boy $b"$ cannot be $\widehat{g}$ (since she is already the $f-$match of $\widetilde{b} \neq b"$); let's label her $g^* := f(b")$. The boy $b"$ was not rejected by anyone in round $\widetilde{n}$ (since he was retained then by $\widetilde{g}$), and he wasn't rejected by g^* in an earlier round of the β-algorithm (otherwise $b"$ would be in $\mathcal{T}$, *and* the conditions that defined $\widetilde{b}$ would be violated). Since all girls preferred by $b"$ would have rejected him prior to his inviting $\widehat{g}$, and since g^* was not in the business of rejecting $b"$ in rounds 1 through $\widetilde{n}$ of the $\beta-$algorithm, it follows that $b"$ ranked $\widehat{g}$ higher than g^*.
However, this means that $f(b") = g^*$ and $f(\widetilde{b}) = \widehat{g}$, even though $\widehat{g}$ refers $b"$ to $\widetilde{b}$, and $b"$ prefers $\widehat{g}$ to g^*. There is thus a boy-girl pair, not matched to each other by f, but who prefer each other over their $f-$partners. This means f is not stable. $\square$

All the above applies if $\mathcal{B}$ and $\mathcal{G}$ have equal cardinality (i.e. they have the same number of members), and if we assume that each member has ranked all the members of the other set in their preference ordering (meaning that everyone would rather get linked to even their least favorite member of the other group than remain unattached – something that could be argued is more likely for a less serious matter, like picking costumes for a fancy dress party, than for marriage). There are many variants of the "marriage game" in which these conditions are not satisfied. Let's now look at a few of those.

Short lists

It might not matter so very much when the "marriage game" is played to decide whose costume yours will have to match for a fancy dress party, or who dances with whom, but there are certainly instances of the game (including, one presumes, marriage itself) where the preference lists of each individual member in $\mathcal{B}$ (resp. $\mathcal{G}$) is far shorter than the list of all members in the "matching set" $\mathcal{G}$ (resp. $\mathcal{B}$). That is, each b (resp. g) lists, in order of preference, those members of $\mathcal{G}$ (resp. $\mathcal{B}$) to which he (resp. she) would like to be matched, but if no match can be worked out with anyone on this list, then b (resp. g) would prefer to be "single" (i.e. unmatched) rather than matched to someone not on his (resp. her) list. In this case we say that the preference lists are *short*.

In many cases, it is also unrealistic to assume that $\mathcal{B}$ and $\mathcal{G}$ have exactly the same number of members.

Both generalizations can be dealt with by a slight variation of the algorithm explained above. We shall artificially "expand" and manipulate the preference lists, and (if necessary) the membership of either $\mathcal{B}$ or $\mathcal{G}$ so that we can apply the previous deferred acceptance algorithm, and then prune back the corresponding matching until it provides a true solution of the problem at hand.

The only acceptable matches are those in which both partners have each other on their short lists. On the (short) preference list of each b, we therefore annotate (e.g. write in red) the g_1, g_2, ... on the list for whom b is *not* on the preference lists of g_1, g_2, ... We do the same for the preference lists of the individual g.
Next, we equalize (if necessary) the numbers of elements of $\mathcal{B}$ and $\mathcal{G}$ by adding "dummies" d_1, d_2, ... (also marked in red) to the smaller set until the two sets have equal numbers of elements.
Then we extend each preference list for all the elements of the extended matching sets by adding in red all the members of the (possibly extended) matching set who were not listed yet; in the case of a dummy d_j, we just fabricate a preference list by listing all the members of the other set, writing everyone in red.
Finally, we make each preference list "neater" by pushing all the red names to the bottom of the list, without changing the relative ranking of the other (black, i.e. non-red) names.

We now have two matching sets with an equal number of members, and everyone's preference list (including red names) lists all the members of their matching set. Now apply whichever version of the deferred acceptance algorithm. This gives a complete list of pairings, some with both partners in black, some with one black and one red partner, and some with both partners in red. Now turn all the red ink into invisible ink. Pairs in which one member was black and the other red now become single black entries. In the course of the algorithm, every unmatched person (black entry who is single in the end) in the proposing set must have run through whole his/her preference list and be turned by everyone; an unmatched person from the invitee set did not get an invitation from anyone on her/his preference list.

The theorems of the previous section apply before we erase all the red entries, and it is not hard to check that what remains after removing the red entries is stable, and optimal in the same sense as stated before.

One-to-multiple matchings

In many cultures (but not all) marriage is considered a one-to-one affair. Even in such cultures we can have interesting instances of a "polygamic" or "polyandric" marriage game – for matchings that are definitely not marriages. An example is the selection of summer interns in companies: since many companies hire more than one summer intern, whereas each (full-time) intern ends up working for one company only, this is a one-to-multiple matching.

These too can be handled by massaging the earlier situation. Suppose the set of companies is $\mathcal{H}$, and for each member h of $\mathcal{H}$, we know their *capacity* $c(h)$, i.e. the number of matches wanted by h. Then we just define a new set $\mathcal{H}'$ which contains, for each h in $\mathcal{H}$, as many proxies (or avatars) for h as indicated by $c(h)$; i.e. $\mathcal{H}' = \cup_{h\in\mathcal{H}}\{a_1(h), a_2(h), \dots, a_{c(h)}(h)\}$. Each of h's avatars inherits h's (probably short) preference list; they all thus have the same preference list.

From this point onwards, we can follow the procedure of the previous section. In the end, we will have a matching of $\mathcal{H}'$ and the intern list (with possibly some of the interns unmatched, as well as some of the company avatars $a_i(h)$). We then "pull back" this matching to $\mathcal{H}$, by assigning to each h all the matches for its avatars. Again, this matching is stable, and optimal in the sense we saw before.

One important instance of the deferred acceptance algorithm in this setting is the process through which medical students are assigned to teaching hospitals as interns.

The paper (also posted on here) "The Origins, History and Design of the Resident Match" published in 2003 in Journal of the American Medical Association **289** 7, pp. 909–912, reviews how this process happened historically. The following is excerpted from its abstract:

> In the early 1990s, competition among hospitals for interns and among medical students for good internships led to increasingly early offers of internships to students. By the 1940s, appointments were often made as early as the beginning of the junior year of medical school. Hospitals thus had little information about students' performance, and students frequently had to make a final decision to accept or reject an offer without knowing which other offers might be forthcoming. From 1945 through 1951, efforts were made to enforce a uniform date for accepting offers. However, students were still faced with offers having very short deadlines [...] Hospitals often had to scramble for available students [...] A centralized clearinghouse was thus developed [...]. This evolved into the current matching program, whose algorithm continues to be updated to take account of changing needs of applicants. [...].

The article reviews how several "Cooperative Plans" were defined empirically, in parallel, which used algorithms to define a matching from preference lists of students and hospitals. With mathematical hindsight, some of these algorithms are now known to be unstable; this was discovered in practice, and such algorithms were abandoned quickly. As the paper states,

> The importance of stability has since become clear. [... Clearinghouses using unstable algorithms] all failed and were abandoned after interested applicants and hospitals learned to circumvent [the match]. In contrast, clearinghouses that produced stable outcomes succeeded and remained in use.

One of the stable algorithms, the "Boston Pool", used essentially a deferred acceptance plan. It became the blueprint for the National Residency Matching Program (NRMP), the 50th anniversary of which was celebrated by the JAMA article.

The article also alludes to a change in the algorithm that happened in 1998. The version that was used until then was essentially a "Hospitals propose" deferred acceptance algorithm, which, as we have seen, produces matchings that are optimal from the hospitals' point of view, within the framework of stable matchings. This was not realized at first; there is some evidence that the procedure, in which the applying students continually got "upgraded" (i.e. they algorithm always retained the best offer on hand for them, rejecting it only when an even better offer came along), was believed to be the most fair possible for the *students*. An extensive discussion ensued in the late 90s, which led to changing the algorithm to what is essentially a "Students propose" version of the deferred acceptance algorithm, with which students are happier. (Note that the NRMP is much more complicated than a simple deferred acceptance algorithm, because there are many added subtleties, some of which are described in the paper.)

It may *still* be possible to game the NRMP or similar algorithms: hospitals could advertise a capacity $\mathtt{c(h)}$ that is greater than their true capacity, in the hope of getting a better crew of residents that way. If you google `game theory`, `hospital`, `intern` and `matching`, then you will probably hit some papers that discuss this ....