Exploring Matching Algorithms with Couples

Algorithm Design & Simulations

Hamsa Sridhar
Osbert Bastani

Prof. Al Roth & Prof. Peter Coles

Economics 2056a Final Project
12/12/2011

1 Introduction

Since the early 20th century, new American medical graduates have been first employed as 'residents' at hospitals before achieving full 'doctor' status. Since the two-sided matching of doctors to hospitals is highly competitive, the market began to unravel in the 1950s, as hospitals began making offers and receiving acceptances earlier and earlier in the matching process. In order to resolve this problem, Roth designed the National Resident Matching Program (NRMP), a centralized clearinghouse for the many-to-one matching of American doctors to hospitals [1]. This matching program used the well-known Deferred Acceptance (DA) algorithm, which always results in a stable matching between hospitals and doctors.

Yet, by the 1970s, the increasing number of married couples who participated in the matching began playing a significant factor in the process. Unlike single doctors, couples require two complementary hospital matches that are located suitably close to each other. There was no method in the old NRMP algorithm for couples to express such joint preferences and so, many couples began to seek jobs outside of the matching system. In response, the Roth-Peranson (RP) algorithm was used to re-design the NRMP to produce matches for single doctors as well as couples [2].

Unlike the DA algorithm, the RP algorithm does not always result in a stable match since there may not even exist a stable matching in a market with complementarities. Nevertheless, in three consecutive years (1993-5), the RP algorithm has successfully found a stable match in the medical market [3]. In light of the steadily rising numbers of couples in the NRMP and other markets, it has become increasingly important to understand when and how such stable matches can be found.

In this paper, we describe an alternate algorithm for finding stable matches in markets with couples. This algorithm is proven to weakly out-perform the RP algorithm, and we show numerically that it in fact performs vastly better with increasing numbers of couples. Furthermore, we also show some interesting numerical results that suggest insights about the problem of existence of a stable match.

2 Stable Matchings in Deferred Acceptance (DA)

We consider a market with a set of hospitals H and a set of single doctors S (no couples). Assume that each hospital has a strict preference relation $\succ_h$ over the set of acceptable doctors who have applied to it, and similarly, each doctor has a strict preference relation $\succ_s$ over the set of acceptable hospitals at which they have been interviewed. A matching μ is a function from $H \cup S$ that maps each $h \in H$ to some set $\{s_k\} \in S \cup \phi$ and each $s \in S$ to a single $h \in H \cup \phi$. Since matchings are two-sided, we have the property that $\mu(s) = h$ if and only if $s \in \mu(h)$ for all $h \in H, s \in S$. Furthermore, one is matched to ϕ if one is unmatched in the algorithm.

A stable matching for single applicants is both individually rational (which we obtain for free since applicants and hospitals can only receive matches that they have specified as acceptable over being unmatched), and satisfies the following property: $\nexists\ s \in S, h \in H$such that $s \succ_h s' \in \mu(h)$ and $h \succ_s \mu(s)$, where $s \notin \mu(h)$. In other words, there does not exist any pair (s, h) where both s and h

would prefer being matched to each other than their current match, incentivizing them to deviate from the matching process. Gale and Shapley have shown that such a many-to-one stable match is guaranteed by the doctor-optimal DA algorithm [4]:

Step 1: Each $s \in S$ applies to his first choice hospital. Each hospital rejects its least preferred doctor applicants in excess of capacity and all unacceptable doctors, and 'holds' the others.

In general,

Step k: Each doctor who was rejected in step (k-1) applies to his next highest choice (if no such acceptable choices remain, he is assigned ϕ). Each hospital considers the new applicants alongside the ones that it is 'holding,' and once again rejects its least preferred doctors in excess of capacity and all unacceptable doctors, and 'holds' the remaining.

Stop: When no further applications are made by the doctors, each hospital is matched to the doctors that it is holding.

3 Roth-Peranson Sequential Couples (SC) Algorithm

The Roth-Peranson (RP) algorithm consists of two parts: the sequential couples (SC) algorithm, followed by a second phase which is not publicly available [3]. If SC succeeds, the resulting match is stable and the RP algorithm terminates. However, if SC fails, the RP algorithm proceeds to the second phase, where doctors who are potentially members of blocking pairs are allowed to apply again. The resulting match then may or may not be stable. Given the confidentiality of the second phase, we shall primarily concern ourselves with SC rather than the entire RP algorithm. Thus, if the SC fails to produce a match, then we assume that no stable match was found.

In addition to the set of hospitals H, and single applicants S, we now introduce the set of couples C where each $c = (f, m) \in C$ consists of a pair of applicants. Couples have a strict preference relation $\succ_c$ over the set of pairs of acceptable hospitals at which they have been individually interviewed. Let us define the set D of all applicants, where $d \in D$ if $d \in S$ or $\exists c \in C$ such that $c = (d,*)$ or $c = (*, d)$. A matching μ is a function from $H \cup S \cup C$ that maps each $h \in H$ to some set $\{d_k\} \in D \cup \phi$, and each $s \in S$ to a single $h \in H \cup \phi$, and each $c \in C$ to a pair of hospitals $(h, h') \in H \times H$. Since matchings are two-sided, we have the property that (i) $\mu(s) = h$ if and only if $s \in \mu(h)$ for all $h \in H, s \in S$, and (ii) $\mu(c) = (h, h')$ if and only if $f \in \mu(h), m \in \mu(h')$ for $c = (f, m)$.

In addition to the previous conditions for stability in single-doctor markets, we must also satisfy: $\nexists\, c = (f, m) \in C,\ (h, h') \in H \times H$ such that $(h, h') \succ_c \mu(c)$ and $f \succ_h d \in \mu(h), m \succ_{h'} d' \in \mu(h')$, and $\mu(c) \neq (h, h')$. In other words, there does not exist a couple and any pair of hospitals that would prefer being matched to each other that their current matches. Furthermore, $\nexists\, d \in D$, $c = (f, m) \in C,\ h \in H$ such that $d \succ_h f \in \mu(h)$ and $h \succ_d \mu(d)$ and $\mu(d) \neq h$. In other words, we reiterate that there does not exist any pair (s, h) where both s and h would prefer being matched to each other than their current match, regardless of whether h is currently matched to a single applicant or a member of a couple.

We now state the SC algorithm from [3]:

Step 1: Run DA on all $s \in S$ and tentatively match them. Fix some order π on the couples $c \in C$ and let $B = \phi$.

Step 2: Allow the couples to apply to their most preferred pair of hospitals $(h, h') \in H \times H$ that has not yet rejected them.

a) If no such pair exists, then c is unmatched and we proceed to the next couple.
b) If either h or h' has been previously applied to by a member of any couple $c' \neq c$, then the algorithm is terminated.
c) If $h = h'$, then $c = (f, m)$ is rejected unless $\exists\, d_1, d_2 \in \mu(h)$ such that $f \succ_h d_1$ and $m \succ_h d_2$, and $d_1 \neq d_2$. Any members rejected from h (who are necessarily single applicants by step 2b) are added to B.
d) If $h \neq h'$, then $c = (f, m)$ is rejected unless $\exists\, d_1 \in \mu(h),\ d_2 \in \mu(h')$ such that $f \succ_h d_1$ and $m \succ_{h'} d_2$. Again, any members rejected from h, h' are necessarily single by step 2b, and are added to B.
e) If (h, h') reject c, then we return to the beginning of Step 2.

Step 3: One by one, allow each rejected single doctor $s \in B$ to apply to his most preferred hospital h that has not yet rejected him.

a) If $B = \phi$, then we stop the algorithm and all current matches are assigned.
b) If no acceptable hospital preferences remain for $s \in B$, then s is unmatched and we proceed to the next $s' \in B$.
c) If an acceptable hospital preference h exists but has ever been applied to by some member of a couple $c \in C$, then the algorithm is terminated.
d) If $d \succ_h s$ for all $d \in \mu(h)$, then s is rejected by h and applies to his next preference (return to Step 3).
e) If s is accepted by h, then assign $\mu(s) = h$. Any member rejected from h (who is necessarily single applicants by step 3b) id added to B (return to Step 3).

It has been shown that the SC algorithm terminates in a finite number of steps, and if it is successful (stops at Step 3a), the resulting matching is stable [3].

4 Concerns over the SC Algorithm

There are several steps in the SC algorithm that appear to prematurely terminate the algorithm despite the possible existence of a stable matching:

In Step 2b, the algorithm is terminated if $c = (m, f) \in C$ applies to $(h, h') \in H \times H$, and a member m' of any $c' = (m', f') \in C$ has previously applied to either h or h'. However, if these hospitals can accept m (and possibly f in the case that $h = h'$) without rejecting m', then this algorithm could still result in a stable match and premature termination would result in unnecessary failure.

Similarly in Step 3c, the algorithm is terminated if $s \in B$applies to any h which has previously been applied to by a member m of a couple c. Once again, it may be that h has sufficient capacity to accommodate m and s, or even that h rejects s. Thus, termination seems premature in this case.

Rather, the algorithm should only terminate if any applicant displaces a member of a couple who has already been assigned in Step 2. Let us for now assume that we can modify the SC algorithm to adjust for this inefficiency. We can still do better by noting that the order π in which the couples apply to their most preferred hospital pairs plays a crucial role, as illustrated by the example below.

Example: Hospitals h_1 and h_2 have capacity for 1 doctor each. h_1 has preferences $f_2 \succ_{h_1} f_1$ and h_2 has preferences $m_2 \succ_{h_2} m_1$. If we allow (f_1, m_1) to apply first, they are tentatively assigned (h_1, h_2). The subsequent application of the more preferred candidates (f_2, m_2) would cause the displacement of (f_1, m_1), thereby terminating the algorithm. If we had instead allowed (f_2, m_2) to apply first, they would have been tentatively assigned (h_1, h_2) and subsequently, (f_1, m_1) would have been rejected as weaker applicants and remained unmatched. Thus, the former ordering causes a termination of the algorithm while the latter results in a stable match.

So, if a couple is displaced by another couple, we may wish to change the couple ordering π. With this intuition in mind, we now propose a new Ordered Couples (OC) algorithm in the next section.

5 New Ordered Couples (OC) Algorithm

Informally, the steps are as follows:

Step 1: Run DA on all single applicants. Fix some intelligent ordering π on the couples based on their rankings at their top preferences, and a number of maximum iterations R, where $R \in [1, |C|!]$.

Step 2: One by one, allow each couple $c = (f, m)$ to apply to acceptable pairs of hospitals until they find a pair $(h, h') \in H \times H$ that accepts them, and tentatively assign them that match. Then, h and h' reject any least preferred candidates that they are holding in excess capacity. On the other hand, if c is not accepted by any acceptable hospital pair (h, h'), then c will be unmatched.

a) If no applicants are displaced from h and h' after assigning c, then we proceed to the next couple c' according to the order π in Step 2.
b) If only single applicants are displaced from h and h' after assigning c, we run the DA algorithm again, allowing the rejected singles to apply to their next preferred hospitals.
    A. If running DA for the two rejected singles causes a member of couple c to be rejected, then the algorithm terminates.

B. If running DA for the two rejected singles causes any member of a previously assigned couple c' to be rejected, go to Step 2c.

C. If running DA for the two rejected singles does not displace any members of previously assigned couples, then tentatively assign everyone their current match and proceed to the next couple c' according to the order π in Step 2.

c) If a member of a previously assigned couple c' is rejected from h and h' (or from Step 2.b.A) after assigning c, we pause the algorithm and consider the new permutation π', achieved from π by swapping the positions of c and c'. We then return to the state of the match when all the couples before c' had been assigned, and resume Step 2 under the new permutation π'. If the permutation π' has been tried before, or if the number of permutations that have been tried exceeds R, then the algorithm terminates.

d) If all couples have been assigned, everyone received their current match and the algorithm stops.

The main differences between the OC algorithm and the SC algorithm are that:

(i) the OC algorithm performs DA on the rejected single applicants before the placement of a new couple, allowing us to significantly relax the condition in Step 3c of SC, and

(ii) the OC algorithm allows up to R different orderings of the couples, which can make a crucial difference in the success of the algorithm as mentioned earlier.

Note that if we set $R = 1$, we retrieve a variant of the SC algorithm, and that $R \leq |C|!$ since there are at most $|C|!$ different permutations of C couples.

Furthermore, consider the initial 'intelligent ordering π.' Given some level of preference correlation among hospitals, it is easy to assign each couple an estimated rank order based on the hospitals' rankings of each couple member. For instance, for each $c = (f, m)$, consider the sets of distinct hospitals $\{h_i\}$ and $\{h_j'\}$ that f and m apply to respectively in their pair preferences. Each $h \in \{h_i\}$ has a ranking for f and similarly for m. If we average these rankings, giving more weight to a high preference ranking from a higher-tier hospital, and add up the averages for f and m, we have a measure by which we can compare the 'quality' of the couples in the match. By allowing the couples to pass through the match by order of quality (top-ranked couples go first), we decrease the chances of a couple displacing a previously-assigned couple.

Although we constructed the OC algorithm independently, it has been recently brought to our attention that a very similar algorithm is implemented in [5], although they treat their ordering π on couples somewhat differently.

6 Properties of the OC Algorithm

Theorem: If the SC algorithm succeeds, then the OC algorithm succeeds and results in precisely the same match if the same initial ordering π of couples was used.

Proof: Note that if the SC algorithm succeeds, we know that (i) no two couples ever applied to the same hospital, and (ii) no rejected single applicant (displaced by a couple) ever applied to a hospital which a couple had applied to. Thus, we eliminate all cases of Step 2 in the OC algorithm except Step 2.b.C. – couples are placed sequentially according to π, and the resulting rejected single applicants are continuously placed using DA, never interfering with the previous placement of couples. First, note that the set of single applicants who were initially placed by DA and never rejected afterwards, receive and maintain the same choices in both algorithms. Let us call the set of rejected single applicants A. Between the placement of couples, if $A \neq \phi$, all members of A are either (i) declared unmatched, or (ii) matched to a new hospital h using DA. The matched members may either (ii.a) hold their position until all the couples have been placed, or (ii.b) lose their position to a new couple at a later time, and re-enter the set A. In cases (i) and (ii.a), the result would have been the same had the applicant remained in A until the end of the match, as is the case in the SC algorithm. In case (ii.b), the applicant re-enters A some number of times, but eventually must fall into category (i) or (ii.a) after the last couple is placed. Thus, the applicant is assigned his most-preferred achievable choice at the end (if he was displaced by a couple at some intermediate stage, then his temporary assignment was clearly not achievable), which is exactly the choice he would have been assigned had he remained in A until the end, as in the SC algorithm. Thus, all single applicants are assigned the same matches in both the SC and the OC algorithms. Furthermore, since the couples are placed sequentially according to π and no couple ever displaces another couple, all couples also receive the same matches. Then, the SC and OC algorithms are equivalent if SC succeeds.

Theorem: If the OC algorithm succeeds, the resulting match is stable.

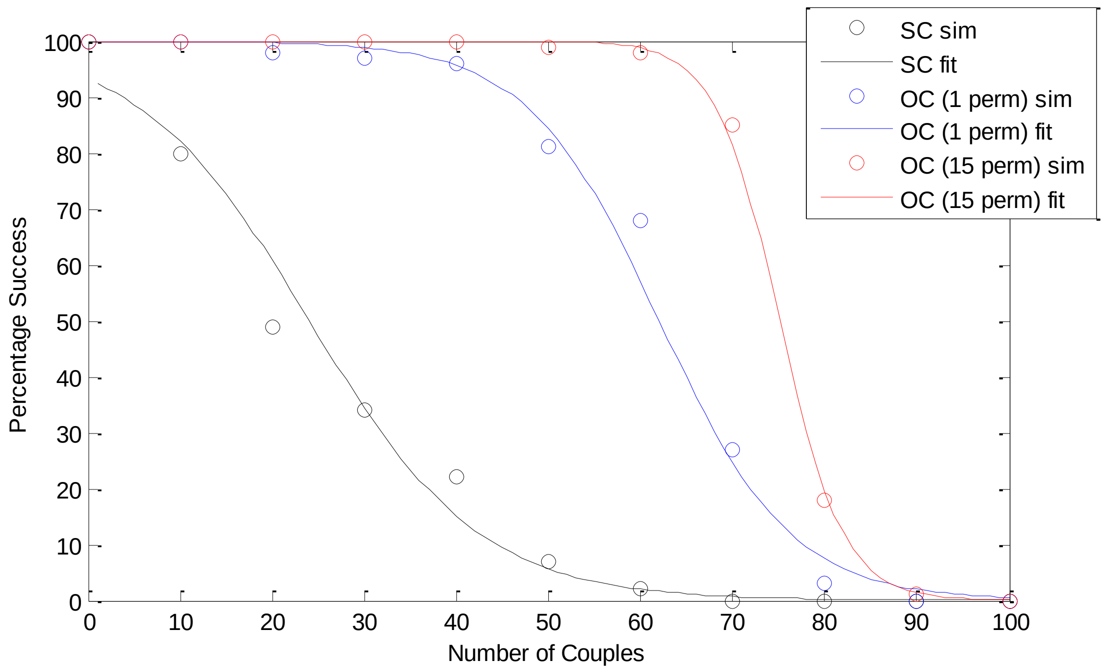
Proof: Say the OC algorithm produces the matching μ. We can take for granted that μ is individually rational since all applicants only apply to acceptable choices in their preference list, and become unmatched if no acceptable preferences remain. All single applicants are placed through DA and thus cannot form blocking pairs by [4]. Note that in the algorithm, each couple applies to their most-preferred hospital pair that has not rejected them yet, and so they cannot form blocking pairs either. Thus, we must have that μ is stable.

7 Simulations

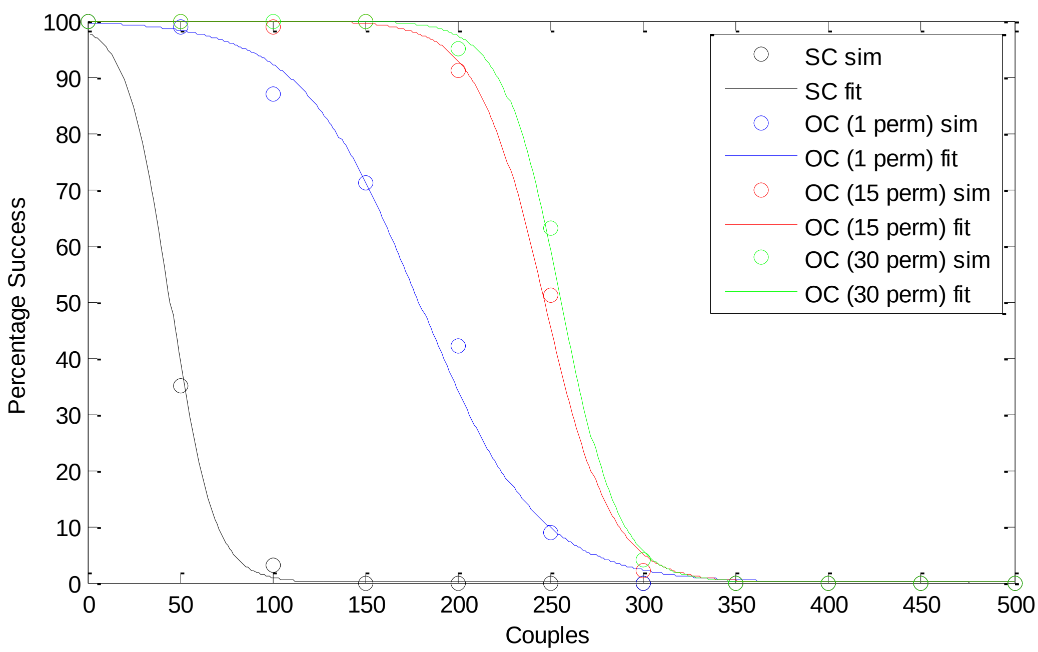
We conducted a series of numerical simulations in MATLAB to better understand the performance of the algorithms, and to examine the probability of finding a stable match as a function of various parameters in the problem. Given a specific set of conditions, in each trial, we randomly generate preference lists and apply the algorithms to determine if a stable match is found. In general, each data point in this section has been averaged over at least 100 trials.

Note that since preferences are random, there is zero correlation between hospital preferences and it does not make sense to assign a 'quality' to any couple. Consequently, every time a previously assigned couple is displaced in the OC algorithm, instead of modifying the previous ordering π, we generate an entirely new random ordering π' in the hopes of achieving faster convergence. Recall, that we fix a cutoff value R for the number of permutations of the couple ordering π.

First, we model a matching for 100 hospitals and 200 applicants, where each hospital has capacity for 3 doctors. Each hospital, applicant, and couple specifies a preference relation among a list of length 20, 10, and 10 preferences respectively. Below, we plot the probability of success for the SC, OC (R = 1), and OC (R = 15) algorithms as a function of the number of couples.

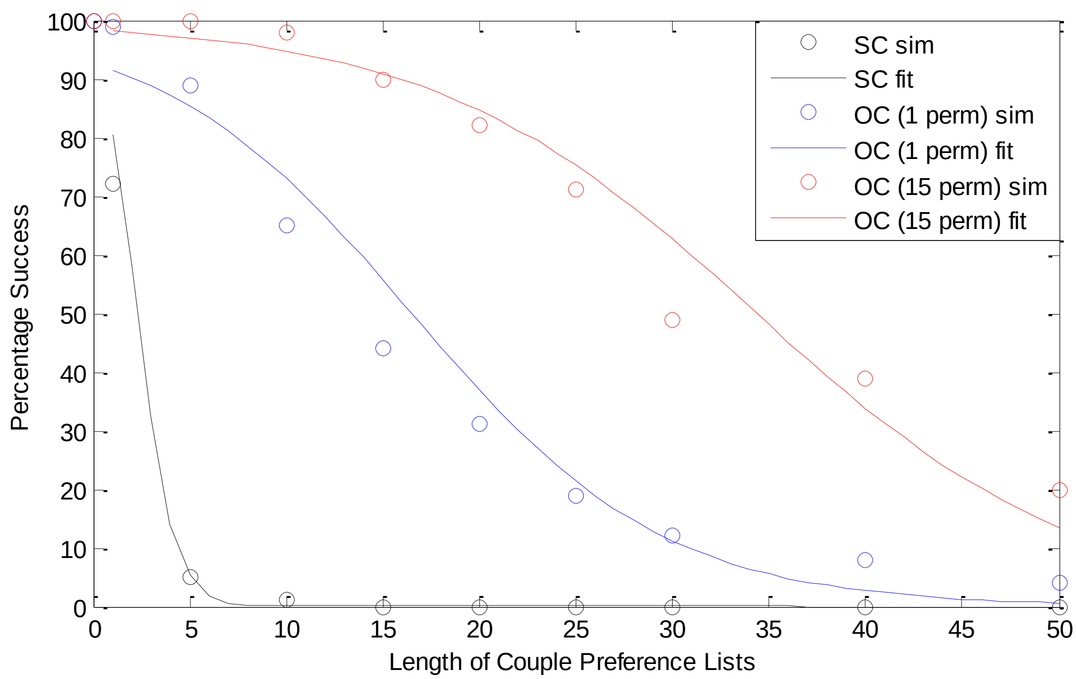


Next, we consider 500 hospitals and 1000 applicants (a five-fold increase) while keeping all other parameters constant.

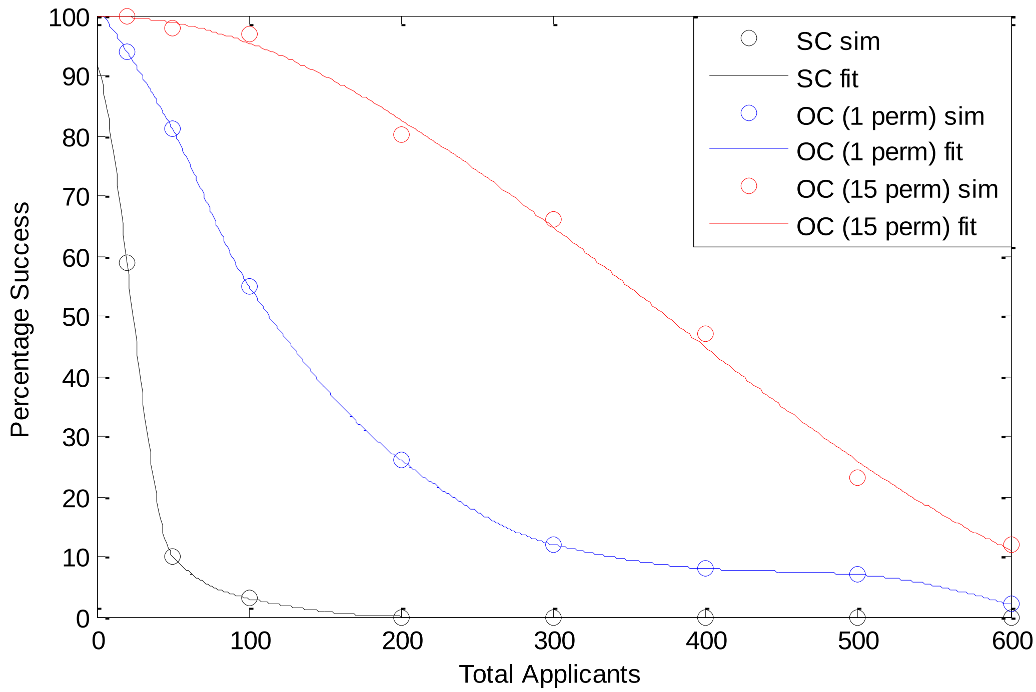


Immediately, we notice that the OC algorithm with even R = 1 greatly outperforms the SC algorithm. With greater values of R, we also find incremental improvements in the probability of finding a stable match. However, the most interesting observation we make is that the graphs seems to follow a logistic function (best fit curves). As the value of R increases, the probability of finding a stable match takes an increasingly sharp dive from 1 to 0. This transition occurs when the number of couples $c^* \approx 75$ in the case of 200 applicants, and when $c^* \approx 250$ in the case of 1000 applicants. It is an interesting problem to consider why such a sharp transition occurs, and to determine an analytical approximation of c^* for a general choice of parameters.

In particular, note that in a small market of size 200, we find a high probability of finding a stable match even when 75% of the applicants are members of couples. In a larger market of size 1000, we only find a high probability of finding a stable match when 50% of the applicants are members of couples, even though the number of hospitals and the total number of positions have grown proportionally. While the authors of [5] simulate that the probability of finding a stable match is constant as a function of the percentage of couples among applicants, they make an important assumption about arbitrarily long preference lists. In fact, as shown in the next plot, the probability of finding a stable match decreases logistically as the allowed length of the couples' preference list increases. In this case, we have 100 hospitals, 200 applicants, and 60 couples:



The negative effect of lengthening couples' preference lists on the probability of finding a stable match is more pronounced in smaller markets than larger ones, since it provides more opportunities for couples to displace each other. Thus, while [5] found that the probability of finding a stable match is constant for any given percentage of couples with arbitrarily long preference lists, we find that the probability of a stable match is higher in smaller markets for any given percentage of couples with a fixed-length preference list. This result is very counter-intuitive as much recent literature has focused on proving that the probability of finding a stable match converges to one in the limit of large markets, rather than small markets. We illustrate this fact in more detail below, where we have plotted the probability of finding a stable match as a function of the number of applicants, holding the percentage of couples fixed at 60%. Remarkably, the curves show the same qualitative behavior as the previous graph which varied the length of couples' preference lists.



8 Conclusions & Future Research

We have presented a new algorithm for two-sided many-to-one matching with couples, which vastly outperforms the current SC algorithm in numerical simulations. We have also found some interesting numerical results about (i) a sharp transition between always finding a stable match to never finding one, as a function of the percentage of couples among applicants, and (ii) how smaller markets can provide an increased probability of finding a stable match with large percentages of couples if the length of couples' preference lists are fixed.

There are many new problems to address as a follow-up of our work. First, we consider the many theoretical implications of the OC algorithm. We hope to address questions such as: (i) whether there can exist a stable match that cannot be achieved through any permutation of couples in the OC algorithm, (ii) what general conditions block the existence of a stable match, and (iii) whether there is an underlying lattice structure for stable matches with couples. Second, we also hope to use our numerical results to: (i) understand the nature of the aforementioned transition in probabilities and (ii) find an analytical approximation for the threshold number of couples c^*, above which the probability of finding a stable match quickly degenerates to zero.

9 References

[1] A. Roth and M. Sotomayor. *Two-Sided Matching: A study in game-theoretic analysis*. Cambridge: Cambridge University Press, 1992.

[2] A. Roth and E. Peranson. The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design. *American Economic Review*, **89**: 748-780, 1999.

[3] F. Kojima, P. Pathak, and A. Roth. Matching with Couples: Stability and Incentives in Large Markets. Working paper, 2010.

[4] D. Gale and L. Shapley. College Admissions and the Stability of Marriage. *American Mathematical Monthly*, **69**: 9-15, 1962.

[5] I. Ashlagi, M. Braverman, and A. Hassidim. Matching with Couples Revisited. Working Paper, 2010.