# Redesign of the National Resident Matching Program

We turn next to think about

- Theory as an input to design
- and why it's not the only input,
- and what some other inputs might be…

To put it another way, we want to think about what makes design difficult, taking as a case study the redesign of the NRMP.

That matching is difficult turns out to be an ancient observation…

# Midrash Rabbah (VaYikra Rabbah)

Translated into English under the editorship of Rabbi Dr. H. Freedman, and Maurice Simon,
Leviticus, Chapters I-XIX translated by Rev. J. Israelstam, Soncino Press, London, 1939
Chapter VIII (TZAV)

A Roman lady asked R. Jose b. Halafta: '**In how many days did the Holy One, blessed be He, create His world**"' He answered: '**In six days**, as it is written, *For in six days the Lord made heaven and earth*, etc.(Ex. XXXI, 17). She asked further: '**And what has He been doing since that time?**' He answered: '**He is joining couples** [proclaiming]: "A's wife [to be] is allotted to A; A's daughter is allotted to B; (So-and-so's wealth is for So-and-so)."'[1] **Said she: 'This is a thing which I, too, am able to do.** See how many male slaves and how many female slaves I have; I can make them consort together all at the same time.' Said he: '**If in your eyes it is an easy task, it is in His eyes as hard a task as the dividing of the Red Sea.**' He then went away and left her. What did she do? She sent for a thousand male slaves and a thousand female slaves, placed them in rows, and said to them: 'Male A shall take to wife female B; C shall take D and so on.' She let them consort together one night. In the morning they came to her; one had a head wounded, another had an eye taken out, another an elbow crushed, another a leg broken; one said 'I do not want this one [as my husband],' another said: 'I do not want this one [as my wife].'



1. M.K. deletes this as irrelevant. But E.J. explains: A's wealth is to be given to B, as a dowry for the former's daughter.

Roth, A. E. and Elliott Peranson, “The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design,” *American Economic Review*, 89, 4, September, 1999, 748-780.

Some NRMP "match variations:” What makes the NRMP different from a simple college admissions model is that it has complications which sometimes cause two positions to be linked to one another, and sometimes cause the number of positions to change.

In the first category of differences are **couples**, who submit rank orders of pairs of programs and must be matched to two positions; and applicants who match to 2nd year positions and have **supplemental lists** which must then be consulted to match them to 1st year positions.

In the second category are requests by residency programs to have an **even or an odd number of matches,** and **reversions of unfilled positions from one program to another.**

These complications matter for two related reasons:

1. They may change the properties of the match; and
2. The clearinghouse algorithm must be designed to accommodate them

Let’s take a look at how we might model couples, for example (keeping in mind that we’ll eventually have to take account of *all* the match variations, not just couples…)

## A More Complex Market: Matching with Couples

This model is the same as the college admissions model, except the set of workers is replaced by a set of applicants that includes individuals and couples.

Denote the set of applicants by $A = A1 \cup C$, where A1 is the set of (single) applicants who seek no more than one position, and C is the set of couples {ai, aj} such that ai is in the set A2 (of husbands) and aj is in the set A3, and the sets of applicants A1, A2, and A3 together make up the entire population of individual applicants, $A' = A1 \cup A2 \cup A3$.

Each couple $c=\{a_i,a_j\}$ in C has preferences over ordered pairs of positions, i.e. an ordered list of elements of FxF. The first element of this list is some $(r_i,r_j)$ in FxF which is the couples' first choice pair of jobs for $a_i$ and $a_j$ respectively, and so forth.

Applicants in the set A1 have preferences over residency programs, and residency programs (firms) have preferences over the individuals in A', just as in the simple model discussed earlier. (That is, firms view the members of a couple as two distinct individuals…)

A matching is a set of pairs in FxA'.

Each single applicant, each couple, and each residency program submits to the centralized clearinghouse a Rank Order List (ROL) that is their stated preference ordering of acceptable alternatives.

A matching $\mu$ is blocked by a single applicant (in the set A1), or by a residency program, if $\mu$ matches that agent to some individual or residency program not on its ROL.

A matching is blocked by an individual couple $(a_i,a_j)$ if they are matched to a pair $(r_i,r_j)$ not on their ROL.

A residency program r and a single applicant a in A1 together block a matching $\mu$ precisely as in the college admissions market, if they are not matched to one another and would both prefer to be.

A couple $c=(a_1,a_2)$ and residency programs $r_1$ and $r_2$ block $\mu$ if the couple prefers $(r_1,r_2)$ to $\mu(c)$, and either $r_1$ and $r_2$ each would prefer to be matched to the corresponding couple member, or if one of them would prefer, and the other already is matched to the corresponding couple member. That is, c and $(r_1,r_2)$ block $\mu$ if

1. $(r_1,r_2) >_c \mu(c)$; and if either
2. $\{(a_1 \notin \mu(r_1)$, and $a_1 >_{r1} a_i$ for some $a_i \in \mu(r_1)$ or $a_1$ is acceptable to $r_1$ and $|\mu(r_1)| < q_1\}$ and either $a_2 \in \mu(r_2)$ or $\{a_2 \notin \mu(r_2)$, $a_2 >_{r2} a_j$ for some $a_j \in \mu(r_2)$ or $a_2$ is acceptable to $r_2$ and $|\mu(r_2)| < q_2\}$

or

3. $a_1 \in \mu(r_1)$ and $\{a_2 \notin \mu(r_2)$, and $a_2 >_{r2} a_j$ for some $a_j \in \mu(r_2)$ or $a_2$ is acceptable to $r_2$ and $|\mu(r_2)| < q_2\}$

A matching is *stable if it is not blocked by any individual or by a an individual and a residency program, or by a couple together with one or two residency programs.*

**Theorem 5.11** (Roth '84): In the college admissions model with couples, the set of stable matchings may be empty.

Proof: by (counter) example.

**Example**--market with one couple and no stable matchings (motivated by Klaus and Klijn, and Nakamura (JET corrigendum 2009 to K&K JET 2005):

Let c=(s1,s2) be a couple, and suppose there is another single student s3, and two hospitals h1 and h2. Suppose that the acceptable matches for each agent, in order of preference, are given by

c: (h1,h2); s3: h1, h2,
h1: s1, s3; h2: s3, s2

Then no individually rational matching μ (i.e. no μ that matches agents only to acceptable mates) is stable. We consider two cases, depending on whether the couple is matched or unmatched.

Case 1: μ(c)=(h1,h2). Then s3 is unmatched, and s/he and h2 can block μ, because h2 prefers s3 to μ (h2)=s2.

Case 2: μ (c)=c (unmatched). If μ (s3)=h1, then (c, h1,h2) blocks μ. If μ (s3)=h2 or μ (s3)=s3 (unmatched), then (s3,h1) blocks μ.

Furthermore, the following example shows that even when the set of stable matchings is non-empty, it may no longer have the nice properties we've come to expect.

Matching with couples (Example of Aldershof and Carducci, '96)
4 hospitals {h1,…h4} each with one position;
2 couples {s1,s2} and {s3,s4}

Preferences:

| h1 | h2 | h3 | h4 | {s1,s2} | {s3,s4} |
|---|---|---|---|---|---|
| S4 | s2 | s2 | s2 | h3h2 | h2h1 |
| S3 | s3 | s4 | s3 | h2h3 | h2h3 |
| | S1 | s1 | | h2h4 | h1h3 |
| | | | | h3h4 | h4h1 |
| | | | | u h3 | h4h3 |
| | | | | u h2 | |
| | | | | u h4 | |
| | | | | h3 u | |
| | | | | h2 u | |

There are exactly two stable matchings: h1,…h4 are either matched to:

| h1 | h2 | h3 | h4 | |
|---|---|---|---|---|
| (s4 | s2 | s1 | s3) | h1, h2, h4, {s1,s2} prefer this |

or to

| | | | | |
|---|---|---|---|---|
| (s4 | s3 | s2 | u) | h3, {s3,s4} prefer this |

So, even when stable matchings exist, there need not be an optimal stable matching for either side, and employment levels may vary.

So we can start to note theorems about simple markets whose conclusions do not carry over to markets with the NRMP match variations (or just with couples).

In a simple matching market:

1. the set of stable matchings is always nonempty

2. the set of stable matchings always contains a "program optimal" stable matching, and an "applicant optimal" stable matching.

3. the same applicants are matched and the same positions are filled at every stable matching.

Similarly, strategic results about simple markets won’t carry over unchanged to the more complex medical market.

Strategic behavior in simple markets (without match variations):

1. In simple markets, *when the applicant proposing algorithm is used*, but not when the hospital proposing algorithm is used, *no applicant can possibly improve his match by submitting an ROL that is different from his true preferences*.

2. In simple markets when the program proposing algorithm is used, every applicant who can do better than to submit his true preferences as his ROL can do so by submitting a truncation of his true preferences.

3. In simple markets, when the program proposing algorithm is used, the only applicants who can do better than to submit their true preferences are those who would have received a different match from the applicant proposing algorithm.

Furthermore, the best such applicants can do is to obtain the applicant optimal match.

# Descriptive Statistics: NRMP

| | 1987 | 1993 | 1994 | 1995 | 1996 |
|---|---|---|---|---|---|
| **APPLICANTS** | | | | | |
| Primary ROL's | **20071** | **20916** | **22353** | **22937** | **24749** |
| Applicants with Supplemental ROL's | **1572** | **2515** | **2312** | **2098** | **2436** |
| Couples | | | | | |
| Applicants who are Coupled | **694** | **854** | **892** | **998** | **1008** |
| **PROGRAMS** | | | | | |
| Active Programs | **3225** | **3677** | **3715** | **3800** | **3830** |
| Active Programs with ROL Returned | 3170 | 3622 | 3662 | 3745 | 3758 |
| Potential Reversions of Unfilled Positions | | | | | |
| Programs Specifying Reversion | **69** | **247** | **276** | **285** | **282** |
| Positions to be Reverted if Unfilled | 225 | 1329 | 1467 | 1291 | 1272 |
| Programs Requesting Even Matching | **4** | **2** | **6** | **7** | **8** |
| Total Quota Before Match | **19973** | **22737** | **22801** | **22806** | **22578** |

Because conclusions about simple markets can have counter examples in the complex medical market, there are points at which we have to rely on computational explorations to see how close an approximation the simple theory provides for the complex market.

Computation proved useful in three places:

1. Computational experiments were used in the algorithm design.
2. Computational explorations of the data from previous years were used to study the effect of different algorithms.
3. Theoretical computation, on simple markets, was used to understand the relation between market complexity and market size.

what class of algorithms we might want to explore?

The **deferred acceptance algorithm** for simple matching models gave us a one pass algorithm; it starts with everyone unmatched, and never cycles – we're probably not going to be able to do that in the complex problem.

What about building a stable matching (if one existed), resolving blocking pairs as we identified them?

E.g. if (m',w') is a blocking pair for a matching μ, a new matching ν can be obtained from μ by *satisfying the blocking pair if m' and w' are matched to one another at ν, their mates at μ (if any) are unmatched at ν, and all other agents are matched to the same mates at ν as they were at μ.*

But even for the simplest models this can cycle, and might not converge to a stable matching (see the discussion of Example 2.4 in Roth and Sotomayor).

In the marriage model there's a way around this problem.

**Theorem 2.33** (Roth and Vande Vate): Let μ be an arbitrary matching for (*M, W,* ***P***). Then there exists a finite sequence of matchings $\mu_1, \mu_2, \dots, \mu_k$, such that $\mu_k$ is stable, and for each $i = 1, \dots, k-1$, there is a blocking pair $(m_i, w_i)$ for $\mu_i$ such that $\mu_{i+1}$ follows from $\mu_i$ by satisfying the pair $(m_i, w_i)$.

Elements of the proof: Let $\mu_1$ be an arbitrary (w.l.o.g individually rational) matching with blocking pair (m1,w1). Let $\mu_2$ be the matching obtained by satisfying the blocking pair, and define the set A(1) = {m1,w1}.

**Inductive assumption**: Let A(q) be a subset of $M \cup W$ such that there are no blocking pairs for $\mu_{q+1}$ contained in A(q), and such that $\mu_{q+1}$ does not match any agent in A(q) to any agent outside of A(q).

Then if $\mu_{q+1}$ isn't stable, there is a blocking pair (m',w') such that at most one of m' and w' is contained in A(q). (If neither of {m',w'} is in A(q), let A(q+1) = A(q)$\cup${m',w'} and let $\mu_{q+2}$ be obtained from $\mu_{q+1}$ by satisfying the blocking pair (m',w').

Otherwise, one of the pair is in A(q), say m' (in the other case the symmetric argument will apply). Let A(q+1) = A(q)$\cup${w'}. Now run the deferred acceptance algorithm, just in the set A(q+1), starting with w' proposing and continuing until a matching is reached with no blocking pairs among the members of A(q+1). The output is $\mu_{q+2}$.

This suggests a new class of algorithms, of which the deferred acceptance algorithm is a special case.

Start with an arbitrary matching $\mu$, and select a subset *A* of agents such that there are no blocking pairs for $\mu$ contained in *A*, and $\mu$ does not match any agent in *A* to any agent not in *A*.

(For example, *A* could be a pair of agents matched under $\mu$, or a single agent, or the set of all men.)

A new player, say woman w, is selected to join *A*. If no man in *A* is part of a blocking pair with woman w, we may simply add her to *A* without changing the matching. Otherwise, select the man m whom woman *w* most prefers among those in A with whom she forms a blocking pair, and form a new matching by satisfying this blocking pair. If there is a woman $w' = \mu(m)$, then she is left unmatched at this new matching, and so there may now be a blocking pair *(w',m')* contained in *A*. If so, choose the blocking pair most preferred by w' to form the next new matching.

The process continues in this way within the set $A \cup \{w\}$, like the deferred acceptance algorithm with women proposing, satisfying the blocking pairs which arise at each step until the process terminates with a matching $\mu_i$ having no blocking pairs within $A_i = S \cup \{w\}$.

The process can now be continued, with the selected set $A_i$ growing at each stage. At each stage, the selected set has no blocking pairs in it for the associated matching $\mu_{I,}$ and so the process converges to a stable matching when $A_k = M \cup W$.

In the deferred acceptance algorithm with men proposing, the initial matching $\mu$ is the one at which all agents are single, and the initial set A is A=W.

In the deferred acceptance algorithm with men proposing, the welfare of the women rises monotonically throughout the algorithm. In this more general class of algorithms there is no parallel, since agents from either side may be introduced into the set A. But the set A itself grows, so the algorithm converges.

So, we'll be looking for an algorithm that accumulates agents not involved in blocking pairs…

**Computational experiments in the algorithm design** (taken from the actual design process of the the applicant-proposing algorithm)

First, a conceptual design was formulated and circulated for comment, based on the family of algorithms explored (for the marriage model) in Roth and VandeVate (1990).

To code this into a working algorithm, a number of choices had to be made, concerning the sequence in which operations would be conducted.

Computational experiments were performed before making sequencing choices.

(Throughout the algorithm design process, progress reports were posted on the web, where they were available to all interested parties. Since designs often have to be adopted my multiple constituencies, the design process might be important…)

- Do sequencing differences cause substantial or predictable changes in the match result (e.g. do applicants or programs selected first do better or worse than their counterparts selected later)?
- Does the sequence of processing affect the likelihood that an algorithm will produce a stable matching?

Experiments to test the effect of sequencing were conducted using data from three NRMP matches: 1993, 1994, and 1995.

The results were that sequencing effects existed, but were unsystematic, and effected on the order of 1 in 10,000 matches.

(In the majority of years and algorithm sequences examined, the match was unaffected by changes in sequencing of algorithm operations, and in the majority of the remaining cases only 2 applicants received different matches.)

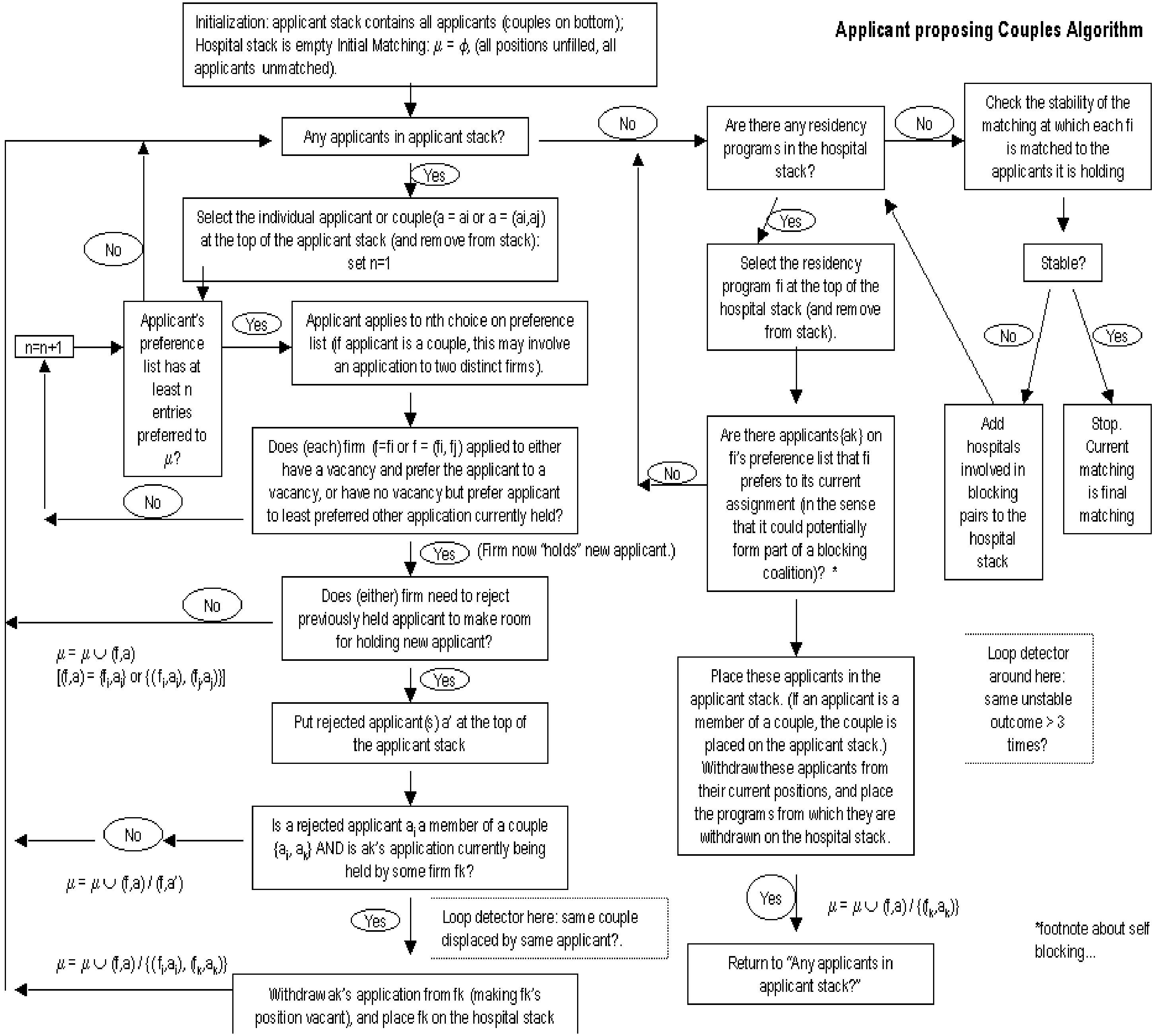
However sequencing decisions did influence the speed of convergence to a stable matching.

Based on the sequencing experiments described above, the following decisions were made pertaining to the design of the applicant proposing algorithm for the NRMP:

1. **All single applicants are admitted to the algorithm for processing before any couples are admitted**.
2. Single applicants are admitted for processing in ascending sequence by applicant code.
3. Couples are admitted for processing in ascending sequence by the lower of the two applicant codes of the couple.

When a program is selected from the program stack for processing, the applicants ranked by the program are processed in ascending order by program rank number (**i.e. in order of the program's preferences**).

## Applicant proposing Couples Algorithm



# Computational Exploration of the Difference Between Program and Applicant Proposing Algorithms

| | 1987 | 1993 | 1994 | 1995 | 1996 |
|---|---|---|---|---|---|
| **APPLICANTS** | | | | | |
| Number of Applicants Affected | 20 | 16 | 20 | 14 | 21 |
| Applicant Proposing Result Preferred | 12 | 16 | 11 | 14 | 12 |
| Program Proposing Result Preferred | 8 | 0 | 9 | 0 | 9 |
| New Matched | 0 | 0 | 0 | 0 | 1 |
| New Unmatched | 1 | 0 | 0 | 0 | 0 |
| **PROGRAMS** | | | | | |
| Number of Programs Affected | 20 | 15 | 23 | 15 | 19 |
| Applicant Proposing Result Preferred | 8 | 0 | 12 | 1 | 10 |
| Program Proposing Result Preferred | 12 | 15 | 11 | 14 | 9 |
| Programs with New Position(s) Filled | 0 | 0 | 2 | 1 | 1 |
| Programs with New Unfilled Positions | 1 | 0 | 2 | 0 | 0 |

If this were a simple market, the small number of applicants whose matching is changed when we switch from hospitals proposing to applicants proposing would imply that there was also little room for strategic behavior when it comes time to state rank order lists.

We can find out if this is also true in the complex market with computational experiments.  It turns out that we don't have to experiment on each individual separately, to put an upper bound on how many individuals could profitably manipulate their preferences.

(For the moment, we treat the submitted preferences as the true preferences—we'll see in a minute why that is justified.)

**Computational experiments to find upper bounds for the scope of strategic behavior**

Truncations *at* the match point (to check if examining truncations is sufficient in the multi-pass algorithm…)

Difference in result for both the program proposing algorithm and the applicant proposing algorithm when *applicant* ROLs truncated *at* the match point:

| 1993 | 1994 | 1995 |
|---|---|---|
| none | 2 applicants improve, same positions filled | 2 applicants improve, same positions filled |

Difference in result for the program proposing algorithm when *program* ROLs truncated *at* the match point:

| 1993 | 1994 | 1995 |
|---|---|---|
| none | none | 2 applicants do worse, same positions filled |

# Difference in result for the applicant proposing algorithm when *program* ROLs truncated *at* the match point:

| 1993 | 1994 | 1995 |
|---|---|---|
| none | 3 applicants do worse, same number of positions filled, but not same positions [3 programs filled one less position, 1 program filled 1 more position, 1 program filled 2 more positions, 1 additional position was reverted from one program to another]. | none |

# Results for Iterative Truncations of Applicant ROL's just *above* the match point

| | 1993 | | | | 1994 | | | |
|---|---|---|---|---|---|---|---|---|
| | Program Proposing Algorithm | | Applicant Proposing | | Program Proposing Algorithm | | Applicant Proposing | |
| | Truncated | Truncated & Improved | Truncated | Truncated & Improved | Truncated | Truncated & Improved | Truncated | Truncated & Improved |
| Run 1 | 17209 | 4546 | 17209 | 4536 | 17725 | 4935 | 17725 | 4934 |
| Run 2 | 4546 | 2093 | 4536 | 2082 | 4935 | 2361 | 4934 | 2359 |
| Run 3 | 2093 | 1036 | 2082 | 1023 | 2361 | 1185 | 2359 | 1183 |
| Run 4 | 1036 | 514 | 1023 | 498 | 1185 | 602 | 1183 | 598 |
| Run 5 | 514 | 258 | 498 | 241 | 602 | 292 | 598 | 287 |
| Run 6 | 258 | 135 | 241 | 116 | 292 | 151 | 287 | 143 |
| Run 7 | 135 | 73 | 116 | 52 | 151 | 75 | 143 | 66 |
| Run 8 | 73 | 48 | 52 | 25 | 75 | 40 | 66 | 31 |
| Run 9 | 48 | 34 | 25 | 12 | 40 | 27 | 31 | 17 |
| Run 10 | 34 | 27 | 12 | 5 | 27 | 18 | 17 | 7 |
| Run 11 | 27 | 24 | 5 | 2 | 18 | 14 | 7 | 3 |
| Run 12 | 24 | 22 | 2 | 0 | 14 | 13 | 3 | 2 |
| Run 13 | 22 | 22 | | | 13 | 13 | 2 | 2 |

The truncation experiments with *applicants*' ROLs yield the following upper bounds for the two algorithms in the years studied.

Upper limit of the number of applicants who could benefit by truncating their lists at one above their original match point:

| | **1987** | **1993** | **1994** | **1995** | **1996** |
|---|---|---|---|---|---|
| Program-Proposing Algorithm | 12 | 22 | 13 | 16 | 11 |
| Applicant-Proposing Algorithm | 0 | 0 | 2 | 2 | 9 |

As expected, more applicants can benefit from list truncation under the program-proposing algorithm than under the applicant-proposing algorithm. Note that the number of applicants who could even potentially benefit from truncating their ROLs under the program-proposing algorithm is in each year almost exactly equal to the number of applicants who received a preferred match under the applicant proposing match (line 2 of Table 2). This suggests that this upper bound is very close to the precise number that would be predicted in the absence of match variations.

The truncation experiments with *programs*' ROLs yield the following upper bounds.

Upper limit of the number of programs that could benefit by truncating their lists at one above the original match point:

| | **1987** | **1993** | **1994** | **1995** | **1996** |
|---|---|---|---|---|---|
| Program-Proposing Algorithm | 15 | 12 | 15 | 23 | 14 |
| Applicant-Proposing Algorithm | 27 | 28 | 27 | 36 | 18 |

As expected, some programs can benefit from list truncation under either algorithm. However, consistently more programs benefit from list truncation under the applicant-proposing algorithm than under the program-proposing algorithm. Note that although the numbers of programs in these upper bounds remain small, they are in many cases about twice as large as the number of programs which received a preferred match at the stable matching produced by the algorithm other than the one being manipulated.

Refined estimate of the upper limit of the number of programs that could improve their results by truncating their own ROL's in 1995 (Based on 50% sample):

| | **Program Proposing Algorithm** | **Applicant Proposing Algorithm** |
|---|---|---|
| Original Results | 23 | 36 |
| Current Estimate (still an upper limit) | 12 | 22 |

Residency programs have another dimension on which they can manipulate; they not only have to report their preferences, but also how many positions they wish to fill. As we've seen in examples of the (simple) college-admissions model, and as in multi-unit auctions, they may potentially benefit from demand reductions.

(And for an impossibility theorem on avoiding capacity manipulation, see Sonmez, Tayfun [1997], "Manipulation via Capacities in Two-Sided Matching Markets," Journal of Economic Theory, 77, 1, November, 197-204.)

Revised Estimate of the Upper Bound of the Number of Programs That Could Improve Their Remaining Matches By Reducing *Quotas*

| | **1987** | **1993** | **1994** | **1995** | **1996** |
|---|---|---|---|---|---|
| Program Proposing Algorithm | 28 | 16 | 32 | 8 | 44 |
| Applicant Proposing Algorithm | 8 | 24 | 16 | 16 | 32 |

This will be worth thinking about again—a small cloud on the horizon—when we consider what temptations may exist for residency programs to hire some of their people early, before the match. If there are such temptations, they may not be counterbalanced by a tendency to do worse in the match, on the contrary, reducing demand may have small spillover benefits in the match for the remaining candidates…

Overall, the striking thing about all these computational results is how small the set of stable outcomes appear to be; i.e. how few applicants or programs are affected by a switch from program proposing to applicant proposing, and how small are the opportunities to misrepresent preferences or capacities.

**But** we don't really understand the structure of the set of stable matchings when there are couples, supplementary lists, and reversion of positions from one program to another. So there's a chance that we're making a big mistake here.

For example, we know that program and applicant optimal stable matchings no longer exist, but we've been studying the set of stable matchings by looking at the outcomes of the program and applicant proposing algorithms. Maybe the set of stable matchings isn't all located between these two matchings when all the match variations are present; maybe the set of stable matchings just appears to be small because we don't know where to look.

Even if our conclusions *are* correct, we'd like to know why. Could it be some spooky interaction between the size of the market and the presence of complications? Or does the core simply get small as the market gets large, even in simple markets?

Two approaches:

- Empirical: examine some simple markets
- Theoretical/computational: explore some artificial simple markets

The Thoracic surgery match is a simple match, with no match variations. It exactly fits the college admissions model; those theorems all apply.

**Descriptive statistics and original Thoracic Surgery match results**

| | **1991** | **1992** | **1993** | **1994** | **1996** |
|---|---|---|---|---|---|
| Applicant ROL's | 127 | 183 | 200 | 197 | 176 |
| Active Programs | 67 | 89 | 91 | 93 | 92 |
| Program ROL's | 62 | 86 | 90 | 93 | 92 |
| Total Quota | 93 | 132 | 141 | 146 | 143 |
| Positions Filled | 79 | 123 | 136 | 140 | 132 |

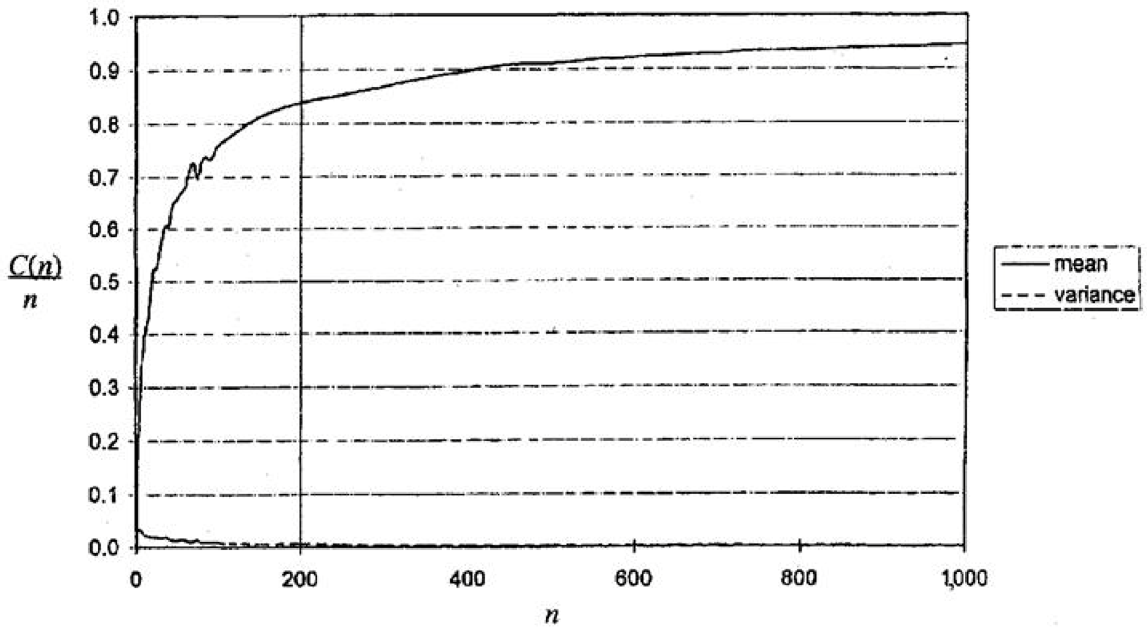
**Difference in Thoracic Surgery results when algorithm changed from program proposing to applicant proposing:**

| 1991 | 1992 | 1993 | 1994 | 1996 |
|---|---|---|---|---|
| none | 2 applicants improve<br>2 programs do worse | 2 applicants improve<br>2 programs do worse | none | none |

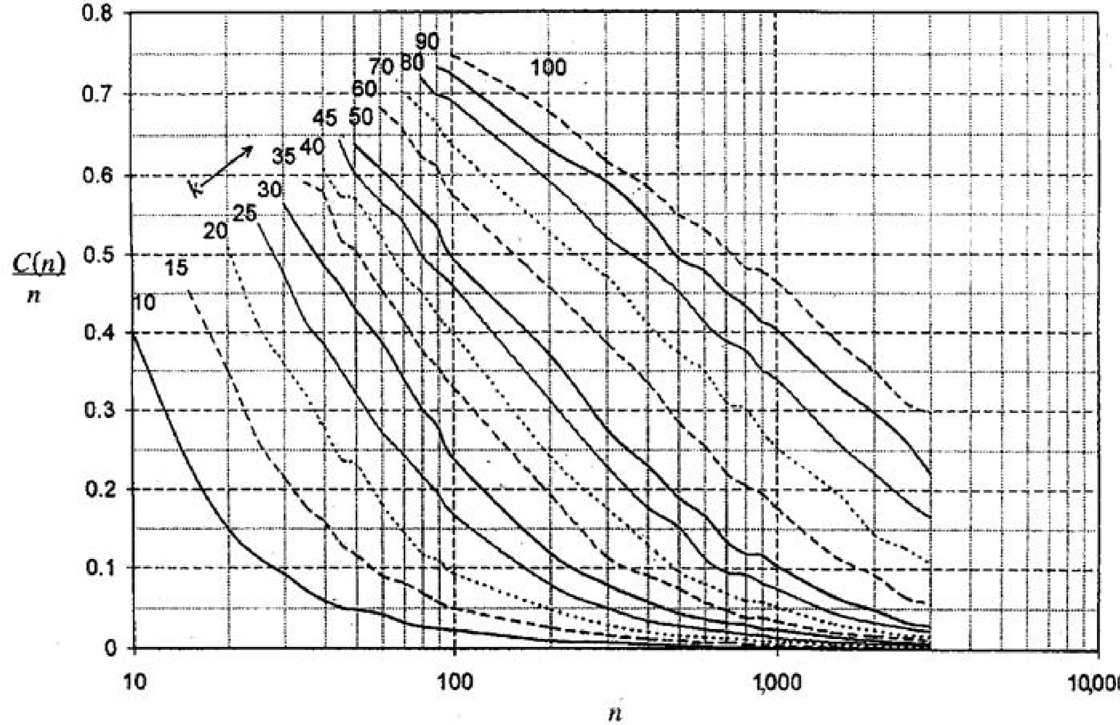
# Theoretical computation on a simple model

Simple model: n firms, n workers, (no couples) uncorrelated preferences, each worker applies to k firms.

C(n) = number of workers matched differently at $\mu F$ and $\mu W$




Large core with k=n: C(n)/n is the proportion of workers who receive different matches at different stable matchings, in a simple market (no couples) with n workers and n firms (each of which employs one worker) when preferences are uncorrelated and each preference list consists of all n agents on the other side of the market. (from Roth and Peranson, 1999) Note that as the market grows large in this way, so does the set of stable matchings, in the sense that for large markets, almost every worker is effected by the choice of stable matching.




Small core of large markets, with k fixed as n grows: C(n)/n is the proportion of workers who receive different matches at different stable matchings, in a simple market with n workers and n firms, when each worker applies to k firms, each firm ranks all workers who apply, and preferences are uncorrelated. (from Roth and Peranson, 1999). Note that for any fixed k, the set of stable matchings grows small as n grows large.

The numerical results show us that C(n)/n gets small as n gets large when k is fixed, (even) for uncorrelated preferences.

And of course, in these simulated markets, we see that the core gets small not because of strategic behavior—these are the true preferences.

This also implies that in large markets it is almost a dominant strategy for every agent to reveal his true preferences—only one in a thousand could profit by strategically mis-stating preferences (if they had full information about all preferences).

**Term paper idea** - follow the developments in finding out whether the following claim is true:

**The set of stable matches (or the core) gets small (in some expected value sense) as the market gets large, if the length of the ROLs doesn't also get large.**

You can start with what we talked about in this class, and then move to recent developments in theory:

- Nicole Immorlica and Mohammad Mahdian, "Marriage, Honesty and Stability," Immorlica, SODA 2005, 53-62.
- Fuhito Kojima and Parag Pathak, "Incentives and Stability in Large Two Sided Matching Markets" American Economic Review, 99(3), 608-627, 2008
- Fuhito Kojima, Parag Pathak, and Alvin E. Roth "Matching with Couples: Stability and Incentives in Large Markets" working paper

**A far more ambitious term paper idea**

Self-blocking couples: A different kind of (modeling) problem is raised by the following partial example:

Let C = {a1, a2} have preferences: (H1, H2), (H2, H3)
Suppose the relevant part of the hospital preferences are
H1: a1, …
H2: a1, a2
H3: a2, …

Consider a (partial) allocation that has C={a1,a2} matched to H2, H3. That is, C gets it's second choice, [(a1,H2), (a2,H3)]
Note that C would prefer the matching at which it was matched to H1,H2, i.e. [(a1,H1), (a2,H2)]
But {C, H1, H2} don't block the original matching because H2 prefers a1 to a2.

But maybe we should think of a modified definition in which {C, H1, H2} does block the original matching, since C can withdraw a2 from H2… As an administrative matter, C wouldn't like to hear that they had gotten their second choice only because they had listed it as acceptable…