Super-Stable Matchings and K-Range Preferences

Quincy Prescott Hughes

May 13, 2022

Advised by Professor William Rosenbaum

Submitted to the
Department of Computer Science
of Amherst College
in partial fulfillment of the requirements
for the degree of
Bachelor of Arts with Honors

Abstract

The Stable Marriage model has been thoroughly explored and many key problems have been found to be NP-hard. Cheng and Rosenbaum [8] have found that some of those problems are fixed-parameter tractable under the parameterized k-range model introduced by Bhatnagar et al. [6]. Under this model, preferences are restricted such that for each man i, all women's rankings of that man differ by no more than $k-1$ (and men's rankings of women are similarly restricted).

In this thesis we explore stable matching with ties and consider super-stable matchings. Super-stable matchings were found in Hu and Garg [14] to have a rotation poset analogous to that of the standard stable matching problem. We provide a correction to the algorithm to find all rotations in an instance, and fully prove the algorithm's correctness by showing termination. By redefining the range of an agent to account for worst-case possible resolution of ties, we also find that under k-range the same fixed-parameter tractability result holds for super-stable matchings.

We implement in Python the algorithm to find all super-stable matchings as well as a method for generating random k-range preference based on iterative adjacent swaps. Using this codebase, we measure key characteristics of the rotation poset given random k-range instances and compare these instances to instances following a tiered model. We find that under k-range the number of rotations and stable matchings is not monotonic with the value of k. There are many nuances in this result which we cannot fully explain. We also find the tiered model to be a poor approximation of k-range, at least when both classes of agents have tiers of the same size. When instances are constructed with k-range preferences on one side and uniformly random preferences on the other, the tiered model seems a reasonable substitute for k-range. We conclude with open questions on the structure of stable matchings with k-range restrictions.

Contents

Chapter 1

Introduction and Background

1.1 Preliminaries and Related Work

In the classical stable marriage problem, there are two sets of agents which are to be assigned to a member of the opposing set. In the simplest case, the cardinality of the sets is equal, and assignments are one to one only. Each agent has ordinal preference rankings over their potential matches. If every agent in the opposing class is included in those rankings, we say the preference lists are complete. In this paper we refer to two types of agents, men and women.

A **matching** is an assignment of men and women to one another. A matching is said to be **stable** if there is no man and woman who would prefer to be with each other than with their current partners. We say that this man and woman form a blocking pair in this matching.

Gale and Shapley [12] show in their seminal paper that in the classical stable marriage problem, there is at least one possible stable matching in the instance. They also give an algorithm to find a stable matching. In their algorithm, one class of agents proposes and the other is proposed to. When we run it with men proposing, the result is the **man-optimal stable matching**. Every man is at least as well off in the man-optimal stable matching as in any other. Any free man is selected, and they propose to their top choice woman, deleting her from his preference list. This woman rejects the engagement if she is currently engaged to a man she prefers, and accepts it otherwise. That man is no longer considered free, and the man who was previously engaged to the woman (if he exists) is now considered free. This algorithm will always terminate and always find a stable matching.

The stable matchings possible in a given instance form an interesting and useful structure. In both the standard SM problem, the stable matchings can be represented as a distributive lattice. Irving and Leather show that this lattice can be transformed to give a compact, algorithmically convenient representation of the stable matchings in an instance. We consider a partial order of stable matchings, with one matching dominating another if and only if it is weakly preferred by all men. At the top of this partial order is the man optimal matching μ_0, and at the bottom is the woman optimal μ_z. The famous Gale-Shapley algorithm can find each of those in $O(n^2)$ time, with n being the number of men/the number of women. The maximum possible number of stable matchings is $2^{\Theta(n)}$ [18, 20]. We can consider the differences between these matchings in terms of **rotations**. A rotation is a transformation from one matching to another, defined by a set of man-woman pairs such that if a matching μ is transformed into μ' by rotation

$$\rho = ((m_1, w_1), (m_2, w_2), \dots (m_l, w_l))$$

$((m_1, w_1), (m_2, w_2), \dots (m_l, w_l)$ are matched in μ, and $(m_1, w_2), (m_2, w_3), \dots (m_l, w_1)$ are matched in μ', and all men and women not in the rotation have the same partners. A rotation is essentially a set of men and women who are "cycled" one partner over to their next stable partner.

We say a rotation is **eliminated** in a matching when we execute the transformation defined by that rotation, and we say a rotation is **exposed** when that transformation can be eliminated (the men are with their partners that define the start of the rotation) and when the new matching that results is stable. This new matching can be expressed as $\mu' = \mu \setminus \rho$. The rotations in an instance can be arranged in a partially ordered set, or poset, ordered by prerequisites. That is, if a rotation ρ is not exposed in any matching unless ρ' is eliminated, then ρ' precedes ρ in the poset. There is a one-to-one correspondence between the stable matchings of an instance and the downsets/antichains of the corresponding **rotation poset** $R(I)$.

In addition to the stable marriage problem which matches men and women one to one, we also have many to one matchings (commonly referred to as the college application problem) and many to many matchings. There's also the stable roommate (SR) problem, in which there is no distinction between men and women and any agent can be matched with any other agent. This paper focuses purely on the one-to-one stable marriage problem, although some results likely can be generalized to other variations.

We also consider **stable matching with ties** (SMT) as well as **stable matching with ties and incomplete lists** (SMTI). When ties are allowed, an agent's preference rankings need not be strict; an agent can be indifferent between two (or more) possible matches. As Manlove [26] first outlines, this gives rise to multiple possible definitions of stability. A matching is **weakly-stable** if there is no man and woman who would strictly prefer to be with each other than with their current partners. It is **strongly-stable** if there is no man and woman such that the man strictly prefers the woman to his current partner and the woman weakly prefers the man to his current partner, or vice versa. Finally, a matching is **super-stable** if there is no man and woman that weakly prefer each other to their current partner. Depending on the context, different definitions of stability could make sense. Weak stability may be sufficient in a case where matching has costs. In the case of the labor market, leaving a workplace for another means significant upheaval for the worker, and significant training costs for the employer. It seems unlikely that such a swap would occur without both agents strictly preferring each other to their current partners due to the costs associated with transfer. Super-stable matchings may seem less relevant, however they are useful in the case where ties in preference lists actually represent unknowns rather than indifference. In this case, a matching being super-stable means that it is stable no matter the true preferences underlying uncertainty.

In SMTI, preference lists can be incomplete. This means that any woman not on a man's list can never be matched with that man, and vice versa. Incomplete preferences can represent two distinct possibilities. First, they could represent partners that are unacceptable to the point where an agent would rather be unmatched than matched with someone off his list. Alternatively, they could represent partners who cannot be matched because of the matching mechanism. In the case of college applications, a student can't attend colleges he did not apply to, so any such colleges would not be on his list, even if he would prefer to be matched with them than unmatched. With SMI, Irving [15] finds that agents can be partitioned into two sets, with one set containing all agents that are matched in no stable matchings, and the other set containing agents that are matched in every stable matching. As a consequence of this, all stable matchings are of the same size. Manlove[26] extends this result to strongly-stable and super-stable matchings in SMTI as well.

Irving introduced an algorithm to find a super-stable matching in an instance if one exists in $O(m)$ time [15]. Irving [15] and Manlove [26] gave an $O(m^2)$ to find a strongly-stable matching if one exists. Spieker [29] proved that super-stable matchings, similar to stable matchings in the case without indifference, form a distributive lattice, and Manlove[26] proved the same property for strongly-stable matchings. Kunsyz et al. [22] characterize the set of all strongly-stable matchings with a rotation poset analogous to that of the classical stable marriage problem, and Kunsyz [21] also gives a polyhedral description of that set and proves that it is integral. Hu and Garg [14] characterize the set of all super-stable matchings with a rotation poset analogous as well.

Unlike super-stable or strongly stable matchings, it is not the case that all weakly-stable matchings in an instance have the same size. Finding a maximum size weakly-stable matching was shown to be NP-hard by Iwama et al. [17]. The best approximation factor given ties on both sides is 3/2. Lam and Plaxton [23] have found the best approximation factor when ties are only on one side to be $1 + 1/e$.

Ashlagi et al. [3] studied the distribution of outcomes for man-optimal stable matching in a model with correlated tiered preferences. In their model, agents have some objective "public score", and preference lists are built top to bottom sampling without replacement from a distribution proportional to these public scores. They look at the average ranks that result from the Gale-Shapley algorithm.

Bhatnagar et al. [6] introduce a few models of parametrically restricted preference models. One such model is the **k-range model**. In this model, there is an objective ranking for each set of agents, and no agent can deviate too far from this objective ranking. This model is motivated by examples such as colleges ranking prospective applicants. They may be use objective metrics like SAT scores as a guide, but have some flexibility within that objective ranking. Unlike the model of Ashlagi et al. [3] not all combinations of preference lists are possible.

We define the **minrank** and **maxrank** of some man m in an instance I with men M and women W as

$$\text{minrank}(m) = \min_{w \in W} P_w(m), \text{maxrank}(m) = \max_{w \in W} P_w(m) \tag{1.1}$$

With $P_w(m)$ denoting m's place in w's list. The minranks and maxranks of women are defined analogously. The range of I is

$$Range(I) = \max_{a \in M \cup W} (\text{maxrank}(a) - \text{minrank}(a)) + 1 \tag{1.2}$$

We say I has k-range preferences if $Range(I) \leq k$. The **range** of an instance is a measure of how similar preferences are. $k = 1$ implies that all agents share the same master preference lists, and $k = n$ is unrestricted.

Cheng and Rosenbaum [8] show that with these restricted preferences, a path decomposition of limited pathwidth can be found and many NP-hard problems in stable matching are fixed-parameter tractable in k. In particular, counting and sampling stable matchings, finding median stable matchings, and finding sex-equal and balanced stable matchings, are all shown to be tractable in their companion paper [7].

1.1.1 Example 1

As a very simple example to illustrate the distinction between the different notions of stability, consider the following simple SMTI instance with three men and three women.

- $m_0 : w_0, w_1$
- $m_1 : w_1, w_2$
- $m_2 : w_2, w_1, w_0$
- $w_0 : m_2, m_0$
- $w_1 : m_0, m_2, m_1$
- $w_2 : m_1, m_2$

This instance admits the following stable matchings:

- $\mu_0 = ((m_0, w0), (m_1, w_1), (m_2, w_2))$
- $\mu_1 = ((m_0, w_0), (m_1, w_2), (m_2, w_1))$
- $\mu_2 = ((m_0, w_1), (m_1, w_2), (m_2, w_0))$

And has the following rotations:

- $\rho_0 = ((m_1, w_1), (m_2, w_2))$
- $\rho_1 = ((m_0, w_0), (m_1, w_2))$

And the following rotation poset:

- $\rho_0 \longrightarrow_2 \rho_1$

Note that this is a classic example of a type one dependency of one rotation on another. ρ_1 can only be elminated when ρ_0 has moved m_1 to his partner in ρ_1. Consider what happens if we make m_2 indifferent between w_0 and w_1. μ_0 is a super-stable matching (and recall super-stability implies strong stability which implies weak stability) as all the men strictly prefer their new partners to their old. μ_1 is only weakly-stable; m_2 is indifferent between his current partner w_1 and w_0, and w_0 strictly prefers m_2 to her current partner m_0. μ_2 is strongly-stable.

We can also consider what happens if w_1 is indifferent between m_0 and m_2, and if both indifferences are included. The following tables summarize the stability of the matchings with these three different cases of indifference. Note that in the case of no indifference, any stable matchings are clearly super-stable as well. Also note that all matchings are weakly-stable in all cases. A matching can be considered weakly-stable if there is some resolution of ties in the matching that is stable, and these instances were derived by adding indifference to an SMI instance that has those stable matchings.

	Super-stable	Strongly-stable	Weakly-stable
μ_0	✓	✓	✓
μ_1	✗	✗	✓
μ_2	✓	✓	✓

Table 1.1: $w_0 \sim_{m2} w_1$

	Super-stable	Strongly-stable	Weakly-stable
μ_0	✓	✓	✓
μ_1	✓	✓	✓
μ_2	✗	✗	✓

Table 1.2: $m_0 \sim_{w_1} m_2$

	Super-stable	Strongly-stable	Weakly-stable
μ_0	✓	✓	✓
μ_1	✗	✗	✓
μ_2	✗	✓	✓

Table 1.3: $w_0 \sim_{m_2} w_1$ and $m_0 \sim_{w_1} m_2$

1.2 Our Contributions

This paper primarily draws upon the work of Rosenbaum and Cheng [8] and Hu and Garg [14].

I expand Rosenbaum and Cheng's [8] work on characterizing the rotation poset of k-range instances to include super-stable matchings. With some changes in definition, the same results hold, and the super-stable matching poset is similarly restricted in size in SMI instances as the stable matching poset is in SM instances.

I add to Hu and Garg's [14] work on characterizing the set of all stable matchings by showing that Algorithm 2 in their paper is presented with an error. Correcting that error leads to no changes in the runtime of the algorithm. I also find that the proof of correctness for the algorithm is insufficient. I provide a proof of the missing condition for termination of the algorithm in this paper. Yet to be done is a brief walkthrough of the algorithm using Example 1 as the instance.

I also constructed a Python codebase for working with the stable marriage problem. This code base implements Manlove's [25]'s algorithm SUPER2 to find the man-optimal/woman-optimal super stable matching for SMTI instances, as well as Hu and Garg's [14] algorithms to find the set of all stable matchings and construct and display the rotation poset. It can be used to run simulations and derive properties of the rotation poset for instances with given size or other properties.

With this codebase we run a number of simulations and look at properties of the super-stable matching poset. Specifically, we analyze the number of super-stable matchings, the number of rotations, the size of the maximum chain, the size of the maximum antichain, and the pathwidth of the poset. The level of indifference has no effect on the high-level trends, which suggests that the structure of the super-stable matching poset in general is very similar to that of the stable matching poset. We compare the tiered model to the k-range model and find it to be a poor approximation when applied to both sides, but when applied to only one side (with the other having uniform preferences) the results are reasonably similar.

Chapter 2

Finding the Super-Stable Matching Rotation Poset

2.1 Updating the Algorithm for Finding all Rotations in a Super-Stable Matching Problem

Hu and Garg's [14] algorithm for finding all the rotations as written has two errors. The original algorithm can be found in Appendix 1. In this section we will show that the algorithm does not always terminate due to two issues and provide solutions. We also update the proof of the algorithm's correctness.

In this section we use the notation $w_i \geq_{m_j} w_k$ to denote that man j weakly prefers woman i to woman k. Strong preference is denoted similarly with $>$, and preference of women among men is defined analogously as well.

The general intuition behind the algorithm still holds. We maintain two graphs G_d and G_c with G_d containing edges that form cycles with no outgoing edges when rotations are exposed, and G_c containing the candidate edges for the new matching when the exposed rotation is eliminated. We also maintain E', the list of edges yet to be considered. The algorithm starts by considering the man-optimal matching, and moves towards to woman-optimal by exposing and eliminating rotations one by one by adding edges according to certain conditions from E' into G_d and G_c and checking for cycles inG_d.

2.1.1 Removing dominated edges from E' every iteration

The first issue is that edges are not removed from E' when the woman is moved above them in a rotation, blocking further rotations from being found. What follows is an example of this occuring, and then the update to the algorithm necessary to fix this issue.

This section uses the following lemma from Hu and Garg [14].

Lemma 2.1 *[14] Let M be a super-stable matching in G. For any successor N of M such that N is also a super-stable matching in G and each $(m, w) \in M$, any edge (m, w') such that $w' \geq_m w$ or (m', w) such that $m >_w m'$ cannot block N.*

Consider the basic, no indifference version of Example 1 run through the algorithm. Recall that there are two rotations,

$$\rho_0 = ((m_1, w_1), (m_2, w_2))$$

$$\rho_1 = ((m_0, w_0), (m_1, w_2))$$

with $\rho_0 \prec \rho_1$.

Lemma 2.2 *The edge (m_2, w_1) is present in E' before the loop.*

(m_2, w_1) is present in E. Since (m_2, w_1) is not an edge in M, it is not in G_d, therefore it is in E' on initialization in line 7. It also does not get removed from E' in line 9, as w_1 prefers m_2 to her current partner, m_1 and m_2 prefers his current partner w_2 to w_1.

Lemma 2.3 *The edge (m_2, w_1) never gets removed from E' in the loop.*

There are two places in which the edge (m_2, w_1) can be removed from E'. The first is inside the if statement on line 13. The condition will never pass, as the outdegree of the connected component containing m_2 will never be 0. This is because the edge (m_2, w_1) is outgoing towards w_1. By construction of the problem, there is no other edge connecting the component (m_0, w_0, m_1, w_1) with the component (m_2, w_2, m_3, w_3). Therefore, w_1 can't be a part of the strongly connected component containing m_2, and therefore the outdegree of the strongly connected component containing m_2 is at least 1. Therefore, the if condition will never be passed.

The other place is the for loop in line 20. For a woman to be multiply engaged, she must have ties on her preference list, else her lowest ranked edge would simply dominate the others, and the others would either not be added to E_c or they would be removed from E_c when the lowest ranked edge is added. Therefore with this example, as there are no ties in her preference list, no edges are removed.

In neither place the edge (m_2, w_1) can be deleted from E'.

Theorem 2.4 *The second rotation ρ_1 is never found, and the algorithm never terminates.*

For the second rotation to ever be found by the algorithm, there must be a strongly connected component (m_2, w_2, m_3, w_3) with no outgoing edges in G_d. This requires that an outgoing edge from m_2 to w_3 gets added to G_d. This edge can't start in G_d, as it is not a part of the original matching M_0. It must be added to G_d in line 12. However, line 12 adds m's top choices in E' to G_d, and as we showed above, the edge (m_2, w_1) is in E'. Since m_2 prefers w_1 to w_3, (m_2, w_3) is never added to G_d, and the second rotation can never be found. Since the algorithm only terminates when $M = M_z$, the algorithm will never terminate as the second rotation is never eliminated and M can never be transformed to M_z. QED

We propose the algorithm is fixed when lines 9 and 10 are swapped. That is, we move line 9 into the repeat loop, deleting edges from E' every iteration of the loop.

Theorem 2.5 *Moving line 9 into the loop starting at line 10 will not prevent the algorithm from finding all rotations.*

By 2.1, we have that no edge deleted in line 9 can ever block a successor matching to the current M. Since the remaining matchings are successors of M (else they would have been eliminated already by the algorithm), then this change will not delete any edges that are part of or block a subsequent stable matching. The latter is proven by 2.1. The former is proven by observing that successor matchings make men worse off and women better off, and deleted edges are either to better partners for men or worse partners for women. So clearly deleted edges can't be part of any successor matchings either. With edges deleted neither blocking nor taking part in any rotations, their deletion won't prevent the algorithm from identifying the rotations in an instance. QED

2.1.2 Preventing adding dominated edges to G_c

The second problem is that edges are added to G_c when they should not be. This issue occurs nondeterministically, as it depends on the order in which men are selected on line 11.

To illustrate this problem, we run an example instance through the algorithm. For sake of clarity, the graphs G_c and G_d are included. Consider the following instance, where square brackets denote indifference:

- $m_0 : w_0, [w_1, w_2]$
- $m_1 : w_2, w_0, w_1$
- $m_2 : w_0, w_1, w_2$
- $w_0 : m_1, m_0, m_2$
- $w_1 : m_0, m_1, m_2$
- $w_2 : m_2, m_0, m_1$

This instance admits the following stable matchings:

- $\mu_0 = ((m_0, w_0), (m_1, w_2), (m_2, w_1))$
- $\mu_1 = ((m_0, w_1), (m_1, w_0), (m_2, w_2))$

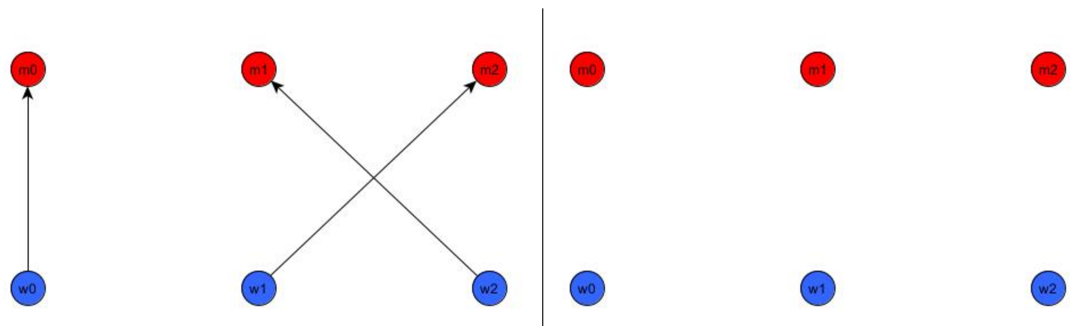
And has the following rotation:

- $\rho_0 = ((m_0, w_0), (m_2, w_1), (m_1, w_2))$

First, in the pre-loop setup of lines 1-9, from this instance we have that E' consists of the following edges:

- (m_0, w_2)
- (m_0, w_1)
- (m_1, w_0)
- (m_2, w_2)

G_c and G_d are given in figure 2.1 to illustrate the example.



(a) G_d (b) G_c

Figure 2.1: Graphs before first iteration of loop

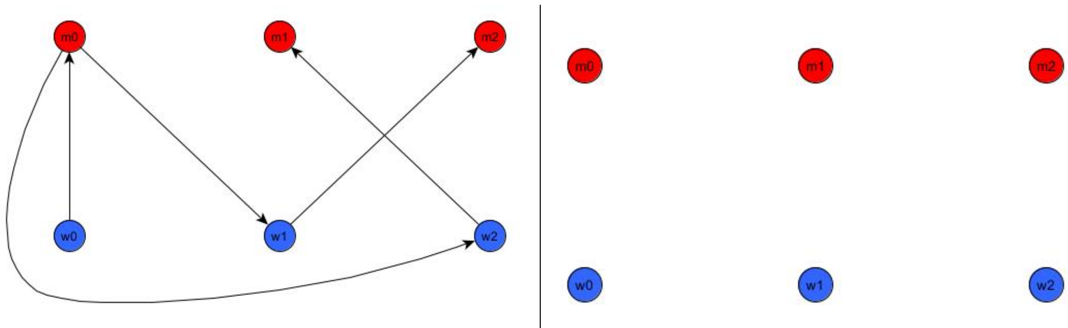
From here, we run the loop from lines 11-20 three times. We assume m_0 is chosen, then m_1, then m_2. The results of each iteration can be seen in figures 2.2, 2.3 and 2.4. From the figures, it's only on the third iteration that the man chosen (m_2) is in a strongly connected component with outgoing edges, and that's the only iteration in which an edge is added to G_c. All 4 of the edges in E' are added to G_d in these iterations.

At this point, we continue running lines 11-20 as men exist with degree 0 in G_c. We assume m_0 is chosen first, then m_1. (m_0, w_1) and (m_1, w_0) are added to G_c.

As we can see in figure 2.5, a second candidate edge (m_0, w_2) is added to G_c incident to w_2. It doesn't dominate (m_3, w_2), so both edges remain.

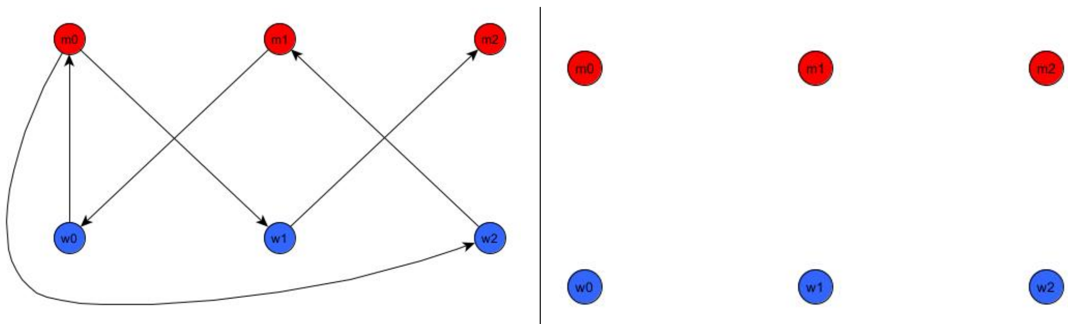
In figure 2.6, each man has an edge in G_c, so the algorithm continues to line 21 and on.

Lemma 2.6 *No edges in G_c are deleted in 21-23.*



(a) G_d (b) G_c

Figure 2.2: Graphs after the first iteration of 11-20, assuming m chosen is m_0



(a) G_d (b) G_c

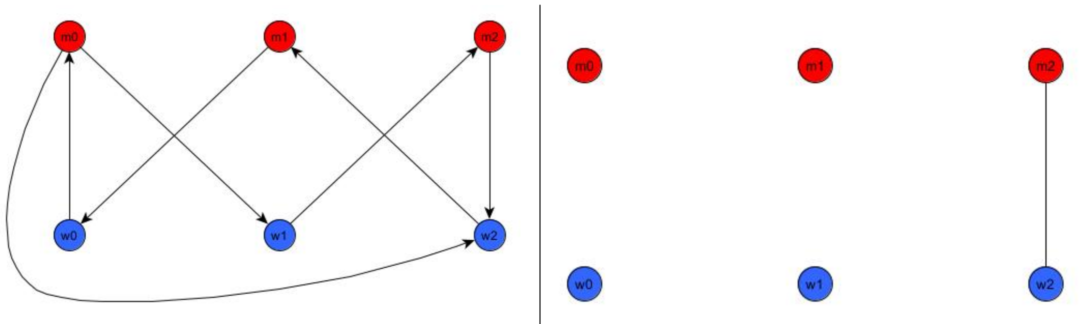
Figure 2.3: Graphs after the second iteration of 11-20, assuming m chosen is m_1

This must be true since no woman is indifferent between multiple edges in G_c or E' (in fact there's no indifference among women at all). This requirement for the definition of "multiple engaged" is made explicit in Lemma 13 in Hu and Garg [14]. QED

This clearly is the result when line 25 is reached for the first time. From here, we can show:

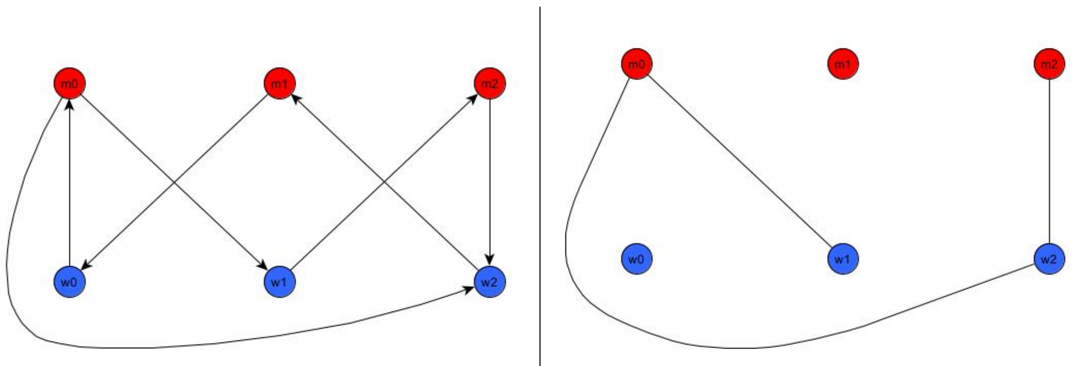
Theorem 2.7 *With this instance, the rotation ρ_0 is never found and the algorithm doesn't terminate.*

ρ_0 includes all men and all women, and so long as w_2 has two edges in G_c, G_c is not a perfect matching on the rotation subgraph and the while loop in 25-33 will not execute. And since all men have edges in G_c, lines 16-17 will never be reached as the while loop in 11-20 will not execute either. Finally, the only remaining place where edges in G_c can be changed is lines 21-23, but by



(a) G_d (b) G_c

Figure 2.4: Graphs after the third iteration of 11-20, assuming m chosen is m_2



(a) G_d (b) G_c

Figure 2.5: Graphs after m_0 is chosen for the second time

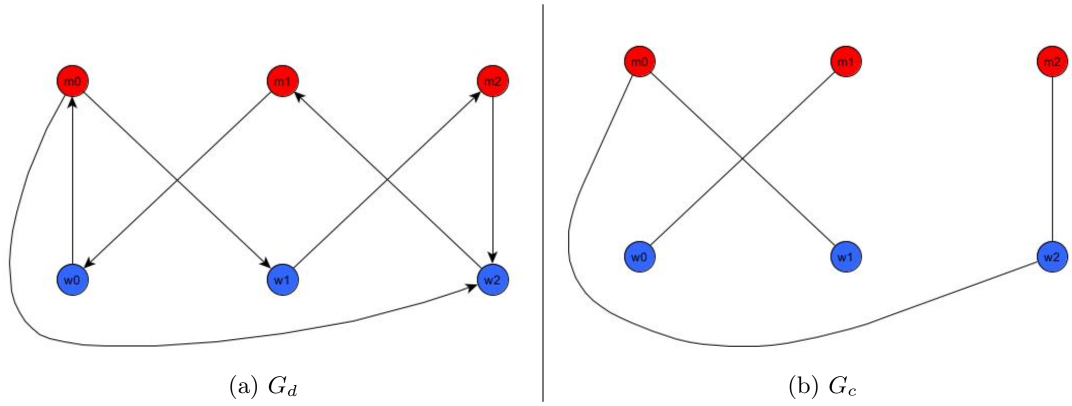
lemma 2.6, no edges will be deleted there. Therefore, the edges in G_c will be unchanged and the algorithm will never terminate. QED

Note that if edges were chosen in a different order, this problem would not occur. We leave it to the reader to show that if m_2 was chosen before m_0 in lines 11-20 and (m_0, w_2) is added to G_c before (m_2, w_2), the algorithm would find the rotation and terminate.

The adjustment to the algorithm to fix this problem is simple: replace lines 14-15 with

- "Add every edge $(m, w) \in E_m$ such that $m >_w M(w)$ and $M(w) >_m w$ and (m, w) is not dominated by any edge (m', w) in E_c to E_c"

Algorithm 1 is the full algorithm with all changes made. Note that any references to this algorithm in the following sections use the line numbers defined here, not the line numbers from



(a) G_d (b) G_c

Figure 2.6: Final state of graphs

the original algorithm in Hu and Garg [14].

2.2 Proving the Correctness of the Algorithm for Finding all Rotations in a Super-Stable Matching Problem

With both corrections implemented, we prove that the algorithm terminates. This section makes use of the following lemmas from Hu and Garg [14], with line numbers updated, and a lemma from Manlove [26]. It's trivial to show that our corrections to the algorithm do not impact them.

Lemma 2.8 *[14] No edge deleted in line 16 can belong to any super-stable matching N dominated by M.*

Lemma 2.9 *[14] No edge deleted in line 20 can belong to any super-stable matching N dominated by M.*

Lemma 2.10 *[26] Let M,N be two super-stable matchings in a given super-stable matching instance. Suppose that, for any agent p,* $(p, q) \in M$ *and* $(p, q) \in N$*, where p is indifferent between q and q', then* $q = q'$ *.*

We also borrow the following terminology from Hu and Garg [14]: a strongly connected component S in a directed graph G is a subgraph that is strongly connected, meaning there is a path

let M_0 be the (unique) man-optimal super-stable matching of G.
let M_z be the (unique) woman-optimal super-stable matching of G.
$M \leftarrow M_0$
let M' contain edge $(m, M(m))$ for each man m such that $M(m) =_m M_z(m)$
let E_d contain all edges of M.
let G_d be the directed graph (V, E_d) such that each edge $(m, w) \in E_d \cap M$ is directed from w to m and every other edge (m, w) is directed from m to w.
$E' \leftarrow E/E_d$
let $E_c = M'$ and $G_c = (V, E_c)$
repeat
  for each $(m, w) \in M$ remove from E' each edge (m', w) such that $m' <_w m$ and each edge (m, w') such that $w' \geq_m w$
  while $(\exists m \in A) deg_{G_c}(m) = 0$ **do**
    add the set E_m of top choices of m from E' to E_d
    if $outdeg(S(m)) = 0$ **then**
      Add every edge $(m, w) \in E_m$ such that $m >_w M(w)$ and $M(w) >_m w$ and (m, w) is not strictly dominated by any edge (m', w) in E_c to E_c
      for each edge m, w of E_c that becomes strictly dominated by some added edge (m', w) remove it from G_c
      remove E_m from E'
    end if
  end while
  for each $m \in A$ such that $outdeg(S(m)) = 0$ **do**
    delete all lowest ranked edges in $E_c \cup E'$ incident to any $w \in S$ such that w is multiply engaged
  end for
  while $\exists S) outdeg(S) = 0$ and E_c is a perfect matching on S **do**
    $M \leftarrow (E_c \cap S) \cup (M \setminus S)$
    $M_i \leftarrow M$
    output M_i
    $i \leftarrow i + 1$
    update G_c and G_d: $E_c \cap S$ contains only edges $(m, M(m))$ such that $M(m) =_m M_z(m)$; an edge $(m, w) \in S$ stays in G_d only if $w = M(m)$ and $rank_w(m) = rank_{m_z}(m)$
  end while
until $(\forall m \in A) rank_M(m) = rank_{M_z}(m)$

Algorithm 1: Updated algorithm for computing a maximal sequence of super-stable matchings from any node to any other node in either direction. An edge (m, w) is outgoing from a component S if $m \in S$ and $w \notin S$. We say a strongly connected component S has outdegree 0 in G if there are no edges outgoing from S in G. We also use $S(m)$ to denote the strongly connected component containing node m.

Consider an SMTI instance I. In this section we aim to prove the following theorem:

Theorem 2.11 *If $M_i \neq M_z$, the algorithm always outputs a matching.*

Assume for sake of contradiction that the algorithm runs forever without finding a rotation. In this case, after some clearly finite number of iterations, each iteration will be identical, as the top choices of all m meeting the conditions of line 22 will not change without E' having edges removed, and E' has finite edges to remove and no way to have edges added. The following proof considers the state of the algorithm in one of these hypothetical iterations. We will show that in this iteration if a rotation exists, it will be found.

Call that rotation $\rho = ((m_i, w_i), (m_{i+1}, w_{i+1}), ...(m_k, w_k))$. Call the (not necessarily unique) matching in which ρ is exposed μ, and let $\mu' = \mu \setminus \rho$.

Lemma 2.12 *If an edge between a man m_i in ρ and his partner w_{i+1} in μ' exists in E', it exists in G_d as well.*

Assume for the sake of contradiction that w_{i+1} is not one of m_i's top choices in E'. Either his (not necessarily unique) top choice (m_i, w_j) is outside ρ or within it. Note that because (m_i, w_j) is in E', it must be the case that it wasn't removed in line 10, so m_i prefers his current partner to w_j and w_j weakly prefers m_i to her current partner. If it's outside ρ, then this pairing blocks μ', as w_j weakly prefers m_i to her current partner (since the edge is in E') and m_i prefers w_j to any of his partners in edges in E', including w_{i+1}. So if (m_i, w_j) is m_i's top choice in E', w_j must be in rotation ρ. Call w_j's next partner in the rotation m_{j-1}.

There are three cases to consider.

First, consider if w_j is indifferent between m_{j-1} and m_i. In this case, (m_i, w_j) blocks μ', since m_i prefers w_j to his next partner w_{i+1} and w_j is indifferent between her current partner and m_i.

Second, we have if w_j prefers m_i to m_{j-1}. (m_i, w_j) still blocks μ' for the same reason as above.

Third, we have the most complicated case: if w_j prefers m_{j-1} to m_i. In this case, we claim that ρ cannot be a rotation in the instance, and instead there are two separate rotations that ultimately lead to the same matching μ'. The intuition behind this case is illustrated by Example 1 above.

Assume the rotation $\rho = ((m_i, w_i), (m_{i+1}, w_{i+1}), \dots (m_j, w_j), (m_{j+1}, w_{j+1})...(m_k, w_k))$.

We claim that $\mu \setminus \rho'$ with $\rho' = ((m_i, w_i), (m_j, w_j), (m_{j+1}, w_{j+1})...)$ is a super-stable matching. Note that ρ' is equal to ρ with the pairs between i and j removed. First note that all men and women in ρ' have the same partners in $\mu \setminus \rho$ as in $\mu \setminus \rho'$ except m_i and w_j. The agents left out of ρ' are matched with the same partners as in μ as well. Therefore, since μ and $\mu \setminus \rho$ are known to be super-stable, if there is any blocking edge in $\mu \setminus \rho'$, at least one of its nodes must be to an agent in $\rho \setminus \rho'$. Consider m_{i+1} and w_{i+1} (an element of $\rho \setminus \rho'$). Since $\mu \setminus \rho'$ is dominated by μ, all women prefer their partners in $\mu \setminus \rho'$ to their partners in μ. Therefore, there's no w_k such that (m_{i+1}, w_k) is blocking in $\mu \setminus \rho'$ but not in μ. If there were an m_k such that (m_k, w_{i+1}) is blocking $\mu \setminus \rho'$, that m_k would have to be in ρ', as all other men have the same partners in μ as in $\mu \setminus \rho'$. This implies that m_k weakly prefers w_{i+1} to his partner in $\mu \setminus \rho'$, w_{k+1}. However, w_{k+1} is also his partner in $\mu \setminus \rho$. Since $\mu \setminus \rho$ is super-stable and (m_k, w_{i+1}) doesn't block that matching, it also can't block $\mu \setminus \rho'$ since m_k must prefer w_{k+1} to w_{i+1}. The same argument holds for all pairings in $\rho \setminus \rho'$.

Since we have a rotation ρ' in the instance, ρ cannot exist as well since two rotations cannot share a pair. This contradicts our assumption.

Therefore we have that w_j, the woman m_i prefers to w_{i+1}, cannot be in the rotation ρ. Since we also showed that w_j cannot be outside the rotation, w_j cannot exist. Therefore, w_{i+1} must be one of m_i's top choices in E'.

Also note that there can be no edge blocking or removing (m_i, w_{i+1}) in lines 14 or 15. Such an edge would also be in G_d outgoing from m_i, and since edges are added to G_d when they are the man's top choice and we showed that w_{i+1} is one of m_i's top choices in E', no such edge can dominate (m_i, w_{i+1}) and prevent it from being added to G_d. And no such edge be in G_c from previous rounds, as G_c is reset to only contain edges present in the woman-optimal matching, and if m_i is in a rotation then he has a different partner in M than in M_z.

So (m_i, w_{i+1}) will have been added in line 14 to G_d and won't be removed in line 15. QED

Lemma 2.13 *If an edge between a man m_i in ρ and his partner in μ' does not exist in E', it exists in G_d.*

w_{i+1} prefers this edge (m_i, w_{i+1}) to the edges from past matchings, as rotations move women to strictly better partners. And since rotations move men to strictly worse, m_i prefers w_i to w_{i+1}. So it can't have been removed in line 10. Nor can it have been removed in line 20, by lemma 13 in Hu and Garg (no edge deleted there is part of a super-stable matching). So the edge must have been removed from E' in line 16, implying that at some point m was in a strongly connected component with outdegree 0 that included the edge (m_i, w_{i+1}). This also implies that line 12 added this edge to G_d. For this edge to have been removed from G_d, either m or w must have been involved in a rotation, as only line 27 removes edges from G_d. Call that rotation ρ'.

Note that when the algorithm found rotation ρ' and removed (m_i, w_{i+1}) from E', $S(m)$ had outdegree 0. Since (m_i, w_{i+1}) was in G_d, it must be that w_{i+1} was a part of the same rotation. It follows that ρ moved m_i to w_{i+1}. This is a contradiction, as by assumption a rotation not yet found moves m_i to w_{i+1}. So the edge can't have ever been removed from G_d after it was added. QED

Lemma 2.14 *Each man m_i in ρ has an edge (m_i, w_{i+1}) in G_d to his partner in the next super-stable matching.*

Either the edge exists in E' or it doesn't. In either case, by 2.13 or by 2.12, we have that the edge exists in G_d. QED

Lemma 2.15 *The nodes in ρ form a strongly connected component with outdegree 0.*

Note that G_d is initialized such that every woman has an edge to her current partner, and whenever a rotation is eliminated, any edges in G_d between partners in the new matching are preserved. This means that at any point in the algorithm, each woman has an edge in G_d to her current partner. Since edges are from woman to man only if the two are partnered in the current matching, each woman only has one outgoing edge, and that edge is to her current partner. This means the nodes in the rotation form a strongly connected component, when combined with 2.14.

For the strongly connected component to either contain nodes outside the rotation or to have outdegree greater than 0, there must be an edge from a man m_i in the rotation to a woman w_k outside of it. This outside woman w_k must be one of man m_i's top choices to have been added to G_d, tied with m_i's next partner w_{i+1}. This edge would be blocking the new matching, as w_k prefers m_i to her current partner (else the edge would have been removed in line 9) and m_i is indifferent between w_k and w_{i+1}, and we have a contradiction.

Therefore, with no outgoing edges from men or women, the nodes in the rotation form a strongly connected component of outdegree 0. QED

Lemma 2.16 *Call the component containing the nodes in ρ S. Let a be the number of men in S. There are no fewer than a edges in E_c incident to any edges in S.*

For G_c to have fewer than a edges within the component, there must be a man in S with no edge in G_c. But since the outdegree of his connected component is 0, the algorithm must run lines 13-16 with (m_i, w_{i+1}) in E_m. Since ρ moves m_i to w_{i+1}, it must be that m_i is strongly preferred by w_{i+1} to her current partner m_{i+1} by lemma 1, and m_i strongly prefers his current partner to w_{i+1} (if he didn't, the current matching would be blocked by (m_i, w_i)). Therefore this edge would be added to G_c. And by 2.8 and 2.9, this edge is not removed from G_c. QED

Lemma 2.17 *G_c must be a perfect matching on S.*

G_c contains edges between men and women only. Each woman can only have one edge in G_c, since having more than one would imply multiple engagement, and those edges would be deleted in lines 20-22. From 2.16, G_c has at least a edges. So the only way for G_c to not be a perfect matching on S, there must be edges connecting men inside S to women outside. But no edge not in G_d can be in G_c, as edges in G_c are either between current partners in μ, or are added in line 14 (which means they were added to G_d in line 12). And we have that S is a strongly connected component with no outgoing edges, so this is impossible. Therefore, since G_c has no fewer than a edges, G_c must have exactly a edges and be a perfect matching on the component. QED

By 2.17 and 2.15, we see that the condition on line 22 of the algorithm is satisfied, and the matching will be found. We have proven 2.11.

2.3 Runtime

Theorem 2.18 *The runtime of the algorithm is still O(mn) after the adjustments is made, with m being the number of edges and n the number of nodes.*

The second adjustment adding to line 14 to when edges are added to G_c clearly has no effect on the runtime.

The other adjustment to the algorithm moving line 9 to line 10 removing edges from E' every iteration weakly increases the running time. However, it is still bounded by $O(mn)$. There are two things to consider here: first, we have the cost of deletions of edges from E', and second we have the cost of checking for when deletions need occur. Edges can never be added back to E', and so at most m edges are deleted. So the amortized cost of deletions is clearly $O(m)$.

When considering checking for deletions, this can be done efficiently using the fact that the lists are ordered. That is, we maintain a pointer on each agent's list, and consider anything above/below that pointer to be deleted. Updating this pointer takes constant time per agent, so this update is $O(n)$ for each iteration, as it must be done for all agents.

Now we must consider how many iterations occur in the outer loop.

Lemma 2.19 *In each iteration of the outer loop, at least one edge is removed from G_c, or at least one edge is removed from E'.*

We prove this by showing that if this is not the case, the algorithm cannot terminate.

With no edges removed from E', no new edges can be added to G_c, as that is done in line 14 only, and edges would have been removed in line 16 if line 14 is executed. So G_c cannot be changed over the course of this iteration.

In addition, no strongly connected components with outdegree 0 can be created in G_d. Edges are added to G_d in line 14, and if those new edges lead to a strongly connected component with outdegree 0, once again line 16 would lead to edges removed from E'. Nor can edges be added to an existing strongly connected component with outdegree 0 for the same reason.

With no changes in G_c (since by assumption edges aren't removed from there) and no changes in the strongly connected components with outdegree 0 in G_d, looking at line 22 we see that any rotation to be found now will have been found in the previous iteration. So no iteration beyond the first can find a rotation if no edge is removed from G_c or removed from E'. If we consider the first iteration, it is plain to see that G_d is initialized with no strongly connected components with outdegree 0.

Furthermore, no rotations will be found in the following iteration. With no changes in G_c or E', there will be no new additions in the following iteration as the same men will pass the condition in line 11 and they will have the same top choices in line 12. No rotation can be found for the same reason as above. It follows by induction that the algorithm will never terminate. Since we proved in 2.11 that the algorithm terminates, this is a contradiction.

Since edges can never be added to E', at most m edges can ever be removed from it. And since edges are removed from E' when they are added to G_c, it follows that at most m edges can be removed from G_c. Therefore, by 2.19, at most $2m$ iterations of the loop can occur.

It follows that the the total runtime of the algorithm within the loop is $O(2mn + m) = O(mn)$. 2.18 has been proven. QED

Chapter 3

Super-Stable Matchings and K-Range

This section uses the following lemma from Manlove [26]:

Lemma 3.1 *[26] Let M,N be two super-stable matchings in a given super-stable matching instance. Suppose that, for any agent $p, (p,q) \in M$ and $(p,q') \in N$, where p is indifferent between q and q' , then $q = q'$.*

3.1 Definitions

We borrow the definition of the **pathwidth** of a rotation poset as the pathwidth of its corresponding Hasse Diagram from Cheng and Rosenbaum [8]

Consider some instance of SMT called I. We define the **lowrank** of a woman w on man m's list $\text{lowrank}_m(w)$ to be the lowest possible ranking she can have if any ties are resolved in her favor if possible. In other words,

$$\text{lowrank}_m(w) = 1 + |\{w' \in W | w' >_m w\}|$$

We define the **highrank** of w on m's list $\text{highrank}_m(w)$ to be the highest possible ranking a woman can have if any ties are resolved against her favor if possible.

$$\text{highrank}_m(w) = 1 + |\{w' \in W | w' \geq_m w\}|$$

We redefine the **minrank** function to return the lowest lowrank woman w has on all mens' lists.

$$\text{minrank}(w) = \min_{m \in M}(\text{lowrank}_m(w))$$

And the **maxrank** function to be the highest highrank woman w has on all mens' lists.

$$\text{maxrank}(w) = \max_{m \in M}(\text{highrank}_m(w))$$

We define all four functions analogously for men.

The **range** of I is defined as

$$range(I) = \max_{a \in M \cup W}(\text{maxrank}(a) - \text{minrank}(a)) + 1$$

Instance I is **k-range** if $\text{Range}(I) \leq k$. We denote the family of instances with n men and women and a range of k as $\text{Range}(k, n)$. With this new definition of minrank and maxrank, we can now extend k-range restrictions to stable matching with indifference.

3.2 Proving bounds on the super-stable matching poset

We showed above that a rotation poset can be found for the super-stable matching problem in $O(mn)$ time, with m being the number of edges and n the number of vertices.

The minranks and maxranks for each man and woman can be found by scanning all the preference lists; therefore k and a minrank function can be found in $O(n^2)$ time.

Theorem 3.2 *$minrank(w) \leq lowrank_m(w) \leq minrank(w) + k - 1$ and $minrank(m) \leq lowrank_w(m) \leq minrank(m) + k - 1$*

3.2 can be shown essentially by the definition of the minrank function and of k.

Lemma 3.3 *Let $I \in Range(k, n)$. Then $\forall i \in [n], i \leq |\{w \in W | minrank(w) \leq i\}| \leq i + k - 1$, and $i \leq |\{m \in M | minrank(m) \leq i\}| \leq i + k - 1$.*

Consider the set of all women with minrank below some i: $\{w \in W | f(w) \leq i\}$. From 3.2, for any man m, each woman in this set must be assigned a rank (or set of ranks) below or equal to $i+k-1$. No more than $i+k-1$ women can be assigned ranks fulfilling this condition on m's list.

Suppose $|\{w \in W | \text{minrank}(w) \leq i\}| < i$. Then, $|\{w \in W | f(w) > i\}| \geq n-i+1$. For a man m, $\text{lowrank}_m(w) > i$ for all women such that $\text{minrank}(w) > i$. At least $n-i+1$ women must have their lowest possible rank be higher than i. There are only $n-i$ ranks from $i+1...n$, and so there is a contradiction. Therefore the bounds proposed must hold (the second set can be shown by a symmetric argument).

Lemma 3.4 *Let $I \in Range(k,n)$. Then for any stable pair (m, w) of I, $|minrank(m) - minrank(w)| \leq 2k-1$.*

We will show that $\text{minrank}(w) \leq \text{minrank}(m) + 2k - 1$.

Let μ be a super-stable matching of I such that $(m,w) \in \mu$. To prove the above, we shall bound $lowrank_m(w)$ from below using $minrank(w)$ and above using $minrank(m)$.

The former is straightforward. By 3.2, $minrank(w) \leq lowrankm(w)$.

Man m strictly prefers $lowrank_m(w) - 1$ women to w. Each such woman w' must be matched to a man m' that she strictly prefers to m (by 3.1), which gives us $lowrank_{w'}(m) < lowrank_{w'}(m')$. Again by 3.1, given the assumption that (m,w) are super-stably matched in I, we need not consider women that share a rank on m's list with w.

$$\text{lowrank}_{w'}(m) \leq \text{minrank}(m) + k - 1 \tag{3.1}$$

by 3.2 so

$$minrank(m') \leq lowrank_{w'}(m') < f(m) + k - 1 \tag{3.2}$$

The number of men who satisfy that property is bounded by:

$$|\{m' | \text{minrank}(m') \leq \text{minrank}(m) + k - 2\}| \leq \text{minrank}(m) + 2k - 3 \tag{3.3}$$

from 3.3. With that bound on the number of men who could be stably matched with the women man m prefers to w, it follows that m weakly prefers at most $minrank(m) + 2k - 2$ women to w so

$$\text{minrank}(w) \leq \text{lowrank}_m(w) \leq \text{minrank}(m) + 2k - 2 \tag{3.4}$$

The rest of the proof is essentially identical as in the case of stable matching without indifference. The remaining arguments can be found in Rosenbaum and Cheng [8], starting with Corollary 7.4.

The main result we find is as follows:

Theorem 3.5 *Given an SMI instance I of size n, there exists an $O(k^2 n + n^2)$ - time algorithm that computes the rotation digraph $G(I)$ and a nice path decomposition X of $G(I)$ with width at most $50k^2$, where $k = Range(I)$.*

This result leads to efficient algorithms for counting and sampling stable matchings, finding median stable matchings, and finding sex-equal and balanced stable matchings, as Rosenbaum and Cheng prove in [8] using results from their companion paper [7].

Chapter 4

Super-Stable Matching Codebase

To answer empirical questions in the realm of stable matchings, a codebase has been developed to generate preference lists and run stable matching algorithms. The codebase was developed using Python 3.7. The algorithms implemented are able to, for any SMTI instance,

- Obtain the reduced GS-lists and man-optimal and woman-optimal super-stable matchings (by extended Gale-Shapely algorithm, as discussed in Manlove [26])
- Find all rotations in the instance (by our Algorithm 1)
- Create the rotation poset (by minimal-differences algorithm discussed in Gusfield and Irving[13] and Hu and Garg[14])
- Check if a given matching is super-stable, strongly-stable, or weakly-stable (by search for blocking pairs)
- Find all stable matchings (using the antichains of the rotation poset)

Random preferences are generated by two systems. First, we consider a model with tiered preferences. In this model, men and women are arranged into tiers. The defining characteristic and restriction given by this model is that all men in a given tier are universally preferred by all women to all men in lower tiers. The same holds for women in a given tier. Without loss of generality, consider the generation of tiered preferences of men over women. We start with master list preferences, with each man having the same preferences over all women. For simplicity, we assume for some man i, i's preferences p_i are

$$p_i = \{w_0, w_1, w_2, w_3...w_{n-1}\} \tag{4.1}$$

The following arguments can also be applied to the women's lists over the men. Since the women all draw their preferences from the same distribution, the ordering of the master list doesn't matter. We partition these preferences into tiers.

$$p_i = \{\{w_0, w_1, w_2, w_3...w_l\}, ...\{w_m, ...w_{n-1}\}\} \tag{4.2}$$

Then, for each man, we shuffle each tier. A similar model is used by Beyhaghi et al. [4], though they only consider the case of two tiers. Note that any instance generated by this method is Range(k) where k is the size of the largest tier.

To generate general k-range preferences, we once again start with master list preferences for all men and women. From there, we undergo a process of iterative adjacent swaps. This process is described in Wilson et al. [30]. For each preference list, a number of swaps to be executed is calculated independently. That number is given by the following formula:

$$C/(\pi^2) * n * k^2 * \log_2(n) \tag{4.3}$$

This formula is adapted from Wilson et al [30]. In that paper they showed the similar formula

$$C/(\pi^2 - o(1)) * n^3 * \log_2(n) \tag{4.4}$$

for some C between 1 and 2 is sufficient to obtain the number of adjacent swaps that will approach the uniform random distribution of permutations of a deck of cards (or any ordered list). For our purposes we ignore the $o(1)$ term.

The general intuition behind the number of swaps being $O(n^3 * \log(n))$ is reasonably straightforward. First note that each element is undergoing a random walk, moving either up or down the list with equal probability (unless at the boundary). After t moves in a random walk, the expected distance from the starting position is $\sqrt{t}$. Therefore, $\Theta(n^2)$ moves are required for the distance from the start to be $\Theta(n)$. On average, it takes $n * \log(n)$ rounds for each individual to be swapped at least once, so we get $O(n^3 * \log(n))$ rounds for the final distribution to approach roughly uniform. Since this process must be conducted for each list, the number of swaps is $O(n^4 * \log(n))$.

Thus we settled on $C/(\pi^2) * n * k^2 * \log_2(n)$ for the distance from the start to be $\Theta(k)$. When a swap is selected, it will only be conducted if doing so doesn't violate the k-range restriction. Therefore, with this shuffling method, the instance will always fall within the k-range restriction (since we start with master preference lists). This method was chosen precisely because of this guarantee.

We chose the number of swaps after careful consideration of what the stationary distribution looks like. First, we note that given sufficiently many swaps, all women are equally likely to find themselves in any position. This should be plainly obvious from the fact that minranks are not fixed, and there's a nonzero probability that the first $(n-1) * n$ swaps send w_0 $n-1$ places to index $n-1$ on every list. So we do not provide an analysis of the true mixing time here, as we don't do nearly enough swaps to reach the true stationary distribution. Given the symmetry of the model, reaching this stationary distribution isn't particularly useful. The structure of the stable matchings is identical regardless of the labels of the agents; any reordering of the master list preferences that we start the process with can be done without changing the ultimate distribution of the stable matchings. Given this, we determined that selecting a number of swaps that gives a man roughly equal probability of moving to any index within his range (assuming that range is fixed) would be sufficient. An agent's range isn't fixed, but again given that the labels placed on agents is irrelevant, we determined this to be enough swaps to get a roughly uniform distribution over the possible non-redundant sets of preferences. Future work is planned to do a more formal analysis of this question.

With these two methods, we obtain random preferences that we can use to test our stable matching algorithms in SM instances. We must introduce indifference for SMT instances to be tested as well.

Indifference is introduced in a relatively straightforward manner. We create a list of all current adjacent pairs; this list is of size $n * (n-1)$. Given male preference lists $P_0..P_n$, this list can be characterized as

$$(w^i_j, w^i_{j+1}) \forall 0 < j < n, i <= n \tag{4.5}$$

Where w^i_j is the jth woman on man i's list.

We shuffle this list, and iterate through it, merging the ranks containing the women j and $j+1$ in man i's list with probability γ if this merge would not violate k-range restrictions. Again, this method was chosen as a simple iterative method that guarantees k-range restrictions are not violated while still producing a reasonably random distribution.

4.1 Runtime

Runtime is clearly $O(n^2)$ for the addition of indifference. The runtime for a swap of adjacent item in a list is constant, so the runtime for swaps and the total runtime for creation of a k-range SMT instance is $O(n^4 * \log(n))$.

4.2 Implementation

The functions to generate and analyze rotation posets were created using standard Python3 libraries, Numpy, Pandas, SageMath, and NetworkX. Since generating preference lists is computationally expensive, we used AWS EC2 instances for that task. The full codebase with which we worked can be found at `https://github.com/qhughes22/super-stable-matching-thesis` for reproducibility. The tools we created which may be useful to others can be found at `https://github.com/qhughes22/SMTI-krange`. All data analysis conducted for this thesis used a sample size of 100.

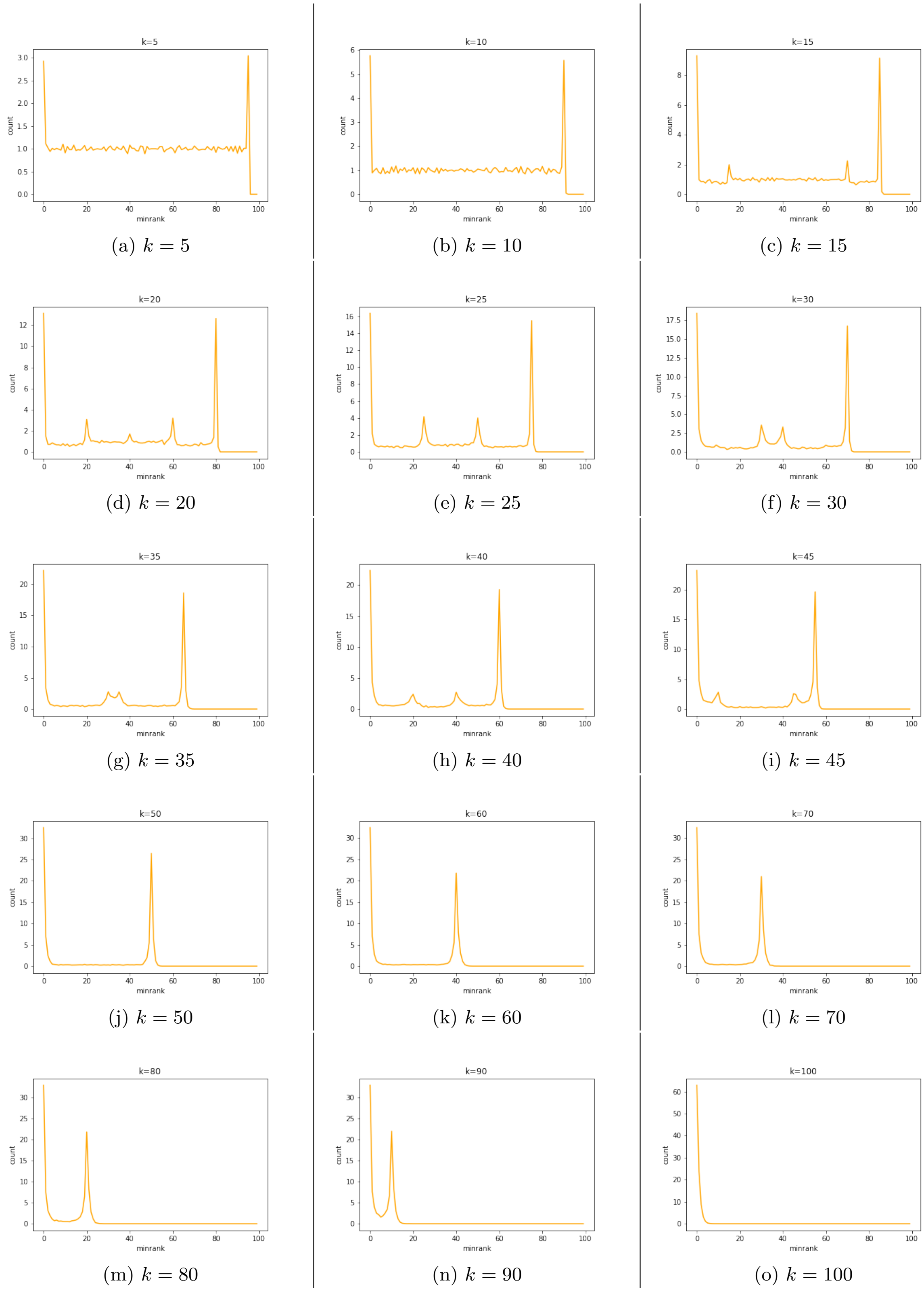
Chapter 5

Super-Stable Matching Empirical Results

5.1 Analysis of the Swapping Method

In this section we analyze the preference lists that result from the swap-based generation method.

Figure 5.1 shows us the distribution of minranks at different values of k. We see a number of different interesting patterns here. First, at all values of k, the global maximum is at 0, and there is a local maximum at $99 - k$. Neither of these results should be surprising. To illustrate why this is largely expected, consider $k = 5$. From the graph we see that on average about three women have minrank 0, three have minrank 95, and the remaining are distributed evenly across the values in between. As the swaps are executed, agents will tend to spread evenly in each direction if they are not at a boundary. An agent can spread $(k-1)/2$ in each direction before their range is restricted. Thus, we might expect the minrank of a woman beginning in index i to be roughly $i - (k-1)/2$. Once the left boundary is factored, we can expect minranks to equal $\max(0, i - (k-1)/2)$. This explains the peak at the left of the distribution. At the right of the distribution, we can reason that any agents with a minrank greater than or equal to $n - k + 1$ is not pushing the bounds of their range, as their maxrank can be no higher than $n - 1$. Such agents would eventually be swapped to have the maximum binding minrank, $n - k$. So with both bounds considered, the minrank of some woman starting in position i would be approximately $\min(n - k, \max(0, i - (k-1)/2))$. Furthermore, this implies $(k-1)/2 + 1$ or $(k+1)/2$ agents have minrank 0.



(a) $k = 5$ (b) $k = 10$ (c) $k = 15$

(d) $k = 20$ (e) $k = 25$ (f) $k = 30$

(g) $k = 35$ (h) $k = 40$ (i) $k = 45$

(j) $k = 50$ (k) $k = 60$ (l) $k = 70$

(m) $k = 80$ (n) $k = 90$ (o) $k = 100$

Figure 5.1: Total count of agents at each minrank for different values of k. Instances were of size 100 with 100 sets of preferences generated.

While this is the case for $k = 5$, we see higher average counts of minrank 0 for higher values of k, contrary to the analysis we just put forth. These calculations are just approximations. Since swaps are only allowed if the restrictions on both agent are met, analyzing the exact distribution over outcomes over single agents is very difficult to do. Nevertheless we put forth a tentative theory as to why there are higher counts of agents at minranks 0 and $n - l$.

Consider the following:

$J = \{i \in \mathbb{N} | 0 \leq i < k\}$ J represents the indices that are able to be occupied by women of minrank 0. With an outsized number of women having minrank 0, there are more women who can be swapped into these indices from the left than from the right, particularly when considering the upper end of the range. In other words, swaps in some index $i \in J$ from the left are more likely to be executed than swaps from the right, as swaps from the left have only one potentially binding k-range restriction, whereas swaps from the right have two agents who could be at the end of their range. This suggests that swaps near the boundary moving a woman's minrank down are more likely to occur than swaps moving her minrank up. This explains what we see in Figure 5.1, with the number of women having minrank 0 being greater than $(k + 1)/2$. The same effect happens at the upper boundary as well.

We can also see evidence of this effect in Figure 5.2. Notice the flatness of the tails of the distributions. The flatness at the lower bound extends past $k/2$. For $k = 20$, indices 0 to approximately 14 have the same average value. This suggests that these indices have the same distributions. If two indices i and $i + 1$ have different distributions but the same means, this implies there's some w_j and w_k such that

$$P(w_j \text{ is in index } i) > P(w_j \text{ is in index } i + 1) \tag{5.1}$$

,

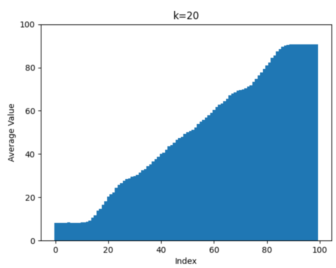
$$P(w_k \text{ is in index } i + 1) > P(w_k \text{ is in index } i) \tag{5.2}$$

and

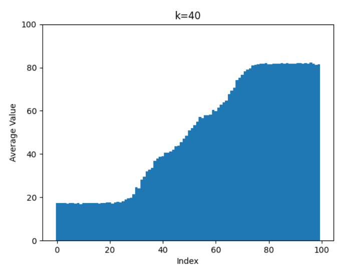
$$j > k \tag{5.3}$$

Given the nature of the swapping method, this is not possible. The only difference between w_j and w_k is their start index, and for w_j to reach index i, she must both pass through index $i+1$ and pass through w_k's starting index. Clearly, these equations cannot hold.

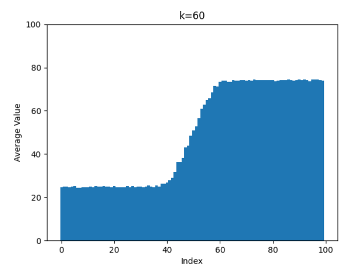
So we see with $k = 40$ the uniformity ends at approximately index 25. There are 23 agents on average with minrank 0 at $k = 40$. So it appears indices 0-25 are primarily populated by the agents w_0 to w_{22} (or whichever agents have minrank 0, these are the most likely). The same goes for the upper bound.



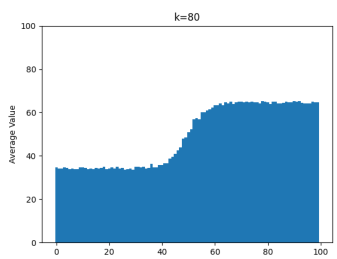
(a) $k = 20$



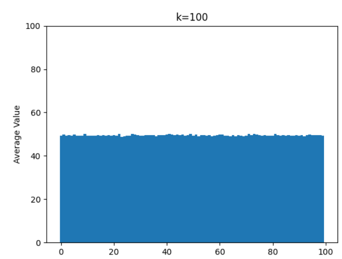
(b) $k = 40$



(c) $k = 60$



(d) $k = 80$



(e) $k = 100$

Figure 5.2: Average value of agents in index after swapping method. Lists began with value i in index i for all i.

Also note that the work of Wilson et al. [30] (and our decision to ignore the $o(1)$ term in the number of swaps) is validated in Figure 5.2, as with $k = n = 100$, the distribition is uniform.

There are other patterns in the distribution of minranks in Figure 5.1 of note as well. For $k \geq 50$, minranks are concentrated at two peaks: 0, and $n - k$, as discussed previously. For $k < 50$, we see other interesting patterns. For nearly all $k < 50$, we see peaks at about $n - 2k$ and $2k$.

Consider $k = 40$, with major peaks at 0 and 60, and minor peaks at 20 and 40. With many agents having minrank 0, during the swapping process it's more likely for an agent to be unable to swap from rank 40 to rank 39 than for an agent to be unable to swap from 41 to 40, as there are more agents at their boundary at 39. This means that agents will concentrate and pool at rank 40. Similarly, a swap between 59 and 60 is less likely to occur than most swaps for the same reason. Agents pool at rank 60, leaving them able to stretch their range into a minrank of 20.

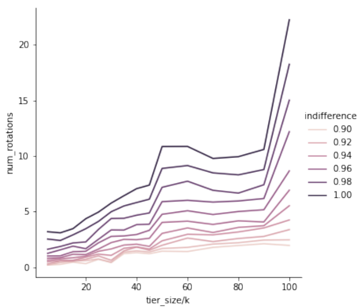
5.2 Analysis of Indifference

In Appendix 2 we can see the distribution of sizes of indifference classes as k and the indifference parameter γ change. The count of agents at each class decreases dramatically as the size of the class increases. Even in the unrestricted case we'd expect exponential decay, as on average only a constant portion of indifference classes of size a end up merging into becoming a class of size $a + 1$, since there's a constant probability of any two adjacent elements to be merged into the same class.

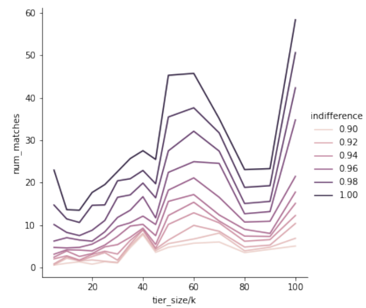
With k-range restrictions, we expect the rate of decrease to be even greater. Consider what happens when two indifference classes S_1 and S_2 in some man m's list are considered to be merged, with S_1 being above S_2. All of the elements in S_1 will go from having $|S_1|$ ranks in m's list to having $|S_1| + |S_2|$ ranks in that list. Effectively, this merging moves an element in S_1 $|S_2|$ places. The same argument holds for elements of S_2. Increases in the sizes of these sets leads to not only more agents having increased range in this list, but also an increase in the degree to which lowranks and highranks (and therefore, probabilistically, minranks and maxranks) are affected. Mergings between larger sets are less likely to satisfy k-range restrictions, so larger sets are unlikely.

5.3 Analysis of Stable Matching Structure

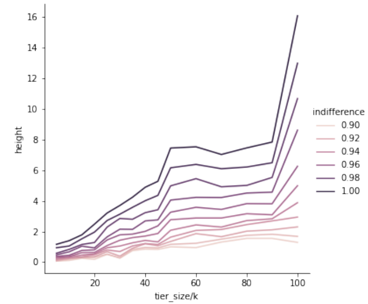
The first question we wanted to answer is how changing the amount of indifference impacted the structure of the stable matchings over varying values of k. We consider five metrics: the number of rotations, the number of super-stable matchings, the length of the maximum antichain in the rotation poset (height), the number of antichains in the rotation poset, and the pathwidth of the rotation poset.



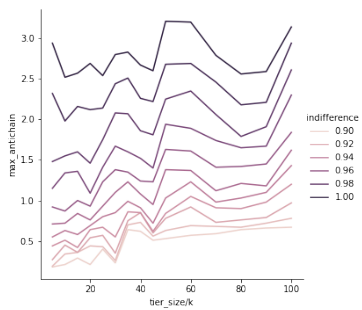
(a) Number of rotations



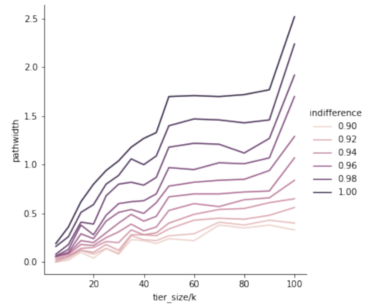
(b) Number of super-stable matchings



(c) Height of rotation poset



(d) Size of maximum antichain in rotation poset



(e) Pathwidth of rotation poset

Figure 5.3: Number of rotations, number of matchings, height of rotation poset, size of maximum antichain, and pathwidth for varying levels of indifference and k at $n = 100$.

Figure 5.3 shows the effect of varying the k parameter and the degree of indifference on five metrics. First notice across the board the same trends tend to hold regardless of the level of indifference. This suggests that super-stable matching rotation posets generally have the same structure as their stable matching counterparts under k-range.

Looking at the number of matches in Figure 5.3 (b) we see that the greatest number of matches is at $k = 100$, the unrestricted case. Uniformly random preferences have more super-stable matchings than random k-range instances. We also notice that from $k = 60$ to $k = 90$, the number of matchings and rotations decreases. This is likely because at $k > n/2$ there is overlap between the large number of agents with minrank 0 and with minrank $n - k$, and this overlap leads to blocking pairs. We provide evidence for this theory in the next section.

For $k < n/2$, we generally see the number of rotations and number of super-stable matchings increasing with k. As each agent has more potential super-stable partners, they find themselves in more rotations and super-stable matchings. There is an exception to this trend at $k = 45$, where the mean number of super-stable matchings sees a dip (and the mean number of rotations doesn't). This is a strange phenomenon that we don't have an explanation for.

In Figure 5.3 (c) we see that the mean height tends to increase with k, with a single exception at $k = 70$. The intuition here is reasonably simple. The height represents the length of a chain of dependencies. Those dependencies can either be type 1 or type 2 as discussed in Gusfield and Irving [13]. With a greater k, there's more possibility for overlap between rotations, and overlap means a greater possibility of dependencies. With small k, the number of potential partners is limited and so the number of rotations an agent can be involved in is limited as well, directly impacting the chance of a type 1 rotation dependence.

Figure 5.3 (d) gives us the size of the maximum antichain. Recall that an antichain is a set of items in a poset such that no item in the set is comparable to another. This metric is roughly a measure of how many rotations there are that are independent of one another. The trends here are less clear. It's worth noting that the maximum for the case without indifference is with a low k of 5. There is also a dip at $k = 45$ as with the number of super-stable matchings. The dip in the size of the maximum antichain may help explain the dip in the number of stable matchings as the number of stable matchings is exponential in the size of the maximum antichain. But why the size of the maximum antichain goes down at k=45 is unclear. It would be interesting to see if such a dip occurs for different values of n and k.

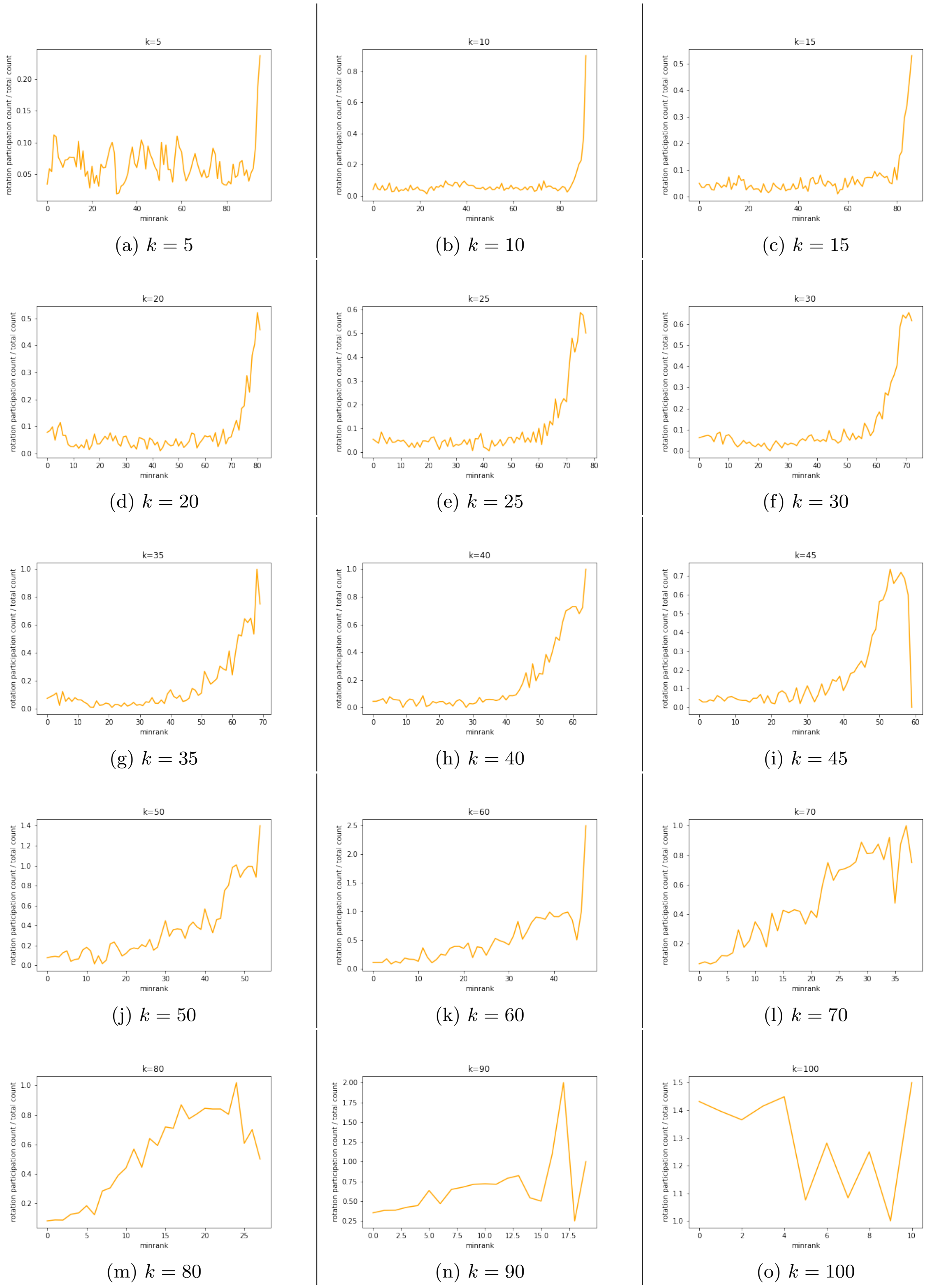
Finally we consider the pathwidth. Recall that Cheng and Rosenbaum [8] found that the pathwidth is bounded by $50k^2$ in the stable matching poset and I extended that bound to the super-stable matching poset. With random instances, the mean pathwidth is no higher than 2.5 even in the unrestricted $k = 100$ case. Random instances apparently tend to have much simpler rotation posets than the theoretical worst case. This also implies that with random instances problems like counting stable matchings can be solved significantly more quickly than the theoretical bound

$O(2^{50k^2} * k^2 * n + n^2)$ that Cheng and Rosenbaum [8] found. It is an open question to what extent preferences that align with the k-range model in practice are similar to the worst case or to our random preferences.

To determine what causes the trends in the number of rotations and the number of super-stable matchings in particular we look at what agents are involved in rotations. Figure 5.4 shows the ratio of occurences of agents of a given minrank in a rotation over the total number of agents with that minrank. In other words, it's the average number of rotations an agent is involved in by minrank. For $k < 50$, we see that agents with minrank near $n-k$ have the greatest high rotation counts. Up to that around that point, minrank has little to no effect on rotation counts.

At ranks $n-k$ through $n-1$, we see mostly agents with minrank $n-k$. These agents are likely to be involved in stable matchings with each other. To understand why, consider an agent m_i with minrank $< n-k$. Agent m_i likely prefers most women with minrank $< n-k$ to any of the women with minrank $n-k$, since women of minrank $n-k$ dominate the bottom k ranks on m_i's list. Therefore, m_i is unlikely to stably match with agents of minrank $n-k$, and also unlikely to block pairings between agents of minrank $n-k$. If we form the reduced GS-list for m_i, there are unlikely to be many agents of minrank $n-k$. We can almost then consider the agents of minrank $n-k$ to be a separate stable matching instance with rotations that are much less likely to be blocked by "outside influence" from higher minrank agents.

The same effect does not occur at minrank 0, despite the symmetry in the preference list generation. For a man m_j with minrank > 0, it's still unlikely that agents of minrank 0 are on their reduced list. But if such a woman w_k is on the reduced GS-list, the edge that results is more likely to be blocking stable matchings, as w_k is likely high up on the reduced list of m_j given her minrank of 0. Since it is hard for m_j to be moved above w_k on his list, this edge is more likely to block potential rotations among the agents of minrank 0 than similar edges are at the right boundary. For $k > 50$, we see the same general trends, though the results seem noisier.



(a) $k = 5$ (b) $k = 10$ (c) $k = 15$

(d) $k = 20$ (e) $k = 25$ (f) $k = 30$

(g) $k = 35$ (h) $k = 40$ (i) $k = 45$

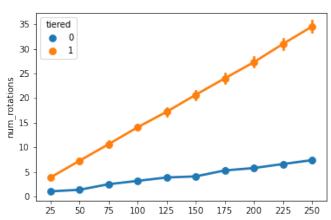
(j) $k = 50$ (k) $k = 60$ (l) $k = 70$

(m) $k = 80$ (n) $k = 90$ (o) $k = 100$

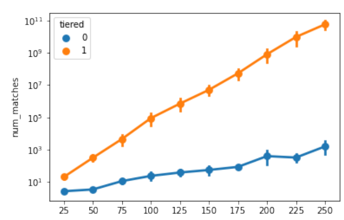
Figure 5.4: Ratio of appearances in rotations and total count of agents at minrank, with $n = 100$.

5.4 Comparing the Tiered Model to the K-Range Model

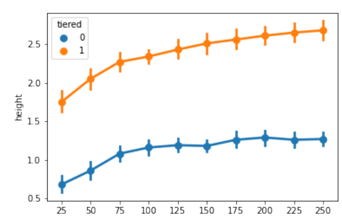
In this section we plot comparisons of the tiered model and the k-range model. There are three questions we consider. First, how do the tiered and k-range models compare when k/the tier size is fixed at a small number and n grows? Second, what happens when n is fixed and k varies? And third, how do the models compare with fixed n and variable k when one side of the preferences are uniformly random. All of these questions are considered without indifference. These questions are considered under the framework of tier size and k being analogous. Since there's no obviously fair/reasonable way to divide into tiers when n is not divisible by k, we only look at tier sizes that are factors of 100. This is a clear practical advantage of k-range: divisibility is not a requirement, and any k can work with any n.



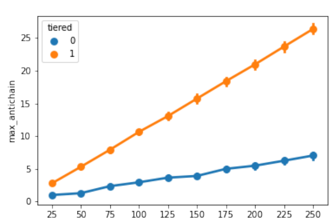
(a) Number of rotations



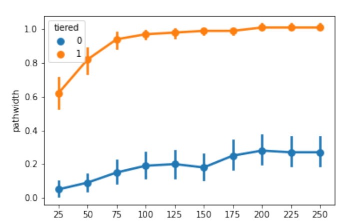
(b) Number of super-stable matchings



(c) Height of rotation poset



(d) Size of maximum antichain in rotation poset



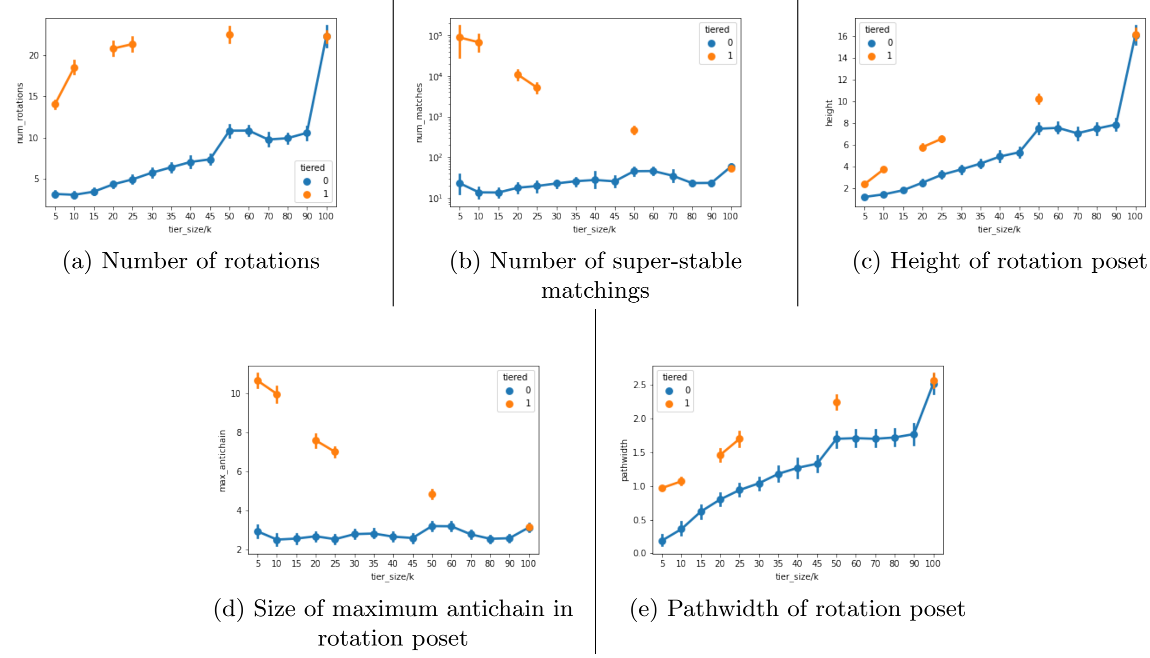
(e) Pathwidth of rotation poset

Figure 5.5: Number of rotations, number of matchings, height of rotation poset, size of maximum antichain, and pathwidth for varying levels of n with $k = 5$ fixed.

We consider the case of fixed k/tier size and growing n. We look at $k = 5$, and see the effect of ranging n from 25 to 250 in increments of 25. In both the tiered and k-range models, the number of rotations increases linearly and the number of stable matchings increase exponentially. The growth is much faster in the tiered model. This makes sense as unlike the tiered model, with k-range we have blocking pairs across minranks as discussed in the previous section.

The size of the maximum antichain increases linearly with both models. It's trivial to see why that would occur for the tiered model. For k-range, with such a small k, it follows that the size of the maximum antichain increases with n in the k-range, as there are more rotations in the instance and these rotations have non-overlapping extents (they affect agents with minranks far enough apart that there's no chance of dependence between the rotations).

The height of the rotation poset is roughly constant for the tiered model, while it increases slightly for k-range. In the tiered model, the height is just the max height of any tier, which is naturally capped at the maximum height possible for instances of size 5. In the k-range, no such cap exists, and we see an increase at what could be a logarithmic rate.



(a) Number of rotations

(b) Number of super-stable matchings

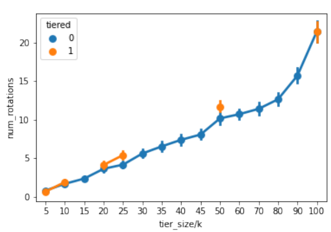
(c) Height of rotation poset

(d) Size of maximum antichain in rotation poset

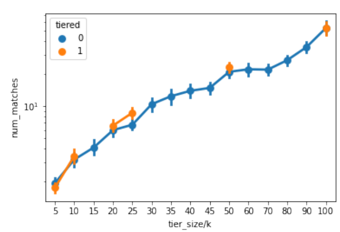
(e) Pathwidth of rotation poset

Figure 5.6: Number of rotations, number of matchings, height of rotation poset, size of maximum antichain, and pathwidth for varying levels of k for both k-range and tiered models

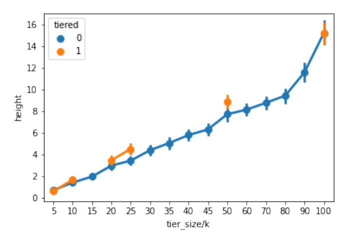
In Figure 5.6 we see the effect of fixing $n = 100$ and varying k. In the tiered model, the number of rotations doesn't differ significantly between $k = 20$, $k = 25$, $k = 50$ and $k = 100$. However, the number of rotations decreases with an increase in the tier size. This is because what rotations are present with smaller k are independent of one another, so the size of the maximum antichain is larger with small k and therefore there are more super-stable matchings for the same number of rotations. Across the board, the tiered model and the k-range model differ in both values and trends. Tiered preferences do not appear to be a good proxy for k-range.



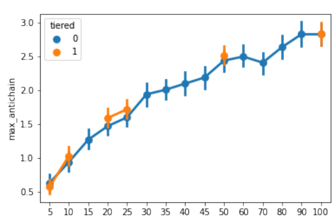
(a) Number of rotations



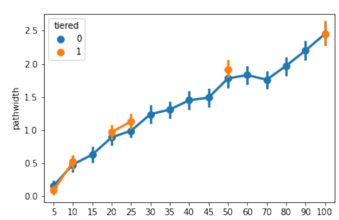
(b) Number of super-stable matchings



(c) Height of rotation poset



(d) Size of maximum antichain in rotation poset



(e) Pathwidth of rotation poset

Figure 5.7: Number of rotations, number of matchings, height of rotation poset, size of maximum antichain, and pathwidth for varying levels of k for one-sided k-range and tiered models

As we see in Figure 5.7 the tier-based model seems a reasonable approximation of the k-range model when one side has uniformly random preferences. In both cases we see what could be a linear increase in all four metrics. As the tier size/k grows, the distributions for the restricted male lists and unrestricted women's lists become more similar, and that might be why we see more rotations (both independent of another and dependent) and matchings. The tiered model has a taller rotation poset as k increases as well as a greater number of rotations and matches, though the differences are small. This result is interesting, and we don't have an explanation for it yet as analysis of these models against uniform preferences is difficult. But the important result, we feel, is that the possibly easier to work with tiered model follows the same trends as k-range and can be used as an approximation for k-range.

Chapter 6

Concluding Remarks and Open Questions

In this thesis we have found that super-stable matchings in SMT follow a structure analogous to stable matchings in SM instances. We've also found that problems related to the rotation poset of a stable matching instance that are fixed-parameter tractable under k-range restrictions are also fixed-parameter tractable when related to the rotation poset of a super-stable matching instance. We provide a method to generate random k-range preferences, and compare the instances given by that method to random tier-based preferences.

There are a number of open questions relating to the subject of this thesis:

- Can the upper bound of $50k^2$ on the pathwidth of the rotation digraph be tightened? The bounds on the pathwidth of the rotation poset for SM instances was shown to be $50k^2$ in Cheng and Rosenbaum [8] and we extended those bounds to the super-stable matching poset in SMTI instances. In practice, we found pathwidth values that were much lower. Do tighter worst-case bounds exist?

- How well does the swapping method for generating k-range preferences approximate sampling from the uniform distribution of all possible k-range instances? We provided an argument based on the work of Wilson et al. [30] that the number of swaps we conduct is roughly sufficient. Is it? What does the uniform distribution of k-range instances look like when we consider the symmetry of the agents and the irrelevance of how we label them and what position they start in? Further work on this question in particular is planned.

- How can we explain the non-monotonicity of the number of matchings/rotations as a function of k under the k-range model? We provided an explanation for the dip in the number of rotations and stable matchings between $k = 50$ to $k = 100$. Does this pattern hold for all $k = n/2$ and all $k = n$? Is our explanation correct? What explains the other trends in our model?

- Do the trends seen with the tiered model hold when tiers of variable sizes are considered? We only considered uniform tier sizes. What if the men and women have different size tiers? In this case there would clearly be more dependency across rotations, as the tiers are no longer separate and matchings within a tier can be blocked from without. What about tiers that vary in size even for a single class?

- Can we consider a k-range model where k is not the same for every agent? Similar to the question above, can we devise a model in which the range of agents are not given the same restrictions? If, say, half of agents have $k = k_1$ and half have $k = k_2$, what can we say about the bounds on the rotation poset? What about different k's for the men and women?

- How would the number of strongly or weakly-stable matchings be affected by varying the range and indifference parameter of an instance? Would our empirical results look similar if we analyzed strongly or weakly-stable matchings instead of super-stable matchings? We'd see more weakly-stable matchings with more indifference clearly, but what about strongly-stable? The super-stable matching rotation poset follows the same trends as the case without indifference on all five metrics. Would the same be true with weakly or strongly-stable matchings?

- Is there a reasonable way to integrate incomplete lists into a k-range model? Incomplete lists as unacceptable partners could be implemented by taking k-range preferences and removing the bottom l agents from the list. This may not have a meaningful effect on the rotation poset, though. What about generating k-range preferences and removing agents at random? Or agents selectively removing agents from their list in a game-theoretic strategic way to maximize their likelihood of getting a favorable match in equilibrium, as studied in Beyhaghi et al [4]? Can we take an instance with incomplete preferences and find the minimum possible range of complete preferences that underly that instance? Are there bounds on the rotation poset with SMI instances under k-range?

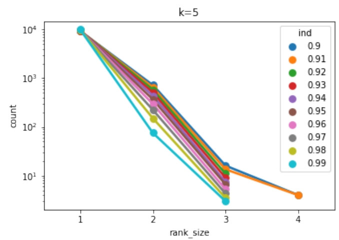
Hopefully further work can answer these questions.

Appendices

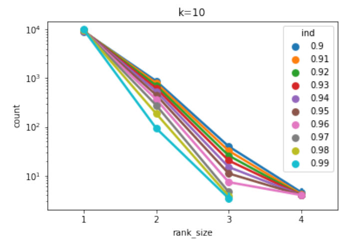
Algorithm 2: Computing a maximal sequence of super-stable matchings

```
let M_0 be the (unique) man-optimal super-stable matching of G.
let M_z be the (unique) woman-optimal super-stable matching of G.
M ← M_0
let M′ contain edge (m, M(m)) for each man m such that M(m) =_m M_z(m)
let E_d contain all edges of M
let G_d be the directed graph (V, E_d) such that each edge (m, w) ∈ E_d ∩ M is
  directed from w to m and every other edge (m, w) is directed from m to w
E′ ← E\E_d
let E_c = M′ and G_c = (V, E_c)
for each (m, w) ∈ M remove from E′ each edge (m′, w) such that m′ ≺_w m and
  each edge (m, w′) such that w′ ⪰_m w
repeat
    while (∃m ∈ A) deg_{G_c}(m) = 0 do
        add the set E_m of top choices of m from E′ to E_d
        if outdeg(S(m)) = 0 then
            add every edge (m, w) ∈ E_m such that m ≻_w M(w) and
            M(m) ≻_m w to E_c
            for each edge (m, w) of E_c that becomes strictly dominated by
            some added edge (m′, w) remove it from G_c
            remove E_m from E′
        end if
    end while
    for each m ∈ A such that outdeg(S(m)) = 0 do
        delete all lowest ranked edge in E_c ∪ E′ incident to any w ∈ S such
        that w is multiple engaged
    end for
    while (∃S) outdeg(S) = 0 and E_c is a perfect matching on S do
        M ← (E_c ∩ S) ∪ (M\S)
        M_i ← M
        output M_i
        i ← i + 1
        update G_c and G_d: E_c ∩ S contains only edges (m, M(m)) such that
        M(m) =_m M_z(m); an edge (m, w) ∈ S stays in G_d only if w = M(m)
        and rank_w(m) ≤ rank_M(w)
    end while
until (∀m ∈ A) rank_M(m) = rank_{M_z}(m)
```

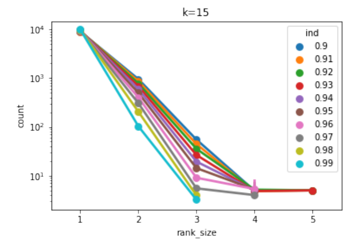
Figure 1: Original algorithm for finding all super-stable matchings from Hu and Garg [14]



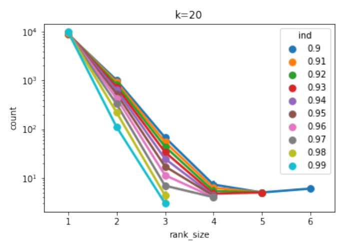
(a) $k = 5$



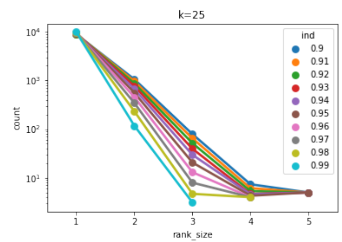
(b) $k = 10$



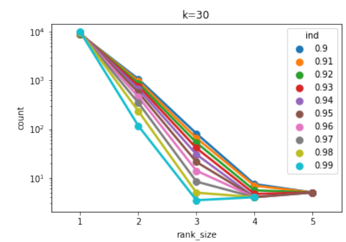
(c) $k = 15$



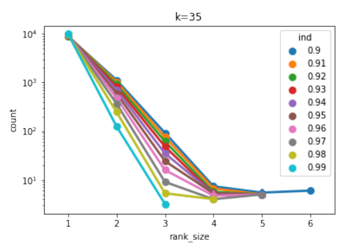
(d) $k = 20$



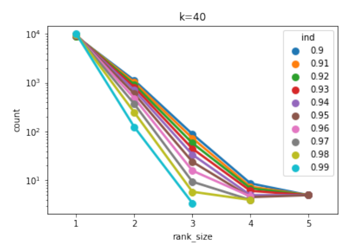
(e) $k = 25$



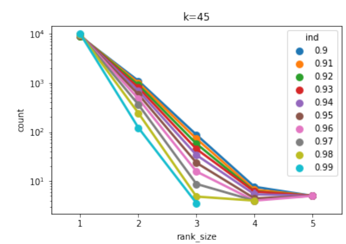
(f) $k = 30$



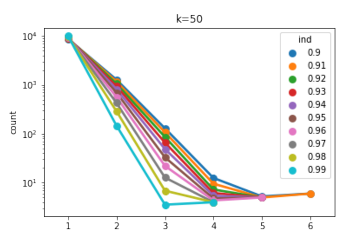
(g) $k = 35$



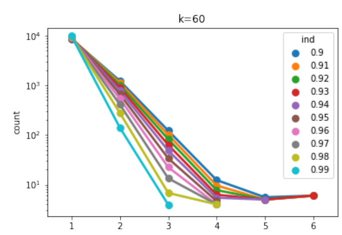
(h) $k = 40$



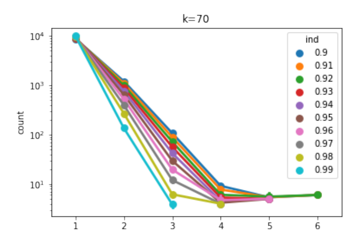
(i) $k = 45$



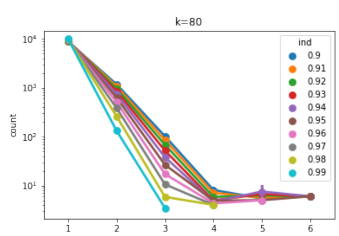
(j) $k = 50$



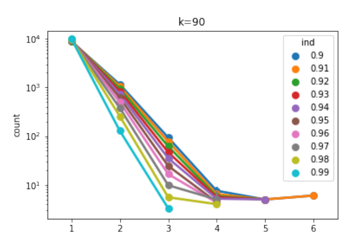
(k) $k = 60$



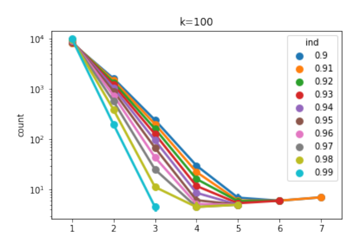
(l) $k = 70$



(m) $k = 80$



(n) $k = 90$



(o) $k = 100$

Figure 2: Count of agents at each rank size for varying indifference parameters and varying restrictions of k. An instance having a single list with an indifference class of size 5 would increase the count by 5.

Bibliography

[1] Strongly Stable Matchings in Time O(nm) and Extension to the Hospitals-Residents Problem | SpringerLink.

[2] David Aldous. Random walks on finite groups and rapidly mixing markov chains. In Jacques Azéma and Marc Yor, editors, *Séminaire de Probabilités XVII 1981/82*, volume 986, pages 243–297. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983. Series Title: Lecture Notes in Mathematics.

[3] Itai Ashlagi, Mark Braverman, Amin Saberi, Clayton Thomas, and Geng Zhao. Tiered Random Matching Markets: Rank is Proportional to Popularity. *arXiv:2009.05124 [cs, econ]*, January 2021. arXiv: 2009.05124.

[4] Hedyeh Beyhaghi, Daniela Saban, and Eva Tardos. Effect of selfish choices in deferred acceptance with short lists. *arXiv:1701.00849 [cs]*, January 2017. arXiv: 1701.00849.

[5] Hedyeh Beyhaghi and Éva Tardos. Randomness and Fairness in Two-Sided Matching with Limited Interviews. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 74:1–74:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ISSN: 1868-8969.

[6] Nayantara Bhatnagar, Sam Greenberg, and Dana Randall. Sampling stable marriages: why spouse-swapping won't work. In *Proceedings of the nineteenth annual ACM-SIAM symposium*

on Discrete algorithms, SODA ’08, pages 1223–1232, USA, January 2008. Society for Industrial and Applied Mathematics.

[7] Christine T. Cheng and Will Rosenbaum. Simple Counting and Sampling Algorithms for Graphs with Bounded Pathwidth. August 2020.

[8] Christine T. Cheng and Will Rosenbaum. Stable Matchings with Restricted Preferences: Structure and Complexity. *arXiv:2008.08480 [cs, math]*, January 2021. arXiv: 2008.08480.

[9] Ágnes Cseh and Klaus Heeger. The stable marriage problem with ties and restricted edges. *Discrete Optimization*, 36:100571, May 2020.

[10] Antoine Desir, Vineet Goyal, and Danny Segev. Assortment Optimization under a Random Swap based Distribution over Permutations Model. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 341–342, Maastricht The Netherlands, July 2016. ACM.

[11] Jean-Paul Doignon, Aleksandar Pekeč, and Michel Regenwetter. The repeated insertion model for rankings: Missing link between two subset choice models. *Psychometrika*, 69:33–54, March 2004.

[12] D. Gale and L. S. Shapley. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. Publisher: Mathematical Association of America.

[13] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. Foundations of Computing. MIT Press, Cambridge, MA, USA, August 1989.

[14] Changyong Hu and Vijay K. Garg. Characterization of Super-Stable Matchings. In Anna Lubiw and Mohammad Salavatipour, editors, *Algorithms and Data Structures*, Lecture Notes in Computer Science, pages 485–498, Cham, 2021. Springer International Publishing.

[15] Robert W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48(3):261–272, February 1994.

[16] Robert W. Irving and Paul Leather. The Complexity of Counting Stable Marriages. *SIAM Journal on Computing*, 15(3):655–667, August 1986. Publisher: Society for Industrial and Applied Mathematics.

[17] Kazuo Iwama and Shuichi Miyazaki. Stable Marriage with Ties and Incomplete Lists. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*, pages 883–885. Springer US, Boston, MA, 2008.

[18] Anna R. Karlin, Shayan Oveis Gharan, and Robbie Weber. A Simply Exponential Upper Bound on the Maximum Number of Stable Matchings. *arXiv:1711.01032 [cs, math]*, November 2017. arXiv: 1711.01032.

[19] Zoltán Király. Linear Time Local Approximation Algorithm for Maximum Stable Marriage. *Algorithms*, 6:471–484, September 2013.

[20] Donald Ervin Knuth. *Mariages stables et leurs relations avec d'autres probl??mes combinatoires: introduction ?? l'analyse math??matique des algorithmes*. Presses de l'Universit?? de Montr??al, Montr??al, 1976. OCLC: 3061053.

[21] Adam Kunysz. An Algorithm for the Maximum Weight Strongly Stable Matching Problem. In Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao, editors, *29th International Symposium on Algorithms and Computation (ISAAC 2018)*, volume 123 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISSN: 1868-8969.

[22] Adam Kunysz, Katarzyna E. Paluch, and Pratik Ghosal. Characterisation of Strongly Stable Matchings. *SODA*, 2016.

[23] Chi-Kit Lam and C. Gregory Plaxton. A (1 + 1/e)-Approximation Algorithm for Maximum Stable Matching with One-Sided Ties and Incomplete Lists. In *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 2823–2840. Society for Industrial and Applied Mathematics, January 2019.

[24] C. L. Mallows. Non-Null Ranking Models. I. *Biometrika*, 44(1/2):114–130, 1957. Publisher: [Oxford University Press, Biometrika Trust].

[25] David F. Manlove. Stable marriage with ties and unacceptable partners, 1999.

[26] David F. Manlove. The structure of stable marriage with indifference. *Discrete Applied Mathematics*, 122(1):167–181, October 2002.

[27] Peter Rabinovitch. *Uniform and Mallows random permutations : inversions, levels & sampling.* Doctor of Philosophy, Carleton University, Ottawa, Ontario, 2012.

[28] Sandy Scott. A Study Of Stable Marriage Problems With Ties. page 184.

[29] Boris Spieker. The set of super-stable marriages forms a distributive lattice. *Discrete Applied Mathematics*, 58(1):79–84, March 1995.

[30] David Bruce Wilson. Mixing times of lozenge tiling and card shuffling Markov chains. *The Annals of Applied Probability*, 14(1):274–325, February 2004. Publisher: Institute of Mathematical Statistics.

[31] L. B. Wilson. An analysis of the stable marriage assignment algorithm. *BIT Numerical Mathematics*, 12(4):569–575, December 1972.