Elicitation and Approximately Stable Matching with Partial Preferences

Joanna Drummond and **Craig Boutilier**
Department of Computer Science
University of Toronto
{jdrummond,cebly}@cs.toronto.edu

Abstract

Algorithms for stable marriage and related matching problems typically assume that full preference information is available. While the Gale-Shapley algorithm can be viewed as a means of eliciting preferences incrementally, it does not prescribe a general means for matching with incomplete information, nor is it designed to minimize elicitation. We propose the use of *maximum regret* to measure the (inverse) degree of stability of a matching with partial preferences; *minimax regret* to find matchings that are maximally stable in the presence of partial preferences; and heuristic elicitation schemes that use max regret to determine relevant preference queries. We show that several of our schemes find stable matchings while eliciting considerably less preference information than Gale-Shapley and are much more appropriate in settings where approximate stability is viable.

1 Introduction

Matching problems are ubiquitous and find application in a variety of domains. One of the most widely studied economic matching problems is the *stable marriage problem* [Gale and Shapley, 1962], in which members of two disjoint groups (colloquially men and women) express preferences for being matched (married) to members of the opposite group. A *stable matching* ensures that specified partnerships offer no incentive for an unmatched pair to defect from the matching. The stable marriage problem is emblematic of a rich set of bipartite matching problems in which elements of each set have some affinity for, or preference over, elements of the other. It has direct application to a variety of problems (e.g., labor market matching, school admissions) [Niederle *et al.*, 2008]; and the classic *Gale-Shapley (GS) algorithm* [Gale and Shapley, 1962], and its variants, can be used to compute stable matchings very effectively.

While algorithms for many forms of stable matching are computationally efficient, the informational burden they place on participants can be severe: they usually require that participants rank all potential partners/matches. Of course, algorithms like GS can be used in an interactive fashion to avoid this, as we outline below, but are not designed specifically to minimize this informational burden. For large-scale matching problems involving hundreds or thousands of options on each side (e.g., hospital-resident matching, or paper-reviewer matching), not only is the cognitive burden on participants high, much of this ranking information will generally be *irrelevant* to the goal of producing a stable matching. For example, in resident matching, one expects some degree of correlation across hospitals in the assessment of the most desirable candidates, and residents to have correlated views on the desirability of hospitals. Thus, more desirable candidates will tend to be matched to more desirable hospitals, and "first-tier" participants (hospitals and residents) are wasting time and mental energy if they offer precise rankings of lower-tier alternatives, while second-tier participants should not bother providing precise rankings of first-tier alternatives.

In this paper, we develop a framework for incremental *preference elicitation* in stable matching problems, as well as procedures for *robust matching* in the presence of incomplete preference information. Our goal is two-fold: first, we want to find stable matchings while requiring that participants specify only relevant information about their preferences; second, because full stability may require considerable preference information, we want to exploit some form of *approximate stability* to further reduce the information burden in practice. To address the latter goal, we use *maximum regret* to bound the potential for defection in a matching with partial preferences, and devise algorithms for computing matchings with *minimal maximum (minimax) regret*. To address the former, we develop elicitation schemes that use the matchings computed via minimax regret to find suitable queries. Our empirical results suggest that our regret-based methods and elicitation heuristics find approximate and exact stable matchings with much less than complete preference information across a variety of preference distributions, and significantly outperform the (interactive) GS algorithm.

2 Background

We first describe the stable matching problem addressed in this paper, briefly discuss prior work on matching with partial preferences and elicitation, then outline the probabilistic preference models used in our experiments.

Stable Matching. In this paper we focus on the classic *stable marriage/matching problem (SMP)* [Gale and Shapley, 1962], which we formulate using the common "marriage scenario." An SMP consists of set of men M and women W, each of size n, and a set of preference orderings associated with each participant: each man $m \in M$ has a strict total *preference order* $\succ_m$ over W, with $w \succ_m w'$ indicating that m prefers w to w' as a potential partner; each $w \in W$ has a similar ordering over M. We use $\succ$ to denote the *set of preference rankings* or *preference profile* of the SMP. For any $q \in M \cup W$, we use R^q to refer q's (*range* of) potential partners, i.e., $R^q = W$ if $q \in M$, and $R^q = M$ if $q \in W$. A *matching* is a function $\mu : M \cup W \to M \cup W$, such that $\mu(q) \in R^q$, and $q = \mu(\mu(q))$. In other words, μ matches men and women in a 1:1 correspondence. Given a matching μ, we say (m, w) is a *blocking pair* if $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$. A blocking pair destabilizes μ since m and w prefer each other to their partners in μ, and thus have incentive to *defect* (or "run away" with each other). A matching is *stable* if it admits no blocking pairs.

The Gale-Shapley (GS) algorithm [Gale and Shapley, 1962] for finding stable matchings proceeds in rounds (we describe the female-proposing variant). Initially, nobody is engaged. At each round, any unengaged woman proposes to the favorite man in her preference order to whom she has not yet proposed. Each man receiving (one or more) proposals accepts the proposal of the proposer (including his current engaged partner, if any) who is highest in his preference ranking—they become engaged—and rejects all other proposals. Once a round is reached where each woman is engaged, the current set of engagements is returned as the matching μ. This mechanism has a number of remarkable properties: it will converge to a stable matching in polynomial time; it is strategy-proof for women (but not men, vice versa if men propose); and among (the lattice of) all stable matchings it gives every woman her most preferred "achievable" partner [Gale and Shapley, 1962; Niederle *et al.*, 2008].

This approach has been extended in a variety of ways and has been applied in many practical settings. Many-to-one extensions are common (e.g., hospital-resident or student-school match [Roth, 1984; Abdulkadiroglu *et al.*, 2005]), and more flexible forms of preferences can be accommodated. For instance, "incomplete" preference lists can be used to express acceptability thresholds: a total order over a subset of partners is specified, with unranked partners deemed unacceptable [Roth *et al.*, 1993]. Indifference (ties) in rankings can also be accounted for [Irving, 1994], and give rise to new forms of stability. For instance, weak stability requires that both partners in a blocking pair *strictly* prefer each other, whereas strong stability allows one of the partners to be indifferent between his/her current and blocking partners. These two extensions can be combined to form the *Stable Matching Problem with Incomplete Lists and Ties (SMP-ILT)*: determining if there is a weakly stable matching in SMP-ILT where all men (or women) are matched is NP-complete [Iwama *et al.*, 1999]; however, SMP-ILT can be addressed heuristically [Gelain *et al.*, 2010; Gent and Prosser, 2002].

Related Work. The problems of incremental preference elicitation and computing stable matchings with partial preference information have received relatively little attention. GS is typically implemented by having participants submit their complete preferences, with the "proposal/acceptance" simulated within the algorithm. However, it can be viewed as an *interactive elicitation algorithm* as well, with proposers and acceptors providing only the information needed to run GS. We are unaware attempts to use early termination of GS to find approximately stable matches. Pini *et al.* [2011] develop several notions of approximate stability, but this is not used to address partial information (and relies on real-valued utilities rather than ordinal preferences). Closest in spirit to our work is that of Rastegari *et al.* [2013], who investigate the problem of minimizing the number of interviews needed to find a stable matching in a labor market. They analyze the problem from several perspectives. They show that finding a minimal certificate (i.e., set of partial preferences that supports an employer-optimal matching) is NP-hard. They also provide an MDP-formulation assuming a prior over preferences (which enumerates all, information states or, in our terminology below, "partial preference profiles"), but is otherwise polynomial in this (exponentially-sized) set. Our work differs in its focus on computing matches that are approximately stable given partial preferences, computationally effective heuristics for elicitation, and anytime evaluation (we also evaluate our schemes experimentally).

Biró and Norman [2012] present a method for stable matching that can be viewed as an elicitation scheme. Individuals interact randomly, and form a new pair if it offers benefit (relative to their current partners). They show that this is likely to result in an egalitarian solution. However, interpreted as an elicitation scheme, it seems to both scale poorly computationally and require far more elicitation rounds than our approach. Other work has considered equilibrium matching with unobservable preferences [Liu *et al.*, 2012]. It is known that the communication complexity of stable matching is $\Omega(n^2 \log n)$ [Chou and Lu, 2010], hence no elicitation scheme can reduce elicitation burden the worst case.

Our work is closely related to previous work on regret-based robust optimization and preference elicitation [Boutilier *et al.*, 2006; Braziunas and Boutilier, 2007; Lu and Boutilier, 2011b]. Specifically, we adapt the methods of Lu and Boutilier [2011b] for regret-based winner determination and elicitation in voting using ordinal preference rankings to our robust stable matching problem.

Probabilistic Preference Models. To capture correlations in preferences, we consider several standard probabilistic models of preferences below. The *Mallows ϕ-model* [Mallows, 1957; Marden, 1995] is parameterized by a modal or *reference ranking* σ and a *dispersion parameter* $\phi \in (0, 1]$, with the probability of a ranking r given by $P(r \mid \sigma, \phi) \propto \phi^{d(r,\sigma)}$, where d is Kendall's τ distance metric. When $\phi = 1$ we obtain the uniform distribution over rankings (i.e., *impartial culture*), and as $\phi \to 0$ the distribution concentrates all mass on σ. The Mallows model (and mixtures thereof) have plausible psychometric motivations and are commonly used in machine learning [Murphy and Martin, 2003; Meila and Chen, 2010; Lebanon and Mao, 2008; Lu and Boutilier, 2011a].

The *riffle independence model* [Huang and Guestrin, 2009] partitions a set of items into two sets: a ranking of each set is generated stochastically (and independently); then a stochastic process is used to interleave or "riffle" the two resulting rankings to produce a combined ranking. The model can also be defined hierarchically, with the same process used to generate the subrankings.

3 Matching with Partial Preferences

In any preference-based matching process, it may be appropriate to elicit only partial information about user preferences, especially when certain preference information is unlikely to be relevant in the computation of the matching or when the costs of elicitation outweigh the benefits derived. However, the partial information elicited may not be sufficient to produce a fully stable matching. In this section, we define a specific form of approximate stability given partial preferences using *minimax regret*. Apart from providing worst-case stability guarantees, we will see that it is a very effective driver of preference elicitation in Sec. 4.

3.1 Minimax Regret

For any $q \in M \cup W$, we assume that q's ordinal preferences are captured by a total order or ranking $\succ_q$ over R^q. (Our methods can be readily adapted to handle ties and unacceptable alternatives in a person's preferences, but we focus on total orders for ease of exposition.) However, we suppose that we have access only to partial information about these preferences. This information takes the form of a *partial ranking* P_q for q, which is a partially order over R^q, or equivalently (the transitive closure of) a consistent collection of *pairwise comparisons* of the form $r \succ_q r'$. Most constraints on preferences, including responses to most natural queries can be represented in this way. A *partial preference profile* $\mathbf{P}$ is a set of a partial rankings $\{P_q : q \in M \cup W\}$. A *completion* of P_q is any total order $\succ_q$ that extends P_q. Let $C(P_q)$ denote the set of completions of P_q and define the completions $C(\mathbf{P})$ of a partial profile $\mathbf{P}$ in the obvious way.

Several special forms of partial preferences are of interest, including *top-k preferences*, in which a user expresses a total order over her k most preferred partners and provides no information over the remaining partners (other than the implicit fact that they are less preferred). *Partitioned preferences* [Lebanon and Mao, 2008] allow a user q to partition potential partners R^q into an ordered set of subsets $R_1, \ldots, R_s$ s.t. (a) for all $i < j \le s$, if $x \in R_i$ and $y \in R_j$ then $x \succ_q y$; and (b) for each $i \le s$, items in R_i are unordered. Top-k preferences are a special case of partitioned preferences and both can be represented as a set of pairwise comparisons.

Example 1. To illustrate consider an example with $n = 4$ men and women in which all men and women have identical preferences. Let $\succ_w$ be the same for each $w \in W$, with $m_0 \succ_w m_1 \succ_w m_2 \succ_w m_3$; and similarly let $w_0 \succ_m w_1 \succ_m w_2 \succ_m w_3$ for each $m \in M$. One possible partial preference ranking for m_i consists of two equal-sized partitions: $P_{m_i} = (w_0, w_1) \succ (w_2, w_3)$. Indeed, this takes the form of partitioned preferences. This partial preference has four possible completions (e.g., $w_1 \succ w_0 \succ w_3 \succ w_2$) including the true underlying full ranking. The partial profile $\mathbf{P}$ consisting of this same "two-by-two" partitioning for each $m \in M, w \in W$ will be discussed below.

Our interest is in *partial-preference stable matching problems (PP-SMPs)*, SMPs in which the user preferences are replaced by partial rankings. Given the uncertainty inherent in a partial preference profile $\mathbf{P}$, one cannot guarantee the existence of a stable matching: while a matching μ may be stable under some completions of $\mathbf{P}$, it may not be under other completions. Thus, intuitively, a solution for a PP-SMP instance should be as stable as possible. We measure the stability of μ by considering the *maximal possible incentive* that some blocking pair might have for "defecting" over all possible realizations of their preferences. While there are several plausible ways in which to measure the incentive for a blocking pair to deviate, we adopt one reasonably natural one here. Specifically, we define the incentive for q to switch from partner r to partner r' to be the *improvement in rank position* in $\succ_q$ attained by the switch. For any total order $\succ_q$ over R^q, let $s_q(r, \succ_q) = n - rank(r, \succ_q)$ be the *Borda score* of r in q's ranking. For any $q \in M \cup W$, preference ordering $\succ_q$, partial ranking P_q and $r, r' \in R_q$, define:

$$Regret(q, r', r, \succ_q) = s_q(r', \succ_q) - s_q(r, \succ_q) \quad (1)$$

$$PMR(q, r', r, P_q) = max_{\succ_q \in C(P_q)} s_q(r', \succ_q) - s_q(r, \succ_q) \quad (2)$$

$Regret(q, r', r, \succ_q)$ is the degree to which r' is preferred to r by q given known preference $\succ_q$. When only partial preferences P_q are known, *pairwise max regret* $PMR(q, r', r, P_q)$ is the greatest degree to which r' might be preferred to r over all possible preference realizations.

Example 2. Using the partial profile described in Example 1, we have $PMR(m_0, w_1, w_2, P_{m_0}) = 3$: m_0's Borda score for w_1 could up to 3 greater than his score for w_2 in some completion of his partial ranking (indeed, this is so in the completion described in Example 1).

Using rank improvement to measure the incentive for an individual to deviate from one partner to another gives us the ability to measure the *degree of instability* of a blocking pair.[1]

Definition 3. *Given a SMP and matching μ, the* (degree of) instability *of pair (m, w) (resp. μ) is:*

$$Inst(m, w, \mu, \succ_m, \succ_w) = \min\{PMR(m, w, \mu(m), \succ_m), PMR(w, m, \mu(w), \succ_w)\} \quad (3)$$

$$Inst(\mu, \succ) = \max_{(m,w)} Inst(m, w, \mu, \succ_m, \succ_w) \quad (4)$$

With complete preferences, (m, w) is a blocking pair iff $Inst(m, w, \mu, \succ_m, \succ_w) > 0$ and μ is stable iff $Inst(\mu, \succ) \le 0$. In this sense, our definition generalizes that of stability on the full information sense.

Notice that we assume the incentive of a pair to deviate is that of the *least willing partner*. This is sensible under the usual assumption of non-transferable utility. But other definitions are possible, including the summing the PMR values of both members of the pair. Similarly, we equate the instability of a matching with that of the maximally unstable pair. The main reason to adopt this definition is that defection of a *single pair* is often sufficient to cause a cascade of

[1] Pini *et al.* [2011] develop several notions of degree of stability using real-valued preferences, and our full (but not our partial) information definition is a special case of their notion of α-stability.

deviations and unravel the matching. Furthermore, these definitions reduce to the usual notion of stability in SMPs given full preferences (which is not generally the case for other definitions, e.g., summing the pairwise max-regret of each individual). However, other definitions of instability require no substantive changes to our underlying conceptual framework or definitions (though the details and analysis of our algorithmic approach below may be sensitive to such changes).

With full information, we are assured a stable matching exists with $Inst(\mu, \succ) \leq 0$. With only a partial profile $\mathbf{P}$, the stability of any matching μ may not be known for sure. To provide guarantees on the quality of a matching, we define the *maximum regret* of μ by considering its worst-case instability over all possible preference completions.

Definition 4. *Given a PP-SMP with partial profile* $\mathbf{P}$*, the* max regret *of matching* μ *and the* minimax regret *of* $\mathbf{P}$ *are:*

$$MR(\mu, \mathbf{P}) = \max_{\succ \in C(\mathbf{P})} Inst(\mu, \succ) \quad (5)$$

$$MMR(\mathbf{P}) = \min_{\mu} MR(\mu, \mathbf{P}); \quad \mu^*_{\mathbf{P}} \in \operatorname*{argmin}_{\mu} MR(\mu, \mathbf{P}). \quad (6)$$

$MR(\mu, \mathbf{P})$ reflects the worst-case degree of instability over possible preference realizations. A *minimax optimal matching* $\mu^*_{\mathbf{P}}$ has the best such stability guarantee, namely $MMR(\mathbf{P})$: for any other matching μ there is *some* completion of preferences for which μ is at least as unstable as the maximum possible instability of $\mu^*_{\mathbf{P}}$. As such, $\mu^*_{\mathbf{P}}$ represents a *robust matching* in the face of preference uncertainty.

Our goal in solving a PP-SMP is as follows: given a partial preference profile $\mathbf{P}$, find the minimax regret optimal match $\mu^*_{\mathbf{P}}$. Notice that if $MMR(\mathbf{P}) \leq 0$, then matching $\mu^*_{\mathbf{P}}$ is in fact stable in the traditional sense, despite the fact that we have a partial profile, a fact we will exploit below to terminate the elicitation process. If the partial profile is complete, then any stable matching has max regret no greater than 0.

Example 5. In Example 1 we see that $MMR(\mathbf{P}) = 1$. Several minimax-regret optimal matchings exist: any matching that pairs each of $\{m_0, m_1\}$ with one of $\{w_0, w_1\}$, and each of $\{m_2, m_3\}$ with one of $\{w_2, w_3\}$, suffices. The stable matching for the true underlying preferences, $\mu_0(m_i) = w_i, \ \forall i$, is one such matching (though it is not "known" to be stable given only the partial profile $\mathbf{P}$).

3.2 Computing Approximately Stable Matchings

The decision variant of computing a minimax-optimal matching with partitioned preferences—and by extension, with arbitrary partial preferences—is NP-complete:

Theorem 6. *Given partitioned partial profile* $\mathbf{P}$*, deciding if some matching* $\mu^*_{\mathbf{P}}$ *has* $MR(\mu^*_{\mathbf{P}}, \mathbf{P}) < k$ *is NP-complete.*

Proof Sketch. PP-SMP is easily seen to be in NP: checking if a match has max regret k takes at most $2n^3$ pairwise max-regret calculations, which can be computed in polynomial time using techniques for finding ranking completions to determine minimax regret in Borda voting with partial votes [Lu and Boutilier, 2011b].

We use a reduction from SMP-ILT (see Sec. 2) to show PP-SMP is NP-hard. Assume an arbitrary SMP-ILT instance of size n, where k is the size of the largest "tie" (group of equally preferred partners) in some preference order. We create a corresponding instance of PP-SMP with $2n + j + 1$ individuals on each side of the market, with $j \leq n^2$. Men and women are broken into three groups, each with distinct partitioned preferences. We describe the men (women are analogous). *Group O*: the original n men in SMP-ILT; *Group I*: $n + 1$ additional men I_m, used to guarantee that no one is matched to an unacceptable partner; *Group T*: j additional men T_m, used to ensure weak stability.

Men in O_m have a partition for each of their tied groups (including singletons) in SMP-ILT, and a partition for all unacceptable partners; these are ordered as in SMP-ILT. All partitions are padded to have size k using women in T_w (arbitrarily). Finally, a partition with all women in I_w is inserted between the least-preferred acceptable and unacceptable partitions (any unused T_ws are placed in this partition as well).

Each man in I_m has the set O_w as his favorite partition. Then he has a unique woman from I_w as his next preferred, with the remaining women in I_w women ordered arbitrarily after this. Women in T_w are ordered arbitrarily below I_w.

Each man in T_m has a unique woman from T_w woman as his favorite, with the remaining women in T_w women ordered arbitrarily below this woman. An arbitrary ordering of I_w follows this, and finally all of O_w comprises a single partition at the bottom of his preference.

It is straightforward to verify that this construction of a partial preference profile $\mathbf{P}$ leads to a PP-SMP problem where a matching $\mu^*_{\mathbf{P}}$ has $MR(\mu^*_{\mathbf{P}}, \mathbf{P}) < k$ iff the original SMP-ILT instance admits a weakly stable matching (we omit details for space reasons). Since finding a weakly stable matching for SMP-ILT in NP-hard, the hardness of PP-SMP follows. ☐

Despite its computational complexity we consider several methods for both exact and heuristic solution of PP-SMP. We first observe that PP-SMP can be formulated as a polynomially sized integer program (IP). Given a matching μ and potential defecting pair (m, w), we can compute the relevant completions of their partial preferences (w.r.t. their partners $\mu(m), \mu(w)$) that maximize their instability $Inst(m, w, \mu)$ in polynomial time. Given a partial profile $\mathbf{P}$, define:

$$PMR(m, w, m', w') = \min\{PMR(m, w, w', P_m), \\ PMR(w, m, m', P_w)\}.$$

The $O(n^3)$ preference completions needed to compute these $O(n^4)$ constants can each be determined in polynomial time. With these constants in hand, we formulate the minimax-optimal matching as the IP in Algorithm 1, using 0-1 matching variables $\mu_{m,w}$.

Algorithm 1 IP for Minimax-optimal Matching

Objective: minimize δ
Subject to:
$\delta \geq PMR(m, w, m', w')(\mu_{m',w} + \mu_{m,w'} - 1), \quad \forall m, m', w, w'$
$\sum_{w \in W} \mu_{wm} = 1, \forall m \in M$
$\sum_{m \in M} \mu_{wm} = 1, \forall w \in W$
$\mu_{wm} \in \{0, 1\}, \forall w \in W, \forall m \in M$

The IP performs reasonably well for small values of n, but becomes impractical beyond $n = 30$ (see Sec. 4). Because PP-SMP is NP-complete, the LP-relaxation tends to produce highly fractional matchings, in contrast to IPs for SMP [Roth *et al.*, 1993]. Our IP works with arbitrary partial profiles; when partial preferences have structure (e.g., partitioned, top-k) we believe more compact IP formulations are possible.

Since our goal is to use minimax regret to support incremental elicitation, we need not compute minimax regret exactly to effectively reach stable matchings, as we will see below. Instead consider a set of related alternative strategies that provide us with an upper bound on minimax regret and can be computed very quickly. Given a partial profile $\mathbf{P}$, let $\succ \in C(\mathbf{P})$ be some completion. Notice that we can readily use GS to compute a stable matching μ for this complete profile $\succ$ and compute its max regret $MR(\mu, \mathbf{P})$ in $O(n^2)$ time. The max regret of the matching so produced provides us with an upper bound on $MMR(\mathbf{P})$, and specifically, if $MR(\mu, \mathbf{P})$ is no greater than zero, μ is in fact a stable matching for *any* completion of $\mathbf{P}$.

The value of this upper bound depends on the completion used to generate the matching μ. We consider several distinct ways of generating a completion, leading to different variants of the *partial preference Gale-Shapley (PPGS)* algorithm for approximating minimax regret. PPGS-R (random) chooses a completion of each user q's partial preference uniformly at random from the set $C(P_q)$. PPGS-k is the same as PPGS-R except that the process is repeated k times and the matching with minimal max regret is returned. If some probabilistic preference model is given, PPGS-ML uses this model to determine the *maximum likelihood completion* of each user's preferences and runs GS on the resulting profile. Finally, PPGS-F is much like PPGS-ML except that an arbitrary *fixed* reference ranking is used to find a maximally consistent completion.

As with exact MMR, special structure in partial preferences can be exploited to approximate MMR. For example, MMR can be bounded quickly with partitioned preferences:

Observation 7. *Given partitioned preferences with largest partition of size p, there is a matching with max regret at most* $p - 1$. *This matching can be found in polynomial time.*

By treating each partition as if they represented "ties" in a full information setting, we can compute a weakly stable matching in polynomial time [Roth *et al.*, 1993] and bound its max regret by $p - 1$. Tighter post-hoc bounds on max regret can be easily computed given the resulting matching μ. We discuss the computational effectiveness of these schemes in the next section.

4 Preference Elicitation

We now turn to interactive elicitation of preferences as a means of reducing cognitive burden. We present a general method that uses minimax regret to drive the elicitation process and empirically compare it to GS in terms of number of queries and rounds required to reach a stable (or approximately stable) matching.

4.1 Elicitation Schemes

We expect preferences in naturally occurring problems to exhibit correlations. Elicitation schemes that exploit this fact can substantially ease the cognitive burden on users, the amount of preference information revealed, and communication costs. Consider again the extreme case where all user preferences are identical (e.g., as illustrated in Example 1): stability requires that the kth-ranked man be matched with the kth-ranked woman. If GS is used for elicitation, low-ranked women (proposers) and men (acceptors) will specify their entire preference ranking, even though only the bottom of their ranking is actually needed. While half of all potential partners are ranked (on average) using GS, a user q using "binary search" can identify the kth-ranked element of R^q without reasoning about her precise ranking. Well-designed queries could further lessen the burden on users, as some queries are intrinsically easier than others.

Algorithm 2 Regret-based Halving: Elicitation

Require: Partitioned preference profile $\mathbf{P}$, threshold τ
loop
  Compute (approximate) $\mu = \mu^*(\mathbf{P})$, compute $MR(\mu, \mathbf{P})$.
  if $MMR(\mathbf{P}) \leq \tau$, done.
  for each $q \in RI(\mu)$ s.t. q not queried this round
    if r and $\mu(q)$ are in the same block B of
    P_q for some $r \in BP(q)$
      Ask q to split B.
    else
      for each $r \in BP(q)$ s.t. r not queried this round
        if q and $\mu(r)$ are in the same block B of P_r
          Ask r to split B.
  if $\nexists P_j \in \mathbf{P}$ s.t. P_j was updated this round
    for each $P_j \in \mathbf{P}$
      Ask j to split j's largest partition B^j_m s.t. $|B^j_m| > 1$.

This example, while suggestive, is unrealistic. Apart from assuming extreme correlation, it presumes the ability to identify the kth-ranked person on each side of the market. In general, knowing which parts of a user's ranking need to be refined requires some prediction of the matching that will result. However, this motivates a new elicitation method, the *regret-based halving (RBH) strategy*, outlined in Alg. 2. RBH works by maintaining a partial profile $\mathbf{P}$ in the form of partitioned preferences for each user q: i.e., a rank-ordered set of blocks $B^q_1, \ldots, B^q_k$, where partners within a block have not yet been compared to each other. At each round of elicitation, RBH asks one or more users q to *split* a single block B^q_j into an upper block U and lower block L of (roughly) equal size such that $r \succ_q r'$ for all $r \in U, r' \in L$. For instance, Fig. 1 shows the evolution of partitioned preferences discovered using RBH for one of the women in Example 1.

To identify appropriate queries at each round, RBH computes (or approximates) the minimax-optimal matching μ relative to $\mathbf{P}$. Users whose preference imprecision is "contributing" to max regret $MR(\mu)$ are queried so as to reduce imprecision in the relevant partition. More specifically, let the *regret-inducing individuals* $RI(\mu)$ be those $q \in M \cup W$ for which there is an $r \in R^q$, s.t. $PMR(q, r, \mu(q), P_q) = MR(\mu)$ and $PMR(r, q, \mu(r), P_r) \geq MR(\mu)$: intuitively, q is a user whose potential preference for new partner r (who potentially reciprocates at least as strongly) determines the max regret of the matching. We call any such r a *blocking partner* for q, and denote the set of such as $BP(q)$ (see Fig. 1 for an illustration). RBH will ask queries only of such individuals, or their potential partners, in a direct attempt to

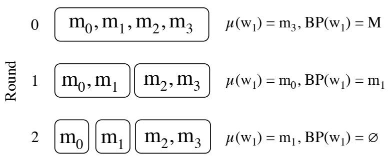

Figure 1: Evolution of preferences for w_1 in Example 1 using RBH.

reduce $MR(\mu)$; see line 4 of Alg. 2. This can be viewed as a form of *current solution* elicitation [Boutilier *et al.*, 2006; Braziunas and Boutilier, 2007; Lu and Boutilier, 2011b], where queries are driven by the regret-optimal solution.

The selection of individuals for querying occurs in lines 4–10. Intuitively, querying the regret-inducing individuals q and their blocking partners has the greatest potential to lower the max regret of the current matching. Lines 4–6 ask q to split the block containing both its current partner $\mu(q)$ and blocking $r \in BP(q)$ when they are in the same block. This will either confirm $\mu(q)$ or r as preferred by q, or will reduce their PMR by a half. If q's blocking partner r is not in the same block as $\mu(q)$, we instead query r, asking him to split the block containing q and $\mu(r)$ if they are in the same block for similar reasons (lines 8–10): note that r's PMR for q is at least as great as that of q for r. In both cases, queries are targeted toward individuals where a single block contains the "relevant" uncertainty required to identify the optimal "attainable" partner, in some loose sense simulating binary search. If neither of these conditions holds for any $q \in RI(\mu)$, we resort to asking each individual to split their largest block to ensure that elicitation does not stall.

Cognitive Costs. Asking a person to "split a block" based on preferences is a very different form of query than those asked by GS. Furthermore, not all splitting queries are equally difficult, since the sizes of the blocks that are split varies (and may contain potential partners that are either very close of very distant from each other in terms of preference). To meaningfully compare RBH with GS, we measure the number and cognitive difficulty of the pairwise comparisons required to answer their queries. We show that, in practice, RBH outperforms GS with respect to the total *cognitive complexity* of queries, as well as the number of queries itself.

One natural measure for comparing how difficult queries are would be to simply consider the number of pairwise comparisons required. Using a Quickselect-style algorithm [Hoare, 1961], a block of size z can be split using $O(z)$ comparisons (assuming reasonable pivot choices). For GS, proposers can select the next best partner to propose to using $O(z)$ comparisons when selecting from z unproposed candidates; and acceptors similarly must make comparisons linear in the number of received proposals. Of course, not all pairwise comparisons are equally difficult. Intuitively, comparing two partners widely separated in one's ranking is easier than comparing two that are close, a fact reflected in many psychometric and behavioral economics models of choice [Louviere *et al.*, 2000; Camerer *et al.*, 2003]. To reflect this, we measure the difficulty of a comparison using the *Luce-Shepard* choice model [Luce, 1959; Shepard, 1959], in which the probability of choosing a lower ranked item over a higher ranked item decreases exponentially with their *separation* or difference in utility. Equating degree of difficulty with choice error, given an underlying ranking $\succ_q$ for person q, temperature $\gamma \geq 0$, and threshold τ, the cost of comparing r with r' is:

ϕ		Avg. Queries per Person		# Rounds
		Proposers	Acceptors	
0.2	GS-elicit	10.07 (0.16)	3.33 (0.35)	23.7 (1.7)
	IP	8.82 (3.03)	8.80 (3.03)	17.8 (6.7)
	PPGS-R	4.25 (0.27)	4.24 (0.21)	14.2 (2.7)
	PPGS-k	4.09 (0.22)	4.12 (0.19)	11.6 (2.4)
	PPGS-F	4.08 (0.20)	4.09 (0.20)	13.8 (3.3)
	PPGS-ML	3.95 (0.18)	3.97 (0.15)	7.7 (1.7)
1.0	GS-elicit	3.15 (0.95)	1.97 (0.97)	24.2 (18.3)
	IP	7.26 (2.02)	7.31 (2.11)	17.5 (4.8)
	PPGS-R	4.06 (0.33)	3.76 (0.37)	21.1 (4.7)
	PPGS-k	3.84 (0.30)	3.38 (0.37)	15.3 (4.3)
	PPGS-F	3.84 (0.36)	3.50 (0.37)	17.2 (4.9)
	PPGS-ML	3.84 (0.32)	3.42 (0.33)	17.6 (4.6)

Table 1: Average number of queries (std. dev.) until max regret 0, Mallows models: $n = 20$, 30 trials.

$$c(r, r') = e^{\gamma(n - \min(|s_q(r, \succ_r) - s_q(r', \succ_r)|, \tau))} \tag{7}$$

(Note: $\gamma = 0$ makes all comparisons equally difficult.) We assess the cognitive complexity of both RBH and GS below.

4.2 Empirical Results

We now describe experiments that test the effectiveness of our elicitation schemes. We draw preference profiles from specific preference distributions, and measure the average number of queries and rounds needed to reach a stable matching (i.e., max regret zero), the cognitive complexity of those queries, and their anytime performance w.r.t. *approximate stability*. We also assess their computational performance.

Number of queries. We first consider small instances with $n = 20$ (i.e., 20 men, 20 women) to allow use of the IP (Alg. 1) to compute true minimax regret. Table 1 shows the average number of queries (over 30 trials) per proposer/acceptor and number of rounds required by different schemes using a Mallows models to generate preferences with $\phi = 1$ (impartial culture) and $\phi = 0.2$ (strongly correlated preferences). GS marginally outperforms all RBH schemes when preferences are completely uncorrelated, but except for IP, the differences are not statistically significant for proposers, and the advantage for acceptors is quite small. With strongly correlated preferences, the RBH schemes (apart from IP) reach a stable matching with *far fewer* queries per proposer, and differ from GS by less than one query per acceptor. In all cases, the RBH schemes (excluding IP) ask fewer than $\log n$ queries per person; thus they even outperform binary search for a *known* target partner (of course, the target is not known *a priori*). The IP scheme works surprisingly poor. This is, in large part, due to the fact that the matchings computed by the IP often do not match individuals to partners in the partitions that they ultimately end up being matched in.

For these small instances, the IP requires roughly 0.5 sec. to compute the MMR matching (and hence generate queries in a given round), but becomes impractical beyond $n = 30$. We do not consider IP further on the larger instances below. The other RBH schemes require 0.01–0.06 sec. on average to generate a matching, compute its max regret, and generate queries. GS is much faster, but does not compute max regret; thus it offers no anytime guarantees (as we discuss below).

ϕ		Avg Queries per Person		# Rounds
		Proposers	Acceptors	
0.2	GS-elicit	125 (0.04)	11.49 (0.45)	273 (5.7)
	PPGS-R	8.62 (0.12)	8.63 (0.14)	258 (28.2)
	PPGS-k	8.28 (0.11)	8.27 (0.14)	141 (13.9)
	PPGS-F	7.92 (0.11)	7.92 (0.10)	93.1 (9.8)
	PPGS-ML	7.49 (0.06)	7.49 (0.06)	31.8 (6.7)
1.0	GS-elicit	5.76 (1.12)	4.52 (1.10)	370 (207)
	PPGS-R	8.07 (0.16)	9.26 (0.34)	700 (44.1)
	PPGS-k	7.62 (0.13)	7.70 (0.29)	393 (46.9)
	PPGS-F	7.30 (0.18)	6.40 (0.15)	342 (28.3)
	PPGS-ML	7.28 (0.19)	6.36 (0.13)	331 (26.9)

Table 2: Average number of queries (std. dev.) until max regret 0, Mallows models: $n = 250$, 30 trials.

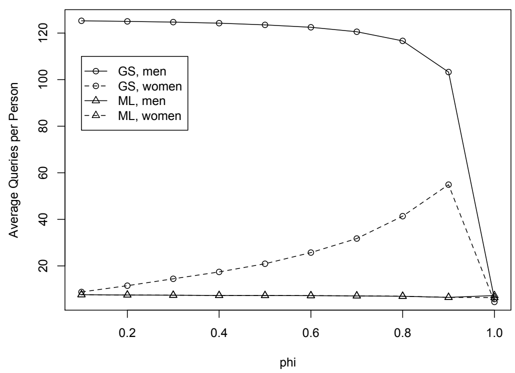

Figure 2: Average number of queries per person: $n = 250$, 20 trials, Solid: proposers, Dashed: acceptors; Circles: GS-elicit, Triangles: RBH-PPGS-ML.

Table 2 shows similar results for $n = 250$, and Fig. 2 shows the performance of PPGS-F and GS as ϕ varies. On these much larger matching problems, PPGS-F and PPGS-ML offer superior performance (both averaging fewer than $\log(n)$ queries per person). With correlated preferences ($\phi = 0.2$) all RBH schemes dramatically outperform GS. Notice that GS is also unbalanced or "unfair," asking proposers 10 times as many queries as acceptors. PPGS-ML has slightly (but statistically significant) better query performance than PPGS-F. It offers much better performance in terms of rounds—an important measure of latency in interactive elicitation—and uses only 31.8 rounds (cf. 272.8 for GS). With completely random preferences, GS has a query advantage over the other schemes, but it is slight (about 1.5 queries for proposers, under 2 for acceptors), and also has much higher variance. GS still requires more rounds than PPGS-ML in this case.

Anytime performance. Computationally, GS takes negligible time per round to compute queries, while PPGS-ML and PPGS-F take approximately 27s. per round when $n = 250$. Of course, the RBH schemes must compute max regret, specifically PMR for alternative partners for each person. However, since the PMR values for person q are independent of those for q', these computations can be fully parallelized. Computing preference completions for each q is the most time-consuming aspect of RBH, so full parallelization can reduce real-time latency by a factor of roughly $2n$. More importantly, computing the max regret of the current matching allows one to determine *approximately stable matchings* at any time and terminate the querying process when max regret reaches an acceptable level. We compare this *anytime performance* of PPGS-ML and GS in Fig. 3.[2]

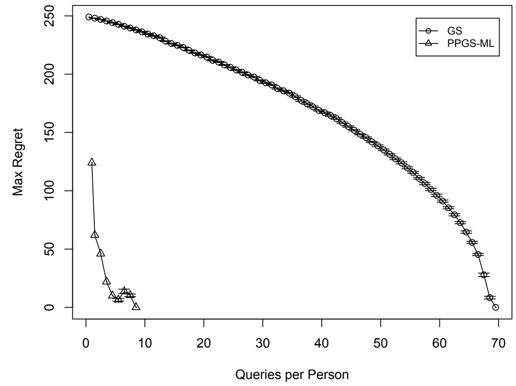

Figure 3: Max regret vs. number of queries (per person): $n = 250$, $\phi = 0.2$, 20 trials, with error bars (low std. dev.).

PPGS-ML displays very desirable behavior, on average roughly halving max regret with each (per person) query, inducing rapid convergence to a stable matching. Critically, it admits early termination of elicitation with high quality matchings. By contrast, GS reduces max regret at an extremely low rate, providing almost no ability to quickly find matchings with low max regret. The ability to determine approximately stable matchings so quickly is a key reason to use our regret-based approaches to elicitation.

Cognitive costs. Simply measuring the number of queries does not consider how *difficult* each query is, which may be misleading. We thus analyze the *cognitive complexity* of each query (see Eq. 7). We set $\gamma = 0.5$ and $\tau = 5$ (and normalize costs by $e^{\gamma n}$ for clarity); results are qualitatively similar for other cost parameters. In the ($n = 250, \phi = 0.2$)-instances above, the cognitive cost of full sorting is 310.34 (per person), assuming Quicksort with perfect pivots. Proposers average a cognitive cost of 1,830.89 using GS, so we instead assume that proposers fully (and perfectly efficiently) sort their lists ahead of time when using GS, giving a cost of 310.34. (This is itself unrealistic, since using GS as elicitation is unlikely to result in proposers sorting their entire list, so true cognitive cost will exceed 310 but fall short of 1830.) Acceptors' average cost for GS is only 13.08, since they make fewer decisions and their comparisons are often widely separated. PPGS-ML's results are much more egalitarian, with proposers and acceptors having cognitive costs of 53.80 and 53.77, respectively. Its average cost of 53.79 is much lower than that of GS—even with *perfect sorting by proposers*—which averages 161.71 per person. Even with $\phi = 1$, while GS asks slightly fewer queries per person, its average cognitive cost is 60.8 (men 121.10, women 0.43) remains greater than the 57.8 for PPGS-ML (54.99 proposers, 57.68 acceptors). This is a compelling reason to use regret-based elicitation.

Riffle models. We now consider a more complicated *riffle model* of preferences. This model makes the realistic assumption that certain (either latent or observable) features correlate user preferences. We assume that every man is one of two

[2]Note that GS does not compute intermediate matchings at each round, so we use the PPGS-ML scheme to find a matching using the information elicited by the GS queries. Other RBH schemes are excluded, but perform similarly to PPGS-ML.

ϕ		Avg Queries per Person: Proposers	Acceptors	# Rounds
0.0	GS-elicit	86.79 (1.32)	33.61 (1.32)	324.9 (39.3)
	PPGS-F	8.57 (0.16)	8.57 (0.21)	232.4 (21.2)
0.1	GS-elicit	86.77 (1.68)	34.46 (1.55)	333.0 (31.0)
	PPGS-F	8.44 (0.26)	8.46 (0.29)	227.0 (22.1)
0.2	GS-elicit	86.37 (1.46)	33.98 (1.38)	312.0 (29.4)
	PPGS-F	8.41 (0.13)	8.41 (0.13)	237.1 (22.9)
1.0	GS-elicit	5.96 (0.99)	4.72 (0.97)	408.4 (172.7)
	PPGS-F	7.41 (0.26)	6.42 (0.21)	338.1 (47.6)

Table 3: Average number of queries (std. dev.) until max regret 0, Gaussian riffle model: $n = 250$, 20 trials.

"types" corresponding to some (for ease of exposition) observable characteristic. For each type, we assume a distinct Mallows model over the men of each type. A woman's preference for men is generated by first drawing a ranking from each of these models, then interleaving them using a riffle process, with a parameter p reflecting the woman's bias toward each type: the full ranking is generated by iteratively placing the top "remaining" item from the Type 1 ranking into the next spot in her full ranking with probability p, and the top item from the Type 2 ranking with probability $1 - p$, until all men have been inserted into the ordering. The bias p towards one type of the other is drawn from an equal-weight mixture of two Gaussians (truncated over $[0, 1]$) with variance $\sigma = 0.1$ and means of 0.25 and 0.75 (so women are unlikely to be "indifferent" between the two types of men). Men's preferences are generated in the same way.

We compare PPGS-F to GS in terms of number of queries to reach a stable matching in this model, using three different dispersion parameters for the underlying Mallows models. (We don't use PPGS-ML since maximum likelihood estimation is less straightforward in this mixture process.) Results in Table 3 show that PPGS-F vastly outperforms GS, when $\phi = 0, 0.1, 0.2$; and it remains competitive with GS when $\phi = 1$. It always requires fewer rounds as well.

MovieLens Models. We next consider matching problems in which preferences are generated from the MovieLens collaborative filtering data set.[3] The MovieLens data set consists of 100,000 ordinal ratings (1–5 scale) of 1682 movies by 943 users. We convert this into preference rankings of users for each other by generating an *affinity score* between pairs of users based on the similarity of their movie ratings.

Let $M(a)$ denoted the set of movies rated by user a, and r_a her rating vector. Given two users a and b, we define their affinity score to be $s(a, b) = \sum_{m \in M(a) \cap M(b)} 5 - |r_a(m) - r_b(m)|$, where 5 reflects the maximum rating. This scores creates very correlated preferences: individuals who rate a larger number of movies will tend to be viewed as more desirable across the population. We also create somewhat less correlated affinities by normalizing scores by the number of movies rated in common: $s^N(a, b) = s(a, b)/|M(a) \cap M(b)|$. With these affinities, we create random matching problems by drawing 250 "men" and 250 "women" uniformly at random, and generating preference rankings for the appropriate side of the market using these real-valued affinities to order potential partners (breaking ties arbitrarily). As expected, the Unnormalized Movie Matching (U-MM) problems using score s exhibit more correlation than their unnormalized counterparts (N-MM) generated using s^N.[4]

Data Set		Avg Queries per Person: Proposers	Acceptors	# Rounds
U-MM	GS-elicit	43.52 (2.07)	26.62 (1.25)	324.9 (39.3)
	PPGS-F	6.36 (0.12)	6.37 (0.14)	249.7 (22.9)
N-MM	GS-elicit	13.23 (1.09)	10.07 (0.99)	255.5 (47.8)
	PPGS-F	8.11 (0.13)	8.09 (0.15)	349.6 (22.3)

Table 4: Average number of queries (std. dev.) until max regret 0, MovieLens Matching Models: $n = 250$, 20 trials.

Table 4 shows the performance of GS and PPGS-F for both U-MM and N-MM, averaged over 20 random matching instances. PPGS-F outperforms GS with respect to number of queries in both settings, but requires more rounds in N-MM. Not surprisingly the performance gap is greater in when preferences are more correlated. PPGS-F similarly outperforms GS with respect to cognitive cost (setting $\gamma = 0.5, \tau = 5$ as above). In U-MM, average cognitive cost per person for GS is 788.52 for proposers, and 3.56 for acceptors. Using PPGS-F, average costs are 55.31 and 55.40 for proposers and acceptors, respectively. Again, note that the cost of fully sorting all alternatives is 310.34 per person. PPGS-F also outperform GS on N-MM problems: GS has an average cognitive cost of 250.04 and 1.35 for proposers and acceptors, respectively, while PPGS-F has costs of 58.84 and 58.59, respectively.

5 Conclusions and Future Work

We have proposed the use of minimax regret as a robustness criterion for stable matching with incomplete information, and developed several heuristic elicitation schemes designed to quickly reduce regret. These schemes compare favorably to the use of Gale-Shapley for interactive elicitation, especially when preferences exhibit some correlation: they reach stable matching with fewer queries, fewer rounds of elicitation, and with lower cognitive cost. Critically for domains where approximate stability is desirable (e.g., when elicitation/interviewing costs or switching/defection costs are high), our elicitation schemes demonstrate vastly superior anytime performance to GS, allowing approximately stable matchings to be found with very little preference information.

Many interesting research directions remain, including: improved procedures for exact MMR computation; new algorithms tuned to specific forms of partial preferences; analysis of additional probabilistic preference models and the use of priors to further improve elicitation performance; the assessment of our methods for different measures of approximate stability; and the extension to other stable matching problems.

Acknowledgments. Drummond was supported by OGS and a Microsoft Research Graduate Women's Scholarship. We acknowledge the support of NSERC, and thank Tyler Lu for generously sharing his code and the reviewers for their helpful suggestions.

[3]See http://www.grouplens.org/node/73, the 100K data set.

[4]Using Kendall's τ-statistic, the average correlation statistics for U-MM are 0.4539 (proposers) and 0.4653 (acceptors); while for N-MM, they are 0.1462 (proposer) and 0.1184 (acceptors).

References

[Abdulkadiroglu *et al.*, 2005] A. Abdulkadiroglu, P.A. Pathak, A. E. Roth, and T. Sönmez. The boston public school match. *American Economic Review*, 95(2):368–371, 2005.

[Biró and Norman, 2012] P. Biró and Gethin Norman. Analysis of stochastic matching markets. *Intl. Journal of Game Theory*, pp.1–20, 2012.

[Boutilier *et al.*, 2006] C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artifical Intelligence*, 170(8–9):686–713, 2006.

[Braziunas and Boutilier, 2007] D. Braziunas and C. Boutilier. Minimax regret-based elicitation of generalized additive utilities. In *Proc. 23rd Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pp.25–32, Vancouver, 2007.

[Camerer *et al.*, 2003] C. F. Camerer, G. Loewenstein, and M. Rabin (eds.). *Advances in Behavioral Economics*. Princeton University Press, Princeton, New Jersey, 2003.

[Chou and Lu, 2010] J. Chou and C. Lu. Communication requirements for stable marriages. In *Proc. 7th Intl. Conference on Algorithms and Complexity (CIAC-10)*, pp.371–382, Rome, 2010.

[Gale and Shapley, 1962] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.

[Gelain *et al.*, 2010] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Local search for stable marriage problems with ties and incomplete lists. In *Proc. 11th Pacific Rim Intl. Conference on Trends in Artificial Intelligence (PRICAI-10)*, pp.64–75, Daegu, Korea, 2010.

[Gent and Prosser, 2002] I. P. Gent and P. Prosser. An empirical study of the stable marriage problem with ties and incomplete lists. In *Proc. 15th European Conference on Artificial Intelligence (ECAI-02)*, pp.141–145, Lyon, 2002.

[Hoare, 1961] C. A. R. Hoare. FIND (algorithm 65). *Communications of the ACM*, 4:321–322, 1961.

[Huang and Guestrin, 2009] J. Huang and C. Guestrin. Riffled independence for ranked data. In *Advances in Neural Information Processing Systems 21*, pp.799–807, Vancouver, 2009.

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

[Iwama *et al.*, 1999] K. Iwama, D. Manlove, S. Miyazaki, and Y. Morita. Stable marriage with incomplete lists and ties. *Proc. 26th Intl. Colloq. on Automata, Languages and Programming (ICALP)*, pp.443–452, Prague, 1999.

[Lebanon and Mao, 2008] G. Lebanon and Y. Mao. Non-parametric modeling of partially ranked data. *Journal of Machine Learning Research*, 9:2401–2429, 2008.

[Liu *et al.*, 2012] Q. Liu, George J. Mailath, A. Postewaite, and L. Samuelson. Matching with incomplete information. Penn Institute for Economic Research WP-12-032, 2012.

[Louviere *et al.*, 2000] J. J. Louviere, D. A. Hensher, and J. D. Swait. *Stated Choice Methods: Analysis and Application*. Cambridge University Press, Cambridge, 2000.

[Lu and Boutilier, 2011a] T. Lu and C. Boutilier. Learning Mallows models with pairwise preferences. In *Proc. 28th Intl. Conference on Machine Learning (ICML-11)*, pp.145–152, Bellevue, WA, 2011.

[Lu and Boutilier, 2011b] T. Lu and C. Boutilier. Robust approximation and incremental elicitation in voting protocols. In *Proc. 22nd Intl. Joint Conference on Artificial Intelligence (IJCAI-11)*, pp.287–293, Barcelona, 2011.

[Luce, 1959] R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

[Mallows, 1957] C. L. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.

[Marden, 1995] J. I. Marden. *Analyzing and Modeling Rank Data*. Chapman and Hall, London, 1995.

[Meila and Chen, 2010] M. Meila and H. Chen. Dirichlet process mixtures of generalized Mallows models. In *Proc. 26th Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pp.358–367. Catalina, 2010.

[Murphy and Martin, 2003] T. B. Murphy and D. Martin. Mixtures of distance-based models for ranking data. *Computational Stats. and Data Analysis*, 41:645–655, 2003.

[Niederle *et al.*, 2008] M. Niederle, A. E. Roth, and T. Sonmez. Matching and market design. In Steven N. Durlauf and Lawrence E. Blume, editors, *The New Palgrave Dictionary of Economics (2nd Ed.)*, volume 5, pp.436–445. Palgrave Macmillan, Cambridge, 2008.

[Pini *et al.*, 2011] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Stability in matching problems with weighted preferences. In *Proc. 3rd Intl. Conference on Agents and Artificial Intelligence (ICAART 2011)*, 2011.

[Rastegari *et al.*, 2013] B. Rastegari, A. Condon, N. Immorlica, and K. Leyton-Brown. Two-sided matching with partial information. In *Proc. 14th ACM Conference on Electronic Commerce (EC'13)*, to appear, Philadelphia, 2013.

[Roth *et al.*, 1993] A. E. Roth, U. G. Rothblum, and J. H. Vande Vate. Stable matchings, optimal assignments, and linear programming. *Mathematics of Operations Research*, 18(4):803–828, 1993.

[Roth, 1984] A. E. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *J. Political Economy*, 92(6):991–1016, 1984.

[Shepard, 1959] R. N. Shepard. Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22(4):325–345, 1959.