Fast distributed almost stable marriages

Rafail Ostrovsky∗ Will Rosenbaum†

August 13, 2014

Abstract

In their seminal work on the Stable Marriage Problem, Gale and Shapley [3] describe an algorithm which finds a stable marriage in \(O(n^2)\) communication rounds. Their algorithm has a natural interpretation as a distributed algorithm where each player is represented by a single processor. The complexity measure of such a distributed algorithm is typically measured by the round complexity, assuming all processors can communicate simultaneously in each round, or in terms of synchronous running time.

Recently, Floréen, Kaski, Polishchuk and Suomela [2] showed that in the special case of bounded preference lists, terminating the Gale-Shapley algorithm after a constant number of rounds results in an almost stable (partial) marriage. Lifting such an approximation to unbounded preference lists remained open.

In this paper, we describe a new distributed algorithm which computes an almost stable marriage in \(O(1)\) communication rounds for unbounded preference lists, so long as the ratio of the lengths of longest to shortest preference lists is bounded by a constant. The synchronous run-time of our algorithm is \(O(n)\) for complete preference lists. To our knowledge, this is the first sub-polynomial round and sub-quadratic time distributed algorithm for any variant of the stable marriage problem with complete preferences.

∗University of California, Los Angeles (Department of Computer Science and Mathematics). Work supported in part by NSF grants 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

†University of California, Los Angeles (Department of Mathematics).
1 Introduction

In their seminal work, Gale and Shapley [3] consider the following problem. A group of $n$ men and $n$ women each rank all of the members of the opposite sex. The men and women (which we collectively call players) wish to form a marriage—a one-to-one correspondence between the men and women—which is stable in the sense that no pair of players mutually prefer each other to their assigned partners in the marriage. Gale and Shapley showed that a stable marriage always exists by giving an explicit algorithm for finding one. The centralized Gale-Shapley algorithm runs in time $O(n^2)$. This run-time is asymptotically optimal for centralized algorithms, as even specifying preferences for all men and women requires storing $2n$ ordered lists of $n$ names.

The Gale-Shapley algorithm has a natural interpretation as a distributed algorithm, where each player is represented by a separate processor which privately holds that player’s preferences. In this case, the input to each processor has size $O(n)$, yet there is still no known distributed algorithm which improves upon the Gale-Shapley algorithm’s $O(n^2)$ run-time for arbitrary preferences in this distributed model. It is known, however, that the Gale-Shapley algorithm terminates after an expected $O(n \log n)$ proposals for inputs chosen uniformly at random (see [10]).

Recently, there has been interest in approximate versions of the stable marriage problem [1, 2, 5, 7], where the goal is to find a marriage which is “almost stable.” There is no consensus in the literature on precisely how to measure almost stability, but typically almost stable matchings have relatively few blocking pairs. Floréen, Kaski, Polishchuk and Suomela (FKPS) show [2] that for bounded preference lists, truncating the Gale-Shapley algorithm after boundedly many communication rounds yields an almost stable matching. More recently, Hassidim, Mansour and Vardi [5] show a similar result in a more restrictive “local” computational model, so long as the men’s preferences are chosen uniformly at random.

Kipnis and Patt-Shamir [7] give an algorithm which finds an almost stable marriage using $O(n)$ communication rounds in the worst case, using a finer notion of approximate stability than we consider. They also prove an $\Omega(\sqrt{n}/\log n)$ communication round lower bound for finding an approximate stable marriage (again for their more restrictive notion of approximation).

Here, we describe a distributed algorithm which finds an almost stable marriage in a constant number of communication rounds for (potentially) unbounded preference lists. In particular, our algorithm works for complete preferences.

**Theorem 1.1** (ASM algorithm guarantee). Let $\varepsilon, \delta > 0$ be fixed and let $P$ be any preferences such that the ratio of the length of longest to shortest preference lists is bounded by a constant. Denote the length of the longest preference list by $d$. Then there exists a distributed algorithm, ASM, which finds a marriage $M$ that is $(1 - \varepsilon)$-stable with probability at least $1 - \delta$ in $O(1)$ communication rounds. The synchronous run-time of ASM is $O(d)$.

Theorem 1.1 extends the result of FKPS [2] to a class of unbounded preferences, with the caveat that our notion of $(1 - \varepsilon)$-stability is coarser than that given in [2]. However, for bounded preferences (the context of [2]) our notion of almost stability agrees with FKPS’s notion up to a constant factor. It is interesting to note that since we use a coarser notion of approximation than Kipnis and Patt-Shamir [7], our algorithm outperforms the $\Omega(\sqrt{n}/\log n)$ round lower bound for their variant of the almost stable marriage problem.

We believe our proof of Theorem 1.1 contains several conceptual innovations which may be of independent interest. The ASM algorithm generalizes the classical Gale-Shapley algorithm in that it allows for multiple simultaneous proposals by the men and acceptances by the women.

---

1In the distributed computational model with complete preferences, each player can broadcast their preferences to all other players in $O(n)$ rounds, after which each player runs a centralized version of the Gale-Shapley algorithm. While this process requires only $O(n)$ communication rounds, the synchronous distributed run-time is still $O(n^2)$ in the worst case.
The men propose and women accept/reject proposals in batches by quantizing their preferences. A marriage is then constructed by finding large matchings among the accepted proposals. We believe this approach for the (almost) stable marriage problem is novel.

In our analysis of ASM we introduce a metric structure on the set of preferences for the stable marriage problem. This metric has the property that given preferences \( P \) and an almost stable marriage \( M \), \( M \) is also almost stable for preferences \( P' \) which are “close” to \( P \). Using this metric, we show that the marriage produced by ASM is almost stable. Specifically, we show that the sequence of proposals, acceptances and rejections produced by ASM is consistent with an execution of the classical Gale-Shapley on (different) preferences which are close to the original preferences for ASM. Thus we achieve the desired approximation guarantee by comparing an execution of our approximate algorithm to an execution of an exact algorithm on a related input. To our knowledge, this technique has not previously been applied to approximately stable matching algorithms.

The organization of the remainder of the paper is as follows. In Section 2, we develop the background necessary to describe the ASM algorithm. In Section 3 we explicitly describe the algorithm ASM and its subroutines. Section 4 contains proofs of the performance guarantees of ASM. Specifically, Theorem 1.1 is an immediate consequence of the run-time guarantee (Theorem 4.1) and approximation guarantee (Theorem 4.3). Finally, in Section 5 we discuss extensions and open problems related to our results.

2 Preliminaries

2.1 The stable marriage problem

We consider the stable marriage problem as originally described by Gale and Shapley with unacceptable partners (cf. [4, 8]). Let \( X \) and \( Y \) be sets of women and men, respectively. For simplicity, we assume \( |X| = |Y| = n \). Each player \( v \in X \cup Y \) holds a preference list or ranking \( P^v \)—a linear order on a subset of the members of the opposite sex. Denote the set of all player’s preferences by \( \mathcal{P} = \{P^v|v \in X \cup Y\} \). We refer to the players \( u \) that appear on \( v \)'s preference list \( P^v \) as \( v \)'s acceptable partners. We call \( \mathcal{P} \) a set of complete if each player ranks all players of the opposite sex. If a man \( m \) precedes \( m' \) on woman \( w \)'s preference list, we write \( m \succ_w m' \), and we say that \( w \) prefers \( m \) to \( m' \). We assume that preferences are symmetric in the sense that if \( m \) appears in \( P^w \), then \( w \) appears in \( P^m \). We define the communication graph \( G = (V, E) \) for a set of preferences \( \mathcal{P} \) to be

\[ V = Y \times X, \quad E = \{ (m, w) | m \in P^w, \ w \in P^m \} \].

A marriage \( M \subseteq E \) is a matching on \( G \)—i.e. a set of edges in \( E \) such that no two edges share a vertex. Given a marriage \( M \) and \( (m, w) \in M \), we call \( m \) and \( w \) partners and write \( p(w) = m \) and \( p(m) = w \). Given preferences and a marriage \( M \), we say that an edge \( (m, w) \notin E \) is a blocking pair if \( (m, w) \notin M \), but \( m \) and \( w \) mutually prefer each other to their partners in \( M \); that is,

\[ m \succ_w p(w) \quad \text{and} \quad w \succ_m p(m) \].

By convention, we assume each unmatched player \( p(v) = \emptyset \) prefers all acceptable partners to being without a partner. A marriage which contains no blocking pairs is stable.

For a communication graph \( G = (V, E) \), we denote the degree of \( v \in V \) by \( \deg v \), which is the number of players that appear on \( v \)'s preference list. We take the parameter \( C \) to be an upper bound for the ratio of maximum to minimum degree vertices \( v \in V \):

\[ C \geq \frac{\max \deg G}{\min \deg G} \].

For the stable marriage problem, \( C \) is at least the ratio of the length of longest to shortest preferences lists. In particular, in the case of complete preferences we may take \( C = 1 \).
2.2 Almost stable marriages

We are primarily concerned with finding marriages which are “almost stable” in the sense that they induce relatively few blocking pairs. We give a definition of almost stability given by Eriksson and Håggström [1], modified to allow for incomplete preference lists.

**Definition 2.1.** Given $\varepsilon \geq 0$, we say that a marriage $M$ is $(1 - \varepsilon)$-stable with respect to preferences $P$ if $M$ induces at most $\varepsilon |E|$ blocking pairs with respect to $P$.

We refer to the problem of finding a marriage which is $(1 - \varepsilon)$-stable for fixed $\varepsilon > 0$ as the **almost stable marriage problem**. We remark that for $\varepsilon = 0$, a 1-stable marriage corresponds precisely to the classical stable marriage definition.

**Remark 2.2.** Again, we reiterate that there is no consensus in the literature on the precise definition of almost stability. For example, FKPS [2] compare the number of blocking pairs to $|M|$, the size of the marriage rather than $|E|$, as we do. Since FKPS only consider bounded preference lists, their notion and our notion of almost stability agree up to a constant factor.

**Remark 2.3.** Kipnis and Patt-Shamir [7] define a pair $(m, w)$ to be $\varepsilon$-blocking if they rank each other an $\varepsilon$-fraction better than their assigned partners. A matching is then almost stable if it does not contain any $\varepsilon$-blocking pairs. Using this definition, Kipnis and Patt-Shamir prove an $\Omega(\sqrt{n}/\log n)$ round lower bound for their version of the almost stable marriage problem. That we are able to achieve an $O(1)$ round algorithm for the almost stable marriage problem using Definition 2.1 bolsters the use of Definition 2.1 for almost stability.

2.3 Computational model

We describe our algorithm in terms of the CONGEST model described by Peleg [9]. In this distributed computational model, each player $v \in X \cup Y$ represents a processor. Given preferences $P$, the communication links between the players are given by the set of edges $E$ in the communication graph $G$. Communication is performed in synchronous rounds. Each communication round occurs in three stages. First, each processor receives messages (if any) sent from its neighbors in $G$ during the previous round. Next, each processor performs local calculations based on its internal state and any received messages. We make no restrictions on the complexity of local computations. Finally, each processor sends short ($O(\log n)$ bit) messages to its neighbors in $G$—the processor may send distinct messages to distinct neighbors. In the CONGEST model, we are exclusively concerned with the number of communication rounds needed to solve a problem—i.e. the **round complexity** of an algorithm.

Since the CONGEST model makes computational assumptions which are unrealistic in practice (for example, each processor is computationally unbounded), we also consider the run-time of ASM with respect to more innocuous computational assumptions. In analyzing the **run-time** of ASM, we shall assume that each processor can perform the following operations in constant time:

1. basic integer arithmetic
2. choose a log $n$-bit integer uniformly at random from a specified range
3. send/receive a single message to/from one neighbor consisting of (a) a short message (e.g., PROPOSE, ACCEPT, REJECT), or (b) the id of a player
4. query a player’s own preferences, specifically “Which player do I rank in position $i$?” and “What is my rank of player $v$?”

Even with respect to these simple operations, the ASM algorithm will be shown to run in linear time in the length of each player’s preference list.
2.4 Almost maximal matchings

In our algorithm, we require a subroutine which finds an “almost maximal matching” in a communication graph. Here we make precise a notion almost maximal matching.

Let \( G = (V, E) \) be a communication graph. A matching \( M \) is a \textit{maximal matching} if it is not properly contained in any larger matching. Equivalently, \( M \) is maximal if and only if every \( v \in V \) satisfies precisely one of the following conditions:

1. there exists a unique \( w \in V \) with \( (v, w) \in M \)
2. for all \( w \in N(v) \) there exists \( v' \in V \) with \( v' \neq v \) such that \( (v', w) \in M \).

We define an “almost maximal” matching to be a matching in which a large fraction of the vertices in \( G \) satisfy one of the above conditions.

**Definition 2.4.** Let \( G = (V, E) \) be a communication graph and \( M \subset E \) a matching in \( G \). For \( 0 < \eta \leq 1 \), we say that \( M \) is \((1-\eta)\)-maximal if the set \( V' \) of vertices not satisfying conditions 1 or 2 above satisfies \( |V'| \leq \eta |V| \).

In 1986, Israeli and Itai [6] described a distributed algorithm which finds a maximal matching in a graph using \( O(\log n) \) communication rounds (where \( |V| = n \)). By truncating Israeli and Itai’s algorithm after a bounded number of steps, we obtain the following theorem. Details of the proof are given in Appendix A.

**Theorem 2.5.** Let \( G \) be a communication graph and \( 0 < \delta, \eta < 1 \). Then there exists a distributed algorithm \( \text{AMM}(G, \delta, \eta) \) which, with probability at least \( 1-\delta \), finds a \((1-\eta)\)-maximal matching in \( G \) using \( O(-\log(\delta\eta)) \) communication rounds. The run-time of this algorithm is \( O(-\log(\delta\eta) \max \deg(G)) \).

**Definition 2.6.** We call a node \( v \in V \) \textbf{unmatched} if \( v \) does not satisfy property 1 or 2 preceding definition 2.4 in the matching found by \( \text{AMM}(G, \delta, \eta) \).

By Theorem 2.5, \( \text{AMM}(G, \delta, \eta) \) induces at most \( \eta |V| \) unmatched players with probability at least \( 1-\delta \).

3 Algorithm description

In this section, we describe in detail the almost stable marriage algorithm, \( \text{ASM} \). We break the main algorithm into subroutines, \( \text{MarriageRound} \) and \( \text{GreedyMatch} \). In Section 3.1 we introduce notation, and describe the internal state of each processor during the execution of the \( \text{ASM} \) algorithm. Section 3.2 contains a description of the main subroutine used in \( \text{ASM} \): \( \text{GreedyMatch} \). Finally, Section 3.3 contains a description of \( \text{ASM} \) and its \( \text{MarriageRound} \) subroutine.

3.1 Notation

In our algorithm, we assume that each player is represented by an independent processor. Each processor has a unique id and a gender (male or female) both of which are known to that processor. At each step of the algorithm, we specify the state of each processor as well as any messages the processor might send or receive. The state of a player \( v \) consists of:

- Quantized preferences \( Q_1, Q_2, \ldots, Q_k \) where we denote \( Q = \bigcup Q_i \). Initially \( Q_1 \) is the set of \( v \)'s \( \deg(v)/k \) favorite men, \( Q_2 \) is her next favorite \( \deg(v)/k \), and so on. We call \( Q_i \) \( v \)'s \( i \)th quantile. For \( m \in Q_i \), we write \( q(m) = i \). If we wish to make explicit the player to whom the preferences belong, we may adorn these symbols with a superscript. For example, \( Q_i^v \) is \( v \)'s \( i \)th quantile. Throughout the execution of the algorithm, elements may be removed from \( Q \) and the \( Q_i \)'s, but elements will never be added to any of these sets.
A partner $p$ (possibly empty). The partner $p$ is $v$'s current partner in the matching $M$ our algorithm constructs. To emphasize that $p$ is player $v$'s partner, we will write $p(v)$. The (partial) matching $M$ produced by the algorithm at any step is given by $M = \{(p(w), w) \mid w \in X, p(w) \neq \emptyset\}$

Additionally, subroutines of our algorithm will require each processor to store the following variables:

- A set $G_0$ of “neighbors” of the opposite sex which correspond to accepted proposals.
- A partner $p_0$ in a matching found in the graph determined by $G_0$.

Thus each player knows their preferences, partners (if any) as well as any accepted proposals from the current round (stored in $G_0$). The men $m \in Y$ hold the following additional information:

- As set $A$ of “active” potential mates, initially set to $Q_1$.

### 3.2 GreedyMatch($Q, k, A, \delta, \eta$)

At the heart of our algorithm is the **GreedyMatch** subroutine (Algorithm 1). In **GreedyMatch**, each processor’s initial state contains a list $Q$ of their rankings of remaining potential mates and a partner $p$ (possibly empty). The men additionally each hold a set $A$ of the women in their best non-empty quantile. **GreedyMatch** works in 5 rounds which are described in Algorithm 1.

**Algorithm 1 GreedyMatch($Q, k, A, \delta, \eta$)**

**Round 1:** The men propose to all women in $A$ by sending each $w \in A$ the message PROPOSE.

**Round 2:** Each women receiving proposals responds with the message ACCEPT to all proposals from her best quantile $Q_i$ from which at least one man proposed in Round 1.

**Round 3:** Let $G_0$ denote the bipartite graph $G_0$ of accepted proposals from Round 2. The players compute a $(1 - \eta)$-maximal matching $M_0$ in $G_0$, using **AMM**($G, \delta, \eta$) (see Theorem 2.5), storing their match in $G_0$ as $p_0$. Players which are unmatched (see Definition 2.6) after the call to **AMM** remove themselves from play by sending the message REJECT to all $v \in Q$ then, if $v$ is male, setting $A^v \leftarrow \emptyset$ and $Q^v \leftarrow \emptyset$.

**Round 4:** Each woman $w$ matched in $M_0$ sends REJECT to all men $m \in Q^w$ in a lesser or equal quantile to her partner $p_0(w)$ in $M_0$ other than $p_0(w)$. She then removes all of these men from $Q^w$ and the corresponding $Q_i^v$. The matched women then set $p \leftarrow p_0$, so the partial matching $M$ now contains the edge $(p_0(w), w)$. Any man $m$ matched in $M_0$ sets $p \leftarrow p_0$ and sets $A \leftarrow \emptyset$. The players $v$ receiving REJECT from some $u$ in the previous round remove $u$ from their preferences: $Q^u \leftarrow Q^u \setminus \{u\}$.

**Round 5:** The men remove all $w$ from whom they received the message REJECT from their preferences $Q$, the various $Q_i$ and $A$. If a man $m$ receives a rejection from his match $p(m)$ from a previous round, he sets $p \leftarrow \emptyset$.

The following lemma follows immediately from the description of **GreedyMatch**.

**Lemma 3.1.** Once a woman $w$ has $p(w) \neq \emptyset$ in some execution of **GreedyMatch**, she is guaranteed to always have $p(w) \neq \emptyset$ after each subsequent execution of **GreedyMatch**, unless she is unmatched in some call to **AMM**. Further, once matched, she will only accept proposals from men in a strictly higher quantile than her current $p(w)$. 
3.3 MarriageRound\((Q, C, k, \delta, \eta)\) and ASM\((P, C, \varepsilon, \delta)\)

In this section, we describe how to use GreedyMatch as a subroutine to find an almost stable marriage. The subroutine MarriageRound (Algorithm 2) calls GreedyMatch directly. For the men, MarriageRound initializes \(A\) to be the (remaining) members of each man \(m\)'s best non-empty quantile \(Q_i\). The idea is to iterate the GreedyMatch subroutine until all men have \(A = \emptyset\). This will occur precisely when all men have either been matched or rejected by all women in \(A\). For a woman \(w\), this will occur when she fails to receive any proposals in an iteration of GreedyMatch. We will argue that \(k\) iterations of GreedyMatch suffice.

The main routine ASM (Algorithm 3) iterates MarriageRound until a suitably large matching \(M\) is found. ASM takes parameters the player's preference list \(P\), an upper bound for the ratio of longest to shortest preference lists \(C\), the desired approximation factor \(\varepsilon\), and the error probability \(\delta\). For simplicity, we assume that \(\varepsilon^{-1} \in \mathbb{N}\).

### Algorithm 2 MarriageRound\((Q, C, k, \delta, \eta)\)

\[
\begin{align*}
i &\leftarrow \min \{i \mid Q_i \neq \emptyset\} \quad \text{(male only)} \\
A &\leftarrow Q_i \quad \text{(male only)} \\
\text{for } i &\leftarrow 1 \text{ to } k \text{ do} \\
&\quad \text{GreedyMatch}(Q, k, A, \frac{\delta}{C^3k^3}, \frac{4}{C^3k^4}) \\
\end{align*}
\]

### Algorithm 3 ASM\((P, C, \varepsilon, \delta)\)

\[
\begin{align*}
k &\leftarrow 12\varepsilon^{-1} \\
\text{for all } i &\leq k \text{ do} \\
&Q_i \leftarrow \{v \mid q(v) = i\} \\
\end{align*}
\]

\[
Q \leftarrow \bigcup_i Q_i, \quad p \leftarrow \emptyset \\
\text{for } i &\leftarrow 1 \text{ to } C^2k^2 \text{ do} \\
&\quad \text{MarriageRound}(Q, p, k, \delta) \\
\end{align*}
\]

4 Performance guarantees

4.1 Round and Run-time

The goal of this section is to prove the following run-time guarantee for ASM.

**Theorem 4.1.** Let \(d = \max \deg G\), the length of the longest preference list. ASM\((P, C, \varepsilon, \delta)\) runs in \(O\left(-\varepsilon^3C^3\log(\varepsilon\delta)\right)\) communication rounds. For fixed \(\varepsilon, \delta > 0\) and \(C \geq 1\), each communication round can be performed in \(O(d)\) time, thus the total run-time is linear in \(d\).

The bulk of the complexity in ASM comes from the GreedyMatch subroutine. Indeed, ASM and MarriageRound essentially just iterate GreedyMatch a specified number of times with the proper parameters.

**Lemma 4.2.** GreedyMatch\((Q, k, \delta, \eta)\) uses \(O\left(-\log(\delta\eta)\right)\) communication rounds. For fixed \(\delta, \varepsilon > 0\), each communication round requires \(O(n)\) operations of the types specified in Section 2.3, thus the run-time of GreedyMatch is \(O(n)\).

**Proof.** In rounds 1, 2, 4 and 5, each player sends at most one message of the type specified in Section 2.3 to each player. Round 3 calls AMM\((G, \delta, \eta)\) once, which terminates after \(O\left(-\log(\delta\eta)\right)\) rounds by Theorem 2.5.
For the second part of the lemma, Round 1 requires sending $O(d)$ messages to other players. Similarly, in Round 2, at players receive at most $O(d)$ messages, and query their preferences at most once for each message. Similarly, Rounds 4 and 5 require only $O(d)$ operations. Finally, by Theorem 2.5, AMM (and hence Round 3) require only $O(d)$ operations. 

Proof of Theorem 4.1. This follows from the definitions of ASM, MarriageRound and Lemma 4.2. Indeed, ASM iterates $\text{MatchingRound}(Q, k, A, \frac{\delta}{C^2}, \frac{4}{C^3})$ a total of $\Theta(\varepsilon^{-2}C^2)$ times.

4.2 Approximation

In this section, we prove the following approximation guarantee.

Theorem 4.3. For any preferences $P$ and numbers $\varepsilon, \delta > 0, C \geq 1$, the marriage $M$ output by $\text{ASM}(P, \varepsilon, \delta, C)$ is $(1-\varepsilon)$-stable with probability at least $1-\delta$.

In order to show that the marriage $M$ output by $\text{ASM}(P, \varepsilon, \delta, C)$ is almost stable, we must show that $M$ induces relatively few blocking pairs. To this end, we consider different groups of players separately and show that each group induces few blocking pairs.

1. A player $v \in X \cup Y$ is matched if they appear in the marriage $M$ produced by ASM.
2. A man $m \in Y$ is rejected if he has been rejected by all women on his preference list.
3. A player $v \in X \cup Y$ is unmatched at the end of $\text{ASM}(P, \varepsilon, \delta, C)$ if they are unmatched in some call to $\text{AMM}(G, \delta, \eta)$ (see Definition 2.6).
4. A man $m \in Y$ is bad if at the end of $\text{ASM}$, he is neither matched, rejected, nor unmatched.

In Section 4.2.1, we bound the number of unmatched and bad players, and hence the number of blocking pairs these players can contribute. In Section 4.2.2, we introduce notions of “close” and “equivalent” preferences which allow us to bound the number blocking pairs contributed by matched and unmatched players in Section 4.2.3. Finally, we prove Theorem 4.3 in Section 4.2.4.

4.2.1 Bounding bad and unmatched players

Lemma 4.4. During an execution of $\text{ASM}$, let $Y^b_i$ denote the set of men which after $i$ iterations of the loop in $\text{ASM}$ are bad. Then for $i < i'$, we have

$$|Y^b_{i'}| \leq |Y^b_i|.$$ 

Thus the sizes of the sets $Y^b_i$ are (weakly) decreasing in $i$.

Proof. To prove the lemma, we will show that the size of the complement of $Y^b_i$ is increasing in $i$. This follows from three observations. First if a man $m$ is rejected, then he will never become un-rejected. Second, if a woman $w$ is matched in some round of $\text{ASM}$, she will remain matched by Lemma 3.1. Thus the number of matched women (and hence the number of matched men) is weakly increasing in $i$. Finally, if a man is ever unmatched, he will remain unmatched. Thus the set of bad men is also weakly decreasing. 

Lemma 4.5. At the termination of $\text{ASM}$, there are at most $\frac{\varepsilon}{3C^3} n$ bad men.

Proof. We prove this by counting the number of rejections sent after $j$ iterations of the loop. Since the women can send a combined total of at most $|E|$ rejections, a lower bound on the number of rejections after $j$ iterations gives an upper bound on the number of iterations until most men are matched, unmatched or rejected.
Suppose there were \( b \) bad men after \( j \) iterations. By Lemma 4.4, this implies that there were at least \( b \) bad men after each iteration up to the \( j \)-th iteration. Note that each bad man \( m \in Y \) must have been rejected by all women in their highest (previously) non-empty quantile \( Q_i^m \). Since each \( Q_i^m \) contains \( \deg m/k \) women, after \( j \) rounds, the women must have sent at least \( jb \min \deg G/k \) total rejections in the first \( j \) rounds. Since combined, the women cannot send more than \( |E| \) rejections, we must have

\[
jb \min \deg G/k \leq |E| \leq n \max \deg G.
\]

Thus, taking \( b = \frac{\varepsilon}{3C}n \), we find

\[
j \leq 3\varepsilon^{-1}Ck \frac{\max \deg G}{\min \deg G} \leq \frac{1}{4}C^2k^2.
\]

Therefore, after \( C^2k^2 \) iterations of the loop in ASM, at most \( \frac{\varepsilon}{3C}n \) men are bad, as desired. \( \square \)

**Lemma 4.6.** At the termination of ASM, there are at most \( \frac{\varepsilon}{3C}n \) unmatched players with probability at least \( 1 - \delta \).

**Proof.** ASM makes \( C^2k^3 \) calls to AMM(\( G_0, \delta/C^2k^3, 4/C^3k^4 \)). Applying Theorem 2.3 and the union bound, the probability that any call yields more than \( 4n/C^3k^3 \) unmatched players is at most \( \delta \). Thus, with probability at least \( 1 - \delta \), there are at most \( 4n/Ck = \frac{\varepsilon}{3C}n \) unmatched players. \( \square \)

### 4.2.2 Close and equivalent preferences

**Definition 4.7.** For a preference structure \( \mathcal{P} \), let \( \mathcal{P}(m, w) \) denote \( m \)'s rank of \( w \) (and symmetrically for \( \mathcal{P}(w, m) \)). Define the metric \( d \) on the set of preferences structures by

\[
d(\mathcal{P}, \mathcal{P}') = \sup \left\{ \frac{|\mathcal{P}(m, w) - \mathcal{P}'(m, w)|}{\deg m}, \frac{|\mathcal{P}(w, m) - \mathcal{P}'(w, m)|}{\deg w} \right\} \quad (m, w) \in E. \]

By convention, we set \( d(\mathcal{P}, \mathcal{P}') = 1 \) if there exist a pair \((m, w)\) that rank each other in \( \mathcal{P} \) but not \( \mathcal{P}' \) or vice versa. We say that \( \mathcal{P} \) and \( \mathcal{P}' \) are \( \eta \)-close if \( d(\mathcal{P}, \mathcal{P}') \leq \eta \).

Intuitively, \( \mathcal{P} \) and \( \mathcal{P}' \) are \( \eta \)-close if all pairs \((m, w)\) rank each other similarly (within \( \eta \deg v \)) in the two preferences. We are now ready to state a key lemma which we will use in proving the approximation guarantee in Theorem 4.3.

**Lemma 4.8.** Suppose \( \mathcal{P} \) and \( \mathcal{P}' \) are \( \eta \)-close and that \( M \) is a \((1 - \varepsilon)\)-stable matching for \( \mathcal{P} \). Then \( M \) is a \((1 - \varepsilon - 4\eta)\)-stable matching for \( \mathcal{P}' \).

**Proof.** To prove the lemma, we compute an upper bound for the number of new blocking pairs incurred by each man and woman by changing their preferences. Note that it suffices to show that \( M \) has at most \( 4\eta |E| \) more blocking pairs relative to \( \mathcal{P}' \) than relative to \( \mathcal{P} \).

Suppose \((m_1, w_j) \in M \). Let \( \mathcal{P}^1_m \) denote the preference structure where all \( m_i \in Y \) with \( i \neq 1 \) have the same preferences as \( \mathcal{P} \), as do all the women, but where \( m_1 \) changes his preferences in accordance with \( \mathcal{P}' \). Since only \( m_1 \)'s preferences differ between \( \mathcal{P}^1_m \) and \( \mathcal{P} \), the only new blocking edges \((m_1, w)\) with respect to \( \mathcal{P}^1_m \) (that is blocking edges which were not blocking with respect to \( \mathcal{P} \) but are blocking with respect to \( \mathcal{P}^1_m \)) must occur because \( m_1 \) preferred \( w_j \) to \( w \) in \( \mathcal{P} \) but prefers \( w \) to \( w_j \) in \( \mathcal{P}' \).

**Claim.** There are at most \( 2\eta \deg m_1 \) women that \( m_1 \) ranks below \( w_j \) in \( \mathcal{P} \) but that rank above \( w_j \) in \( \mathcal{P}' \).
To prove the claim, note that by \( \eta \)-closeness, \( m_1 \)'s rank of \( w_j \) can decrease by at most \( \eta \deg m_1 \). On the other hand, for \( w \neq w_j \), \( m_1 \)'s rank of \( w \) can increase by at most \( \eta \deg m_1 \). Thus, only women \( w \) which rank within \( 2\eta \deg m_1 \) of \( w_j \) can change their relative rank with \( w_j \).

By the claim, \( M \) can have at most \( 2\eta \deg \bigcup_{m} P_m \) more blocking pairs relative to \( P_m^1 \) than \( P \). To prove the lemma, we work by iteratively changing players preferences one at a time. Specifically, let \( P'_m \) be the preference structure where \( m_i' \) with \( i' \leq i \) have preferences from \( P' \) while the remaining players have preferences from \( P \). The above argument shows that \( P'_m \) can have at most \( 2\eta \deg m_1 \) more blocking pairs than \( P_m^{i-1} \). Applying the analogous procedure for updating the women’s preferences (after the men’s preferences have been updated) we find that that at most

\[
\sum_{v \in X \cup Y} 2\eta \deg v = 4\eta |E|
\]

new blocking pairs can be created by changing preferences from \( P \) to \( P' \).

**Definition 4.9.** We say that two preferences \( P \) and \( P' \) are \( k \)-equivalent if they have the same \( k \)-quantiles. That is, for each \( v \in X \cup Y \), and each \( i \in [k] \) we have \( Q^i_v = Q^i_{v'} \).

**Lemma 4.10.** If preferences \( P \) and \( P' \) are \( k \)-equivalent, then they are \((1/k)\)-close.

**Proof.** Since \( P \) and \( P' \) have the same \( k \)-quantiles, for each pair \((m, w)\), \( P(m, w) \) and \( P'(m, w) \) must reside in the same \( k \)-quantile. In particular, this implies that \(|P(m, w) - P'(m, w)| \leq \deg(m)/k\). Similarly for \( P(w, m) \) and \( P'(w, m) \).

**Corollary 4.11.** If \( P \) and \( P' \) are \( k \)-equivalent, and \( M \) is a \((1 - \varepsilon)\)-stable matching for \( P \), then \( M \) is a \((1 - \varepsilon - 4/k)\)-stable matching for \( P' \).

### 4.2.3 Bounding blocking pairs from matched and unmatched players

In this section, we prove an approximation guarantee for ASM. Blocking pairs in the output \( M \) of ASM come from three sources: (1) blocking pairs from matched and rejected players, (2) unmatched players, and (3) bad players. We bound the number of blocking pairs of type (1) by showing that \( M \) (along with any rejected players) contains no blocking pairs with respect to preferences \( P' \) which are \( k \)-equivalent to \( P \). We bound the number of blocking pairs of type (2) by appealing to Lemmas 4.5 and 4.6.

We define \( P' \) by following the sequence of proposals, acceptances and rejections in an execution of ASM. In particular, we will ensure that the sequence of messages is consistent with an execution of the (extended) Gale-Shapley algorithm with preferences \( P' \). For each player \( v \in X \cup Y \), it suffices to define an order on the quantiles \( Q^i_v, \ldots, Q^k_v \). The construction of \( P' \) for the men and women are different.

**Men’s preferences:** Suppose \( m \in Y \) has a sequence of matches in his \( i \)th quantile \( Q^m_i \) given by \( w_1, w_2, \ldots, w_j \). That is \( m \) is matched with \( w_1, \ldots, w_j \) at some point in the execution of ASM in that order. Then we define an order on \( Q^m_i \) to have \( w_1 \succ w_2 \succ \cdots \succ w_j \). The remaining \( w \in Q^m_i \) satisfy \( w_j \succ w \), but their relative order is arbitrary. That is, in \( P' \) each man prefers each woman he is paired with in the (temporal) order in which they are matched. He prefers these matches to all other women in \( Q^m_i \).

**Women’s preferences:** Suppose \( w \in X \) is paired with \( m \in Q^w_i \) at some step in ASM. From the description of ASM, women can be paired with at most one man in each of her quantiles in a given execution (see Lemma 4.1). We define \( P' \) on \( Q^w_i \) by \( m \succ m' \) for all \( m' \in Q^w_i \), but the order is otherwise arbitrary.

**Lemma 4.12.** \( P \) and \( P' \) are \( k \)-equivalent.
Proof. This is immediate from the definition of $P'$, as only the preferences within each quantile are modified.

Lemma 4.13. Let $M$ be the (partial) marriage found by ASM, and $G'$ the induced subgraph of $G$ consisting of matched and rejected players. Then $M$ contains no blocking pairs in $G'$ with respect to preferences $P'$.

Proof. Suppose to the contrary, that $(m, w)$ is a blocking pair in $G'$ with respect to $P'$. First, consider the case where $m$ is rejected—i.e., $p(m) = \emptyset$. Since $m \in G'$ is rejected, $w$ must have rejected $m$ in some previous call to GreedyMatch. By Lemma 3.1 and the definition of $P'$, this implies that $w$ is matched with someone she prefers to $m$, contradicting that $(m, w)$ is a blocking pair.

Now suppose $p(m) = w' \neq w$. By Lemma 3.1 $w$ could not have rejected $m$ in any call to GreedyMatch (and in particular, $m$ and $w$ could not previously been each others' partner). Consider the last GreedyMatch in which $m$ proposed to $w$. If $m$ was paired with another woman $w''$ in this GreedyMatch, then we must have $w'' = w'$, for otherwise $m$ would have proposed to $w$ again after being rejected by $w''$. Thus, by the definition of $P'$, $m$ prefers $w'$ to $w$, a contradiction. If $m$ is unpaired after this call to GreedyMatch, then either $w$ was paired with someone she prefers to $m$, or $m$ and $w$ are unpaired. In either case, we arrive at a contradiction.

4.2.4 Proof of Theorem 4.3

We are now ready to prove the approximation guarantee for ASM given in Theorem 4.3.

Proof of Theorem 4.3. Let $M$ be the marriage output by $ASM(P, C, \varepsilon, \delta)$. We must show that with probability at least $1 - \delta$, $M$ is $(1 - \varepsilon)$ stable with respect to preferences $P$. Let $V_b$ and $V_u$ denote the sets bad and unmatched players, respectively. By Lemmas 4.5 and 4.6, we have $|V_b|, |V_u| \leq \frac{\varepsilon}{3C} n$. The number of blocking pairs incident with $V_b$, say, is bounded by

$$\max \deg G |V_b| \leq \max \deg G \frac{\varepsilon}{3C} n \leq \frac{\varepsilon}{3} n \min \deg G \leq \frac{\varepsilon}{3} |E|.$$

A similar inequality holds for blocking pairs incident with $V_u$, hence combined these two sets contribute at most $2\varepsilon |E|/3$ blocking pairs.

Let $P'$ be the preferences described in Section 4.2.3. By Lemma 4.13 the only blocking pairs in $M$ with respect to $P'$ must be incident with $V_b$ or $V_u$, hence $M$ is $(1 - 2\varepsilon/3)$-stable with respect to $P'$. Thus, for $k = 12\varepsilon^{-1}$, applying Corollary 4.11 and Lemma 4.12 we find that $M$ is $(1 - 2\varepsilon/3 - \varepsilon/3)$-stable with respect to $P$, as desired.

Theorem 1.1 is an immediate consequence of Theorems 4.1 and 4.3.

5 Commentary

A reasonable critique of ASM is that it requires the parameter $C \geq \frac{\max \deg G}{\min \deg G}$. Since $C$ encodes some global information about the structure of the communication graph $G$, using $C$ as a parameter in ASM is somewhat unnatural.

Open Problem 5.1. Can Theorem 1.1 be extended to not depend on the parameter $C$?

In our analysis of ASM, we use $C$ to bound the number of potential blocking pairs induced by unmatched and bad players. Since we do not control which players are unmatched or bad, we trivially bound the number of blocking pairs from an unmatched or bad player by $\max \deg G$. It is conceivable that by locally ensuring that players with relatively long preferences lists (and
hence the potential to contribute to many blocking pairs) are not unmatched or bad, we can avoid using $C$ as a parameter to obtain a generalization of Theorem 1.1.

Since the run-time of ASM is linear in the input size for each processor, it is asymptotically optimal for algorithms which require sequential access to the input. However faster algorithms may exist if each processor is granted random access to its preference list.

**Open Problem 5.2.** Is there a sub-linear time distributed algorithm for the almost stable marriage problem using random access to players’ preferences?

## References

1. Kimmo Eriksson and Olle Häggström. Instability of matchings in decentralized markets with various preference structures. *International Journal of Game Theory*, 36(3):409–420, March 2008.

2. Patrik Florén, Petteri Kaski, Valentin Polishchuk, and Jukka Suomela. Almost Stable Matchings by Truncating the Gale–Shapley Algorithm. *Algorithmica*, 58(1):102–118, 2010.

3. D Gale and L S Shapley. College Admissions and the Stability of Marriage. *The American Mathematical Monthly*, 69(1):pp. 9–15, 1962.

4. Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms*, volume 54. MIT press Cambridge, 1989.

5. Avinatan Hassidim, Yishay Mansour, and Shai Vardi. Local computation mechanism design. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 601–616. ACM, 2014.

6. Amos Israeli and Alon Itai. A fast and simple randomized parallel algorithm for maximal matching. *Information Processing Letters*, 22(2):77–80, 1986.

7. Alex Kipnis and Boaz Patt-Shamir. a note on distributed stable matching. In *Proceedings of the 28th ACM symposium on Principles of distributed computing*, pages 282–283, New York, NY, USA, 2009. ACM.

8. David Manlove. *Algorithmics of matching under preferences*. World Scientific Publishing, 2013.

9. David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000.

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

## A Almost Maximal Matchings

Israeli and Itai’s [6] algorithm for finding a maximal matching works by identifying a sparse subgraph of $G$, then finding a large matching $M_1$ in the sparse subgraph. The edges and incident vertices of $M_1$, as well as remaining isolated vertices, are removed from $G$ resulting in a subgraph $G_1$. The process is iterated, giving a sequence of subgraphs $G_1, G_2, \ldots$ and matchings $M_1, M_2, \ldots$, until $G_k = \emptyset$. At this point, $M = \bigcup_{i=1}^{k} M_i$ is a maximal matching. We give pseudocode for Israeli and Itai’s main subroutine, which we call MatchingRound, in Algorithm 1. In [6], Israeli and Itai prove the following performance guarantee for MatchingRound.
Algorithm 4 MatchingRound($G$): Finds a large matching in a graph

1: Each $v \in V$ picks a neighbor $w$ uniformly at random, forms oriented edge $(v, w)$.
2: Each $v \in V$ with $\deg_G(v) > 0$ picks one in-coming edge $(w, v)$ uniformly at random, deletes remaining in-edges. Let $G'$ be the (undirected) graph formed by the chosen edges with orientation ignored.
3: Each $v \in V$ with $\deg_{G'}(v) > 0$ chooses one incident edge $(v, w)$ uniformly at random.
4: The matching $M_1$ consists of edges $(v, w) \in G'$ which were chosen by both $v$ and $w$ in the previous round. $G_1 = (V_1, E_1)$ is the induced subgraph of $G$ formed by removing all vertices contained in $M_1$ and any remaining isolated vertices from $G$.
5: Output $(G_1, M_1)$.

Lemma A.1. (Israeli and Itai [6]) There exists an absolute constant $0 < c < 1$ such that on input $G = G_0 = (V_0, E_0)$, the resulting graph $G_1 = (V_1, E_1)$ found by MatchingRound satisfies $E(|V_1|) \leq c |V_0|$.

Remark A.2. The graph $G_1$ found by MatchingRound consists precisely of those vertices (and induced edges) which do not satisfy properties 1 or 2 preceding Definition 2.4. Thus, $M_1$ is a $(1 - |V_1|/|V_0|)$-maximal matching.

Proof of Theorem 2.5. For fixed $k \in \mathbb{N}$, let $G = G_0, G_1, G_2, \ldots, G_k$ be the graphs and $M_0 = \emptyset, M_1, \ldots, M_k$ the matchings defined inductively by $(G_i, M_i) = \text{MatchingRound}(G_{i-1})$. Denote $M = \bigcup_{i=1}^k M_i$. By Lemma A.1 we have

$$E(|G_k|) \leq c^k |G|.$$ 

Choosing $k = O(\log(\delta^{-1} \eta^{-1}))$ and applying Markov's inequality, we find

$$\Pr(|G_k| \geq \eta |G|) \leq \delta.$$ 

Thus, AMM($G, \delta, \eta$) can be defined by iterating MatchingRound$k = O(\log(\delta^{-1} \eta^{-1}))$ times. The run-time follows from observing that MatchingRound($G$) can be performed in time $O(\max \deg G)$. 

\hfill \Box