On the Stable Matchings that can be Reached
When the Agents Go Marching in One by One

Christine T. Cheng
University of Wisconsin-Milwaukee

November 19, 2014

Abstract

The Random Order Mechanism (ROM) can be thought of as a sequential version of Gale and Shapley's deferred-acceptance (DA) algorithm where agents are arriving one at a time, and each newly arrived agent has an opportunity to propose. Like the DA algorithm, ROM can be implemented in polynomial time. Unlike the DA algorithm, it is possible for ROM to output a stable matching that is different from the man-optimal and woman-optimal stable matchings.

We say that a stable matching \( \mu \) is ROM-reachable if ROM can output \( \mu \). In this paper, we investigate computational questions related to ROM-reachability. First, we show that there is an efficient algorithm for determining if ROM can reach a non-trivial stable matching - i.e., one different from the man-optimal and woman-optimal stable matchings. However, if we wish to determine if ROM can reach a particular stable matching then the problem becomes NP-complete. We then study two restricted versions of this problem. In the first version, we consider stable matchings that can be reached by ROM in a “direct” manner. We show that they are computationally easy to recognize. In the second version, we restrict the class of stable matchings to what we call extreme stable matchings and prove that computational complexity of determining if they are ROM-reachable depends on the number of unstable partners of the agents.

1 Introduction

Since Gale and Shapley's seminal publication [10] on stable matchings, economists, mathematicians and computer scientists alike have flocked into the field. The subject is rich and deep – four books [14, 11, 20, 16] and hundreds, if not thousands, of papers on stable matchings have been published. As a solution concept, it is also widely used in practice – many centralized matching markets such as those for NRMP [19], the Boston Public School Match [2] and the New York City High School Match etc.[1]\(^1\) aim to match agents from two sides of the market in a stable way.

Our initial interest in stable matchings, however, had a more mundane reason. We simply found Gale and Shapley's deferred-acceptance (DA) algorithm to be a lot of fun. Our students share this enthusiasm whenever we have lectured on this topic. The reason it seems is how the agents behave in the algorithm. Their actions more or less capture what most people do in practice. An “active” agent (classically a “man”) will initiate offers starting with the person from the other group that he prefers the most. A “passive” agent (classically a “woman”), on the other hand, will just wait for offers but will still act in a self-interested way. The students are taken aback

\(^1\)See also the references in [17].
though when they learn that the DA algorithm can only produce two kinds of stable matchings – the man-optimal/woman-pessimal and woman-optimal/man-pessimal stable matchings – even when an instance has an exponential number of stable matchings. In a few of these occasions, they have asked if there are other algorithms where both men and women can make proposals, and whether such an algorithm might output a stable matching that is less biased towards one side of the matching. The answer to their question turns out to be “yes”, and determining the stable matchings the algorithm can reach is the subject of our investigation.

Ma proposed the Random Order Mechanism (or ROM for short) in 1996 [15] as a variant to Roth and Vande Vate’s [21] work on random paths to stability. It works as follows: Let $\pi$ be an ordering of the agents chosen uniformly at random. Think of the agents as arriving in a room (or a market) one at a time. In between arrivals, the room is closed so that a stable matching can be found for the agents in the room. The initial stable matching $\mu_0$ is empty. Let $\mu_{i-1}$ denote the stable matching obtained prior to the arrival of $\pi(i)$. When $\pi(i)$ enters the room, $\mu_{i-1}$ may or may no longer be a stable matching of the instance consisting of the $i$ agents in the room. If the former is true, $\mu_i$ is just $\mu_{i-1}$; if the latter is true, $\pi(i)$ must form a blocking pair with one of the agents in the room. Resolve this in a best response manner. That is, among all the agents that $\pi(i)$ forms a blocking pair, $\pi(i)$ is matched to the agent she prefers the most, say $a$. Now if $a$ had a partner in $\mu_{i-1}$, this partner is now unmatched and may create new blocking pairs. Let her resolve in a best response manner again. This process is repeated until a stable matching is obtained. Set $\mu_i$ to be this stable matching. The final stable matching, $\mu_{|\pi|}$, formed after all the agents have arrived must then be a stable matching of the original instance.

We note that the step of determining if some agent $b$ is part of some blocking pair of an existing matching $\mu$ and then resolving it in a best response manner can be simulated by a procedure that is reminiscent of the DA algorithm: $b$ goes through her preference list and proposes to those who are currently in the market starting with the agent she prefers the most. The first agent to accept her proposal forms a blocking pair with $b$ and is the agent that $b$ prefers the most among all those that form a blocking pair with her. If no agent accepts her proposal, the current matching is stable. We have been referring to $\pi(i)$ as a woman but $\pi(i)$ can be a man too. Hence, we can think of ROM as a sequential version of the DA algorithm where agents from both sides of the market have an opportunity to propose. Like the DA algorithm, ROM can also be implemented in polynomial time. Unlike the DA algorithm, it is possible for ROM to output a stable matching different from the man-optimal and woman-optimal stable matchings. Ma [15] used an example of Knuth’s to show that ROM can produce six out of the ten stable matchings of the instance.

Let us say that a stable matching $\mu$ is reachable by ROM or ROM-reachable if there is an ordering $\pi$ of the agents so that when ROM processes $\pi$, the output is $\mu$. One of the most useful properties of ROM-reachable stable matchings is due to Cechlárová [9] and Blum et al. [8]. It states that every ROM-reachable stable matching must have at least one agent matched to their best stable partner. Other properties can be also found in [8] and [7]; nonetheless, a nice characterization of ROM-reachable stable matchings is still missing. In this paper, our goal is to address computational questions about ROM-reachability.

**Our Results.** Let $I$ be an instance with $n$ agents, and let us refer to a stable matching of $I$ as non-trivial if it is different from the man-optimal and woman-optimal stable matchings. First, we ask a basic question – Is there a non-trivial stable matching of $I$ that is ROM-reachable? We show that the problem can be answered in polynomial time. All we have to do is run at most $n$ permutations on ROM. If ROM can reach a non-trivial stable matching then one of these runs will also output a non-trivial stable matching of $I$. 

2
However, if we want to determine if ROM can reach a particular stable matching of $I$, then we prove that problem becomes NP-complete, even in the case when every agent has a preference list of length at most 4. Our result answers an open problem in [16]. In our reduction, the stable matching of interest, $\mu$, has many disjoint sub-matchings where no agent is matched to their best stable partner. To reach these sub-matchings, ROM has to first form sub-matchings that are not part of $\mu$ and then use them as stepping stones to reach the said sub-matchings. It is this two-step process that makes the problem difficult because the intermediate sub-matchings are intertwined with each other.

Next, we say that a stable matching $\mu$ is strongly ROM-reachable if there is some input permutation $\pi$ of $I$’s agents so that the output of ROM is $\mu$, and $\mu_1 \subseteq \mu_2 \subseteq \cdots \subseteq \mu_{|\pi|} = \mu$. That is, once a pair of agents is part of a $\mu_i$, it is part of the remaining stable matchings until ROM ends. Anecdotally, many of the ROM-reachable stable matchings we have found are also strongly ROM-reachable. In the third part of our paper, we characterize the strongly ROM-reachable stable matchings using directed graphs. We then present an efficient algorithm that recognizes these kinds of stable matchings. Our characterization makes use of subgraphs of jealousy graphs defined recently by Hoffman et al. [12] to obtain more refined bounds for the convergence time of random better response dynamics. It is interesting that jealousy graphs are also relevant to ROM.

Finally, we consider a class of stable matchings we call extreme stable matchings. They are the stable matchings where every pair has one agent matched to his/her best stable partner and the other to his/her worst stable partner. Unlike the stable matching of interest in our first NP-completeness reduction, these stable matchings do not have sub-matchings that lie in the “middle”. We show that when each agent has at most one unstable partner in $I$ (i.e., the agent and the unstable partner are never matched in a stable matching of $I$), every extreme stable matching of $I$ is strongly ROM-reachable. However, there is an instance where some agents have two unstable partners, and this instance has an extreme stable matching that is not reachable by ROM. Using this instance as a gadget, we then prove that when agents have two or more unstable partners, determining if an extreme stable matching of $I$ is ROM-reachable is NP-complete. These results are highly unusual in that we know of no computational problems on stable matchings where the unstable pairs of the instance determine the complexity of the problem.

**Related work.** We have presented ROM as a sequential version of the DA algorithm where the starting matching is the empty matching and agents from both sides of the matching are allowed to propose. Ma [15] also allowed ROM to start at an arbitrary matching. He based ROM on Roth and Vande Vate’s proof [21] that the random better response dynamics converges to a stable matching with probability 1. Interestingly, ROM is quite different from the random better response dynamics in at least two ways. First, starting with an empty matching, the latter can reach every stable matching of an instance [21]. Such a property does not hold for ROM. Second, there are instances where the random better response dynamics can take exponential time to converge to a stable matching [3]. In contrast, ROM always reaches a stable matching in a polynomial number of steps.

In [8], Blum et al. sought to model the dynamics of senior-level labor markets (e.g., head football coaches of US college teams, etc.). Assume a stable matching already exists for the firms and workers in the market. Then some workers retire and some firms open up new positions. Blocking pairs involving unmatched firms can now exist. The paper studied how the market can restabilize itself using the DA algorithm. Their results describe how stable matchings change from one iteration of ROM to the next. Biro et al. [7] extended their work to the stable roommates setting.
Other probabilistic mechanisms for generating stable matchings have been studied in the past (see [3], [16] and references therein). Perhaps the one that is most similar to ROM is Employment by Lotto (EBL) by Aldershof et al. [4], which is just the random serial dictatorship (RSD) applied to the stable matchings setting. Like ROM, the input of EBL is a random permutation \( \pi \) of the agents. Initially, \( S_0 \) consists of all the stable matchings of the instance. In the \( i \)th iteration, \( S_i \) is reduced to the set of stable matchings where \( \pi(i) \) is matched to his/her best stable partner in \( S_{i-1} \). The algorithm ends when all the agents in \( \pi \) have been processed and outputs \( S_{|\pi|} \). The recent work by Aziz et al. [5] on RSD in the one-sided matching setting imply that there is an efficient algorithm for determining if a particular stable matching can be reached by EBL. What is computationally hard is determining the probabilities induced by EBL on the stable matchings.

Lastly, many researchers have proposed different notions of “fair” stable matchings. Klaus and Klijn [13] argued that while both ROM and EBL do not guarantee end-state fairness (i.e., they may not output the stable matchings of the instance with equal probability), they are procedurally fair because “the sequence of moves for the agents is drawn uniformly at random.”

2 Preliminaries

Stable Marriage with Incomplete Lists (SMI) instances model two-sided matching markets. One side consists of “men”, the other of “women”. Each agent has a preference list that ranks members from the opposite group the agent has deemed acceptable in a linear order. A pair \((m, w)\) is acceptable if \(m\) and \(w\) appear in each other’s preference lists. A matching \(\mu\) is a set of acceptable man-woman pairs so that every agent is part of at most one pair. The matching has a blocking pair \((m, w)\) if (i) \(m\) is unmatched or \(m\) prefers \(w\) to his partner in \(\mu\) and (ii) \(w\) is unmatched or \(w\) prefers \(m\) to her partner in \(\mu\). A goal in two-sided matching markets is to find stable matchings, which are matchings with no blocking pairs, because the agents are less likely to break their assignments.

Throughout this paper, we will assume that in every SMI instance \(I\), an agent \(a\) is in another agent \(b\)'s preference list if and only if \(b\) is in \(a\)'s preference list. The two agents are stable partners if they are matched to each other in some stable matching of \(I\); otherwise, they are unstable partners. Additionally, \(b\) is \(a\)'s best (worst) stable partner if among all of his/her stable partners \(b\) is his/her most (least) preferred one. Gale and Shapley [10] showed that when it is the men who propose in their algorithm, the result is the man-optimal/woman-pessimal stable matching – that is, every man is matched to his best stable partner and simultaneously every woman is matched to her worst stable partner. On the other hand, when the women are the ones who propose in their algorithm, the output is the woman-optimal/man-pessimal stable matching which is defined similarly. A simple corollary of this result is that when \(b\) is \(a\)'s best stable partner then \(a\) is \(b\)'s worst stable partner.

Since the number of men and the number of women in \(I\) need not be the same, some agents may be unmatched in a stable matching of \(I\). The Rural Hospitals Theorem [18] states that when an agent is unmatched in one stable matching of \(I\), the agent will be unmatched in all stable matchings of \(I\). Thus, the set of matched agents is the same for all stable matchings of \(I\). The set can be easily determined by computing the man-optimal stable matching of \(I\).

Below is the pseudocode for Ma’s Random Order Mechanism or ROM [15]. Let \(I\) consist of \(n\) agents. When \(S\) is a subset of \(I\)'s agents, we use \(I_{|S|}\) to denote the SMI instance obtained by restricting \(I\) to the agents in \(S\). We say that \(a\) is a blocking agent of a matching \(\mu_i\) of \(I_{|S|}\) if it is part of a blocking pair of \(\mu_i\). We also say that \(b\) is the best blocking partner of \(a\) if \(b\) is the one that \(a\) prefers the most among all agents that form a blocking pair with \(a\). Let \(\pi\) be a permutation of \(I\)'s agents chosen uniformly at random.
\begin{align*}
\text{ROM}(\pi, I) \\
S & \leftarrow \emptyset, \mu_0 \leftarrow \emptyset \\
\text{for } i = 1 \text{ to } n \\
& a_i \leftarrow \pi(i), S \leftarrow S \cup \{a_i\} \\
& \mu_i \leftarrow \mu_{i-1} \\
\text{while } a_i \text{ is a blocking agent of } \mu_i \text{ with respect to } I_{|S} \\
& \text{let } b_j \text{ be the best blocking partner of } a_i \\
& a_z \leftarrow a_i \\
& \text{if } b_j \text{ is matched in } \mu_i \\
& \text{let } a_i \text{ now denote the partner of } b_j \\
& \mu_i \leftarrow \mu_i - \{(a_i, b_j)\} \\
& \mu_i \leftarrow \mu_i \cup \{(a_z, b_j)\} \\
\text{return } \mu_n
\end{align*}

**Fact 1** ([21, 16]) ROM(\pi, I) always terminates and outputs a stable matching of I.

The while loop runs in $O(|I|)$ time, where $|I|$ is the size of instance I, so ROM(\pi, I) runs in $O(n|I|)$ time. Let us call a stable matching $\mu$ of I reachable by ROM or ROM-reachable if there is some permutation $\pi$ of its agents so that $\mu$ is the output of ROM(\pi, I). Ma noted that if $\pi$ consists of all the women first followed by all the men, the output of ROM is the man-optimal stable matching of I; similarly, if $\pi$ consists of all the men first followed by all the women, the output of ROM is the woman-optimal stable matching of I. Unlike Gale and Shapley’s algorithm, however, ROM can in some cases output stable matchings different from the man-optimal and woman-optimal stable matchings. For example, Ma presented an SMI instance that had 10 stable matchings, six of which are reachable by ROM. Several researchers have derived necessary conditions for a stable matching to be ROM-reachable. Here are some of them:

**Fact 2** (Cechlárová [9], Blum et al. [8]) Suppose SMI instance I has $n$ agents, and $\pi$ is an ordering of I’s agents.

(i) Let $\pi(n) = a$. Then $a$ is matched to his/her best stable partner in $\mu_n$, the output of ROM(\pi, I).

(ii) Let $b \neq a$. If the partner of $b$ in $\mu_{n-1}$ is one of his/her stable partners in I, then $b$ will remain matched to this partner in $\mu_n$. Otherwise, $b$ is matched to his/her best stable partner in $\mu_n$ if $b$ has the same gender as $a$ and to his/her worst stable partner in $\mu_n$ if $b$ has the opposite gender as $a$.

We shall now use the above fact to prove our first result. For each agent $a$, let $\pi_a$ denote a permutation that consists first of an ordering of the agents of the same gender as $a$ except for $a$, followed by an ordering of the agents of the opposite gender as $a$, and then ending with $a$. Let $\mu(a)$ refer to the partner of $a$ in the stable matching $\mu$.

**Lemma 1** Suppose there is a permutation $\pi$ of I so that ROM(\pi, I) outputs a non-trivial stable matching of I. Let $a$ be the last agent in $\pi$. Then ROM($\pi_a$, I) will also output a non-trivial stable matching of I.

**Proof** Denote the output of ROM(\pi, I) as $\mu$. Without loss of generality, assume $a$ is a woman, and let $\mu_W$ be the woman-optimal stable matching of I. Then $\mu(a) = \mu_W(a)$ according to Fact 2
(i). But since \( \mu \neq \mu_W \), there must be some man \( b \) such that \( \mu(b) \neq \mu_W(b) \). In particular, \( b \) prefers \( \mu(b) \) over \( \mu_W(b) \) since \( \mu_W(b) \) is his worst stable partner in \( I \). Additionally, \( \mu(b) \) has to be a stable partner of \( b \) in \( I_a \), the instance obtained from \( I \) by removing agent \( a \); otherwise, according to Fact 2 (ii), \( \mu(b) = \mu_W(b) \).

Now consider what happens in ROM(\( \pi_a, I \)). At the end of iteration \( n-1 \), the stable matching is the man-optimal stable matching of \( I_a \). Hence, \( b \) is matched to \( \mu(b) \) or somebody he prefers over \( \mu(b) \) since \( \mu(b) \) is one of his stable partners in \( I_a \). At iteration \( n \), agent \( a \) arrives and a sequence of proposals are made by the women until a stable matching is obtained. Since \( b \) is already matched, he will accept a new offer only if it came from a woman he prefers over his current partner. In other words, \( b \) will always be matched throughout iteration \( n \) and his partner will stay the same or get better and better. Thus, at the end of iteration \( n \), \( b \) has to be matched to \( \mu(b) \) or somebody he prefers over \( \mu(b) \); that is, his partner cannot be \( \mu_W(b) \). On the other hand, since \( a \) is the last agent to arrive, \( a \) has to be matched to \( \mu_W(a) \). It follows that the outcome of ROM(\( \pi_a, I \)) is neither the man-optimal nor woman-optimal stable matchings. \( \square \)

We emphasize that the lemma does not say that ROM(\( \pi_a, I \)) and ROM(\( \pi, I \)) have the same outputs; rather, if ROM(\( \pi, I \)) outputs a non-trivial stable matching then so will ROM(\( \pi_a, I \)).

**Theorem 1** Suppose \( I \) has \( n \) agents. Then checking if ROM can reach a non-trivial stable matching of \( I \) takes \( O(n^2|I|) \) time.

**Proof** First, determine the man-optimal and woman-optimal stable matchings of \( I \). Then for each agent \( a \), construct a permutation \( \pi_a \) and run ROM(\( \pi_a, I \)). If one run outputs a non-trivial stable matching of \( I \), return “yes”; otherwise if the outputs of all the runs are just the trivial stable matchings of \( I \), return “no”. The correctness follows from the previous lemma. Since there are \( n \) permutations to consider and ROM can be implemented in \( O(n|I|) \) time, the whole procedure takes \( O(n^2|I|) \) time. \( \square \)

### 3 ROM-Reachability is NP-complete

In this section, we prove the NP-completeness of ROM-Reachability: Given a stable matching \( \mu \) of instance \( I \), is there a permutation \( \pi \) of \( I \)'s agents so that when ROM processes \( \pi \), the output is \( \mu \)? Consider the SMI instance \( I^* \) below. The men are \( m_i \) and \( a_{i1}, a_{i2} \), \( i = 1, 2, 3 \) while the women are \( w_i \) and \( b_{i1}, b_{i2}, i = 1, 2, 3 \).

\[
\begin{align*}
m_1: & \quad w_1 \quad w_2 \quad b_{11} \quad w_3 \\
m_2: & \quad w_2 \quad w_3 \quad b_{21} \quad w_4 \\
m_3: & \quad w_3 \quad w_1 \quad b_{31} \quad w_2 \\
a_{11}: & \quad b_{11} \quad b_{12} \\
a_{12}: & \quad b_{12} \quad b_{11} \\
a_{21}: & \quad b_{21} \quad b_{22} \\
a_{22}: & \quad b_{22} \quad b_{21} \\
a_{31}: & \quad b_{31} \quad b_{32} \\
a_{32}: & \quad b_{32} \quad b_{31}
\end{align*}
\]

\[
\begin{align*}
w_1: & \quad m_2 \quad m_3 \quad m_1 \\
w_2: & \quad m_4 \quad m_1 \quad m_2 \\
w_3: & \quad m_1 \quad m_2 \quad m_3 \\
b_{11}: & \quad a_{12} \quad a_{11} \quad m_1 \\
b_{12}: & \quad a_{11} \quad a_{12} \\
b_{21}: & \quad a_{22} \quad a_{21} \quad m_2 \\
b_{22}: & \quad a_{11} \quad a_{12} \\
b_{31}: & \quad a_{32} \quad a_{31} \quad m_3 \\
b_{32}: & \quad a_{31} \quad a_{32}
\end{align*}
\]

Let \( \alpha_1 = \{(m_1, w_1), (m_2, w_2), (m_3, w_3)\}, \alpha_2 = \{(m_1, w_2), (m_2, w_3), (m_3, w_1)\}, \alpha_3 = \{(m_1, w_3), (m_2, w_1), (m_3, w_2)\} \). Notice that \( \alpha_1, \alpha_2, \alpha_3 \) are exactly the stable matchings of the sub-instance of
I* when restricted to the agents m_i’s and w_i’s. Both \(\alpha_1\) and \(\alpha_3\) are ROM-reachable stable matchings of this sub-instance because they are the man-optimal and woman-optimal stable matchings respectively. However, \(\alpha_2\) is not ROM-reachable for this sub-instance since none of the m_i’s nor the w_i’s are matched to their best stable partners in the sub-instance.

For \(j = 1, 2, 3\), let \(\beta_{j1} = \{(a_{j1}, b_{j1}), (a_{j2}, b_{j2})\}\) and \(\beta_{j2} = \{(a_{j1}, b_{j2}), (a_{j2}, b_{j1})\}\). Notice also that \(\beta_{j1}\) and \(\beta_{j2}\) are exactly the stable matchings of the sub-instance of \(I^*\) when restricted to \(a_{j1}, a_{j2}, b_{j1}, b_{j2}\). It is easy to check that the stable matchings of \(I^*\) are exactly of the form \(\alpha_i \cup \beta_{1k_i} \cup \beta_{2k_i} \cup \beta_{3k_i}\) where \(i \in \{1, 2, 3\}\) and \(k_1, k_2, k_3 \in \{1, 2\}\). Of interest to us is the stable matching \(\mu^* = \alpha_2 \cup \beta_{12} \cup \beta_{22} \cup \beta_{32}\).

**Proposition 1** When \(\pi = m_1, m_2, m_3, w_1, w_2, w_3, b_{11}, a_{11}, a_{12}, b_{12}, a_{21}, a_{22}, b_{22}, b_{31}, a_{31}, a_{32}, b_{32}\), ROM(\(\pi, I\)) outputs \(\mu^*\).

**Lemma 2** Let \(\pi\) be a permutation of the participants of \(I^*\). Suppose ROM(\(\pi, I^*\)) outputs \(\mu^*\). Then the following must be true:

(i) For \(j = 1, 2, 3\), among \(a_{j1}\), \(a_{j2}\), \(b_{j1}\) and \(b_{j2}\), the last agent to appear in \(\pi\) is \(b_{j1}\) or \(b_{j2}\).

(ii) There must be some \(j \in \{1, 2, 3\}\) so that \(b_{j1}\) appears first and \(b_{j2}\) appears last in the ordering of \(a_{j1}, a_{j2}, b_{j1}\) and \(b_{j2}\) in \(\pi\).

**Proof** To prove (i), among \(a_{j1}\), \(a_{j2}\), \(b_{j1}\) and \(b_{j2}\), let c be the last agent to appear in \(\pi\). Let \(\mu'\) be the stable matching prior to ROM processing c. Suppose \(c = a_{j1}\). If \(b_{j2}\) is unmatched in \(\mu'\), \((a_{j2}, b_{j2})\) is a blocking pair of \(\mu'\). Since the only person \(b_{j2}\) can be matched to is \(a_{j2}\), \((a_{j2}, b_{j2})\) must belong to \(\mu'\). Consequently, either \((m_j, b_{j1}) \in \mu'\) or \(b_{j1}\) is unmatched in \(\mu'\). When ROM finally processes \(a_{j1}\), he will get matched to \(b_{j1}\) so that \(\beta_{j1}\) is part of the stable matching at the end of this iteration. Since none of the remaining agents after \(a_{j1}\) in \(\pi\) can form a blocking pair with \(a_{j1}\), \(a_{j2}\), \(b_{j1}\) and \(b_{j2}\) when the latter are matched according to \(\beta_{j1}\), \(\beta_{j1}\) will be a sub-matching of the output of ROM. This contradicts our assumption that ROM(\(\pi, I^*\))’s output is \(\mu^*\). Thus, \(c \neq a_{j1}\).

The same reasoning applies as to why \(c \neq a_{j2}\) so \(c = b_{j1}\) or \(b_{j2}\).

To prove (ii), first we note that during the execution of ROM(\(\pi, I^*\)) there must be a step where some \(m_j\) is temporarily matched to \(b_{j1}\) even though the two may not be matched to each other in the stable matching for that iteration. Otherwise, let \(\pi'\) be the permutation obtained from \(\pi\) by removing the \(a_{jk}\)’s and \(b_{jk}\)’s. Let \(I'\) be the instance obtained from \(I^*\) by removing the same set of agents. Then it must be the case that ROM(\(\pi', I'\)) can simulate how ROM(\(\pi, I^*\)) matched the agents in \(I'\) so that the output of ROM(\(\pi', I'\)) is a sub-matching of ROM(\(\pi, I^*\)). But the former will only output \(\alpha_1\) or \(\alpha_3\) and therefore \(\mu^*\) cannot be the output of ROM(\(\pi, I^*\)). Since this is a contradiction, some \(m_j\) must be temporarily matched to \(b_{j1}\) during the execution of ROM(\(\pi, I^*\)).

Now, consider an arbitrary \(b_{j1}\). If it appears second or third in the ordering of \(a_{j1}, a_{j2}, b_{j1}\) and \(b_{j2}\) in \(\pi\), \(b_{j2}\) must appear last in the ordering because of (i). Thus, either \(a_{j1}\) or \(a_{j2}\) appears first. When ROM begins to process \(b_{j1}\), at least one of these men will be matched to \(b_{j1}\) immediately so \(b_{j1}\) will never be temporarily matched to \(m_j\) in this iteration. In the later iterations, \(b_{j1}\) may change her partner from \(a_{j1}\) to \(a_{j2}\) but other than this change \(b_{j1}\) will never be matched to any one else.

Suppose \(b_{j1}\) appears last in the ordering of \(a_{j1}, a_{j2}, b_{j1}\) and \(b_{j2}\) in \(\pi\). Using the same reasoning in the first paragraph, \((a_{j1}, b_{j2})\) will be part of the stable matching just before ROM processes \(b_{j1}\) while \(a_{j2}\) is unmatched. Thus, when ROM processes \(b_{j1}\), she will be immediately matched to \(a_{j2}\) and \(b_{j2}\) will be the resulting sub-matching until the end of ROM. Throughout the execution of ROM, \(b_{j1}\) will never be matched to anyone else.
Thus, the only way for \( b_{j1} \) to be temporarily matched to \( m_j \) is for her to appear first in the ordering of \( a_{j1}, a_{j2}, b_{j1}, b_{j2} \) in \( \pi \) and consequently for \( b_{j2} \) to appear last because of (i). Since this must be true for some \( b_{j1} \), (ii) follows.

Now consider an arbitrary 3-SAT instance \( \Phi \) with \( n \) variables \( x_1, x_2, \ldots, x_n \) and \( q \) clauses \( C_1, C_2, \ldots, C_q \). Our goal is to construct an SMI instance \( I_\Phi \) so that a particular stable matching of \( I_\Phi \) is reachable by ROM if and only if \( \Phi \) is satisfiable. For each variable \( x_j \), create the sub-instance

\[
\begin{align*}
a_{j1}: & \quad b_{j1} \quad b_{j2} \\
\alpha: & \quad a_{j2} \quad a_{j1} \quad \cdots \\
a_{j2}: & \quad b_{j2} \quad b_{j1}
\end{align*}
\]

where the dots indicate that \( b_{j1} \) and \( b_{j2} \) may have more agents in their preference lists. Let us call \( b_{j1} \) and \( b_{j2} \) twins and \( a_{j1} \) and \( a_{j2} \) the counterparts of \( b_{j1} \) and \( b_{j2} \). For each clause \( C_i \), create the sub-instance

\[
\begin{align*}
m_{i1}: & \quad w_{i1} \quad w_{i2} \quad z_{i1} \quad w_{i3} \quad w_{i1} \\
m_{i2}: & \quad w_{i2} \quad w_{i3} \quad z_{i2} \quad w_{i1} \quad w_{i2} \\
m_{i3}: & \quad w_{i3} \quad w_{i1} \quad z_{i3} \quad w_{i2} \quad w_{i3} \\
\end{align*}
\]

where \( z_{ik}, k = 1, 2, 3 \), is based on the \( k \)th literal in \( C_j \). If this literal is \( x_j \), set \( z_{ik} \) to \( b_{j1} \) and add \( m_{ik} \) to the end of \( b_{j1} \)’s preference list; otherwise, if the literal is \( \overline{x}_j \), set \( z_{ik} \) to \( b_{j2} \) and add \( m_{ik} \) to the end of \( b_{j2} \)’s preference list. Thus, when we restrict \( I_\Phi \) to the participants associated with \( C_j \) and its variables, the sub-instance looks just like the instance \( I^* \) we considered with the exception that \( b_{j2} \) may sometimes play the role of \( b_{j1} \) and vice versa. For \( i = 1, \ldots, q \), let

\[
\begin{align*}
\alpha_{i1} = \{ (m_{i1}, w_{i1}), (m_{i2}, w_{i2}), (m_{i3}, w_{i3}) \} \\
\alpha_{i2} = \{ (m_{i1}, w_{i2}), (m_{i2}, w_{i3}), (m_{i3}, w_{i1}) \} \\
\alpha_{i3} = \{ (m_{i1}, w_{i3}), (m_{i2}, w_{i1}), (m_{i3}, w_{i2}) \}
\end{align*}
\]

Define \( \beta_{j1} \) and \( \beta_{j2} \) as before for \( j = 1, \ldots, n \). Again, it is straightforward to verify that the stable matchings of \( I_\Phi \) are exactly of the form \( \alpha_{1g_1} \cup \alpha_{2g_2} \cup \cdots \cup \alpha_{qg_q} \cup \beta_{1k_1} \cup \beta_{2k_2} \cup \cdots \cup \beta_{nk_n} \) where each \( g_i \in \{1, 2, 3\} \) and each \( k_j \in \{1, 2\} \).

Let \( \mu^{**} = \alpha_{12} \cup \alpha_{22} \cup \cdots \cup \alpha_{q2} \cup \beta_{12} \cup \beta_{22} \cup \cdots \cup \beta_{n2} \). When we restrict \( \mu^{**} \) to the agents associated with clause \( C_j \) and its literals, \( \mu^{**} \) is just like \( \mu^* \). It is, however, much trickier for ROM to reach \( \mu^{**} \) because \( b_{j1} \) and \( b_{j2} \) can be part of other sub-instances. The two women cannot simultaneously help obtain their sub-instances’ “middle” stable matchings since one of them has to appear last in the ordering of \( a_{j1}, a_{j2}, b_{j1}, b_{j2} \). We are now ready to prove our main result.

**Theorem 2**  The 3-SAT instance \( \Phi \) has a satisfying assignment if and only if there is a permutation \( \pi \) of \( I_\Phi \)'s agents so that the output of \( \text{ROM}(\pi, I_\Phi) \) is \( \mu^{**} \).

**Proof**  Let \( f \) be a satisfying assignment of \( \Phi \). We now order the agents of \( I_\Phi \) based on \( f \). Initially set \( \pi_f \) to the empty sequence. For \( i = 1 \) to \( q \), add \( m_{i1}, m_{i2}, m_{i3}, w_{i1}, w_{i2}, w_{i3} \) to the end of \( \pi_f \). We call this the first part of the sequence. Next, for \( j = 1 \) to \( n \), if \( f(x_j) = 1 \), add \( b_{j1} \) to the end of \( \pi_f \); otherwise add \( b_{j2} \) to the end of \( \pi_f \). We call this the second part of the sequence. Finally, for \( j = 1 \) to \( n \), if \( f(x_j) = 1 \), add \( a_{j1}, a_{j2}, b_{j2} \) to the end of \( \pi_f \); otherwise, add \( a_{j1}, a_{j2}, b_{j1} \) to the end of \( \pi_f \). We call this the third part of the sequence.

Now consider what happens in \( \text{ROM}(\pi_f, I_\Phi) \). After ROM processes the first part of \( \pi_f \), the resulting stable matching is \( \alpha_{13} \cup \alpha_{23} \cup \cdots \cup \alpha_{q3} \). Next, ROM processes the second part of \( \pi_f \). Suppose \( b_{j3} \) is one of the women in this sequence. Let \( C_{i1}, C_{i2}, \ldots, C_{ik} \) be the clauses that have \( x_j \) as a literal. Without loss of generality, assume that \( x_j \) is their first literal so that \( m_{i1}, m_{i2}, \ldots, m_{i1} \) appear
after \(a_{j2}\) and \(a_{j1}\) in \(b_{j1}\)’s preference list. Consider the beginning of the iteration that processes \(b_{j1}\). There are two possible cases:

1. \(\alpha_{i3}\) is part of the current stable matching. Then \((m_{i1}, b_{j1})\) is a blocking pair of \(\alpha_{i3}\). As a result, \(m_{i1}\) rejects \(w_{i3}\), and \(w_{i3}\) will in turn propose to her second choice who then accepts her proposal. Blocking pairs will continue to get resolved until \(\alpha_{i2}\) replaces \(\alpha_{i3}\) as a sub-matching of the current stable matching and \(b_{j1}\) is unmatched.

2. \(\alpha_{i3}\) is not part of the current stable matching. This implies that in a prior iteration, case (1) happened (via the agent associated with the second or third literal of \(C_{i1}\)) and \(\alpha_{i2}\) already replaced \(\alpha_{i3}\) as a sub-matching. Since none of the women in the second part of \(\pi_f\) can form a blocking pair with the agents in \(\alpha_{i1}, \alpha_{i2}\) is still a sub-matching of the current stable matching. Furthermore, \(b_{j1}\) will just skip over \(m_{i1}\) and remain unmatched.

Thus, after \(b_{j1}\) considers \(m_{i1}\), \(\alpha_{i2}\) is a sub-matching of the current stable matching and \(b_{j1}\) is unmatched. Similar results apply as \(b_{j1}\) considers \(m_{i1}, \ldots, m_{i4}\) so that at the end of the iteration that processes \(b_{j1}\), \(\alpha_{i2}\) has replaced \(\alpha_{i3}\) for all clauses \(C_i\) that have \(x_j\) as a literal. Additionally, \(b_{j1}\) is unmatched.

When \(b_{j2}\) instead of \(b_{j1}\) is in the second part of the sequence, \(\alpha_{i2}\) will replace \(\alpha_{i3}\) for all clauses \(C_i\) that have \(x_j\) as a literal at the end of the iteration that processes \(b_{j2}\). Additionally, \(b_{j2}\) is unmatched. Hence, once ROM processes the second part of \(\pi_f\), the resulting stable matching is \(\alpha_{i2} \cup \alpha_{i3} \cup \cdots \cup \alpha_{i4}\) because \(f\) is a satisfying assignment of \(\Phi\). All of the women that appears in the second part of \(\pi_f\) are unmatched.

Finally, after ROM processes the third part of \(\pi_f\), it is easy to see that \(\beta_{i2} \cup \cdots \beta_{i4}\) becomes part of the output. Moreover, \(\alpha_{i2} \cup \alpha_{i3} \cup \cdots \cup \alpha_{i4}\) remains unchanged because none of the agents in the third part of \(\pi_f\) forms a blocking pair with the agents of this sub-matching. We have shown that \(\mu^{**}\) is the output of ROM(\(\pi_f, I_2\)).

We prove the converse next. Suppose ROM(\(\pi, I_2\)) outputs \(\mu^{**}\). When we restrict \(I_2\) to the agents associated with \(C_i\) and its variables, the instance is just like our first example \(I\). Similarly, when we restrict \(\mu^{**}\) to the same set of agents, the stable matching looks just like \(\mu^{*}\) of \(I\). Thus, a result like Lemma 2 should apply to the ordering of the agents in \(\pi\). We restate part \((ii)\) as follows:

(iii) For \(i = 1, \ldots, q\), there is some \(k \in \{1, 2, 3\}\) so that \(z_{ik}\) appears first and her twin appears last in the ordering of \(z_{ik}\), her twin and their counterparts in \(\pi\).

We now construct a truth assignment \(f_\pi\) as follows: for \(j = 1, \ldots, n\), Set \(f_\pi(x_j)\) to 0 if \(b_{j1}\) is the last agent to appear in \(\pi\) among \(a_{j1}, a_{j2}, b_{j1}, b_{j2}\); otherwise set \(f_\pi(x_j)\) to 1. We know from \((i)\) that when \(f_\pi(x_j)\) is set to 1, \(b_{j2}\) is the last agent to appear in \(\pi\) among \(a_{j1}, a_{j2}, b_{j1}, b_{j2}\). Additionally, from \((ii')\), we know that \(f_\pi\) has set one of the literals in \(C_i\) to 1 for \(i = 1, \ldots, q\). Thus, \(f_\pi\) is a satisfying assignment for \(\Phi\).

**Corollary 1** ROM-Reachability is NP-complete even in the restricted case when all agents have a preference list of length at most 4.

**Proof** Given 3-SAT instance \(\Phi\) with \(n\) variables and \(q\) clauses, we created an instance \(I_2\) that has \(4n + 6q\) agents so that \(\Phi\) is satisfiable if and only if a particular stable matching of \(I_2\) is reachable by ROM. Thus, 3-SAT is polynomially reducible to ROM-Reachability. Additionally, it is easy to verify that ROM-Reachability is in NP. It follows that ROM-Reachability is NP-complete.

But it is also known that 3-SAT is NP-complete even in the special case when each literal appears twice among the clauses – i.e., there are two clauses that contain \(x_i\) and two other clauses that contain \(\pi_i\) for \(i = 1, \ldots, n\) [6]. When \(\Phi\) is such an instance, then in \(I_2\) both \(b_{j1}\) and \(b_{j2}\)
have exactly two unstable partners in their preference lists, for \( j = 1, \ldots, n \). The restriction on ROM-Reachability follows. □

4 Strongly ROM-Reachable Stable Matchings

Recall that a stable matching \( \mu \) of \( I \) is strongly ROM-reachable if there is a permutation \( \pi \) of the agents of \( I \) so that (i) ROM(\( \pi, I \)) outputs \( \mu \) and (ii) \( \mu_1 \subseteq \mu_2 \subseteq \cdots \subseteq \mu_{|\pi|} = \mu \) where \( \mu_i \) is the stable matching at the end of iteration \( i \) of ROM(\( \pi, I \)). Call \( \pi \) a permutation associated with \( \mu \). Notice that the definition implies that once an agent is matched in some \( \mu_i \), his or her partner must be the same one as in \( \mu \) and remains so until the end of ROM. Intuitively, it is easier to determine if a stable matching is strongly ROM-reachable because ROM can build it one pair at a time.

**Proposition 2** Let \( \mu \) be a strongly ROM-reachable stable matching of \( I \), and let \( \pi \) be a permutation associated with \( \mu \). Suppose \((m, w) \in \mu, \pi(k) = m, \pi(k') = w \) and \( k < k' \). Then for \( i = k, k + 1, \ldots, k' - 1, m \) is unmatched in \( \mu_i \) while for \( i = k', k' + 1, \ldots, |A|, (m, w) \in \mu_i \).

In [12], Hoffman et al. defined the jealousy graph of a stable matching \( \mu \), \( J(\mu) \) as follows: The vertices of \( J(\mu) \) are the pairs in \( \mu \), and there is a directed edge from the pair \((m, w)\) to another pair \((m', w')\) whenever \( m' \) prefers \( w \) to \( w' \) or \( w' \) prefers \( m \) to \( m' \). In this section, we consider a “labeled” version of \( J(\mu) \). Let \( L \) be a labeling that assigns each agent of \( I \) as lucky or unlucky. We say that \( L \) respects \( \mu \) if for every pair in \( \mu \) one agent is labeled lucky while the other is labeled unlucky. For such a labeling, denote as \( J_L(\mu) \) the graph whose vertices are the pairs in \( \mu \) such that there is a directed edge from \((m, w)\) to \((m', w')\) if \( w \) is an unlucky agent and \( m' \) prefers \( w \) to \( w' \) or \( m \) is an unlucky agent and \( w' \) prefers \( m \) to \( m' \). Thus, \( J_L(\mu) \) is a subgraph of \( J(\mu) \) and keeps only the edges “caused” by the agents labeled unlucky by \( L \). Here now is our characterization of strongly ROM-reachable stable matchings based on labeled jealousy graphs.

**Theorem 3** A stable matching \( \mu \) of \( I \) is strongly ROM-reachable if and only if there is a labeling \( L \) that respects \( \mu \) such that \( J_L(\mu) \) is acyclic.

**Proof** Suppose \( \mu \) is a strongly ROM-reachable stable matching of \( I \). Let \( \pi \) be a permutation that is associated with \( \mu \). For each pair \((m, w) \in \mu \), label the agent that appears first in \( \pi \) as unlucky and the agent that appears later in \( \pi \) as lucky. For the unmatched agents, arbitrarily label them as lucky or unlucky. Call the labeling \( L^* \). We argue that \( J_{L^*}(\mu) \) is acyclic next.

Order the pairs of \( \mu \) based on when the pairs’ lucky agents appeared in \( \pi \). Denote the ordering as \( p_1, p_2, \ldots, p_{|\mu|} \). Thus, among all pairs in \( \mu \), \( p_1 \)’s lucky agent appeared first in \( \pi \) as unlucky, and the agent that appears later in \( \pi \) as lucky. For the unmatched agents, arbitrarily label them as lucky or unlucky. Call the labeling \( L^* \). We argue that \( J_{L^*}(\mu) \) is acyclic next.

Let us now prove the converse. Suppose \( J_L(\mu) \) is acyclic. To prove that \( \mu \) is strongly ROM-reachable, we need to show that there is a permutation that is associated with \( \mu \). Let \( p_1, p_2, \ldots, p_{|\mu|} \) be a topological ordering of \( J_L(\mu) \). Construct \( \pi \) by making its \((2j - 1)\)st agent be the unlucky agent in \( p_j \) and the \((2j)\)th agent be the lucky agent in \( p_j \) for \( j = 1, \ldots, |\mu| \). Then add any unmatched agents in \( \mu \) to the end of the sequence. Consider ROM(\( \pi, I \)) next.
Claim: During the execution of ROM(\(\pi, I\)), \(\mu_{2j-1} = \mu_{2j-2}\) while \(\mu_{2j} = \mu_{2j-2} \cup \{p_j\}\) for \(j = 1, \ldots, |\mu|\).

Proof of claim: It is clear that \(\mu_1\) is an empty matching while \(\mu_2 = \{p_1\}\). So assume that the claim is true for \(j = 1, \ldots, t'\). Without loss of generality, let \(m\) be an unlucky agent in \(p_{t'+1} = (m, w)\). When ROM processes \(m\), \(m\) can propose to the women on his list that are also in \(p_1, \ldots, p_{t'}\). These may include women that he prefers less over \(w\). If such a woman prefers \(m\) to her current partner, it would mean that there is a directed edge from \(p_{t'+1}\) to \(p_k\) for some \(k < t' + 1\) in \(J_L(\mu)\), a contradiction. Thus, every woman that \(m\) proposes to rejects him. It follows that at the end of iteration \(2t' + 1\), \(\mu_{2t'+1} = \mu_{2t'}\) and \(m\) is unmatched.

When ROM processes \(w, w\) will begin by proposing to the men on her list that she prefers over \(m\) and who are also in \(p_1, \ldots, p_{t'}\). But every such man \(m'\) must prefer his current partner over \(w\) because not doing so will mean that \((m', w)\) is a blocking pair of \(\mu\). Since this cannot be the case, every man that \(w\) proposes to before \(m\) rejects her. Thus, \(w\) will propose to \(m\) and he will accept because he is unmatched. At the end of the iteration \(2t' + 2\), \(\mu_{2t'+2} = \mu_{2t'} \cup \{p_{t'+1}\}\). By induction, we have shown that the claim is true.

Finally, when ROM processes an unmatched agent \(a\) in \(\mu\), none of the agents \(a\) proposes to will accept the proposal since it would mean that \(\mu\) has a blocking pair. Thus, after ROM has processed all the agents in \(p_1, \ldots, p_{|\mu|}\), the stable matching is \(\mu\) and will remain so until the end of the ROM. We have shown that \(\pi\) is a permutation that accompanies \(\mu\) so \(\mu\) is a strongly ROM-reachable stable matching of \(I\). □

So how do we take advantage of Theorem 3 to determine if a stable matching \(\mu\) is strongly ROM-reachable? There are at least \(2^{|\mu|}\) labelings of the agents of \(I\) that respect \(\mu\) so the brute force method of checking if one of the labelings \(L\) yields an acyclic \(J_L(\mu)\) is infeasible. First, we note that strongly ROM-reachable stable matchings are made up of strongly ROM-reachable stable sub-matchings.

Lemma 3 Suppose \(\mu\) is a strongly ROM-reachable stable matching of \(I\). Let \(\mu' \subseteq \mu\). Then \(\mu'\) is also a strongly ROM-reachable stable matching of \(I_{|\mu'|}\), the instance obtained by restricting \(I\) to the agents in \(\mu'\).

Proof Since \(\mu\) is a strongly ROM-reachable stable matching, by Theorem 3 there is a labeling \(L\) that respects \(\mu\) such that \(J_L(\mu)\) is acyclic. Additionally, \(\mu'\) has to be a stable matching of \(I_{|\mu'|}\); otherwise, if it has a blocking pair then so will \(\mu\). Restrict \(L\) to the agents in \(\mu'\) and call it \(L'\). Clearly, \(L'\) respects \(\mu'\) and \(J_{L'}(\mu')\) is acyclic since it is a subgraph of \(J_L(\mu)\). It follows that \(\mu'\) is a strongly ROM-reachable stable matching of \(I_{|\mu'|}\). □

Second, we define the notion of a sink agent whose name is meant to suggest that it behaves like the sink node of a directed acyclic graph. We shall say that a stable matching \(\tau\) of \(I\) (not necessarily strongly ROM-reachable) has a sink agent if

(i) there is a man \(m\) so that \(\tau(m) = w\) is his best stable partner in \(I\) and for any other pair \((m', w') \in \tau\), \(m'\) does not prefer \(w\) to his partner \(w'\) or
(ii) there is a woman \(w\) so that \(\tau(w) = m\) is her best stable partner in \(I\) and for any other pair \((m', w') \in \tau\), \(w'\) does not prefer \(m\) to her partner \(m'\).

In (i), we say \(m\) is a sink agent of \(\tau\) while in (ii) \(w\) is a sink agent of \(\tau\).

Lemma 4 Every strongly ROM-reachable stable matching \(\mu\) of \(I\) has a sink agent.
**Proof** Let $\pi$ be a permutation that is associated with $\mu$. Consider the very last matched agent that appears in $\pi$. Without loss of generality, let this agent be $m$ who is matched to $w$ in $\mu$, and let $\pi(i) = m$. This means that $\mu_i = \mu$ but that $w$ is unmatched in $\mu_{i-1}$. The latter implies that for every other pair $(m', w')$ in $\mu$, $m'$ preferred $w'$ to $w$. Hence, $m$ is a sink agent of $\mu$. If the last matched agent that appears in $\pi$ is a woman $w$, a similar proof will show that $w$ is a sink agent of $\mu$. □

Our algorithm for determining if a stable matching is strongly ROM-reachable is patterned after the standard algorithm for topologically sorting a directed acyclic graph.

CheckDirectROM$(\tau, I)$

Set $i = 1$, $I_1 = I$ and $\tau_1 = \tau$.
While $\tau_i$ has a sink agent $a$
    let $p_i$ be the pair that consists of $a$ and $\tau_i(a)$
    label $a$ as lucky and $\tau_i(a)$ as unlucky
    $\tau_{i+1} \leftarrow \tau_i - \{p_i\}$ and $I_{i+1} \leftarrow I_{|\tau_{i+1}|}$
    $i \leftarrow i + 1$
If $\tau_i$ is empty return ("yes"; $p_1, p_2, \ldots, p_{|\tau|}$)
Else return ("no"; $\tau_i$)

**Theorem 4** CheckDirectROM correctly determines if a stable matching $\tau$ of $I$ is strongly ROM-reachable in $O(|\tau| \times |I|)$ time. In each case, the algorithm also returns a certificate that can be used to verify that the algorithm’s answer is correct.

**Proof** Since every $\tau_i$ is a subset of $\tau$, if $\tau$ is a strongly-ROM reachable stable matching of $I$, then every $\tau_i$ is a strongly ROM-reachable stable matching of $I_{|\tau_i|}$ according to Lemma 3. By Lemma 4, every $\tau_i$ has a sink agent. Thus, the algorithm is correct in concluding that when some $\tau_i$ has no sink agent, the input $\tau$ is not a strongly ROM-reachable stable matching. The lack of a sink agent in $\tau_i$ is evidence that $\tau$ is not a strongly ROM-reachable stable matching.

Let $|\tau| = n$. Now suppose that $\tau_1, \ldots, \tau_n$ have sink agents. Let $L$ be the labeling that assigns each sink agent in $p_i$ as lucky and the partner as unlucky; the unmatched agents are labeled arbitrarily. By the definition of sink agents, $p_n, p_{n-1}, \ldots, p_1$ is a topological ordering of $J_L(\mu)$ because $p_i$ will not have any edges to $p_{i+1}, \ldots, p_n$ in $J_L(\mu)$. Thus, $\mu$ is a strongly ROM-reachable stable matching of $I$, and the permutation $\pi$ based on $p_n, p_{n-1}, \ldots, p_1$ as described in the proof of Theorem 3 can be used to verify that ROM($\pi, I$) outputs $\tau$.

Finally, to determine if $\tau_i$ has a sink agent, we run the Gale-Shapley algorithm to identify every agent’s best stable partner. The algorithm can be implemented in $O(|I|)$ time. Next, for each agent $a$ matched to their best stable partner, we check if there is a person who prefers $\tau_i(a)$ over their current partner in $\tau_i$. We can do this by going through the preference list of $\tau_i(a)$ and, for each person $b$ that appears in this list, we compare the rank of $\tau_i(b)$ and $\tau_i(a)$ in $b$’s preference list. Using the appropriate data structure so that rank-checking can be done in $O(1)$ time, this step can again be implemented in $O(|I|)$ time. But there can be $n$ $\tau_i$’s so implementing CheckDirectROM takes $O(|\tau| \times |I|)$ time. □
5 Extreme Stable Matchings

A stable matching is an extreme stable matching if for every pair in the stable matching, either the man or the woman is matched to his/her best stable partner (and consequently the other person is matched to his/her worst stable partner). These stable matchings are interesting because they do not have the “middle” sub-matchings like the \( \alpha_{i2} \)’s that \( \mu^* \) had when we proved the NP-completeness of ROM-Reachability. Are all extreme stable matchings ROM-reachable?

**Theorem 5** Let \( I \) be an SMI instance where every agent has at most one unstable partner. Then every extreme stable matching \( \mu \) of \( I \) is strongly ROM-reachable.

**Proof** Let \( \mu \) be an extreme stable matching of \( I \). For each pair \((m, w) \in \mu\), label the agent matched to his/her worst stable partner as unlucky and the other agent as lucky.\(^2\) Call the labeling \( L \). Clearly, \( L \) respects \( \mu \). To prove the theorem, we will show that \( J_L(\mu) \) is acyclic.

Suppose this is not the case and the pairs \((a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)\) form a directed cycle in \( J_L(\mu) \). (For the rest of the proof, addition and subtraction operations are modulo \( k \).) First, we note that for any two consecutive pairs in this cycle \((a_j, b_j)\) and \((a_{j+1}, b_{j+1})\), the unlucky agent in \((a_j, b_j)\) cannot be a stable partner in \( I \) with the agent of opposite gender in \((a_{j+1}, b_{j+1})\). To see this, assume without loss of generality that \( a_j \) is the unlucky agent in \((a_j, b_j)\) so that \( b_j \) is the worst stable partner of \( a_j \). The edge from \((a_j, b_j)\) to \((a_{j+1}, b_{j+1})\) implies that \( b_{j+1} \) prefers \( a_j \) over \( a_{j+1} \). If \( a_j \) and \( b_{j+1} \) are stable partners, \( a_j \) must prefer \( b_{j+1} \) over \( b_j \) making \((a_j, b_{j+1})\) a blocking pair of \( \mu \). Since this cannot be the case, \( a_j \) and \( b_{j+1} \) are not stable partners in \( I \).

Let us now consider \((a_1, b_1)\) and \((a_2, b_2)\). Again, without loss of generality, suppose \( a_1 \) is the unlucky agent in \((a_1, b_1)\). Then \( a_1 \) and \( b_2 \) are unstable partners in \( I \). Next, consider \((a_2, b_2)\) and \((a_3, b_3)\). If \( b_2 \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\). Applying the same argument around the cycle, we conclude that for each pair \((a_j, b_j)\), \( a_j \) is the unlucky agent in \((a_2, b_2)\) then \( b_2 \) and \( a_3 \) are also unstable partners in \( I \). But \( b_2 \) can have at most one unstable partner by assumption so \( a_2 \) must be the unlucky agent in \((a_2, b_2)\).

Let \( \mu' \) be the matching obtained by removing the pairs \((a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)\) from \( \mu \) and replacing them with \((a_1, b_2), (a_2, b_3), \ldots, (a_k, b_1)\). We will now argue that \( \mu' \) has to be a stable matching of \( I \) too. Suppose \( \mu' \) is not a stable matching so that it has a blocking pair. Clearly, one of the agents in the blocking pair must be from the set \( \{a_1, \ldots, a_k, b_1, \ldots, b_k\} \); otherwise, the same pair will be blocking \( \mu \) as well. Additionally, none of \( b_1, \ldots, b_k \) are part of the blocking pair since each one is matched to her first choice. So suppose the blocking pair is \((a_j, b_j')\). Now, the partner of \( b_j' \) in \( \mu \) and \( \mu' \) are the same but \( a_j \) prefers \( b_j \), his partner in \( \mu \), over his partner \( b_{j+1} \), his partner in \( \mu' \). Furthermore, \( b_j \) and \( b_{j+1} \) are next to each other in \( a_j \)’s preference list. Thus, if \((a_j, b_j')\) is a blocking pair of \( \mu' \), then it is also a blocking pair of \( \mu \). Since the latter cannot be true, \( \mu' \) has no blocking pairs and must be a stable matching of \( I \).

But we already noted that \( a_j \) and \( b_{j+1} \) are unstable partners in \( I \). It must be the case then that \( J_L(\mu) \) is acyclic and, consequently, \( \mu \) is strongly ROM-reachable. \( \square \)

In the next lemma, we show that when we relax the condition on Theorem 5 and allow agents to have two unstable partners in \( I \), an extreme stable matching of \( I \) may no longer be ROM-reachable.

---

\(^2\)If \((m, w)\) is a fixed pair – i.e., they are matched to each other in all of the stable matchings of \( I \), then they are each other’s best and worst stable partners. Arbitrarily label one as lucky and the other as unlucky.
Lemma 5 There is an SMI instance whose agents have at most two unstable partners, and this instance has an extreme stable matching that is not ROM-reachable.

Proof Consider the following SMI instance $I$:

$\begin{align*}
m_1: & \quad w_1 \quad w_2 \\
m_2: & \quad w_2 \quad w_1 \\
m_3: & \quad w_1 \quad w_3 \quad w_4 \quad w_8 \\
m_4: & \quad w_2 \quad w_4 \quad w_3 \quad w_7 \\
m_5: & \quad w_7 \quad w_5 \quad w_6 \quad w_2 \\
m_6: & \quad w_8 \quad w_6 \quad w_5 \quad w_1 \\
m_7: & \quad w_7 \quad w_8 \\
m_8: & \quad w_8 \quad w_7
\end{align*}$

For $i = 1, 2, 3, 4$, let $\alpha_{i1} = \{(m_{2i-1}, w_{2i-1}), (m_{2i}, w_{2i})\}$ and $\alpha_{i2} = \{(m_{2i-1}, w_{2i}), (m_{2i}, w_{2i-1})\}$. It is easy to verify that the man-optimal stable matching is $\alpha_{11} \cup \alpha_{21} \cup \alpha_{31} \cup \alpha_{41}$, the woman-optimal stable matching is $\alpha_{12} \cup \alpha_{22} \cup \alpha_{32} \cup \alpha_{42}$, and that the set of stable matchings of $I$ is $\{\alpha_{1j_1} \cup \alpha_{2j_2} \cup \alpha_{3j_3} \cup \alpha_{4j_4} | j_1, j_2, j_3, j_4 \in \{1, 2\}\}$. In other words, each agent has exactly two stable partners, and every stable matching of $I$ is an extreme stable matching. Furthermore, the agents $m_3, m_4, m_5, m_6, w_1, w_2, w_7, w_8$ all have two unstable partners. We will now prove that $\tau^* = \alpha_{11} \cup \alpha_{22} \cup \alpha_{32} \cup \alpha_{41}$ is not reachable by ROM.

Let $M_1$ and $W_1$ denote the set of men and women who are matched to their best stable partners in $\tau^*$. Thus, $M_1 = \{m_1, m_2, m_7, m_8\}$ while $W_1 = \{w_3, w_4, w_5, w_6\}$. If there is a permutation $\pi$ of $I$’s agents so that ROM($\pi, I$) outputs $\tau^*$, the last agent in $\pi$ must belong to $M_1 \cup W_1$.

Suppose that the last agent in $\pi$ is $m_1$. Consider the instance prior to ROM processing $m_1$, $I_{-\{m_1\}}$. Since there is one more woman than man in the instance, at least one woman is unmatched in all the stable matchings of $I_{-\{m_1\}}$. In this case, it is $w_3$. (The reader can verify this by computing the man-optimal matching of $I_{-\{m_1\}}$.) According to Fact 2(b), this means that when ROM finally processes $m_1$, the woman $w_3$ will be matched to her man-optimal stable partner in $I$, which is $m_3$. Thus, the output of ROM($\pi, I$) is not $\tau^*$. A similar argument can be used to show why none of the agents in $M_1 \cup W_1$ can be the last agent in $\pi$. It follows that $\pi$ does not exist. □

By using the above instance as a gadget, we can prove the next theorem. The proof is similar to that of Theorem 2.

Theorem 6 Let $I$ be an SMI instance where agents can have two or more unstable partners. Let $\mu$ be an extreme stable matching of $I$. Then determining if $\mu$ is ROM-reachable is NP-complete.

Proof Let $\Phi$ be a 3-SAT instance with $n$ variables $x_1, x_2, \ldots, x_n$ and $q$ clauses $C_1, C_2, \ldots, C_q$. For each variable $x_i$, create the sub-instance

$\begin{align*}
a_{i1}: & \quad b_{i1} \quad b_{i2} \quad \cdots \quad b_{i1} \quad a_{i2} \quad a_{i1} \\
a_{i2}: & \quad b_{i2} \quad b_{i1} \quad \cdots \quad b_{i2} \quad a_{i1} \quad a_{i2}
\end{align*}$

where the dots indicate that $a_{i1}$ and $a_{i2}$ may have more agents in their preference lists. For each clause $C_j$, create the sub-instance

$\begin{align*}
a_{j1}: & \quad b_{j1} \quad b_{j2} \quad \cdots \quad b_{j1} \quad a_{j2} \quad a_{j1} \\
a_{j2}: & \quad b_{j2} \quad b_{j1} \quad \cdots \quad b_{j2} \quad a_{j1} \quad a_{j2}
\end{align*}$
where $z_{jk}$, $k = 1, 2, 3$ is based on the $k$th literal in $C_j$. If this literal is $x_i$, set $z_{jk}$ to $a_{i1}$; otherwise, set $z_{jk}$ to $a_{i2}$. Add $w_{j3}$ or $w_{j4}$ to to the preference list of $z_{jk}$ depending on whose preference list $z_{jk}$ appears in. For $j = 1, \ldots, m$, let

$$
\tau_j^* = \{(m_{j1}, w_{j1}), (m_{j2}, w_{j2}), (m_{j3}, w_{j3}), (m_{j4}, w_{j4}), (m_{j5}, w_{j5}), (m_{j6}, w_{j6}), (m_{j7}, w_{j7}), (m_{j8}, w_{j8})\}
$$

and for $i = 1, \ldots, n$ let $\beta_i = \{(a_{i1}, b_{i1}), (a_{i2}, b_{i2})\}$. It is easy to verify that

$$
\mu^{**} = \tau_1^* \cup \tau_2^* \cup \ldots \cup \tau_m^* \cup \beta_{i1} \cup \beta_{i2} \cup \ldots \cup \beta_{i1}
$$

is a BW-stable matching of $I_{\Phi}$.

Claim: The 3-SAT instance $\Phi$ has a satisfying assignment if and only if there is a permutation $\pi$ of $I_{\Phi}$'s agents such that the output of $\text{ROM}(\pi, I_{\Phi})$ is $\mu^{**}$.

We omit the proof of the claim as it is very similar to that of Theorem 2. Given 3-SAT instance $\Phi$ with $n$ variables and $q$ clauses, we have created an instance $I_{\Phi}$ that has $4n + 16q$ agents so that $\Phi$ is satisfiable if and only if $\mu^{**}$, a BW-stable matching of $I_{\Phi}$, is reachable by ROM. The theorem follows. □

6 Conclusion

We investigated the stable matchings that Ma’s Random Order Mechanism (ROM) [15] can reach starting from the empty matching. Since ROM induces a probability distribution on a instance’s set of stable matchings, we were equivalently interested in the stable matchings that are in the support of ROM. In the first half of the paper, we showed that it is computationally easy to determine if some non-trivial stable matching is in the support of ROM but it is NP-complete to determine if a particular stable matching lies in the support of ROM.

In the second half of the paper, we introduced the notion of a strongly ROM-reachable stable matchings which are stable matchings that ROM can reach in a “direct” manner. We provided a nice characterization and presented an efficient recognition algorithm for these stable matchings. Interestingly, strongly ROM-reachable stable matchings are also relevant to the Employment by Lotto (EBL) mechanism we described in the introduction. Suppose $\mu$ is a strongly ROM-reachable stable matching, and $\pi$ is the permutation found by CheckDirectROM. It is not difficult to show that when the reverse of $\pi$, $\pi^r$, is the input to the Employment by Lotto (EBL), the output is again $\mu$. That is, in the context of strongly ROM-reachable stable matchings, ROM and EBL are “equivalent” to each other.

Question: What are the stable matchings $\mu$ for which there is a permutation $\pi$ so that $\text{ROM}(\pi, I) = \text{EBL}(\pi^r, I) = \mu$? Do they have to be strongly ROM-reachable? What precisely are the stable matchings that are both ROM-reachable and EBL-reachable?
Lastly, we defined the class of extreme stable matchings and showed that the computational complexity of determining if ROM can output an extreme stable matching is dependent on the number of unstable partners of the agents. One interesting avenue of research is to investigate the stable matchings that can be reached by ROM when agents are allowed to enter as well as leave the market.

**Question:** Might some stable matchings which were not reachable by ROM in our current setting be reachable in the setting where agents are also allowed to leave?

**Acknowledgments.** I would like to thank David Manlove, Péter Biró, Daniel Moeller and Ágnes Cseh whose insightful questions, comments and suggestions helped shape the direction of this research.

**References**


