Probably Correct Optimal Stable Matching for Two-Sided Markets Under Uncertainty

Andreas Athanasopoulos
University of Neuchatel

Anne-Marie George
University of Oslo

Christos Dimitrakakis
University of Neuchatel

Abstract

We consider a learning problem for the stable marriage model under unknown preferences for the left side of the market. We focus on the centralized case, where at each time step, an online platform matches the agents, and obtains a noisy evaluation reflecting their preferences. Our aim is to quickly identify the stable matching that is left-side optimal, rendering this a pure exploration problem with bandit feedback. We specifically aim to find Probably Correct Optimal Stable Matchings and present several bandit algorithms to do so. Our findings provide a foundational understanding of how to efficiently gather and utilize preference information to identify the optimal stable matching in two-sided markets under uncertainty. An experimental analysis on synthetic data complements theoretical results on sample complexities for the proposed methods.

1 Introduction

We explore the problem of two-sided matching markets, where two distinct groups of agents must be matched with one another [20]. Such markets have diverse real-world applications, ranging from labor markets, like online crowd-sourcing platforms such as Amazon Mechanical Turk, to online dating services [19, 8, 9]. In these contexts, the challenge lies in designing a matching algorithm that respects the preferences of both sets of agents.

In their seminal work, Gale and Shapley introduce the concept of stable matchings [8]. Here, agents are matched one-to-one according to ordinal preferences such that no two agents have the incentive to deviate from the proposed solution. Gale and Shapley's Deferred Acceptance algorithm guarantees a stable matching that is optimal for one side of the market. This optimal stable matching always exists and is unique, based on one side's preferences. Many works have considered solutions for this and related problems of finding matchings under known preferences [17, 22]. However, in online settings, agents often lack certainty about their preferences.

More recently, research has focused on cases where agents' preferences are learned over time. This line of work, initiated by Das and Kamenica, frames the problem as an online decision-making challenge, likening it to a *multi-agent multi-armed bandit* problem where agents must compete for potential matches. In the centralized version of this problem, a platform matches agents according to their preferences, and the agents receive noisy feedback that updates those preferences. In this setting, there is an inherent tension between exploration and exploitation: the platform must decide whether to exploit current estimates of the best stable match or continue exploring to refine agents' preferences. Liu et al. formalize this dilemma using regret measures that balance this trade-off.

In this work, we focus on the centralized version of the stable marriage model, but approach the problem from a pure exploration standpoint. Our primary goal is to *efficiently* learn the optimal stable matching w.r.t. the side of the market with uncertain preferences. This perspective emerges from the observation that algorithms designed for the regret-minimization setting, which explore

18th European Workshop on Reinforcement Learning (EWRL 2025).

potential stable matches, can hinder the exploration of critical agent pairs that contribute to stability. Accordingly, we move beyond the regret-minimization framework to address the problem:

How can we efficiently identify the optimal stable matching with high probability?

Our contributions can be summarised as follows:
(1) Action Space. We do not limit our algorithms to stable matchings during exploration, so as to be able to learn player preferences. This is because constraining ourselves to stable matchings may never lead to discovering the optimal stable match (Section 3.2).
(2) Solution concept. We introduce the concept of a *probably correct optimal stable matching* (Section 3.3). This special case of probably approximately correct (PAC) solutions requires the output matching to be optimal with high probability.
(3) Uniform exploration strategy. We begin our analysis with an algorithm that uniformly samples all available agent pairs, similar to the Explore-Then-Commit (ETC) strategy. We demonstrate that this can produce the optimal stable matching with high probability and provide a bound on its sample complexity (Section 4).
(4) Action elimination algorithms. Next, we explore an algorithm based on the concept of action elimination (Section 5), which improves sample efficiency and reduces dependence on instance-specific parameters. Additionally, we enhance sample complexity by modifying the stopping criterion, enabling the algorithm to terminate when sufficient information is gathered (Section 6).
(5) Adaptive sampling. We study a strategy that adaptively samples agent pairs, improving the complexity in practice (Section 7).
(6) Experimental evaluation. Finally, we compare the performance of our algorithms on several synthetic instances (Section 8).

2 Related Work

The *stochastic* MAB problem, introduced by [23], has been extensively studied in the past few decades, becoming a fundamental tool for analyzing online decision-making problems under uncertainty. In its classic version, a learner selects one arm in each round and receives a numeric reward drawn from an unknown distribution associated with the selected arm. The goal of maximising the learner's cumulative reward requires a balanced solution of the exploration-exploitation dilemma: finding the best arm requires exploration, while receiving good rewards requires exploitation. Another popular setting is the pure exploration problem [5, 18, 6], where the learner's objective is to identify the best action (arm to pull) regardless of the cost incurred by choosing suboptimal actions but with a minimal number of samples. This problem is usually studied in the Probably Approximately Correct (PAC) setting, where the learner aims to identify the approximately best arm with high probability. This paper focuses on the pure exploitation setting and the presented solutions liken ETC and UCB strategies. In contrast to the PAC setting, we aim at identifying the best action with high probability, not just an approximation. For a comprehensive overview of MAB, we refer to the book by Lattimore and Szepesvári.

The stable marriage problem, is a well studied problem in two-sided markets in which agents from the two sides are matched such that no pair of agents prefers to deviate from the matching [17, 8], making the solution a *stable matching*. In earlier work on stable matchings, preferences of agents in the market are assumed to be known. The problem of two-sided markets with unknown preferences has recently attracted considerable attention in various settings. Das and Kamenica were the first to introduce the marriage problem with unknown preferences on both sides of the market, framing it as a MAB problem. In their work [4], they empirically studied several algorithms in specific preference settings. A follow-up work [15] studied a variant of the problem where one side of the market has unknown preferences and first introduced the notion of Player Stable Optimal/Pessimal regret for the agents in the market. They studied an ETC-like algorithm, proving sublinear bounds for both notions of regret, but requiring additional knowledge of the reward differences of the arms. They also study a UCB-like approach with sublinear bounds on the Player Pessimal Stable regret. However, they identified fundamental issues with their UCB algorithm in achieving sublinear optimal stable regret for the agents. Another research direction studied the decentralized setting where the agents act independently in the market [16, 13, 21, 2]. Kong et al. introduced a Thompson sampling algorithm for the decentralized market, and they also highlight issues with their algorithm in achieving sublinear optimal/pessimal stable regret in the centralized setting. Overall, developing algorithms with sublinear optimal regret remains an open problem. To bridge this gap, other lines of research study the centralized problem with transferable utilities to guarantee stability [3, 11]. In this work, we instead focus on approaching the problem in a pure exploration setting, aiming to efficiently identify a correct optimal stable matching with high probability. To the best of our knowledge, the pure exploration problem has not been explored in this setting.

3 Problem Setting

In this section, we first introduce all necessary definitions of agents, their preferences and stability of matchings for the classic two-sided matching markets. We then describe the problem of learning an optimal stable matching in two-sided matching markets with unknown preferences.

3.1 Stable Matchings in Two-Sided Markets

Agents: We consider two distinct sets of agents, *players* $\mathcal{P} = \{p_1, \cdots, p_N\}$ and *arms* $\mathcal{A} = \{a_1, \cdots, a_K\}$, each having N and K elements, respectively, where $N \leq K$. Let the set of all agents be $\mathfrak{A} = \mathcal{P} \cup \mathcal{A}$. We will also refer to the players as the *left* and the arms as the *right* side of the market.

Preferences: Each agent's preferences are represented as a complete list of agents from the opposite side. More specifically, the *preferences* π_p of a player p are given as a permutation over the arms, i.e., $\pi_p = (a_{p_1}, \cdots, a_{p_K}) \in \mathbf{P}(\mathcal{A})$, where $\mathbf{P}(\mathcal{A})$ is the set of permutations over $\mathcal{A}$. We say player p prefers arm $\pi_p(i)$ to arm $\pi_p(j)$ if $i < j$. Sometimes, we express these preference by the *order relation* $\succ_{\pi_p}$ where $\pi_p(i) \succ_{\pi_p} \pi_p(j)$ if $i < j$. Here we might omit the subscript, simply writing $\succ$, if it is clear from the context. Further, we define the *rank of an arm* a w.r.t. π_p as the position of a in p's preference list, i.e., $r_p(a) := |\{a_j \in \mathcal{A} : a_j \succ_{\pi_p} a\}|$. Similarly, we can define preferences π_a, the corresponding order relation $\succ_{\pi_a}$, and rank function r_a of an arm $a \in \mathcal{A}$ over the set of players $\mathcal{P}$.

Matchings: In the stable marriage model with players $\mathcal{P}$ and arms $\mathcal{A}$, a *matching* is a set of player-arm pairs $m \subseteq \mathcal{P} \times \mathcal{A}$ that are pairwise disjoint, representing the pairs of agents that are *matched*. We say an agent $i \in \mathfrak{A}$ is *unmatched* under m if there exists no pair in m that involves i. By a slight abuse of notation, we define the equivalent functional representation $m : \mathfrak{A} \to \mathfrak{A} \cup \{\perp\}$ of a matching m, where $m(i) = j$ and $m(j) = i$ for the pair $(i, j) \in m$, and $m(i) = \perp$ if $i \in \mathfrak{A}$ is unmatched. Let $\mathcal{M}$ denote the set of all possible matchings.

Stability: In order for a matching m to align with the agents' preferences, Gale and Shapley proposed stability [8] as a notion of equilibrium in the market. A matching m is *stable* if there is no pair of agents who prefer to be matched with each other compared to their match under m. More formally, m is a stable matching, if there exists no *blocking pair* $(p, a) \in \mathcal{P} \times \mathcal{A}$, i.e., no pair $(p, a) \notin m$ such that (1) $a \succ_{\pi_p} m(p)$ or p is unmatched, and (2) $p \succ_{\pi_a} m(a)$ or a is unmatched.

In Gale and Shapley's Deferred Acceptance (DA) algorithm [8], the agents on one side of the market sequentially propose to the other side, while the other side is temporarily matched with the most preferred agents so far until all agents are matched. They demonstrate that not only does a stable matching always exist for any instance of the marriage problem, but also that multiple stable matchings can exist; we denote this *collection of stable matchings* by $\mathcal{S}$. The stable matching $m_s^\star$ produced by the algorithm is optimal for the proposing side of the market and pessimal for the side that received the proposals, in the sense that the agents of the sets are matched with their most/least preferred partner among any feasible stable matching in $\mathcal{S}$. In the remainder of this paper, we refer to the unique optimal stable matching, $m_s^\star$, w.r.t. the players' preferences as the *optimal stable matching*.

3.2 Two-Sided Markets with Unknown Player Preferences

In this work, we consider a setting where the players $p \in \mathcal{P}$ are initially uncertain about their preferences π_p. Then the players gradually learn their preferences through noisy feedback from matchings that are imposed by a centralised matching algorithm $\mathbf{A}$.

More formally, the learning process is performed in T rounds where in each round t the algorithm $\mathbf{A}$ selects a matching m_t for the agents. Then, each player $p \in \mathcal{P}$ receives a stochastic reward $X^t_{p,m_t(p)} \in [0, 1]$, distributed according to an unknown distribution $\mathbb{P}_{p,m_t(p)}$ with mean $\mu_{p,m(p)}$. We assume that a player p truly prefers arm a_1 to arm a_2 if $\mu_{p,a_1} > \mu_{p,a_2}$, while we denote the difference of their expected rewards by $\Delta_{p,a_1,a_2} = \mid \mu_{p,a_1} - \mu_{p,a_2} \mid > 0$.

After each round t, the players update their preferences $\hat{\pi}_p(t)$ based on the estimate of the expected rewards $\hat{\mu}_{p,a}(t)$ (sample mean) and report it to the algorithm. The objective of the algorithm is to efficiently identify the optimal stable matching $m_s^\star$ with respect to the (initially unknown) preferences π_p with high probability. Thus, at each round t, the algorithm aims to select a matching $m_t \in \mathcal{M}$ to gather relevant information.

To summarize the process, at every round $t \in \{1, \ldots, T\}$:

1. The algorithm $\mathbf{A}$ selects a matching $m_t \in \mathcal{M}$.
2. The players $p \in \mathcal{P}$ receive a reward $X^t_{p,m_t(p)}$.
3. The players update $\hat{\mu}_{p,m_t(p)}(t)$ and report it to $\mathbf{A}$.
4. The algorithm $\mathbf{A}$ might terminate the process (according to a stopping criterion) and return a matching.

The returned matching should be the true optimal stable matching $m_s^\star$ with high probability.

Note that here we do not require the action selected by $\mathbf{A}$ to be a *stable* matching. Since preferences of agents are uncertain, we cannot be certain which matchings are stable. Yet related work restricts $\mathbf{A}$ to select only matchings that are stable with respect to confidence bounds [4, 15, 3]. Because we can compute confidence bounds on the estimated means μ_p, we could identify blocking pairs with high probability. However, the following example, which Liu et al. use to demonstrate that their UCB-like algorithm fails to achieve sublinear player-optimal stable regret [15], shows that excluding actions of matchings with high-probability blocking pairs will in some cases not allow us to identify the optimal stable matching. This motivates our setting in which the action space of algorithm $\mathbf{A}$ is the set of all matchings $\mathcal{M}$ disregarding stability.

Example 1 (Liu et al. [15])**.** *Consider a market with three agents on each side and the following true agents' preferences.*

Players	*Arms*
$\pi_{p_1} : a_1 \succ a_2 \succ a_3$	$\pi_{a_1} : p_2 \succ p_3 \succ p_1$
$\pi_{p_2} : a_2 \succ a_1 \succ a_3$	$\pi_{a_2} : p_1 \succ p_2 \succ p_3$
$\pi_{p_3} : a_3 \succ a_1 \succ a_2$	$\pi_{a_3} : p_3 \succ p_1 \succ p_2$

Here, the only stable matchings are the player-optimal stable matching $m_{s1} = \{(p_1, a_1), (p_2, a_2), (p_3, a_3)\}$ and the arm-optimal stable matching $m_{s2} = \{(p_1, a_2), (p_2, a_1), (p_3, a_3)\}$ which is player-pessimal.

Now assume that players p_1 and p_2 are certain about their preferences and correctly report them, while player p_3 is uncertain about the order of a_1 and a_3, but p_3's current estimates of the sample means $\hat{\mu}_{p_3,a_1}$, $\hat{\mu}_{p_3,a_2}$ and $\hat{\mu}_{p_3,a_3}$ yield $\hat{\pi}_{p_3} : a_1 \succ a_3 \succ a_2$. Under these preferences, the only stable matching is m_{s2}. Further, the only matchings that could be stable w.r.t any possible preference $\hat{\pi}_{p_3} \in \mathbf{P}(\mathcal{A})$ of p_3, are m_{s1} or m_{s2}.

In this situation, if the matching algorithm $\mathbf{A}$ is only permitted to select potentially stable matchings, i.e., m_{s1} or m_{s2}, the player p_3 will receive a sample from $\mathbb{P}_{p_3,a_3}$. This serves p_3 to grow more certain about the estimate $\hat{\mu}_{p_3,a_3}$. However, the uncertainty about $\hat{\mu}_{p_3,a_1}$ persists. Thus, if $\hat{\mu}_{p_3,a_1} > \mu_{p_3,a_3}$ then with high probability the updated estimated preference order $\hat{\pi}_{p_3}$ will remain $a_1 \succ a_3 \succ a_2$, i.e., yielding the same situation as before. Thus, no matter the number of samples, with high probability the returned matching is m_{s2} — the player-pessimal stable matching!

In contrast, if $\mathbf{A}$ is permitted to sample any matching, it could also sample a matching that matches p_3 to a_1 in order to grow more certain about the estimate $\hat{\mu}_{p_3,a_1}$ and eventually determining $\hat{\mu}_{p_3,a_3} > \hat{\mu}_{p_3,a_1}$ and correctly returning m_{s1}.

While this richer action space of all matchings (regardless of stability) allows us to circumvent the shortcomings outlined in Example 1, it also permits and even encourages the selection of matchings that are known with high probability not to be stable. Although this approach efficiently identifies a true stable matching (and is indeed necessary, as the example shows), it might not be desirable in some applications to implement such "unfair" matchings as intermediate actions. In our case, we decide to focus on the pure exploration setting, aiming only to quickly identify the optimal stable matching.

3.3 Probably Correct Optimal Stable Matching

We introduce the notion of *Probably Correct Optimal Stable Matching* (PCOS). This concept is similar to the 'Probably Approximately Correct' (PAC) setting [6], where the goal is to find an ϵ-optimal arm with high probability using as few samples as possible in the context of MAB. Note that in the context of stable matchings it is non-trivial to define what an *approximation* of the optimal stable matching is. Consider Example 1 and assume for player p_3 the gap between true mean rewards $\Delta_{p_3,a_1,a_3} = \mid \mu_{p_3,a_1} - \mu_{p,a_3} \mid$ is less than ϵ. Then even an ϵ-approximation of the true means is not sufficient to identify the true optimal stable matching m_{s1} and instead m_{s2} is returned. In case one is tempted to consider approximations w.r.t. the sum of players' rewards over their matches we remark that here the approximation value of m_{s_2} towards m_{s_1} depends on the reward gaps Δ_{p_1,a_1,a_2} and Δ_{p_2,a_1,a_2} which might be arbitrarily large. We thus leave more extensive discussions around approximate solutions to future work and instead focus on the following solution concept.

Definition 1 (Probably Correct Optimal Stable Matching)**.** *We say that an algorithm* $\mathbf{A}$ *is a* δ*-PCOS algorithm with sample complexity* T*, if it outputs the optimal stable matching,* $m_s^\star$*, with probability at least* $1 - \delta$ *after at most* T *sampled matchings.*

Remark 1. *We measure the sample complexity in terms of the number of matchings performed by algorithm* $\mathbf{A}$*.*

As correctly identifying the optimal stable matching is closely related to correctly identifying the agent's preferences we introduce the following two notions.

Definition 2 (Completely Correct Preferences)**.** *We say that the estimated preferences* $\hat{\pi}$ *of players are completely correct if they are exactly the same as the true preferences* π*, i.e.,* $\hat{\pi}[i] = \pi[i] \; \forall \; i \in [1, \dots, K]$.

Definition 3 (Partially Correct Preferences up to an Arm)**.** *We say that the estimated preferences* $\hat{\pi}$ *of players are partially correct up to an arm* $a \in \mathcal{A}$ *if they are correct up to the position* $r_p[a]$ *with the true preference* π*, i.e.,* $\hat{\pi}[i] = \pi[i] \; \forall \; i \leq r_p[a]$.

We formally state the relation between the probability of the DA algorithm returning the optimal stable matching and the preference estimates being correct.

Proposition 1. *Let* $m_{\hat{\pi}}$ *be the output of the DA algorithm running using estimated preferences* $\hat{\pi}$ *for the agents* $\mathfrak{A}$*. The probability that* $m_{\hat{\pi}}$ *is equal to the optimal stable matching* $m_s^\star$ *for the true preferences* π *is at least as high as the probability of those estimates* $\hat{\pi}$ *being correct, i.e.:*

$$P(m_{\hat{\pi}} = m_s^\star) \geq P\left(\hat{\pi}_i = \pi_i \; \forall \; i \in \mathfrak{A}\right).$$

Proof. This follows directly from the fact that the Deferred Acceptance algorithm with correct preferences always returns the optimal stable matching $m_s^\star$. □

4 Naive Uniform Exploration

We begin our analysis by considering an algorithm that uniformly explores every available pair of agents, similarly to the ETC algorithm in the regret minimization setting [15]. Although ETC can achieve sublinear player-optimal stable regret, this does not always imply convergence to the correct optimal stable matching. Here, we provide a uniform sampling strategy that can identify the true optimal stable matching with high probability.

Our Naive Uniform Exploration (NUE) Algorithm uniformly samples matchings such that each pair of agents is sampled for a fixed number times, depending on the minimum reward difference between the arms. After the exploration rounds, the algorithm estimates the preferences of the players using the sample mean of the rewards for the arms and commits to the matching produced by the DA algorithm using these estimated preferences.

Algorithm 1 Naive Uniform Exploration (NUE)

```
Require: $\delta > 0, \mathcal{P}, \mathcal{A}, \Delta_{\min} = \min_{p \in \mathcal{P}, i,j \in \mathcal{A}} \Delta_{p,i,j}, \{\pi_a\}_{a \in \mathcal{A}}$
1: $h = \lceil \frac{2\ln(2KN/\delta)}{\Delta^2_{\min}} \rceil$
2: for $t$ in $\{1, \cdots, hK\}$ do
3:     $m(p_i) = a_j$ , $j = (t + i - 2 \mod K) + 1 \; \forall i \in \{1, \ldots, N\}$
4:     Sample $m$ and update $\hat{\mu}_{p_i, m(p_i)}(t) \; \forall i \in \{1, \ldots, N\}$
5: end for
6: $\hat{\pi}_p = \arg \operatorname{sort}_{a \in \mathcal{A}} \hat{\mu}_{p,a} \; \forall p \in \mathcal{P}$
7: $m_{\hat{\pi}} = DA(\{\hat{\pi}_p\}_{p \in \mathcal{P}}, \{\pi_a\}_{a \in \mathcal{A}})$
8: return $m_{\hat{\pi}}$
```

Theorem 1. *Let $\Delta_{\min} = \min_{p \in \mathcal{P}, i,j \in \mathcal{A}} \Delta_{p,i,j}$. Algorithm 1 is a δ-PCOS algorithm, and the number of matchings is bounded by:*

$$O(K \frac{\ln(KN/\delta)}{\Delta^2_{\min}}). \tag{1}$$

Proof Sketch. The sample complexity follows from the definition of the algorithm, as $O(Kh) = O(K \ln(KN/\delta)/\Delta^2_{\min})$.

We now prove that Algorithm 1 is an δ-PCOS algorithm, i.e., $\Pr(m_{\hat{\pi}} = m^\star_s) \leq 1 - \delta$. Proposition 1 implies that $\Pr(m_{\hat{\pi}} \neq m^\star_s) \leq \Pr(\bigcup_{p \in \mathcal{P}} \hat{\pi}_p \neq \pi_p)$. Using the union bound over the set of players $\mathcal{P}$ and arms $\mathcal{A}$ we have $\Pr(m_{\hat{\pi}} \neq m^\star_s) \leq \sum_{p \in \mathcal{P}} \sum_{i=1}^N \Pr(\hat{\pi}_p[i] \neq \pi_p[i])$. Thus it is sufficient to show $\Pr(\hat{\pi}_p[i] \neq \pi_p[i]) \leq \delta/NK$.

The event that $\hat{\pi}_p[i] \neq \pi_p[i]$ can only occur if the player p wrongly orders two consecutive arms a and a', i.e., $\hat{\mu}_{p,a} \leq \hat{\mu}_{p,a'}$ when $\mu_{p,a} \geq \mu_{p,a'}$. By construction of the sampled matchings (see line 3 in Algorithm 1), every player p receives h many rewards for every arm. Similar to the proof of Theorem 6 in [6] and by using Hoeffding's inequality, we have that $\Pr(\hat{\mu}_{p,a} \leq \hat{\mu}_{p,a'}) \leq \Pr(\hat{\mu}_{p,a} \leq \mu_{p,a} - \Delta_{p,a,a'}/2) + \Pr(\hat{\mu}_{p,a'} \geq \mu_{p,a'} + \Delta_{p,a,a'}/2) \leq 2\exp(-\frac{\Delta^2_{\min}}{2} h) \leq \delta/NK$ which concludes the proof. □

5 Elimination Algorithm

In this section, we propose an elimination algorithm, similar to the one in [6] for MAB, which improves sample complexity and does not require prior knowledge of the reward differences $\Delta_{p,a,a'}$ of the agents. The key idea of our Elimination Algorithm (Algorithm 2) is to successively eliminate player-arm pairs (p, a) when the position of arm a in player p's preference list can be determined with high probability. Eliminating every player-arm pair can eventually guarantee the correct estimation of the preferences of players with high probability, and thus identification of the optimal stable matching, as implied by Proposition 1.

Algorithm 2 operates in rounds, where in each round t, we sample matchings such that each player $p \in \mathcal{P}$ receives a reward from every non-eliminated arm from the set of available arms S_p, exactly ones (line 5). For this we compute a *minimal matching cover*, i.e., a cardinality-minimal set of matchings $\mathfrak{M}$ that cover all remaining pairs $\{(p, a) \in \mathcal{P} \times \mathcal{A} \mid a \in S_p\} \subseteq \bigcup_{m \in \mathfrak{M}} m$. We denote $\mathfrak{M}(X)$ to be a minimal matching cover on a given set of pairs X.

In the case of a bipartite graph, finding a minimal matching cover can be framed as a minimum edge coloring problem, where the edges of one color correspond to a matching in the matching cover [7]. According to the Kőnig-Hall Theorem, the optimal number of colors required is equal to the maximum degree of the graph — in our case the maximal degree $deg_t = \Delta(G(E_t))$ of the bipartite graph $G(E_t)$ with edges E_t corresponding to available pairs $\{(p, a) \in \mathcal{P} \times \mathcal{A} \mid a \in S_p\}$. Thus, in every round t, we sample deg_t matchings (see also Appendix C). .

The algorithm terminates once we eliminate all arms for every player, i.e., $S_p = \emptyset$ for all $p \in \mathcal{P}$. We use the elimination rule that eliminates a (p, a) pair when the arm a has no overlapping confidence interval $C_{p,a}$ with other available arms in S_p for player p. Together with the use of anytime confidence intervals[1], the elimination rule can guarantee the position of arm a in the preference list of player p with high probability.

Algorithm 2 Elimination Algorithm

```
Require: $\delta > 0, \mathcal{P}, \mathcal{A}, \{\pi_a\}_{a\in\mathcal{A}}$
 1: $S_p = \mathcal{A} \ \forall p \in \mathcal{P}$
 2: $t = 1$
 3: $\hat{\mu}_{p,a}(t) = 0 \ \forall p \in \mathcal{P}, a \in \mathcal{A}$
 4: while $| \cup_{p\in\mathcal{P}} S_p | \geq 1$ do
 5:     Sample all matchings $m \in \mathfrak{M}\left(\{(p,a) \in \mathcal{P} \times \mathcal{A} \mid a \in S_p\}\right)$
 6:     Update $\hat{\mu}_{p,a}(t) \ \forall p \in \mathcal{P}$ and $a \in S_p$
 7:     $B_t = \sqrt{\frac{\ln(4KNt^2/\delta)}{2t}}$
 8:     $C_{p,a} = [\hat{\mu}_{p,a}(t) \pm B_t] \ \forall p \in \mathcal{P}$ and $a \in S_p$
 9:     $S_p = S_p \setminus \{a : C_{p,a} \cap C_{p,j} = \emptyset \ \forall j \in \mathcal{A} \setminus \{a\}\} \ \forall p \in \mathcal{P}$
10:     $t = t + 1$
11: end while
12: $\hat{\pi}_p = \arg\mathrm{sort}_{a\in\mathcal{A}} \hat{\mu}_{p,a} \ \forall p \in \mathcal{P}$
13: return $DA(\{\hat{\pi}_p\}_{p\in\mathcal{P}}, \{\pi_a\}_{a\in\mathcal{A}})$
```

As a starting point for the overall sample complexity of Algorithm 2, we first calculate how many samples $t_{p,a}$ that are sufficient to eliminate an arm a from a player p's list of available arms S_p as in line 9 of the algorithm.

Lemma 1. *For a player $p \in \mathcal{P}$ and arm $a \in \mathcal{A}$, let $\Delta_{p,a} = \min_{a' \in \mathcal{A}\setminus\{a\}} \Delta_{p,a,a'}$. With probability at least $1-\delta$, the number of samples $t_{p,a}$ needed to eliminate an arm a from S_p in line 9 of Algorithm 2, is at most*

$$t_{p,a} = O\left(\frac{\ln(KN/\delta\Delta_{p,a})}{\Delta_{p,a}^2}\right). \tag{2}$$

Proof Sketch. Let $\mathcal{E}$ denote the event that, for all time steps t the expected rewards $\mu_{p,a}$ lie in the confidence interval $CI_{p,a}(t)$ for every pair of player p and arm a i.e. $\mu_{p,a} - B_t \leq \hat{\mu}_{p,a}(t) \leq \mu_{p,a} + B_t \ \forall \ t, \ a \in \mathcal{A}, \ p \in \mathcal{P}$. Using Hoeffding's inequality, we can show that the values $B_t = \sqrt{\frac{\ln(4KNt^2/\delta)}{2t}}$ in Algorithm 2 correspond to bounds of any-time confidence intervals $CI_{p,a}(t) = [\hat{\mu}_{p,a}(t) - B_t, \hat{\mu}_{p,a}(t) + B_t]$. Consequently, $\mathcal{E}$ is true w.p.a. $1-\delta$.

Now assume $\mathcal{E}$ is true. For a player p consider two arms a, a' with $\mu_{p,a} > \mu_{p,a'}$. Under the event $\mathcal{E}$ and for t such that $\Delta_{p,a,a'} > 4B_t$, the arms will have no overlapping confidence intervals as $(\hat{\mu}_{p,a} - B_t) - (\hat{\mu}_{p,a'} + B_t) \geq \mu_{p,a} - \mu_{p,a} - 4B_t > 0$. So in order for an arm a to have no overlapping confidence intervals with any other arm we need $\Delta_{p,a} = \min_{a'\in\mathcal{A}\setminus\{a\}} \Delta_{p,a,a'} > 4B_t$, which holds for $t_{p,a} = O\left(\ln(KN/\delta\Delta_{p,a})/\Delta_{p,a}^2\right)$ using a similar analysis with [6].

Consequently, after $t_{p,a}$ samples of all arms, no confidence interval $CI_{p,a'}(t) \forall a' \in \mathcal{A} \setminus \{a\}$ overlaps with $CI_{p,a}(t)$ and a is eliminated from S_p, with probability at least $1-\delta$. □

We can now state the sample complexity and show Algorithm 2 is a δ-PCOS algorithm, with a detailed proof in Appendix D.

Theorem 2. *Let $\mathcal{P}(S) = \{r \subseteq S : \Delta(G(S)) - \Delta(G(S \setminus r)) = 1\}$ denote the set of pairs that we have to eliminate to reduce the degree of the graph $G(S)$ with $S = \{(p,a) \in \mathcal{P} \times \mathcal{A}\}$. Let also $r_0 = \emptyset$ and*

$$r_i = \arg \min_{r\in\mathcal{P}(S\setminus\bigcup_{j<i} r_j)} \max_{(p,a)\in r} t_{p,a} \ \forall i = 1, \ldots, K$$

Algorithm 2 is a δ-PCOS algorithm, and with probability at least $1-\delta$, the number of matching samples is bounded by:

$$O\left(\sum_{s=1}^{K} \max_{(p,a)\in r_s} t_{p,a}\right) \tag{3}$$

[1] Anytime confidence intervals ensure valid coverage probabilities at any time t.

Proof Sketch. Using Hoeffding's inequality, we can show that the values B_t (in Algorithm2) define any-time confidence intervals, i.e., the event $\mathcal{E}$, where at any time-step t the expected rewards $\mu_{p,a}$ lie in the confidence interval $C_{p,a}(t)$ for every pair of player p and arm a, holds with probability at least $1-\delta$.

Under the event $\mathcal{E}$, once two confidence intervals are not overlapping for some arms a, a', we can determine their relative order in player p's preferences. This still holds true even when taking more samples of one of the arms, as the confidence intervals only shrinks. We can thus eliminate the pair (p, a) once the order of the arm a towards all other arms can be determined with high probability. Thus, upon termination of the algorithm, we will have determined the correct preferences $\hat{\pi}_p = \pi_p \forall p \in \mathcal{P}$ and by Proposition 1, Algorithm 2 will output $m_s^\star$ with w.p.a. $1-\delta$.

For the sample complexity, consider that at any time t we sample every available pair, $\{(p, a) \in \mathcal{P} \times \mathcal{A} \mid a \in S_p\}$, using the matchings from a minimal matching cover of the bipartite graph with edges corresponding to currently available pairs (see line 5). Such a minimal matching cover has deg_t many matchings. Thus, once the maximal degree is reduced, less matchings have to be sampled.

We thus consider $s = 1, \ldots, K$ phases where each phase corresponds to the elimination of the subset of pairs r_s to reduce the degree of the graph. So under the event $\mathcal{E}$, we can use Lemma 1 to define r_s as the set of pairs with the lowest sample complexity i.e:

$$r_i = \arg \min_{r \in \mathcal{P}(S \setminus \bigcup_{j<i} r_j)} \max_{(p,a)\in r} t_{p,a} \; \forall i = 1, \ldots, K$$

with

$$\mathcal{P}(S) = \{r \subseteq S : \Delta(G(S)) - \Delta(G(S \setminus r)) = 1\}$$

denote the subsets of pairs that can reduce the degree of the graph. Note that here the $r_1, \ldots, r_K$ form a partition of all player-arm pairs.

In every phase s, we perform $t_s - t_{s-1}$ iterations, where t_s are sufficient number of steps to eliminate r_s determined by the player-arm pair $(p, a) \in r_s$ that takes the longest to eliminate i.e: $t_s = \max_{(p,a)\in r_s} t_{p,a}$ with $t_0 = 0$.

Finally, in each step t, every matching cover consists of $K - s + 1$ matchings. Thus, under the event $\mathcal{E}$ witch holds w.p.a. $1 - \delta$, the total number of matching samples is given by:

$$\sum_{s=1}^{K}(K-s+1)(t_s - t_{s-1}) = \sum_{s=1}^{K} t_s = O\left(\sum_{s=1}^{K} \max_{(p,a)\in r_s} t_{p,a}\right)$$

□

Remark 2. *Note, that in the case where $N = K$, the set of pairs r_s that we have to eliminate, corresponds to a perfect matching m_s as in this case the graph of available pairs is regular at each phase s.*

Remark 3. *We can construct a worst-case instance, where Algorithm 2 uniformly samples all player-arm pairs. In particular, if the differences in the expected rewards are equal i.e. $\Delta_{p,a} = \Delta \; \forall p \in \mathcal{P}, a \in \mathcal{A}$, leading to sample complexity:*

$$O\left(\sum_{s=1}^{K} \max_{(p,a)\in r_s} t_{p,a}\right) = O\left(K \frac{\ln(KN/\delta\Delta)}{\Delta^2}\right).$$

6 Improved Elimination Algorithm

In this section, we propose an improved version of the elimination algorithm, Algorithm 3, based on the observation that to identify the optimal stable matching, we only need to correctly estimate the preferences of the players up to the position of the matching partner in the optimal stable matching. We formalize this in the Lemma below.

Lemma 2. *Let $m_s^\star$ be the true optimal stable matching according to preferences π. The output, $m_{\hat{\pi}}$, of the DA algorithm using preferences $\hat{\pi}$ that are partially correct up to $m_{\hat{\pi}}(p)$ for every player p, i.e., $\hat{\pi}_p[i] = \pi_p[i] \; \forall \, i \leq r_p[m_{\hat{\pi}}(p)] \; \forall \, p \in \mathcal{P}$, is equal to the true optimal stable matching $m_s^\star$.*

Proof. The DA algorithm sequentially executes proposals from players starting with their most preferred arm and never backtracks to a previously made proposal. Further, it halts once the player optimal stable matching has been found. Consequently it only considers the matching partner $m_{\hat{\pi}}(p)$ and all higher ranked arms in the optimal stable matching according to $\hat{\pi}$. So if the preferences π_p are correct up to the positions of the $m_{\hat{\pi}}$ then is equal to the true optimal stable matching $m_s^\star$. □

We can modify the stopping criteria of the Elimination Algorithm of the previous section according to Lemma 2. Algorithm 3 terminates when it eliminates the arms up to the stable matching partner for every player. Specifically, after each round t, we calculate an estimate of the player optimal stable matching $\hat{m}_t$ from the DA algorithm using the estimated preferences $\hat{\pi}$ from the sample means. The algorithm terminates if for all players $p \in \mathcal{P}$, every the stable matching partner $\hat{m}_t(p)$ and all higher ranked arms have been eliminated. As the algorithm proceeds the stable matching $\hat{m}_t$ changes until we eventually can reach a state where our termination criteria holds.

Algorithm 3 Improved Elimination Algorithm

```
Require: $\delta > 0, \mathcal{P}, \mathcal{A}, \{\pi_a\}_{a \in \mathcal{A}}$
 1: $t = 1, S_p = \mathcal{A} \; \forall p \in \mathcal{P}, S = S_p$
 2: $\hat{\mu}_{p,a}(t) = 0 \; \forall p \in \mathcal{P}, a \in \mathcal{A}$
 3: while $| \; S \; | \geq 1$ do
 4:     Sample all matchings $m \in \mathfrak{M}\left(\{(p,a) \in \mathcal{P} \times \mathcal{A} \mid a \in S_p\}\right)$
 5:     Update $\hat{\mu}_{p,a}(t) \; \forall p \in \mathcal{P}$ and $a \in S_p$
 6:     $B_t = \sqrt{\frac{\ln 4KNt^2/\delta}{2t}}$
 7:     $C_{p,a} = [\hat{\mu}_{p,a}(t) \pm B_t] \; \forall a \in S_p, p \in \mathcal{P}$
 8:     $S_p = S_p \setminus \{a : C_{p,a} \cap C_{p,j} = \emptyset \; \forall j \in \mathcal{A} \setminus \{a\}\}$
 9:     $\hat{\pi}_p = \arg\operatorname{sort}_{a \in \mathcal{A}} \hat{\mu}_{p,a} \; \forall p \in \mathcal{P}$
10:     $m_t = DA(\{\hat{\pi}_p\}_{p \in \mathcal{P}}, \{\pi_a\}_{a \in \mathcal{A}})$
11:     $S = \cup_{p \in \mathcal{P}} \{a \in S_p : a \succeq_{\hat{\pi}_p} m_t(p)\}$
12:     $t = t + 1$
13: end while
14: return $m_t$
```

Theorem 3. *Let $\mathcal{P}(S) = \{r \subseteq S : \Delta(G(S)) - \Delta(G(S \setminus r)) = 1\}$ denote the set of pairs that we have to eliminate to reduce the degree of the graph $G(S)$ with $S = \{(p,a) \in \mathcal{P} \times \mathcal{A}\}$, $r_0 = \emptyset$ and*

$$r_i = \arg \min_{r \in \mathcal{P}(S \setminus \bigcup_{j<i} r_j)} \max_{(p,a) \in r} t_{p,a} \; \forall i = 1, \ldots, K$$

Let also $t_{\max} = \max_{(p,a) \in \mathcal{P} \times \mathcal{A} : a \succeq_p m_s^\star(p)} t_{p,a}$, and $n \in \{1, \ldots, K\}$ be the index s.t. $\max_{(p,a) \in r_{n-1}} t_{p,a} \leq t_{\max} \leq \max_{(p,a) \in r_n} t_{p,a}$.

Algorithm 3 is a δ-PCOS algorithm, and with probability at least $1 - \delta$, the number of matching samples is bounded by:

$$O\left(\sum_{s=1}^{n-1} \max_{(p,a) \in r_s} t_{p,a} + (K - n + 1) t_{\max}\right) \tag{4}$$

Proof. Algorithm 3 is essentially the same as Algorithm 2 with a different stopping criterion (see lines 3 and 11), and many of arguments from the proof of Theorem 2 transfer. In particular, at the time of elimination of an arm a from S_p, the arm can be correctly ordered w.r.t. all other arms in p's preferences with high probability. Thus at the time of termination, with probability at least $1 - \delta$, the condition of Lemma 2 is satisfied and Algorithm 3 outputs the correct optimal stable matching $m_s^\star$.

For the sample complexity, we can again consider phases s where we eliminate a subset of pairs r_s after at most $t_s = \max_{(p,a) \in r_s} t_{p,a}$ number of samples, under the event $\mathcal{E}$. However, with the stopping criterion in Algorithm 3, we might not have to eliminate every $r_1, \ldots, r_K$ before stopping. In fact, by Lemma 2 we only need to eliminate the optimal matching partner and all higher ranked arms for every player i.e. the pairs $\{(p,a) \in \mathcal{P} \times \mathcal{A} : a \succeq_p m_s^\star(p)\}$. For this, a maximal number of samples $t_{\max}$ from every pair are needed.

Now, consider the last phase n in which all arms $a \succeq_p m_s^\star(p)$ are either eliminated from S_p or sampled $t_{\max}$ many times, i.e., $t_{n-1} \leq t_{\max} \leq t_n$. The algorithm terminates within phase n, with a total number of matching: $\sum_{s=1}^{n-1}(K-s+1)(t_s - t_{s-1}) + (K-n+1)(t_{\max} - t_{n-1}) = \sum_{s=1}^{n-1}(K-s+1)t_s + (K-n+1)t_{\max}$. To conclude, under the event $\mathcal{E}$ witch holds w.p.a. $1-\delta$, the algorithm terminates with the total number of matchings bounded by:

$$O\left(\sum_{s=1}^{n-1} \max_{(p,a)\in r_s} t_{p,a} + (K-n+1)t_{\max}\right)$$

□

Remark 4. *We can construct a worst-case instance, where Algorithm 3 uniformly samples all player-arm pairs. In particular, if the differences in the expected rewards are equal i.e.* $\Delta_{p,a} = \Delta \; \forall p \in \mathcal{P}, a \in \mathcal{A}$*, leading to sample complexity:*

$$O\left(K \frac{\ln\left(KN/\delta\Delta\right)}{\Delta^2}\right).$$

As for Algorithm 2, Algorithm 3 will, in the worst case, have the same sample complexity as a strategy that uniformly samples pairs until they are eliminated, e.g., when all expected reward gaps are equal, i.e., $\Delta_{p,a} = \Delta \; \forall p \in \mathcal{P}, a \in \mathcal{A}$ (see also Remark 3). However, in practice, we expect Algorithm 3 to perform much better and terminate earlier, particularly when the reward gaps are larger for higher-ranked consecutive arm pairs.

7 Adaptive Sampling Algorithm

The algorithms introduced in the previous sections uniformly sample the arms for each player, until their confidence intervals are sufficiently separated. Here we propose an approach to adaptively sample player-arm pairs at every round.

Algorithm 4, leverages insights from Lemma 2 to dynamically define the set of arms requiring further exploration, denoted as S_p. Specifically, in each round t, we estimate the optimal stable matching $\hat{m}_t$ from the empirical preferences $\hat{\pi}$. According to Lemma 2, for this matching to be correct, the arms must be accurately ranked for the players up to the partner in the optimal stable matching. Consequently, in each round t and for each player p, we sample the set of arms in order to distinguish the confidence intervals of the arms up to the stable matching partner $m_t(p)$ (see line 11). Due to the adaptive selection of arms, each arm has a distinct confidence margin $B_{p,a}$ (line 5), which varies based on the number of times we sample a player-arm pair $t_{p,a}$.

Algorithm 4 Adaptive Sampling Algorithm

```
Require: $\delta > 0, \mathcal{P}, \mathcal{A}, \{\pi_a\}_{a\in\mathcal{A}}$
 1: $t_{p,a} = 0, S_p = \mathcal{A} \; \forall p \in \mathcal{P}$
 2: $\hat{\mu}_{p,a}(t) = 0 \; \forall p \in \mathcal{P}, a \in \mathcal{A}$
 3: while $| \cup_{p\in\mathcal{P}} S_p | \geq 1$ do
 4:     Sample all matchings $m \in \mathfrak{M}\left(\{(p,a) \in \mathcal{P} \times \mathcal{A} \mid a \in S_p\}\right)$
 5:     Update $\hat{\mu}_{p,a}(t)$ and $t_{p,a} \; \forall p \in \mathcal{P}$ and $a \in S_p$
 6:     $B_{p,a} = \sqrt{\frac{\ln 4KNt_{p,a}^2/\delta}{2t_{p,a}}} \forall a \in \mathcal{A}, p \in \mathcal{P}$
 7:     $C_{p,a} = [\hat{\mu}_{p,a} \pm B_{p,a}] \; \forall a \in \mathcal{A}, p \in \mathcal{P}$
 8:     $\hat{\pi}_p = \arg\operatorname{sort}_{a\in\mathcal{A}} \hat{\mu}_{p,a} \; \forall p \in \mathcal{P}$
 9:     $m_t = DA(\{\hat{\pi}_p\}_{p\in\mathcal{P}}, \{\pi_a\}_{a\in\mathcal{A}})$
10:     $A[p] = \{a \in \mathcal{A} : a \succeq_{\hat{\pi}_p} m_t(p)\} \forall p \in \mathcal{P}$
11:     $S_p = \{a \in \mathcal{A} \mid \exists a' \in \mathcal{A} \setminus \{a\}:$
             $C_{p,a'} \cap C_{p,a} \neq \emptyset$ and $\{a, a'\} \cap A[p] \neq \emptyset\} \forall p \in \mathcal{P}$
12:     $t = t + 1$
13: end while
14: return $m_t$
```

Theorem 4. *Algorithm 4 is a δ-PCOS algorithm.*

Proof. First, note that, as shown for Algorithm 2 and 3, the $B_{p,a}$ define anytime confidence intervals, i.e., the event $\mathcal{E}$ — where the expected reward lies within the confidence intervals at any time t for all pairs— holds with probability at least $1-\delta$.

So under $\mathcal{E}$, consider a round where the estimate of the stable matching m_t is incorrect, witch occurs only when at least one player p has wrong preferences $\hat{\pi}_p$ up to the stable match. This implies that at least one arm a with $a \succeq_{\hat{\pi}_p} m_t(p)$ has an overlapping confidence interval with another arm a', defining the active set of arms in round t, e.g., $S_p = \{a, a'\}$. In this situation, the algorithm samples the arms in the active set S_p until their confidence intervals are sufficiently separated or until we find ourselves with a different matching $m_{t'}$ and a new active set of arms. Since our confidence intervals only shrink, our algorithm will eventually terminate with the correct preferences up to the stable match, and thus with the correct optimal stable matching with probability at least $1-\delta$. □

The adaptive selection of agent pairs complicates the analysis of sample complexity, which we leave for future work. Instead, we test the algorithms in practice in the following section.

8 Simulations

In this section, we perform simulations on random instances to further evaluate the performance of our algorithms for fixed $\delta = 0.1$. The rewards for each pair are drawn from a Bernoulli distribution, i.e., $X_{p,a} \sim \text{Bern}(\mu_{p,a})$, with $\mu_{p,a} = \mathbb{E}[X_{p,a}]$. We perform different experiments by varying the number of agents with $N = K$, while we explore two different reward settings: (1) random expected rewards, and (2) random expected rewards with decreasing gaps. For each experiment, we generate 100 random instances. The code for this work is available at `https://github.com/****`.

In both reward settings, we randomly generate the preferences of the agents π for each instance. To create the respective expected rewards μ for the pairs, we first sample $K-1$ arbitrary reward gaps $\Delta_{p,i,i+1}$ for each player p and $i = 1, \ldots, K-1$ from a Dirichlet distribution $\text{Dir}(\alpha = 1)$, while we set $\Delta_{p,0,1} = 0$. For computational reasons, we normalize these values $\Delta_{p,i,i+1}$ to ensure that none exceed 0.05. In **Reward Setting 1**, we set the expected rewards to $\mu_{p,\pi_p[i]} = \sum_{j \le K-i} \Delta_{p,j,j+1}$, while in **Reward Setting 2**, we first sort the reward gaps in increasing order. The second setting ensures reduced sample complexity for the algorithms that employs the stopping rule according to our Lemma 2, as the preferences up to the stable match are easier to distinguish, i.e., $\Delta_{p,\pi[i],\pi[i+1]} \ge \Delta_{p,\pi[i+1],\pi[i+2]}\ \forall i = 1, \ldots, K-1$.

We compare the **(1) Elimination Algorithm**, **(2) Improved Elimination Algorithm**, and **(3) Adaptive Sampling Strategy**. In addition, we consider a variant of the NUE algorithm, the **(4) Uniform Sampling Strategy** that uniformly samples every pair until there are no overlapping confidence intervals, as the algorithm introduced in Section 4 requires knowledge of the minimum reward difference Δ and has a fixed sample complexity.

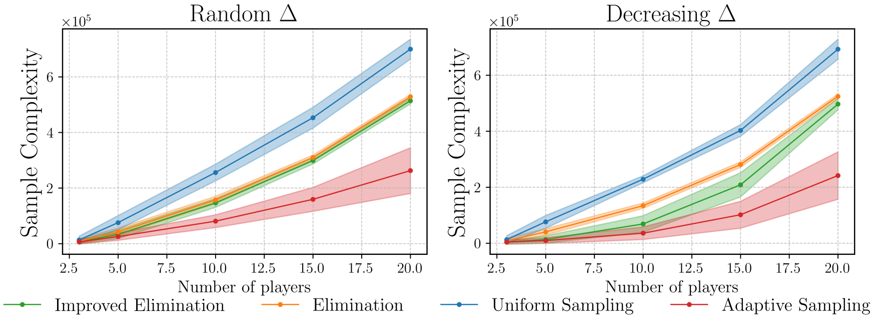

Figure 1: Sample complexity for the proposed algorithms for the two different reward settings, averaged over the runs.

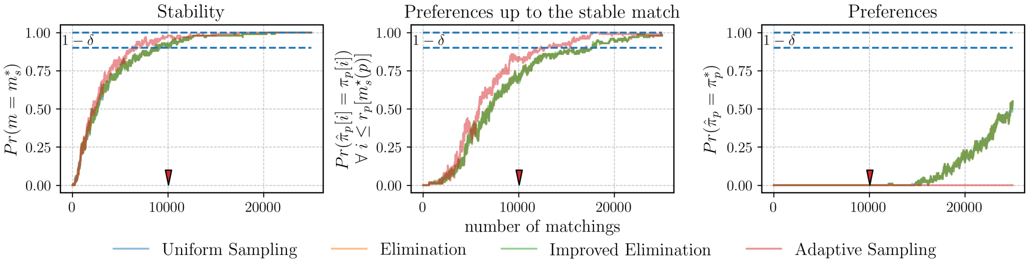

Figure 2: Any-time performance of the algorithm for the first instance with 20 agents on each side. The figure illustrates the average number of times the algorithms are able to identify (left) the optimal stable matching, (middle) the correct preferences up to the stable match for every player, and (right) the correct preferences for every player, after each matching.

8.1 Sample Complexity

First note, that in every experiment the algorithms always return the correct stable matching, similar to the study on MAB [12]. We further discuss the anytime performance of the algorithms in Section 8.2. In Figure 1, we present the average and the standard deviation of the sample complexity over the instances, for both preference settings, respectively.

In the first setting, we observe that the elimination algorithms behave similarly. This is because the randomly generated preferences do not allow the Improved Elimination Algorithm to terminate early. In the second setting, however, the Improved Elimination Algorithm outperforms the standard Elimination strategy. Additionally, the Uniform Sampling Strategy requires more samples, even with 20 players, where Remark 3 indicates similar sample complexity, as the differences in expected rewards are similar ($\Delta \approx 0.05$). This occurs because our theoretical sample complexity measures the sufficient number of samples, while in practice some arms can be eliminated earlier. Finally, the Adaptive Sampling Strategy outperforms all other algorithms in both settings, as it dynamically refines exploration based on the agents' preferences.

8.2 Anytime Performance

Now we study the performance of the algorithms at each time step t, similar to the approaches described in [12, 1] for the MAB. More specifically, after each time step t, we can check whether the DA algorithm using the current preferences, outputs the correct $m_s^\star$. We also checked if the preferences are completely correct, and correct up to the stable match, respectively. In Figure 2, show the results averaged over 100 runs for Reward Setting 1 with 20 players. The results for the remaining experiments can be found in Appendix E.

First, note that the curves for the two elimination algorithms overlap, as the only difference among them lies in the stopping criterion. The same is also true for the uniform sampling strategy. This is because elimination begins after the plotted time window (after 54.000 matchings), so the algorithms uniformly sample arms until that point. On the other hand, the adaptive algorithm quickly reduces the number of matchings to be explored, as indicated by the red arrow, which marks the average time when we need to explore less agents. This allows the algorithm to efficiently identify the preferences up to the stable pair and, consequently, the optimal stable matching. In addition, the algorithm fails to accurately estimate the preferences, as it focuses only on exploring the arms crucial for stability. Finally, we can also observe that the probability of achieving the optimal stable matching is greater than the other metrics, as indicated by Proposition 1 and Lemma 2.

9 Conclusion & Future Work

In this work, we consider the stable marriage problem under uncertain preferences on one side of the market. Our objective was to develop algorithms that efficiently identify the true optimal stable matching with high probability. To this end, we proposed the novel concept of a Probably Correct

Optimal Stable Matching and present several algorithms with theoretical analyses of correctness and sample complexity. Finally, we support our theoretical results with an empirical evaluation of our algorithms' performance in practice.

There are several interesting directions for future research. One promising area is the hardness analysis, particularly establishing a lower bound on the sample complexity for the pure exploration problem. Additionally, analyzing the principles of the Elimination Algorithm in the regret minimization setting [15] and examining its relationship to the pure exploration problem is an important direction. Another straightforward extension is to investigate how these algorithms perform when both sides of the market are uncertain about their preferences (discussed in Appendix A). The sample complexity of the Adaptive Sampling Strategy remains an open question, while experiments showcase its superiority over the other strategies. Finally, exploring alternative solution concepts, such as popular matchings [10] and median stable matchings [22], or studying alternative models for two-sided markets [20], can potentially broaden the applicability of our approach.

References

[1] P Auer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem.

[2] Soumya Basu, Karthik Abinav Sankararaman, and Abishek Sankararaman. 2021. Beyond $log^2(T)$ regret for decentralized bandits in matching markets. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, Virtual, 705–715. https://proceedings.mlr.press/v139/basu21a.html

[3] Sarah H. Cen and Devavrat Shah. 2022. Regret, stability & fairness in matching markets with bandit learners. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 151)*, Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (Eds.). PMLR, Virtual, 8938–8968. https://proceedings.mlr.press/v151/cen22a.html

[4] Sanmay Das and Emir Kamenica. 2005. Two-sided bandits and the dating market. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence* (Edinburgh, Scotland) *(IJCAI'05)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 947–952.

[5] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. 2002. PAC Bounds for Multi-armed Bandit and Markov Decision Processes. In *Proceedings of the 15th Annual Conference on Computational Learning Theory (COLT '02)*. Springer-Verlag, Berlin, Heidelberg, 255–270.

[6] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. 2006. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *J. Mach. Learn. Res.* 7 (dec 2006), 1079–1105.

[7] Harold N. Gabow and Oded Kariv. 1978. Algorithms for edge coloring bipartite graphs. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing* (San Diego, California, USA) *(STOC '78)*. Association for Computing Machinery, New York, NY, USA, 184–192. https://doi.org/10.1145/800133.804346

[8] D. Gale and L. S. Shapley. 1962. College Admissions and the Stability of Marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15. http://www.jstor.org/stable/2312726

[9] Guillaume Haeringer and Myrna Wooders. 2011. Decentralized Job Matching. *International Journal of Game Theory* 40 (02 2011), 1–28. https://doi.org/10.1007/s00182-009-0218-x

[10] Chien-Chung Huang and Telikepalli Kavitha. 2013. Popular matchings in the stable marriage problem. *Information and Computation* 222 (2013), 180–194.

[11] Meena Jagadeesan, Alexander Wei, Yixin Wang, Michael I Jordan, and Jacob Steinhardt. 2023. Learning equilibria in matching markets with bandit feedback. *J. ACM* 70, 3 (2023), 1–46.

[12] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. 2014. lil' UCB : An Optimal Exploration Algorithm for Multi-Armed Bandits. In *Proceedings of The 27th Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 35)*, Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári (Eds.). PMLR, Barcelona, Spain, 423–439. `https://proceedings.mlr.press/v35/jamieson14.html`

[13] Fang Kong, Junming Yin, and Shuai Li. 2022. Thompson Sampling for Bandit Learning in Matching Markets. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria, 3164–3170. `https://doi.org/10.24963/ijcai.2022/439` Main Track.

[14] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press, Cambridge, UK.

[15] Lydia T. Liu, Horia Mania, and Michael I. Jordan. 2020. Competing Bandits in Matching Markets. arXiv:1906.05363 [cs.LG]

[16] Lydia T. Liu, Feng Ruan, Horia Mania, and Michael I. Jordan. 2021. Bandit Learning in Decentralized Matching Markets. arXiv:2012.07348 [cs.LG]

[17] David F Manlove. 2013. *Algorithmics of Matching Under Preferences*. WORLD SCIENTIFIC, Singapore. `https://doi.org/10.1142/8591` arXiv:https://www.worldscientific.com/doi/pdf/10.1142/8591

[18] Shie Mannor and John N Tsitsiklis. 2004. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research* 5, Jun (2004), 623–648.

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

[20] Alvin E. Roth and Marilda A. Oliveira Sotomayor. 1990. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, Cambridge, UK.

[21] Abishek Sankararaman, Soumya Basu, and Karthik Abinav Sankararaman. 2021. Dominate or Delete: Decentralized Competing Bandits in Serial Dictatorship. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 130)*, Arindam Banerjee and Kenji Fukumizu (Eds.). PMLR, Virtual, 1252–1260. `https://proceedings.mlr.press/v130/sankararaman21a.html`

[22] Chung-Piaw Teo and Jay Sethuraman. 1998. The geometry of fractional stable matchings and its applications. *Mathematics of Operations Research* 23, 4 (1998), 874–891.

[23] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3-4 (1933), 285–294.

A Uncertainty for both sides of the market.

In the main text, we discuss the case where only one side of the market is uncertain about its preferences. An interesting research direction is to consider the scenario where both sides of the market have uncertain preferences.

Here, we discuss how our results can potentially generalize in this setting. To begin with, our Proposition 1 also holds in this case, as we consider every agent in the market. The only difference is that we need to adjust the bounds in the proofs to account for the event that every agent has correctly estimated their preferences. Therefore, there must be additional factors based on the number of agents inside the logarithm.

Regarding the algorithms, the analysis of the uniform sampling strategy is straightforward, as one can obtain enough samples based on the minimum reward difference Δ for both sides. For the case of the elimination algorithm, adjustments can also be made if a player-arm pair is eliminated when both agents are able to separate the confidence intervals w.r.t the agents of the other side of the market.

Additionally, the stopping rule used in the Improved Elimination Algorithm, as outlined in Lemma 2, can also be modified to consider preferences that are correct up to any stable match for both sides of the market. This can also be used to adjust the adaptive sampling strategy.

B Proof of Theorem 1

Proof. We define the event C_p as the event in which one player $p \in \mathcal{P}$ correctly estimates their preferences, and the event C the event in which all players correctly estimate their preferences. Additionally, we let the event $\mathcal{E}$ represent the case that the algorithm correctly outputs the optimal stable matching.

Step 1 - bounding $P(\mathcal{E})$
From Proposition 1 we know that:

$$P(\mathcal{E}) \geq P(C) = 1 - P(C^c) = 1 - P(\bigcup_{p\in\mathcal{P}} C_p^c),$$

where C_p^c denote the opposite event of C_p. Using the union bound we have that:

$$P(\mathcal{E}) \geq 1 - \sum_{p\in\mathcal{P}} P(C_p^c).$$

In order for our algorithm to be δ-PACOS, i.e., to bound $P(\mathcal{E}) \geq 1-\delta$, we need:

$$P\left(C_p^c\right) \leq \delta/N \quad \forall p \in \mathcal{P}.$$

Step 2 - bounding $P\left(C_p^c\right)$
The event C_p for a player p to correctly estimate their preference is equal to the intersection of the events that all consecutive arms (according to the preference) are correctly ordered, i.e., $Pr(C_p) = Pr(\bigcap_{i=1}^{K-1} C_{p,i,i+1})$. Here $C_{p,i,i+1}$ denotes the event that the i-th ranked agent a and $(i+1)$-th ranked agent a' in $\hat{\pi}_p$ are correctly ordered, i.e., $\mu_{p,a} > \mu_{p,a'}$. The opposite event C_p^c can be bounded using the union bound as:

$$\begin{aligned} P\left(C_p^c\right) &= P\left(\bigcup_{i=1}^{K-1} C_{p,i,i+1}^c\right) \\ &\leq \sum_{i=1}^{K-1} P\left(C_{p,i,i+1}^c\right) \leq K \max_i P\left(C_{p,i,i+1}^c\right). \end{aligned}$$

In order for $P(C^c) \leq \delta$ to hold we next aim to establish the bound $\max_{p,i} P\left(C_{p,i,i+1}^c\right) \leq \delta/NK$.

Step 3 - bounding $P\left(C_{p,i,i+1}^c\right)$
Consider two consecutive arms, a_i and a_j, with $\Delta_{p,i,j} = \mu_{p,i} - \mu_{p,j} > 0$ for an arbitrary player p. The event $C_{p,i,j}^c$ that the algorithm incorrectly orders the arms a_i and a_j after h samples from each arm, occurs when $\hat{\mu}_{p,i}(h) \leq \hat{\mu}_{p,j}(h)$. Using the Hoeffding inequality, we can upper bound the probability of the event $C_{p,i,j}^c$ as follows, omitting the indexing for player p for simplicity:

$$\begin{aligned} &\Pr(C_{p,i,j}^c) = \Pr(\hat{\mu}_i(h) \leq \hat{\mu}_j(h)) \\ &\leq \Pr\left(\hat{\mu}_i(h) < \mu_i - \frac{\Delta_{i,j}}{2}\right) + \Pr\left(\hat{\mu}_j(h) > \mu_j + \frac{\Delta_{i,j}}{2}\right) \\ &\leq 2\exp\left(-\frac{\Delta_{i,j}^2}{2}h\right) \end{aligned}$$

Step 4 - sample complexity
In order to bound $\max_{p,i} P\left(C_{p,i,i+1}^c\right) \leq \delta/NK$ we need:

$$2\exp\frac{-h\Delta_{min}^2}{2} \leq \delta/NK \Rightarrow h \geq \frac{2\ln(2KN/\delta)}{\Delta_{\min}^2},$$

where $\Delta_{\min} = \min_{p,i,j} \Delta_{p,i,j}$

To conclude, w.p. at leat $1-\delta$, the sample complexity of the algorithm is

$$O(hK) = O(\frac{K\ln(KN/\delta)}{\Delta_{\min}^2}).$$

□

C The minimum matching cover

This section explains how we match agents in each round of our process in order to efficiently sample all pairs (p,a) (line 5 of Algorithm 2). Specifically, in each round t, we sample every non-eliminated pair, i.e., $(p,a) \in \mathcal{P} \times S_p$, exactly once using the minimum number of matchings.

Now, consider the bipartite graph $G_t = \{V, E_t\}$, where the vertices $V = \{\mathcal{P}, \mathcal{A}\}$ represent the players and arms, and the edges E_t connect the non-eliminated pairs $(p,a) \in \mathcal{P} \times S_p$ at round t. The objective is to cover all available pairs using the minimum number of matchings, i.e., a cardinality-minimal set of matchings $\mathfrak{M}$ that covers all remaining pairs.

In bipartite graphs, this problem can be formulated as a *minimum edge coloring problem*, where matchings correspond to edges of the same color [7]. According to the Kőnig-Hall Theorem, for a bipartite graph G, the minimum number of colors required for edge coloring is equal to the maximum degree $\Delta(G)$ of its vertices. Thus, we need $\Delta(G_t)$ matchings to cover all edges in the graph at every round t.

Our method can be implemented using any edge coloring algorithm. In the experiments section, we use a recoloring edge coloring algorithm that recolors edges using augmenting paths when no available colors are left, as described in [7].

D Proof of Theorem 2

Proof.
step 1 - Anytime confidence intervals

We start our proof by noticing that the values $B_t = \sqrt{\frac{\ln(4KNt^2/\delta)}{2t}}$ in Algorithm 2 correspond to bounds of any-time confidence intervals $CI_{p,a}(t) = [\hat{\mu}_{p,a}(t) - B_t, \hat{\mu}_{p,a}(t) + B_t]$. More specifically, with probability at least $1-\delta$, at any time-step t, the expected rewards $\mu_{p,a}$ lie in the confidence interval $CI_{p,a}(t)$ for every pair of player p and arm a. Let us denote this event by:

$$\mathcal{E} = \bigcap_{p\in\mathcal{P}} \bigcap_{a\in\mathcal{A}} \bigcap_{t=1}^{\inf} \{\mid \hat{\mu}_{p,a}(t) - \mu_{p,a} \mid \leq B_t\} \tag{5}$$

To prove that $\Pr(\mathcal{E}) \geq 1-\delta$, we first consider the individual events $\mathcal{E}_{p,a}(t)$ that the true mean $\mu_{p,a}$ lies in the confidence interval $CI_{p,a}(t)$ for some player p, arm a, and time t. We can bound the probability of the complement event $\mathcal{E}^c_{p,a}(t)$, i.e., $\Pr[\mid \hat{\mu}_{p,a}(t) - \mu_{p,a} \mid > B_t]$ using the Hoeffding inequality and $B_t = \sqrt{\frac{\ln(4KNt^2/\delta)}{2t}}$ as:

$$\Pr[\mathcal{E}^c_{p,a}(t)] = \Pr[\mid \hat{\mu_i}(t) - \mu_i \mid > B_t] \leq 2\exp(-2{B_t}^2 t) \leq \frac{\delta}{2NKt^2} \tag{6}$$

Using the union bound over all rounds,

$$\Pr[\bigcup_{t=1}^{\inf} \mathcal{E}^c_{p,a}(t)] \leq \sum_{t=1}^{\inf} \Pr[\mathcal{E}^c_{p,a}(t)] \leq \frac{\delta}{NK} \tag{7}$$

Using the union bound over all possible players and arms, we can bound the probability that the mean estimate for some arm is outside the confidence intervals, i.e.:

$$\Pr[\mathcal{E}^c] \leq \sum_{p\in\mathcal{P}} \sum_{a\in\mathcal{A}} \Pr[\bigcup_{t=1}^{\inf} \mathcal{E}^c_{p,a}(t)] \leq \delta \tag{8}$$

So at any time t, for every player, all arms have estimated rewards within their respective confidence intervals with probability at least $1-\delta$:

$$\Pr[\mathcal{E}] \geq 1-\delta \tag{9}$$

step 2 - Correctness argument
Now we will prove that our algorithm outputs the correct optimal stable matching $m_s^\star$ with probability at least $1-\delta$. To prove this, we first demonstrate that as long as the event $\mathcal{E}$ holds, our algorithm returns the correct ranking for every player. This is sufficient for obtaining the correct match. Then according to Proposition 1, we can compute the correct stable optimal matching using the DA algorithm based on these preferences.

In the following, we drop the p subscript to simplify notation. From 9) we know that event $\mathcal{E}$ holds with high probability. In that case, the following hold:

1. First, our algorithm will eventually terminate, as B_t approaches zero as t increases. This ensures that there will be no overlapping arms for every player.

2. Second, consider two arbitrary arms a_i and a_j with $\mu_i \geq \mu_j$. If the arms have no overlapping confidence intervals, i.e., $\hat{\mu}_i - \hat{\mu}_j \geq 2B_t$, we can determine their relative ranking, i.e., $\mu_i > \mu_j$, as follows:

$$\begin{aligned} \mu_i - \mu_j &\geq \hat{\mu}_i(t) - B_t - (\hat{\mu}_j(t) + B_t) && (10)\\ &\geq \hat{\mu}_i(t) - \hat{\mu}_j(t) - 2B_t && (11)\\ &\geq 0. && (12) \end{aligned}$$

   Thus, we conclude that $\mu_i \geq \mu_j$ as required.

3. Third, if an arm is eliminated at time t, it maintains its relative order according to the mean estimates when we continue to sample the remaining arms, as the confidence intervals of the other arms only shrink as t increases.
   To illustrate this, consider two arms a_i and a_j with $\Delta_{i,j} = \mu_{a_i} - \mu_{a_j} > 0$, without loss of generality. Let also a_i be eliminated at some round t while we continue to sample a_j. Since arm a_i is eliminated, we know that $\hat{\Delta}_{i,j} = \hat{\mu}_i(t) - \hat{\mu}_j(t) \geq 2B_t$.
   In some $t' > t$, where we continue to sample a_j, we have a new estimate of the expected rewards, $\hat{\mu}_j(t')$. Now consider the worst-case scenario where at round t we underestimate the expected rewards and at round t' we overestimate them, leading to:

$$\begin{aligned} \hat{\mu}_j(t') - \hat{\mu}_j(t) \leq \mu + a_{t'} - (\mu - a_t) \leq 2B_t \Rightarrow && (13)\\ \hat{\mu}_j(t') \leq \hat{\mu}_j(t) + 2B_t. && (14) \end{aligned}$$

   Therefore, the relative position of the arm remains the same as:

$$\begin{aligned} \hat{\mu}_i(t) - \hat{\mu}_j(t') &\geq \hat{\mu}_i(t) - \hat{\mu}_j(t) - 2B_t && (15)\\ &= \hat{\Delta}_{i,j} - 2B_t \geq 0 \Rightarrow && (16)\\ \hat{\mu}_{a_i}(t) &\geq \hat{\mu}_{a_j}(t'). && (17) \end{aligned}$$

   Thus, the eliminated arm maintains its relative position according to the mean estimates as long as the average rewards are between the confidence intervals.

4. Fourth, when an arm is eliminated, we know its relative position compared to all other arms with high probability, as it has no overlapping confidence intervals with any other arm.

5. Finally, when all arms are eliminated for a player, we know the relative position between every arm, and thus we have the correct preferences for the player with high probability.

To conclude, using Proposition 1 our algorithm outputs the correct optimal stable matching with probability at least $1-\delta$.

step 3 - Sample Complexity
We now begin the analysis of the sample complexity by first presenting a Lemma that bounds the sufficient number of samples required to eliminate an arbitrary arm for a player. We then derive the

overall sample complexity by accounting for the elimination process, as outlined in the main text.

step 3.1 - Sample Complexity to eliminate an arm $t_{p,a}$
Now we present a detailed proof for Lemma 1.

Lemma 1. *For a player* $p \in \mathcal{P}$ *and arm* $a \in \mathcal{A}$, *let* $\Delta_{p,a} = \min_{a' \in \mathcal{A} \setminus \{a\}} \Delta_{p,a,a'}$. *With probability at least* $1 - \delta$, *we need at most* $t_{p,a}$ *samples to eliminate an arm* a *from* S_p *in line 9 of Algorithm 2:*

$$t_{p,a} = O\left(\frac{\ln\left(KN/\delta\Delta_{p,a}\right)}{\Delta_{p,a}^2} \right). \tag{18}$$

Proof. First, consider some fixed player p and drop the subscript p for simplicity. Let us take two arbitrary arms a_i and a_j with $\mu_i \geq \mu_j$. In order for the arms to have non-overlapping confidence intervals, we want:

$$\hat{\Delta}_{i,j} = \hat{\mu}_i(t) - \hat{\mu}_j(t) > 2B_t \tag{19}$$

If the event $\mathcal{E}$ holds for all the arms and time steps, the mean estimates of the arms are within the confidence intervals.
So for an arbitrary arm a_k we have:

$$\mu_k + B_t \geq \hat{\mu}_k(t) \geq \mu_k - B_t \tag{20}$$

and thus:

$$\hat{\mu}_i(t) \geq \mu_i - B_t \quad \textit{and} \quad -\hat{\mu}_j(t) \geq -(\mu_j + B_t) \tag{21}$$

So the event of equation 19 is guaranteed to occur if $\Delta_{i,j} > 4B_t$ as it implies that:

$$\hat{\Delta}_{i,j} = \hat{\mu}_i(t) - \hat{\mu}_j(t) \geq (\mu_i - B_t) - (\mu_j + B_t) = \Delta_{i,j} - 2B_t \Rightarrow \tag{22}$$

$$\hat{\Delta}_{i,j} > 2B_t \tag{23}$$

By solving for the minimum value of t for which $\Delta_{i,j} > 4B_t$, we can upper bound the number of samples sufficient for the arms to have non-overlapping confidence intervals. Similarly to the analysis in [6] we have:

$$t_{a_i,a_j} = O\left(\frac{\ln\left(NK/\delta\Delta_{i,j}\right)}{\Delta_{i,j}^2} \right) \tag{24}$$

Now we can apply this result to each player p. In order for an arm a to be eliminated it must have non-overlapping CI with every other arms of the player p. So, under the event $\mathcal{E}$ witch holds w.p.a. $1 - \delta$, the sufficient number of samples $t_{p,a}$ are:

$$t_{p,a} = O(\frac{\ln\left(NK/\delta\Delta_{p,a}\right)}{\Delta_{p,a}^2}) \tag{25}$$

where $\Delta_{p,a} = \min_{a' \in \mathcal{A} \setminus \{a\}} \Delta_{p,a,a'}$, as it has to resolve the uncertainty of the closest arm.

□

step 3.2 - Overall sample complexity
Here we measure sample complexity in terms of the number of matchings that are sufficient for outputting the correct optimal stable match with high probability.

In each round t, our algorithm samples every non-eliminated player-arm pair exactly once. The number of matchings in each round t corresponds to the degree $\Delta(G_t)$ of the underlying bipartite graph $G_t = \{V, E_t\}$, with vertices $V = \{\mathcal{P}, \mathcal{A}\}$ and the edges E_t connect the non-eliminated pairs $\{(p,a) \in \mathcal{P} \times \mathcal{A} \mid a \in S_p\}$. To reduce the number of matchings in each round, we have to reduce the degree of the graph.

We thus consider $s = 1, \ldots, K$ phases where each phase corresponds to the elimination of the subset of pairs r_s to reduce the degree of the graph. So under the event $\mathcal{E}$, we can use Lemma 1 to define r_s as the set of pairs with the lowest sample complexity i.e.:

$$r_i = \arg \min_{r \in \mathcal{P}(S \setminus \bigcup_{j<i} r_j)} \max_{(p,a) \in r} t_{p,a} \; \forall i = 1, \ldots, K$$

with

$$\mathcal{P}(S) = \{r \subseteq S : \Delta(G(S)) - \Delta(G(S \setminus r)) = 1\}$$

denote the subsets of pairs that can reduce the degree of the graph. Note that here the $r_1, \ldots, r_K$ form a partition of all player-arm pairs.

In every phase s, we perform $t_s - t_{s-1}$ iterations, where t_s are sufficient number of steps to eliminate r_s determined by the player-arm pair $(p, a) \in r_s$ that takes the longest to eliminate i.e.: $t_s = \max_{(p,a) \in r_s} t_{p,a}$ with $t_0 = 0$.

Finally, in each step t, every matching cover consists of $K - s + 1$ matchings. Thus, under the event $\mathcal{E}$ witch holds w.p.a. $1 - \delta$, the total number of matching samples is given by:

$$\sum_{s=1}^{K} (K - s + 1)(t_s - t_{s-1}) = \sum_{s=1}^{K} t_s = O\left(\sum_{s=1}^{K} \max_{(p,a) \in r_s} t_{p,a}\right)$$

To conclude our Algorithm 2 outputs the correct stable matching with probability at least $1 - \delta$ after at most:

$$O\left(\sum_{s=0}^{K} \max_{(p,a) \in m_s} t_{p,a}\right) = O\left(\sum_{s=0}^{K} \max_{(p,a) \in m_s} \frac{\ln\left(KN/\delta\Delta_{p,a}\right)}{\Delta_{p,a}^2}\right)$$

matching samples. □

E Simulations

In Figures 3 and 4, we present the results on the anytime performance of the algorithm, similar to the subsection 8.2, for different numbers of players and both settings, respectively.

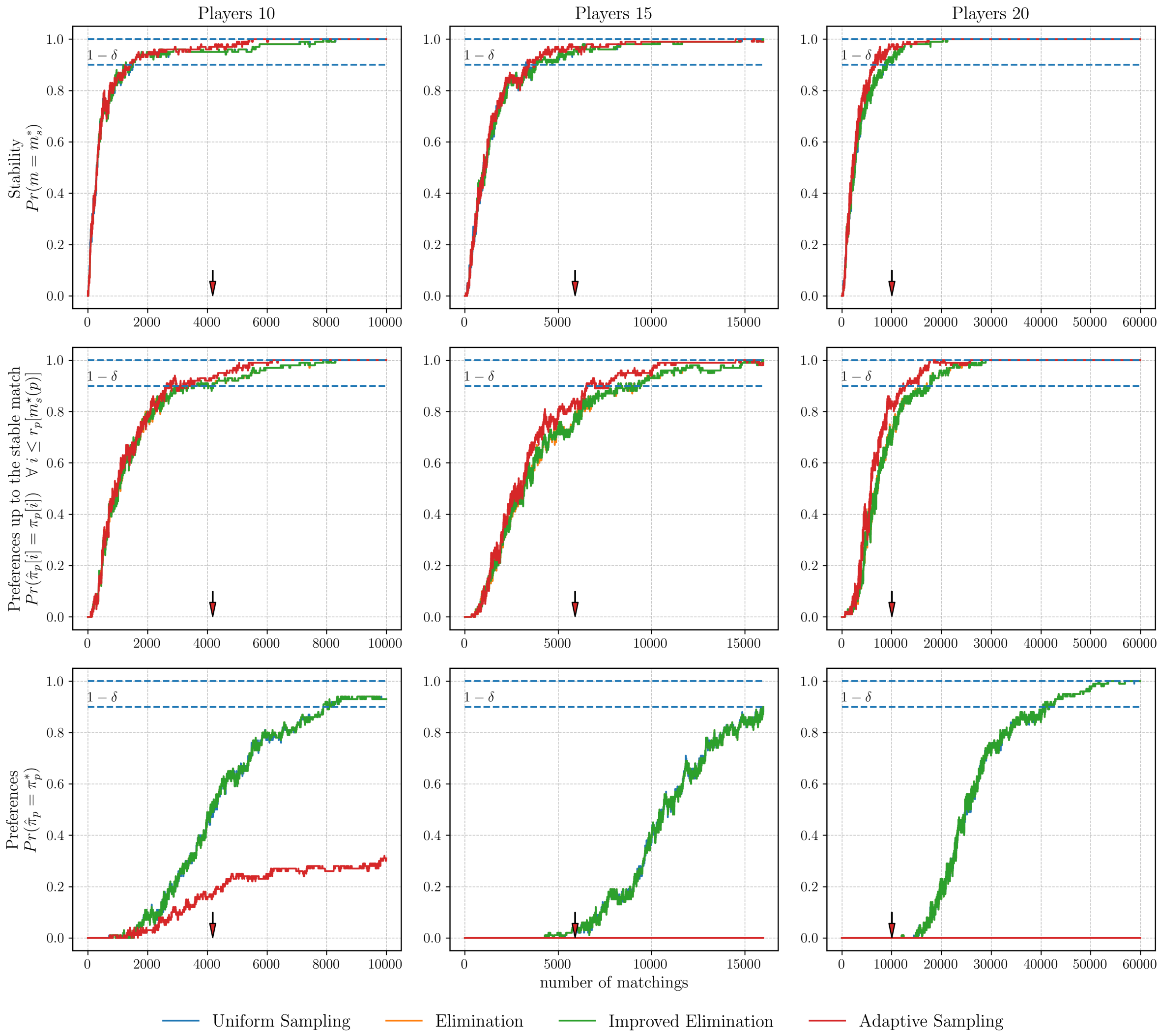

Figure 3: Anytime performance of the algorithm for the first instance setting with varying numbers of agents on each side. The figure illustrates the average number of times the algorithm identifies (left) the optimal stable matching, (middle) the correct preferences up to the stable match for each player, and (right) the correct preferences for every player after each matching.

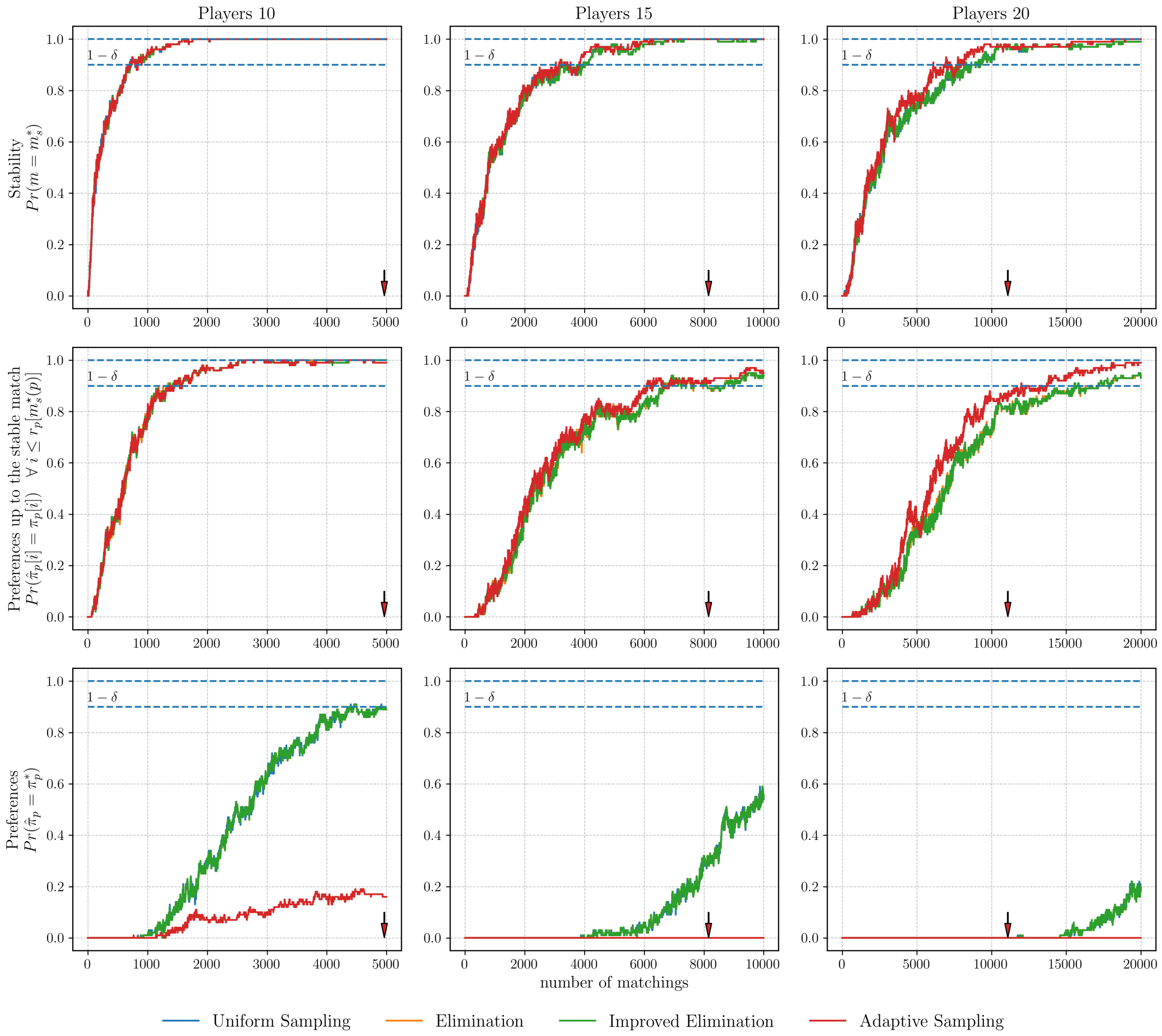

Figure 4: Anytime performance of the algorithm for the second instance setting with varying numbers of agents on each side. The figure illustrates the average number of times the algorithm identifies (left) the optimal stable matching, (middle) the correct preferences up to the stable match for each player, and (right) the correct preferences for every player after each matching.