Converging to Stability in Two-Sided Bandits: The Case of Unknown Preferences on Both Sides of a Matching Market

Gaurab Pokharel, Sanmay Das
Virginia Tech
Alexandria, VA 22305, USA
{gaurab, sanmay}@vt.edu

Abstract

We study the problem of repeated two-sided matching with uncertain preferences (two-sided bandits), and no explicit communication between agents. Recent work has developed algorithms that converge to stable matchings when one side (the proposers or agents) must learn their preferences, but the preferences of the other side (the proposees or arms) are common knowledge, and the matching mechanism uses simultaneous proposals at each round. We develop new algorithms that converge to stable matchings for two more challenging settings: one where the arm preferences are no longer common knowledge, and a second, more general one where the arms are also uncertain about their own preferences. In our algorithms, agents start with optimistic beliefs about arms' preferences, updating these preferences over time, and combining beliefs about preferences with beliefs about the value of matching when choosing whom to propose to.

1 Introduction

The classic literature on two-sided matching (Gale and Shapley 1962; Roth and Xing 1997; Haeringer and Wooders 2011, e.g.), encompassing applications including long- and short-term labor markets, dating and marriage, school choice, and more, has typically focused on situations where agents are aware of their own preferences. The problem of learning preferences while participating in a repeated matching market first started receiving attention in the AI literature in the work of Das and Kamenica [2005], and the general idea of two-sided matching under unknown preferences has since been studied in economics and operations research as well (Lee and Schwarz 2009; Johari et al. 2022). This area of research has received renewed attention in the last few years, along with novel theoretical insights into convergence properties of upper-confidence-bound (UCB) style algorithms (Liu et al. 2021; Kong, Yin, and Li 2022; Zhang, Wang, and Fang 2022).

The two-sided matching problem involves agents on two sides of a market who have preferences for each other but cannot communicate explicitly. The goal is to create a matching process that ensures *stability*, where no pairs of agents would rather be matched with each other over their current match. The existence of such matchings was famously demonstrated constructively in the Gale-Shapley algorithm (Gale and Shapley 1962) which structured around one side of the market *proposing* and the other side choosing whether to accept proposals. This theory has been applied to various markets, such as matching medical students to residencies (Roth and Peranson 1999) and students to schools (Abdulkadiroğlu et al. 2005), with the assumption that agents know their own preferences. Considerable interest has also been shown in the AI community regarding two-sided matching in the presence of various constraints, such as diversity constraints (Aziz, Biró, and Yokoo 2022; Benabbou et al. 2018). A version of the problem, called *two-sided bandits*, was introduced in Das and Kamenica [2005], where agents engage in repeated matching without prior knowledge of preferences, and the impact of the matching mechanism on *convergence* to stability was studied. Recent work has also focused on computing the probability that a matching is stable or finding the matching with the highest probability of being stable under different models of preference uncertainty (Aziz et al. 2020).

The exploration-exploitation dilemma that characterizes bandit problems is further complicated in two-sided bandits by uncertainty on not just the values of the "arms" of the bandit, but also uncertainty on whether the arms will accept or reject your attempt to pull them (your proposal). This additional uncertainty arises from the multi-agent nature of the problem; instead of there being just one player, there are multiple players competing for the arms, and the arms also have preferences associated with the players. A popular solution choice for this type of problem involves the explore-then-commit style of algorithms (Pagare and Ghosh 2023; Zhang and Fang 2024). This flavor of solution involves a first phase of "learning" preferences, and a second phase using what has been learned in the prior one to commit to a choice. This requires the entire agent population coordinating on when the exploration phase ends and the commitment phase begins. We are instead interested in approaches where learning happens simultaneously with an ongoing matching process. One such algorithm, CA-UCB (Liu et al. 2021), assumes that at each round all the agents on the receiving side get to view all their proposals before deciding which one to accept. The algorithm provably converges to stability under the assumptions that (1) all arms have complete knowledge

of their preferences, and (2) these preferences are common knowledge, so the proposers also know them. The algorithm is a variation of the well-known family of UCB algorithms.

Contributions In this paper, we are interested in the general case where neither the agents nor the arms start off with knowledge of their own preferences. This problem is significantly more complex because it adds a layer of uncertainty that could potentially lead to agents converging to an unstable equilibrium more easily than in the case with one-sided uncertainty and common knowledge of the other side's preferences. We tackle this in two steps. First, we relax the common knowledge assumption and assume only that arms are aware of their own preferences.

Recent works have used frameworks where agents communicate with each other to find a matching in this setting (Zhang, Wang, and Fang 2022; Zhang and Fang 2024). While there are approaches which allow for and design explicit communication frameworks (Zhang, Wang, and Fang 2022), and also others where players only communicate indirectly to the extent of carefully orchestrating "collisions" (Zhang and Fang 2024), we resolve the question of convergence in the more traditional non-cooperative setting of stable matching when arm preferences are not common knowledge. Our algorithm uses the concept of "plausible sets" (arms that an agent might be able to "win" in the next time step) from the CA-UCB algorithm (Liu et al. 2021). We show how to define and update these sets in order to develop a provably convergent algorithm (which we call Optimistic CA-UCB or OCA-UCB). The case with two-sided uncertainty is more challenging. We introduce a framework (the Probabilistic Conflict Avoiding - Simultaneous Choice Algorithm) that maintains and uses an additional optimistic estimate on the probability of winning each possible contentious situation. We show how to instantiate the algorithm with both UCB and Thompson Sampling style estimates, and empirically demonstrate convergence to stability. Finally, we experimentally evaluate the speed of convergence as a function of the number of agents in the market and the level of preference heterogeneity.

2 Setting

There are N proposers (henceforth players) and K proposees (henceforth arms). To ensure that each player can be matched, we assume that $N \leq K$ (Liu, Mania, and Jordan 2020; Liu et al. 2021; Basu, Sankararaman, and Sankararaman 2021; Sankararaman, Basu, and Abinav Sankararaman 2021). The set of players is denoted by $\{p_i\}_{i=1}^{N}$ and the set of arms by $\{a_k\}_{k=1}^{K}$. For each p_i, there is a distinct (unknown) mean reward μ_i^k associated with each arm a_k with unit variance. If $\mu_i^k > \mu_i^j$, we say that p_i prefers a_k over a_j denoted by $a_k \succ_{p_i} a_j$. Similarly, the arms also have distinct mean rewards associated with each player. The arms may or may not have this information available to them a priori depending on the setting in consideration.

At each time-step $t \in \{1, \cdots, T\}$, player p_i attempts to pull an arm a_k. Let $A_i(t)$ and $\bar{A}_i(t)$ denote, respectively, the arm that player p_i attempted to pull, and the arm that player p_i successfully pulled at time t. If multiple players attempt to pull the same arm, a conflict will arise. If the arms are aware of their preferences, the player that is the most preferred by the arm will be picked in the event of a conflict (if the arms are not fully aware of their preferences, then the chosen player will depend on the arm's decision-making algorithm). At the end of each time-step, if a player p_i is successfully matched with an arm i.e. $\bar{A}_i(t) = m_i(t)$, it will receive a stochastic reward $X_i^{m_i(t)} \sim \mathcal{N}(\mu_i^{m_i(t)}, 1)$. Players that fail to successfully pull an arm will receive a reward of 0 and $\bar{A}_i(t) = \emptyset$. The final matching is made public to all the agents at the end of each time step.

We use the classic notion of a stable matching: a bipartite matching from the group of players to the group of arms such that no player and arm prefer each other over their current matching i.e $\nexists(p_i, m_i(t))$ such that p_i prefers an arm $a_j \neq m_i(t)$ and a_j also prefers p_i over its current match. There may be more than one stable match. Our primary focus in this work is on convergence to stability. Another quantity we study is maximum Player Pessimal Regret. We use the Gale-Shapley algorithm with arms as proposers in order to calculate the player-pessimal stable match (Gale and Shapley 1962). Denote the reward received by player p_i in a player pessimal match by X_i. Then, we define maximum player-regret at each time-step as $R(t) = \max_{i \in N} \left[X_i - X_i^{m(t)} \right]$ which is the maximum difference (over players) between the reward a player received in its current match and the reward it would have received in a stable matching that is player-pessimal.

In this work we consider matching markets with varying levels of information availability. As is standard in models of learning in two-sided matching, we assume no explicit communication between players on the proposing side. Throughout the rest of the paper we assume that the players do not have access to their own preferences and need to learn them over time. As mentioned above, we consider several scenarios in terms of arm's preferences. For convenience, we name those as follows:

- Scenario 0: Players have access to arm preferences. Arms have access to arm preferences. We call this the **Arm Preferences Common Knowledge (APCK)** model.
- Scenario 1: Players do not have access to arm preferences. Arms have access to arm preferences. We call this the **Arm Preferences Known but Private (APKP)** model.
- Scenario 2: Players do not have access to arm preferences. Arms do not have access to arm preferences. We call this the **Arm Preferences Unknown (APU)** model.

The APCK model has been studied and a decentralized solution approach is described in the Conflict Avoiding Upper Confidence Bound (CA-UCB) algorithm (Liu et al. 2021). Our main contribution is to detail decentralized approaches to solutions for Scenarios 1 and 2. We begin with a quick recap of CA-UCB in the APCK model.

3 APCK Model and CA-UCB

In a recent paper, Scenario 0 was studied (Liu, Mania, and Jordan 2020). Subsequently, a solution was proposed that is decentralized, the CA-UCB algorithm (Liu et al. 2021). CA-UCB finds a stable match in Scenario 0 by avoiding conflicts using the notion of a "plausible set." The plausible set for a player p_i at time-step t is defined as $S_i(t) := \{a_k : p_i \succ_{a_k} p_j, \text{ where } \bar{A}_j(t-1) = a_k\}$ i.e. the set of all arms a_k such that at time-step $(t-1)$ another player $p_j \neq p_i$ successfully pulled the arm, but p_i knows that a_k prefers p_i over p_j. This set also includes the arms the player successfully pulled (if any) and all the arms that were unmatched at the previous time-step. This makes it so that p_i only attempts arms where there is a possibility of it successfully pulling the arm, avoiding conflicts. Then, p_i attempts to pull the arm with the highest UCB value among $S_i(t)$.

4 APKP Model and OCA-UCB

Note that the CA-UCB approach works only when both players and arms have full knowledge of the arms' preferences, which players use to construct $S_i(t)$. When players lack this information, they must learn their positions in the arms' preference orderings and recreate plausible sets. Recent game-theoretic approaches address this scenario. For instance, (Etesami and Srikant 2024) allows proposers to maintain and update probability distributions over potential arms based on feedback from previous rounds. Accepted proposals increase the likelihood of future proposals to that arm, while rejections decrease it. These methods guarantee convergence to a stable matching without centralized coordination or full knowledge of preferences, achieving logarithmic regret in *hierarchical* (Etesami and Srikant 2024) markets. Similarly, some approaches use mood-based state variables—like "content," "discontent," and "watchful"—to govern each player's strategy at each time-step (Shah, Ferguson, and Marden 2024). Based on utility feedback, proposers decide whether to stick with their current match or experiment with new proposals. This ensures that over repeated interactions, proposers learn their preferences and the system converges to the proposer-optimal stable match, even in fully decentralized and information-limited markets. We present a much simpler approach to the solution which requires "collisions" with other players at most once for the players to learn arm preferences. We call it OCA-UCB *(Optimistic CA-UCB)* which exploits the fact that the arms still have full knowledge of their own preferences in order to achieve this.

Following the general structure of the CA-UCB algorithm, all players initially set their UCB estimates for the arms to ∞. However, now in addition to the UCB estimates, each player will also keep track of their position in the arms' preference orderings. This can be done simply by having each player maintain two sets: $O_{(i)}^{k_h}$ the set of all players that p_i believes are ranked higher than itself in a_k's preference ordering, and $O_{(i)}^{k_l}$ which is the set of all players that p_i believes are ranked lower than itself. Initially, for each player p_i, these sets are initialized to $O_{(i)}^{k_h} = \emptyset$ and $O_{(i)}^{k_l} = \{p_j\}_{j=1}^{N} \setminus p_i$. Thus, each player starts with the optimistic belief that it is the most preferred by each of the arms.

Notice, when updating plausible sets, the only information that p_i requires is its' relative positioning in a_k's preference with respect to p_j if $\bar{A}_j(t-1) = a_k$. This can be accomplished using the two sets described above. Since each player starts out with an optimistic view of itself in the arms' preference orderings, conflicts are almost inevitable. This is because according to each player, all the arms are in the plausible set, even when they ought not to be. Suppose conflict occurs between two players, say p_i and p_j, for an arm a_k at time-step t, and player p_j wins the conflict, then p_i's belief with respect to p_j and a_k gets updated. More precisely, $O_{(i)}^{k_h} = O_{(i)}^{k_h} \cup \{p_j\}$ and $O_{(i)}^{k_l} = O_{(i)}^{k_l} \setminus \{p_j\}$. At future time steps, p_i can use this information when considering a_k in its plausible set. Functionally the rest of the algorithm is the same as CA-UCB. The only difference is that, at the end of each time-step, the players make use of the matching information and update $O_{(i)}^{k_h}$ and $O_{(i)}^{k_l}$ accordingly.

Eventually, each of the players will have complete sets that represent the arms' true preferences with respect to the other players. In fact, we show theoretically that this approach to the solution does not affect the convergence guarantee of the CA-UCB algorithm in the APCK Model. First, let us define some terminology. A triplet $Q_{ij}^k = (p_i, p_j, a_k)$ is *inconsistent* if p_i believes $p_i \succ_{a_k} p_j$ but the opposite is true. Then, let us define a plausible set constructed by p_i to be inconsistent with respect to p_j and a_k if $a_k \in S_i(t)$, denoted by $\hat{S}_{ij}^k(t)$. Finally, let us define an inconsistent plausible set as being *inconsequential* at time t if $\arg\max_m\{\text{UCB}_i^m : a_m \in S_i(t)\} \neq k$ i.e. p_i does not attempt a_k OR if $\arg\max_m\{\text{UCB}_j^m : a_m \in S_j(t)\} \neq k$ i.e. p_j does not attempt a_k. We begin the proof of the convergence guarantee by first presenting two lemmas:

Lemma 4.1. *At each time-step, inconsistent sets $\hat{S}_{ij}^k(t)$ are either inconsequential or get resolved such that $S_i(t+1)$ is no longer inconsistent.*

Proof. Consider an inconsistent triplet $Q_{ij}^k = (p_i, p_j, a_k)$ such that $\bar{A}_j(t-1) = a_k$. Then, at time t, $\hat{S}_{ij}^k(t)$- the plausible set constructed by p_i will be inconsistent. One of three things must happen at time t with regard to $\hat{S}_{ij}^k(t)$. (1) p_j attempts a_k either due to λ, or because $\arg\max_m\{\text{UCB}_j^m : a_m \in S_j(t)\} = k$. p_i also attempts a_k because $\arg\max_m\{\text{UCB}_i^m : a_m \in \hat{S}_{ij}^k(t)\} = k$. This leads to a conflict, resulting in the inconsistency getting resolved via p_i receiving feedback about matching information. (2) p_j attempts a_k but p_i does not. This implies $\arg\max_m\{\text{UCB}_i^m : a_m \in \hat{S}_{ij}^k(t)\} \neq k$, which in turn implies $\hat{S}_{ij}^k(t)$ is inconsequential with respect to p_j and a_k. (3) p_j does not attempt a_k but p_i does. This implies $\hat{S}_{ij}^k(t)$ is inconsequential with respect to p_j and a_k. However, two further distinct sub-cases follow from this. First, if $\bar{A}_i(t) = a_k$

then p_i was the most preferred among a_k's incoming pull requests. The plausible set constructed at $(t+1)$ is no longer inconsistent with respect to p_j and a_k because a_k will be included in it by virtue of it being successfully pulled by p_i at t, and not because of p_i's belief Q^k_{ij}. And second, if $\bar{A}_i(t) \neq a_k$ then a_k will not be included in plausible set constructed at $(t+1)$ as p_i will lose a conflict with $M(a_k)$ and p_i's belief will be updated with respect to $M(a_k)$ and a_k if it has not already. □

Lemma 4.2. *Eventually all inconsistent plausible sets constructed are resolved or are inconsequential.*

Proof. First, begin by noting that there are only finitely many inconsistent triplets in any given APKP model. Then, a simple recursive argument will suffice to show that each one gets resolved or becomes inconsequential. The first and second cases from Lemma 4.1 are base cases. The first case corrects p_i belief altogether, and in the second case the incorrect belief never gets used in any significant way. In the first sub-case of case three, if a_k is in-fact p_i's best possible achievable arm, then it will continue getting matched with a_k regardless of it's wrong belief (inconsequential w.r.t a_k). Nonetheless, if $\exists p_j$ at $(t+1)$ such that triplet $Q^k_{ij} = (p_i, p_j, a_k)$ exists, then we can apply Lemma 4.1 recursively. And finally, in the second sub-case of case three from Lemma 4.1, p_i loses the conflict to some other player which implies that some belief gets updated resulting in the resolution of some triplet. However, if $\exists p_j$ at $(t+1)$ such that triplet $Q^k_{i,j} = (p_i, p_j, a_k)$ exists, then again we can apply Lemma 4.1 recursively. □

Theorem 4.1. *Under the belief update scheme described above, OCA-UCB shares CA-UCB's guarantee of convergence to stability.*

The proof of Theorem 4.1 is the immediate consequence of lemmas 4.1, and 4.2. Once all the inconsistent plausible sets are either resolved, or are inconsequential, the best arm picked from each player's plausible set will be the same as the one picked from CA-UCB's plausible set. This implies that this approach to updaing beliefs has no effect on CA-UCB algorithm's guarantee on convergence. □
We present empirical evidence on the performance of OCA-UCB in Section 6.

5 APU Model and PCA-SCA

The previous section looked at the APKP model where the agents did not have access to arms' preferences. Instead, the players relied on the fact that the arms had perfect knowledge about their own preferences and used deterministic feedback obtained to recreate something similar to the plausible set from the CA-UCB algorithm. However, in the APU model, even arms lack knowledge of their preferences, making feedback received from them noisy, invalidating the approach from the previous section.

Given this dilemma of unreliable feedback associated with conflict results, we must formulate an approach in which agents can pick the best possible arms while avoiding conflicts. To accomplish this, we introduce a notion of **highest expected reward**. The players keep track of estimated conflict win probabilities and attempt arms that maximize the product of reward estimates and win probability. We detail the algorithmic approach to using this heuristic in Algorithm 1 (PCA-SCA).

Algorithm 1: **Probabilistic Conflict Avoiding-Simultaneous Choice Algorithm (PCA-SCA)**

```
1:  procedure PCA-SCA(λ ∈ [0, 1))
2:      for t ∈ {1, ··· , T} do
3:          for i ∈ {1, ··· , n} do
4:              if t = 1 then
5:                  Set reward to ∞ for all arms
6:                  Set Z_win = 1 for all arms
7:                  Sample j ∈ [1, K] uniformly at random
8:                  Set A_i(t) ← a_j
9:              else
10:                 D_i(t) ~ Ber(λ)
11:                 if D_i(t) = 0 then
12:                     A_i(t) ← GET-BEST-ARM(p_i, t)
13:                 else
14:                     A_i(t) ← A_i(t − 1)
15:         for a_j ∈ {a_1, ··· , a_k} do
16:             p_win ← RESOLVE-CONFLICT(a_j, t)
17:             r_player, r_arm ~ SAMPLE-REWARD (p_win , a_j)
18:             UPDATE-ARM-REWARDS(a_j, r_player)
19:             UPDATE-PLAYER-REWARDS(p_win, r_arm)
20:             for p_i ∈ {p_1, ··· , p_n} do
21:                 UPDATE-PROBABILITY (p_i, a_j, p_win)

1:  procedure GET-BEST-ARM (Player p_i, Time t)
2:      for a_j ∈ {a_1, ··· , a_k} do
3:          p_prev ← Player that pulled a_j at (t − 1)
4:          REWARDS[j] ← p_i tracked R for a_j
5:          Z[j] ← probability of winning: p_i against p_prev for a_j
6:      j ← arg max (REWARDS ∘ Z)
7:      return a_j

1:  procedure RESOLVE-CONFLICT (Arm a_i, Time t)
2:      players ← Requesting players at time t
3:      values ← Arm tracked rewards of players
4:      player ← arg max_{p_i ∈ players} (values)
5:      return player
```

The algorithm is parameterized by $\lambda \in [0, 1)$, used to introduce a random delay mechanism to reduce the likelihood of conflicts (Liu et al. 2021; Kong, Yin, and Li 2022). This is controlled in line 10 where each player draws a Bernoulli Random Variable $D_i(t)$ with expectation λ. If $D_i(t) = 0$, the player attempts the arm with the highest expected reward, else it will attempt the arm that it did in the previous time step.

Initially, like CA-UCB, each player sets the reward estimate for the arms to ∞. Unlike CA-UCB however, the players also keep track of some probability estimates. Each probability estimate $\mathcal{Z}^{(i)}_j(a_k)$ represents player p_i's belief about how likely it is that it will win a conflict against player p_j for arm a_k. Each of these entries are initially set to 1. (line 5 and 6 in Algorithm 1). With these initial beliefs, at each subsequent time step, the players will attempt to pull the arm that maximizes the product of the reward estimate and the probability of winning (line 6 of procedure GET-BEST-ARM).

Another important distinction is how the arms pick the players. The arms no longer know their preferences and need to learn them. So, each arm will also keep track of beliefs about the payoffs associated with the players. Then, at the end of each time step, the arms will pick the player that it believes will give the highest reward (similar to how arms behave in the simultaneous choice mechanism of (Das and Kamenica 2005)). A player and arm both receive a reward and update their beliefs when an arm is pulled successfully. No reward is given if an arm is not pulled or if a pull request is not received. Players update their beliefs about winning after all matches are made.

One thing to note is that the arms might not have fully accurate estimates for *all* the players in the given horizon. However, the arms only need accurate estimates for the players who have the arm in their 'achievable set' i.e. the set of players that an arm can form a stable match with given their respective preference orderings. The algorithm structure is such that the arms are only picking from the set of available proposals. As long as the players get an accurate enough estimation of their preferences in the APU model, the arms will get the information necessary about the subset of players required to make the match stable.

Finally, the structure of the algorithm allows us to use different methods to keep track of beliefs about expected rewards and the probability of conflict wins. In this paper we use two: first, UCB, as used by (Liu et al. 2021), and second, a Thompson Sampling variant, as used by (Kong, Yin, and Li 2022).

Using UCB

We use the Upper Confidence Bound (UCB) approach to keep track of reward estimates, as used in multi-armed bandit literature by (Auer, Cesa-Bianchi, and Fischer 2002). We use UCB in the PCA-SCA algorithm and refer to it as PCA-UCB. Players must keep track of beliefs about payoffs and the probability of winning conflicts. The UCB heuristic can be used to estimate both, while the arms only needing to estimate rewards. Equation 1 describes how beliefs about reward payoffs are estimated.

$$\text{UCB}_i^k(t) = \begin{cases} \infty & \text{if } N_i^k(t) = 0 \\ \hat{\mu}_i^k + \sqrt{\frac{3\log t}{2N_i^k(t-1)}} & \text{otherwise} \end{cases} \quad (1)$$

where $N_i^k(t)$ is the number of times player p_i has pulled arm a_k at time-step t, and $\hat{\mu}_i^k$ is the empirical mean tracked by player p_i for arm a_k. Next, to estimate the probabilities of conflict wins, we use a similar quantity that shares the optimism property of UCB.

$$\mathcal{Z}_i^j(a_k) = \begin{cases} 1 & \text{if } n_i^j = 0 \\ \frac{w_i^j(a_k)}{n_i^j(a_k)} + \sqrt{\frac{3\log(t)}{2\cdot n_i^j(a_k)}} & \text{otherwise} \end{cases} \quad (2)$$

where $\mathcal{Z}_i^j(a_k)$ represents player p_i's belief about the probability of winning a conflict against player p_j for arm a_k. $w_i^j(a_k)$ is the number of times player p_i has won this conflict and $n_i^j(a_k)$ is the total number of times this conflict has happened thus far. We upper-censor $\mathcal{Z}_i^j(a_k)$ at 1. At the end of each time step, belief estimates are updated once rewards are sampled and matching information made public. If a player wins a conflict, the corresponding probability estimate increases, otherwise, it decreases. This is handled by the UPDATE-PROBABILITY function in Algorithm 1.

Using Thompson Sampling

Thompson Sampling (Thompson 1933) (TS) is another approach to solving the MAB problem which has seen a recent resurgence in the literature (Agrawal and Goyal 2012; Chapelle and Li 2011; Kong, Yin, and Li 2022). In the PCA-SCA algorithm, TS can be used to keep track of both the beliefs about rewards as well as probability estimates for winning a conflict. The arms also use TS to keep track of reward estimates. Henceforth, we will refer to this approach as the PCA-TS algorithm. The players and arms keep track of the same information as in UCB, i.e. the total reward obtained for an arm/player, the total number of pulls of an arm, and the total number of conflict wins against each player. The distinction is in how this information is used to update beliefs.

First, we estimate the rewards. We assume the variance ($\sigma^2 = 1$) in rewards of the arms and the players are known. Then, we update the mean and precision ($\tau = \frac{1}{\sigma^2}$) as:

$$\mu_{\text{new}}, \tau_{\text{new}} = \frac{\tau_0\mu_0 + \tau\sum_{i=1}^{n} x_i}{\tau_0 + n\tau}, \tau_0 + n\tau \quad (3)$$

where $\{\mu_{\text{new}}, \tau_{\text{new}}\}$ are the new mean and precision, $\{\mu_0, \tau_0\}$ are the old mean and precision, $\tau = 1$ is the known true precision, and $\sum x_i$ is the sum of rewards for the agent in question. This reward estimate is analogous for both the players and the arms. When an agent needs an estimate for a reward for a particular player or an arm, it samples from $\mathcal{N}(\mu_0, \frac{1}{\tau_0})$ where μ_0 and τ_0 is the corresponding mean and precision tracked by the agent for the specific player/arm. A Bernoulli distribution is used to keep track of the probability of a player p_i winning a conflict against player p_j for arm a_k.

$$\mathcal{Z}_i^j(a_k) = \frac{\omega_i^j(a_k)}{\omega_i^j(a_k) + \nu_i^j(a_k)} \quad (4)$$

where $\omega_i^j(a_k)$ is the number of times p_i has won the conflict and $\nu_i^j(a_k)$ is the number of times p_i has lost the conflict against player p_j for arm a_k.

6 Simulation Results

We run simulation experiments where (1) preferences are uniformly random on both sides of the market with varying market sizes $N = K \in \{5, 10, 15, 20\}$; (2) player preference heterogeneity is varied with market size $N = K = 10$. The maximum reward an agent can get is $(K + 1)$, when matched with its most preferred arm, and the minimum is 1, when matched with the least preferred. The reward gaps (of the means) between consecutively ranked agents are kept

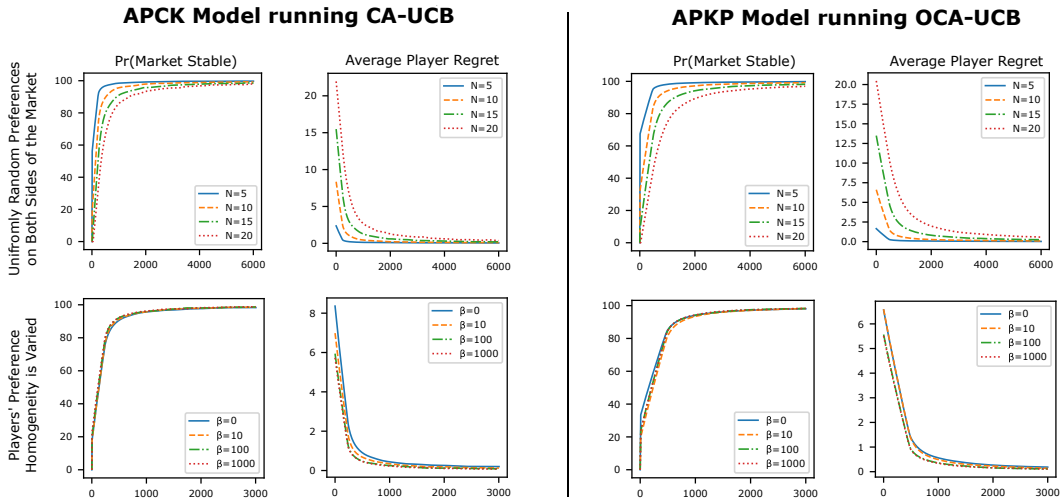

Figure 1: The left sub-figures show the APCK models using CA-UCB and the right sub-figures show the APKP model using OCA-UCB. Results are averaged over 1000 runs. The first row varies market size with uniformly random preferences, the second row varies player preference heterogeneity while keeping market size constant at $N = K = 10$. OCA-UCB converges to stability even with uncertainty on both sides, though it takes longer (expected because of increased complexity)

constant at $\Delta = 1$. To control the level of heterogeneity in player preferences, we use a method due to (Ashlagi, Kanoria, and Leshno 2017), as also used by (Liu et al. 2021). The level of heterogeneity is controlled by a parameter $\beta \in [0, 10, 100, 1000]$. Heterogeneity in preferences decreases with β, and as $\beta \to \infty$ all players have the same preferences with probability 1. To translate β to preference orderings we sample the mean reward μ_i^k of arm a_k for player p_i using the following process:

$$\begin{aligned} x_k &\overset{i.i.d}{\sim} \text{Uniform([0,1])} \\ \epsilon_{k,i} &\overset{i.i.d}{\sim} \text{Logistic(0, 1)} \\ \overline{\mu}_i^k &= \beta \cdot x_k + \epsilon_{k,i} \\ \mu_i^k &= \#\{j : \overline{\mu}_i^j \leq \overline{\mu}_i^k\} \end{aligned} \tag{7}$$

These random utilities μ_i^k are mapped so that the gap in rewards between consecutively ranked arms is kept constant at $\Delta = 1$. The hyper-parameter λ is set to 0.9.[1]

To objectively quantify the quality and rate of convergence, we introduce a notion of a "convergence proxy": It measures, in sliding windows of size $\mathcal{X}$, the percentage of time-steps where market stability was greater than θ. It effectively measures the tendency of a market to converge within the next $\mathcal{X}$ time-steps. We calculate the "convergence proxy" line for PCA-UCB vs PCA-TS for random preferences and varied market sizes. This allows us to objectively visualize the convergence behavior of the two approaches and draw conclusions about their behaviors.

We run simulations in each of the aforementioned scenarios 1000 times and monitor the market state. More specifically, we take a snap-shot of the maximum player regret and market stability every 10 time-steps for each experiment and average the results over all the runs. To evaluate the trend in the quantities we study, Loess smoothing was run on the data points to yield the graphs presented in the following sections.

OCA-UCB in the APKP model

Recall that, in this scenario the players must learn their own position in the arms' preference order and their own preferences. We compare the performance of this model with the APCK model using CA-UCB. Both models were run with their respective algorithms, with 6000 time-steps for random preferences and 3000 time-steps for varied preference homogeneity. The results are shown in Figure 1.

The figures in the first row of Figure 1 show results from the experiments where agents' preferences are uniformly random. We can see that OCA-UCB converges to stability in the APKP model, with increasing market sizes corresponding to slower convergence. This trend in similar to the APCK model running CA-UCB, albeit OCA-UCB convergence is the APKP model is slightly slower.

The second row in Figure 1 details experimental results for varying player preference heterogeneity. As can be seen from the APCK Model running CA-UCB, decrease in player preference heterogeneity in the APCK model does not have any effect on the convergence of CA-UCB. This trend carries over to OCA-UCB in the APKP model, with convergence slightly slower. This shows that in the setting with

[1]There are alternative methods for controlling heterogeneity in player preferences, for example notions like uncoordinated/coordinated markets (Ackermann et al. 2008). The method we use corresponds to uncoordinated markets when β is low. However, coordinated markets are different, with correlation defined on the edges of the matching graph rather than the vertices. We find that the convergence time in these edge-correlated cases is actually comparable to when preferences are uniformly random, compared with node-correlation, where convergence time increases (see Supplement).

varying levels of player preference homogeneity, OCA-UCB does not show any noticeable difference in the rate at which the markets converge to stability.

PCA-UCB and PCA-TS in the APU Model

Recall that, in this scenario, both the players and the arms do not have access to any preference information. Our proposed approach is to make use of a simultaneous choice algorithm, where at each time-step the players attempt the arm with the *highest expected reward*. As the arms become more confident about their reward estimates for players, the probability estimates that the players keep track of will be more representative of true probabilities. Given the structure of Algorithm 1 (PCA-SCA), we propose two methods in Section 5 to keep track of agents' beliefs: UCB (PCA-UCB), and Thompson Sampling (PCA-TS). In this section, we detail the results of our experiments when using these approaches.

Using PCA-UCB The results of the experiments using PCA-UCB in the APU model are summarized in Figure 2. We run PCA-UCB for 20,000 time steps for markets with random preferences on both sides, and for 10,000 time steps for when player preference heterogeneity is varied. The former results are in the first row of the figure, whereas the results of the latter are detailed in the second row of Figure 2.

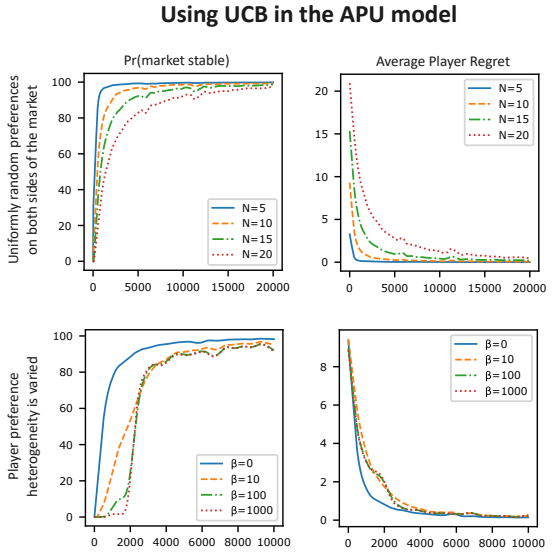

Figure 2: Results of experiments run in the APU model using UCB to keep track of beliefs: first row has preferences uniformly random, second row has varied player preference heterogeneity with $N = K = 10$. Key Takeaways: the markets converge to stability in expectation. Using UCB, there is a dependence on player preference heterogeneity and the time taken to convergence. If you compare this figure against Figure 4, Thompson Sampling converges to stability slightly faster than UCB and smoother, showing no dependence on player preference heterogeneity.

Firstly, notice that the APU model using PCA-UCB takes longer to stabilize compared to the APKP model using OCA-UCB, as seen in Figures 1 and 2. This is due to the fact that both players and arms need to learn their beliefs from scratch. Regardless, when market preferences are random, the trends in both figures are similar, with market stabilization probabilities approaching 1 and player regret tending to 0. This shows that PCA-UCB can find stable matches in a decentralized fashion with unknown preferences on both sides of the market when preferences are random.

The second row of sub-figures in Figure 2 shows what happens when the players' preference heterogeneity is varied.There is a trend associated with the time taken to stabilize with changes in the parameter β, with higher values of beta (more homogeneity) leading to slower convergence. We hypothesize that, given the structure of the algorithm, this is largely due to the fact that, with more preference homogeneity, it takes longer for players to learn to avoid conflicts, because the arms' own uncertainty allows optimism to prevail for longer among more players. This is further corroborated by Figure 3 which shows the average number of conflicts in the experiments where β is varied as a function of time. We can see that increases in the values of β correspond to more conflicts in the early stages of the algorithm. Ultimately though, markets still converge to stability for all the values of β across all 1000 runs, with the conflict counts also tending to 0. Hence, this demonstrates that PCA-UCB can find stable matches when the players have varying levels of preference heterogeneity as well.

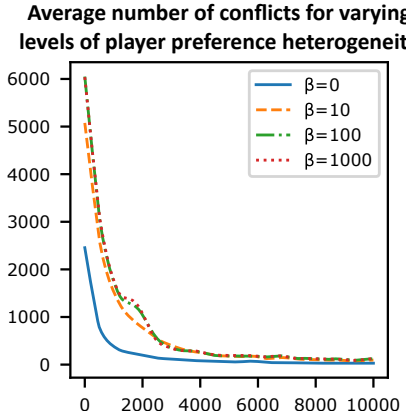

Figure 3: The number of conflicts as a function of time for different values of β when using PCA-UCB in the APU model. As time progresses, the number of conflicts decreases and after a certain point they converge to 0 (converge to stability). Higher values of β result in an overall higher number of conflicts when using UCB. This explains the trend seen in Figure 2 where we can see that the probability of market stability depends on the value of β when player preference heterogeneity is varied.

Using PCA-TS Results for when PCA-TS was used in the APU model are shown in Figure 4. Like Figure 2 (the UCB results), the first row in Figure 4 is when the agent preferences are sampled uniformly at random, and the second row is where the players' preference heterogeneity is varied.

Results from PCA-UCB (Figure 2) and PCA-TS (Figure 4) show similarities in market stability and player regret for all market sizes with uniformly random preferences, with larger markets stabilizing slower. However, PCA-TS

does not show dependency on β when player preference heterogeneity is varied, likely due to quicker convergence of Thompson beliefs. We can also see that PCA-TS converges faster and more smoothly than PCA-UCB. We will discuss this particular property in further detail in the next subsection. The results demonstrate that Algorithm 1 can find a stable match in the APU model using both UCB and Thompson Sampling.

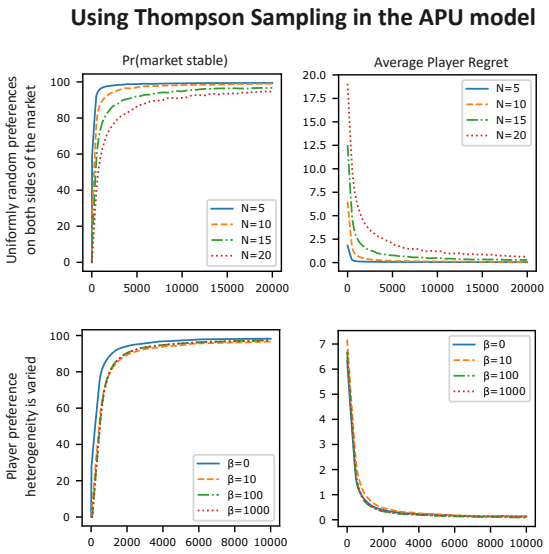

Figure 4: Results of experiments run in the APU model using Thompson Sampling to keep track of beliefs. The first row shows market stability and average player regret when preferences are uniformly random on both sides of the market whereas the second row is for when player preference heterogeneity is varied. All the markets stabilize and player regret tends to 0. Performance was better than UCB in terms of convergence rate and smoothness.

Convergence Comparison: UCB vs. Thompson Sampling When looking at Figures 2 and 4, we can see that the Thompson Sampling approach of keeping track of beliefs appears better than UCB in terms of both speed and smoothness of convergence. To quantify this behavior, we turn to the *convergence proxy* we defined earlier, this time comparing the stability graphs of PCA-UCB and PCA-TS. We set $\mathcal{X} = 1000, \theta = 90$ and generate these lines, the results of which are summarized in Figure 5. Each color on the graph corresponds to the market size with the solid line representing PCA-UCB and the dotted line representing PCA-TS. We can see that the lines associated with Thompson Sampling reach 1.0 (i.e. 100% of the time-steps in question have stability $> 90\%$) quicker and tend to stay that way.

In comparison, the convergence proxy for UCB reaches 1.0 slower for each of the market sizes, and often the proxy value dips before going back up again. By nature of the way beliefs are kept track of, whenever a player chooses to explore, it causes a disturbance in the believed preference ordering of the player. When the internal belief state for a player changes, the player's set of arms in consideration changes, leading to a different matching. This behavior

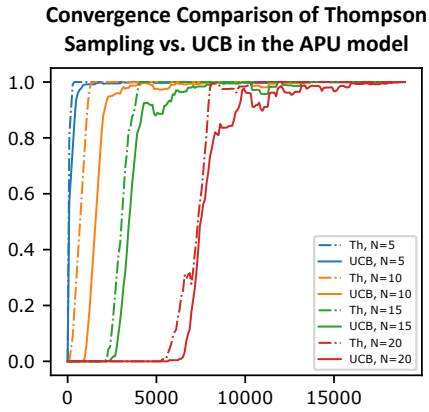

Figure 5: Convergence Comparison of UCB and Thompson Sampling in the APU model with varying market sizes and uniformly random preferences on both sides of the market. The Thompson Sampling approach (dotted) converges slightly faster and stays that way compared to the UCB (solid) which tends to dip time and again before ultimately reaching convergence.

causes the match to be unstable at some time-steps. This is more frequent in the early stages of the algorithm and becomes less frequent as confidence increases about player preferences. PCA-TS does not exhibit this dipping effect and reaches 1.0 in an overall much smoother manner. The combination of these two reasons suggests that Thompson Sampling might be a better approach for keeping track of agent beliefs in the APU model.

7 Conclusion and Future Work

In this paper, we investigated two-sided matching with uncertain preferences in two challenging settings: the APKP model (where only arms know their preferences) and the APU model (where neither side initially knows preferences). Building on prior work that assumes arms' preferences are common knowledge, we extended the analysis and designed algorithms that learn preferences on both sides without direct communication.

In the APKP model, we showed that players can reliably learn their positions in each arm's preference ordering, enabling convergence to a stable match only slightly slower than CA-UCB. In the more complex APU model, we proposed an algorithm where each player selects arms based on highest expected reward, taking into account both conflict probabilities and reward estimates. Empirically, we found that both UCB and Thompson Sampling converge to a stable match, with Thompson Sampling doing so more quickly and smoothly—paralleling its advantages in simpler multi-armed bandit settings.

While we proved convergence in the APKP model, the APU setting is more challenging. Our results suggest that stable matching can still be achieved when all preferences are unknown, but formal theoretical guarantees for APU remain an important direction for future work.

Acknowledgements

We are grateful for funding from NSF awards 2127752 and 2402856.

References

Abdulkadiroğlu, A.; Pathak, P. A.; Roth, A. E.; and Sönmez, T. 2005. The Boston public school match. *American Economic Review*, 95(2): 368–371.

Ackermann, H.; Goldberg, P. W.; Mirrokni, V. S.; Röglin, H.; and Vöcking, B. 2008. Uncoordinated Two-Sided Matching Markets. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, EC '08, 256–263. New York, NY, USA: Association for Computing Machinery. ISBN 9781605581699.

Agrawal, S.; and Goyal, N. 2012. Analysis of Thompson Sampling for the multi-armed bandit problem. In *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings.

Ashlagi, I.; Kanoria, Y.; and Leshno, J. D. 2017. Unbalanced Random Matching Markets: The Stark Effect of Competition. *Journal of Political Economy*, 125(1): 69–98.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2): 235–256.

Aziz, H.; Biró, P.; and Yokoo, M. 2022. Matching market design with constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 12308–12316.

Aziz, H.; Biró, P.; Gaspers, S.; de Haan, R.; Mattei, N.; and Rastegari, B. 2020. Stable Matching with Uncertain Linear Preferences. *Algorithmica*, 82(5): 1410–1433.

Basu, S.; Sankararaman, K. A.; and Sankararaman, A. 2021. Beyond $log^2(T)$ regret for decentralized bandits in matching markets. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 705–715. PMLR.

Benabbou, N.; Chakraborty, M.; Ho, X.-V.; Sliwinski, J.; and Zick, Y. 2018. Diversity Constraints in Public Housing Allocation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 973–981.

Chapelle, O.; and Li, L. 2011. An empirical evaluation of Thompson Sampling. *Advances in Neural Information Processing Systems*, 24.

Das, S.; and Kamenica, E. 2005. Two-Sided Bandits and the Dating Market. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, 947–952. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Etesami, S. R.; and Srikant, R. 2024. Decentralized and Uncoordinated Learning of Stable Matchings: A Game-Theoretic Approach. (arXiv:2407.21294). ArXiv:2407.21294 [cs, eess].

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

Haeringer, G.; and Wooders, M. 2011. Decentralized job matching. *International Journal of Game Theory*, 40(1): 1–28.

Johari, R.; Li, H.; Liskovich, I.; and Weintraub, G. Y. 2022. Experimental design in two-sided platforms: An analysis of bias. *Management Science*, 68(10).

Kong, F.; Yin, J.; and Li, S. 2022. Thompson Sampling for Bandit Learning in Matching Markets. *ArXiv Preprint*. ArXiv:2204.12048 [cs].

Lee, R.; and Schwarz, M. 2009. *Interviewing in Two-Sided Matching Markets*. Cambridge, MA.

Liu, L. T.; Mania, H.; and Jordan, M. I. 2020. Competing Bandits in Matching Markets. *arXiv:1906.05363 [cs, stat]*. ArXiv: 1906.05363.

Liu, L. T.; Ruan, F.; Mania, H.; and Jordan, M. I. 2021. Bandit Learning in Decentralized Matching Markets. *J. Mach. Learn. Res.*, 22(1).

Pagare, T.; and Ghosh, A. 2023. Two-Sided Bandit Learning in Fully-Decentralized Matching Markets. In *ICML 2023 Workshop The Many Facets of Preference-Based Learning*.

Roth, A. E.; and Peranson, E. 1999. The redesign of the matching market for American physicians: Some engineering aspects of economic design. *American economic review*, 89(4): 748–780.

Roth, A. E.; and Xing, X. 1997. Turnaround Time and Bottlenecks in Market Clearing: Decentralized Matching in the Market for Clinical Psychologists. *Journal of Political Economy*, 105(2): 284–329.

Sankararaman, A.; Basu, S.; and Abinav Sankararaman, K. 2021. Dominate or Delete: Decentralized Competing Bandits in Serial Dictatorship. In Banerjee, A.; and Fukumizu, K., eds., *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, 1252–1260. PMLR.

Shah, V.; Ferguson, B. L.; and Marden, J. R. 2024. Learning Optimal Stable Matches in Decentralized Markets with Unknown Preferences. (arXiv:2409.04669). ArXiv:2409.04669 [cs, eess].

Thompson, W. R. 1933. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3/4): 285–294.

Zhang, Y.; and Fang, Z. 2024. Decentralized Two-Sided Bandit Learning in Matching Market. In *The 40th Conference on Uncertainty in Artificial Intelligence*.

Zhang, Y.; Wang, S.; and Fang, Z. 2022. Matching in Multi-arm Bandit with Collision. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.