Bandit Learning in Decentralized Matching Markets

Lydia T. Liu　　　　　　　　　　　　　　　　　LYDIATLIU@CS.BERKELEY.EDU

Feng Ruan　　　　　　　　　　　　　　　　　　FENGRUAN@CS.BERKELEY.EDU

Horia Mania　　　　　　　　　　　　　　　　　　HMANIA@CS.BERKELEY.EDU

Michael I. Jordan　　　　　　　　　　　　　　　JORDAN@CS.BERKELEY.EDU
Department of Electrical Engineering and Computer Sciences and Department of Statistics
University of California
Berkeley, CA 94720-1776, USA

Abstract

We study two-sided matching markets in which one side of the market (the players) does not have a priori knowledge about its preferences for the other side (the arms) and is required to learn its preferences from experience. Also, we assume the players have no direct means of communication. This model extends the standard stochastic multi-armed bandit framework to a decentralized multiple player setting with competition. We introduce a new algorithm for this setting that, over a time horizon T, attains $\mathcal{O}(\log(T))$ stable regret when preferences of the arms over players are shared, and $\mathcal{O}(\log(T)^2)$ regret when there are no assumptions on the preferences on either side. Moreover, in the setting where a single player may deviate, we show that the algorithm is incentive compatible whenever the arms' preferences are shared, but not necessarily so when preferences are fully general.

Keywords: Online learning, Multi-armed bandits, Stable matching, Two-sided markets

1. Introduction

A fundamental question at the intersection of learning theory and game theory is as follows: *how should individually rational agents act when they have to learn about the consequences of their actions in the same uncertain environment?* While there has been a long line of work on learning in games (Fudenberg and Levine, 1998; Hu et al., 1998; Littman, 1994), recent developments in statistical learning theory and online learning have opened the door to a new line of work that aims to quantify precisely the amount of data players require to achieve good performance in games with stochasticity. The problems studied are motivated by a broad range of modern applications, from modeling competition among firms (Mansour et al., 2018; Aridor et al., 2019) to implementing protocols for wireless networks (Liu and Zhao, 2010; Cesa-Bianchi et al., 2016; Shahrampour et al., 2017). A particularly salient application is the online marketplace[1], where two sides of a market need to be matched

---

1. Examples include online labor markets (Upwork, TaskRabbit, Handy), online crowdsourcing platforms (Amazon Mechanical Turk), online dating services (Match.com) and peer-to-peer sharing platforms (Airbnb).

and market participants have uncertainty about their preferences, leading to a concomitant need for exploration and statistical learning.

The multi-armed bandit is a core learning problem that models decision-making under uncertainty: a player is faced with a choice among K actions—"arms"—each of which is associated with a reward distribution, and the goal is to learn which action has the highest reward, doing so as quickly as possible so as to be able to reap rewards even while the learning process is underway. Even in the more complex setting involving multiple players participating in a two-sided matching market, the bandit problem can be extended to model how players simultaneously learn and acquire information about their preferences, while satisfying economic constraints imposed by the need to realize a matching. Such a blend of bandit learning with two-sided matching markets was introduced by Das and Kamenica (2005), who formulated a problem in which the players and the arms form the two sides of the market, and each side has preferences over the other side. Das and Kamenica (2005) explored possible algorithms via numerical simulations. Liu et al. (2020) studied a refinement of this problem setting and proposed the first algorithm with theoretical guarantees. In contradistinction to the classical formulation of matching markets, the preferences of the players are assumed to be unknown a priori and must be learned from the rewards that are received when arms are pulled successfully. Compared to prior work studying multi-player bandits, the problem formulation we consider introduces an aspect of scarcity and competition—when multiple players attempt to pull the same arm, there is a conflict, and only the player that is most preferred by that arm receives a reward.

Liu et al. (2020) focused on a *centralized* setting in which the players are able to communicate with a central platform that computes matchings for the entire market. They defined a notion of regret called *stable regret*, which is the average reward a player obtains less the rewards achieved under a stable matching with respect to the true preferences of the market. It was shown in this setting that an algorithm that combines the upper confidence bound principle from the bandit literature (Lai and Robbins, 1985) with the Gale-Shapley algorithm from the matching market literature (Gale and Shapley, 1962) can achieve low stable regret.

While Liu et al. (2020) discussed a decentralized version of the problem, where the actions of the players cannot be coordinated by a central platform, and studied a simple explore-then-commit algorithm for this setting, finding a viable algorithm for the decentralized case was left as an open problem. The decentralized setting is arguably a more useful formulation in practice. Indeed, most online marketplaces are decentralized, that is, there is no central clearinghouse and players are unable to coordinate their actions with each other directly. However, players may observe limited information about past matchings, such as their own conflicts.

New theoretical challenges arise in the decentralized setting, in both the design and the analysis of algorithms. Given that players may use past matchings to inform their current play (e.g., to avoid conflicts), a player who has statistical uncertainty about their preferences over arms may impose externalities on other players not only at the current time step but also into the future. In essence, the decentralized formulation more fully exposes the challenges of the economic and learning aspects of the problem.

We propose a solution for the decentralized version of the two-sided matching bandit problem. Our primary contribution is a new multiplayer bandit algorithm, *Decentralized*

Conflict-Avoiding Upper Confidence Bound (CA-UCB), that is guaranteed to yield for all players a stable regret that grows polylogarithmically with the number of rounds of interaction between players and arms, also known as the time horizon, T. In particular, to prove this regret guarantee we roughly showed that the market converges to a stable matching at a polylogarithmic rate. When the arms have the same preferences over players we offer a better guarantee. In this case we prove that the stable regret grows at most logarithmically with the time horizon. Informally, we can state our results as follows.

Theorem 1 (Informal main results) *Suppose we have a market with N players and L arms, with arbitrary preferences, and let Δ be the minimum absolute gap between the mean rewards of different arms. Then, if all players run the CA-UCB algorithm for T steps, the probability that the market is unstable at time T is $\mathcal{O}(\log(T)^2/T)$ (see Theorem 6). Moreover, the players' stable regret satisfies*

$$R(T) = \mathcal{O}\left(\rho^{N^4}\frac{\log(T)^2}{\Delta^2}\right), \text{ for some } \rho > 1. \tag{Corollary 7}$$

When the arms have the same preferences over players, the players' stable regret satisfies

$$R(T) = \mathcal{O}\left(N^2 L\frac{\log(T)}{\Delta^2}\right). \tag{Theorem 2}$$

Moreover, if $N-1$ players implement the CA-UCB algorithm, the remaining player cannot significantly improve their regret by running a different algorithm (Proposition 16).

The CA-UCB algorithm is simple and does not require communication between players. There are two features of this algorithm that enable players to avoid conflicts. Firstly, when implementing this algorithm a player observes the actions of other players in the previous round and avoids attempting an arm if that arm was previously pulled by a better player for it. Secondly, players randomly decide whether to choose the same arm as at the previous time step or to make a new decision. When players implement our method conflicts can still occur, but our analysis shows that the expected number of conflicts would be small.

The rest of the paper is organized as follows: In Section 2, we review the matching bandits problem, following the presentation in Liu et al. (2020), and fully specify the decentralized setting that is our focus. In Section 3, we motivate and introduce the algorithm that is the subject of our regret analyses in Sections 4 and 5. In Section 6, we discuss the incentive compatibility of this algorithm, showing one positive and one negative result. Our theoretical guarantee on the performance of CA-UCB exhibits an exponential dependence on the size of the market. In Section 7 we show empirically that this dependence is an artifact of our analysis; CA-UCB performs much better in practice than these results suggest. In Section 8, we survey the related literature, and in Section 9, we present a thorough discussion of our results, as well as avenues for future work.

2. Problem Setting

We consider a multiplayer multi-armed bandit problem with N players and L stochastic arms, with $N \leq L$. We denote the set of players by $\mathcal{N} = \{p_1, p_2, \ldots, p_N\}$ and the set of arms by $\mathcal{K} = \{a_1, a_2, \ldots, a_L\}$. At time step t, each player p_i attempts to pull an arm $m_t(i) \in \mathcal{K}$.

When multiple players attempt to pull the same arm, only one player will successfully pull the arm, according to the arm's preferences via a mechanism we detail shortly. Then, if player p_i successfully pulls arm $m_t(i)$ at time t, they are said to be *matched* to $m_t(i)$ at time t and they receive a stochastic reward, $X_{m_t}^{(i)}(t)$, sampled from a 1-sub-Gaussian distribution with mean $\mu_{m_t(i)}^{(i)} > 0$.

For each player p_i we assume $\mu_j^{(i)} \neq \mu_{j'}^{(i)}$ for all distinct arms, a_j and $a_{j'}$. If $\mu_j^{(i)} > \mu_{j'}^{(i)}$, we say that player p_i *truly prefers* a_j to $a_{j'}$, and denote this as $a_j \succ_{p_i} a_{j'}$.

Each arm a_j has a fixed, known, and strict preference ordering over all the players, $\succ_{a_j}$. In other words, $p_i \succ_{a_j} p_{i'}$ indicates that arm a_j prefers player p_i to player $p_{i'}$. If two or more players attempt to pull the same arm a_j, there is a *conflict* and only the most preferred player successfully pulls the arm to receive a reward; the other player(s) $p_{i'}$ is said to be *unmatched* and does not receive any reward, that is, $X_{m_t}^{(i')}(t) = 0$.

A *stable matching* (Gale and Shapley, 1962) of players and arms is one where no pair of player and arm would prefer to be matched with each other over their respective matches. Given the full preferences of the arms and players, arm a_j is called a *achievable match* of player p_i if there exists a stable matching according to those preferences such that a_j and p_i are matched. We say a_j is the *optimal match* of player p_i if it is the most preferred achievable match. Similarly, we say a_j is the *pessimal match* of player p_i if it is the least preferred achievable match. We denote by $\overline{m}$ and $\underline{m}$ the functions from $\mathcal{N}$ to $\mathcal{K}$ that define the optimal and pessimal matches of a player according to the true preferences of the players and arms.

In the decentralized matching setting, a notion of *stable regret*, as introduced in Liu et al. (2020), is useful for analyzing the performance of learning algorithms. We consider a player's *player-pessimal stable regret*, where the baseline for comparison is the mean reward of the arm that is the player's pessimal match.[2] It is defined as follows for player p_i:

$$\underline{R}_i(T) := T\mu_{\underline{m}(i)}^{(i)} - \sum_{t=1}^{T} \mathbb{E}X_{m_t}^{(i)}(t). \tag{1}$$

The above notion of stable regret considers regret from the perspective of the players only, that is, we are primarily interested in how the players perform with respect to their stable arms over time. Focusing on the welfare of one side of the market is consistent with the stable matching literature, in particular that on school choice, where one side of the market (the schools) are said to have "priorities", rather than "preferences", for the other side of the market (the students), and it is the students' welfare that is of primary interest (Abdulkadiroğlu and Sönmez, 2003; Abdulkadiroğlu et al., 2006).[3] Recently, (Cen and Shah, 2021) studied fairness and social welfare in the context of matching markets.

In order to fully specify the problem we need to clarify what information the players have access to. We consider the following decentralized setting:

---

2. We can define analogously the *player-optimal stable regret* corresponding to the player's optimal match, denoted $\overline{R}_i(T)$. The player-pessimal stable regret and player-optimal stable regret tend to coincide in many real-world markets, such as in unbalanced random matching markets (Ashlagi et al., 2017b) where the stable matching is essential unique. This is as well the case when players are globally ranked. In this work, we focus on the player-pessimal stable regret.

3. We thank a reviewer for pointing out this connection to the economics literature.

Decentralized with Conflict Information At each round, each player attempts to pull an arm, with the choice of arm based on only their rewards and observations from previous rounds. At the end of the round, all players can observe the winning player for each arm. They can see their own rewards only if they successfully pull an arm. They cannot see the rewards of other players. We also assume that all players know, for each arm, which players are ranked higher than themselves.[4]

3. Algorithm: Decentralized Conflict-Avoiding UCB

In the single-player multi-armed bandit (MAB) the player must explore different arms in order to identify the arms with the highest mean payoff. At the same time, the player must keep selecting arms that seem to give high payoff in order to accumulate a large reward over time. The upper confidence bounds (UCB) algorithm offers an elegant solution to this exploration-exploitation dilemma. As the name suggests, UCB maintains upper confidence bounds on the arms' mean payoffs and selects the arm with the largest upper confidence bound. Then, the UCB algorithm updates the upper confidence bound corresponding to the selected arm according to the reward observed.

In the aforementioned decentralized model, however, a player cannot implement UCB obliviously of other players' actions given the possibility of conflicts. Let us discuss this issue from the perspective of player p_1. Suppose p_1 chooses arm a_1, and suppose player p_2 chooses a_1 at the same time. Then, if a_1 prefers p_2 over p_1, a conflict arises and player p_1 receives no reward. In addition to not receiving a reward, in this case, player p_1 does not learn anything new about the distribution of rewards offered by arm a_1. Therefore, in the decentralized case players must balance exploration and exploitation while avoiding conflicts that they would lose.

To see intuitively how p_1 can achieve such conflict avoidance let us assume that there are only two players and that all arms prefer p_2. Then, from the perspective of p_2, the problem is identical with the single-player MAB problem and therefore p_2 can achieve small regret by using the standard UCB method. Since p_2 aims to minimize their own regret, p_2 will sample the arm that gives them the highest mean payoff most of the time. More precisely, there can be at most $\mathcal{O}(\log(T))$ time steps when p_2 does not sample the best arm for themselves.

On the other hand, p_1 must minimize the number of times they select the same arm as p_2 because they would lose the conflicts with p_2. Because most of the time player p_2 chooses the best arm for themselves, the following simple heuristic allows player p_1 to avoid choosing the same arm as p_2 most of the time: player p_1 should not select the arm p_2 chose at the previous time step.

It turns out that this conflict-avoidance heuristic, combined with the UCB method, gives rise to an algorithm that provably achieves small regret for all players. We call this method *Decentralized Conflict-Avoiding Upper Confidence Bound*, or *CA-UCB* for short, and detail it in Algorithm 1. Before introducing our algorithm, let us first introduce some notation

---

4. This assumption allows for a cleaner analysis of our algorithm. Our results can be generalized to the setting where players do not know this information initially because the arms know their own preferences and the conflicts between players are resolved deterministically. It is sufficient for each player to assume in the beginning that they are the most preferred player by every arm. Then, each lost conflict reveals which players are more preferred by which arms. This procedure would introduce at most LN^2 conflicts.

Algorithm 1 CA-UCB with random delays

Input: $\lambda \in [0, 1)$
1: **for** $t = 1, \ldots, T$ **do**
2:     **for** $i = 1, \ldots, N$ **do**
3:         **if** $t = 1$ **then**
4:             Set upper confidence bound to ∞ for all arms.
5:             Sample an index $j \sim 1, \ldots, L$ uniformly at random. Sets $A_t^{(i)} \leftarrow a_j$.
6:         **else**
7:             Draw $D^{(i)}(t) \sim Ber(\lambda)$ independently.
8:             **if** $D^{(i)}(t) = 0$ **then**
9:                 Update plausible set $S^{(i)}(t)$ for player p_i:

$$S^{(i)}(t) := \{a_j : p_i \succ_{a_j} p_k \text{ or } p_i = p_k, \text{ where } \bar{A}^{(k)}(t-1) = a_j\}.$$

10:                 Pulls $a \in S^{(i)}(t)$ with maximum upper confidence bound. Sets $A_t^{(i)} \leftarrow a$.
11:             **else**
12:                 Pulls $A_{t-1}^{(i)}$. Sets $A_t^{(i)} \leftarrow A_{t-1}^{(i)}$.
13:         **if** p_i wins conflict **then**
14:             Update upper confidence bound for arm $A_t^{(i)}$.

---

for the players' actions. We use $A^{(i)}(t)$ to denote the player p_i's attempted arm at time t, and $\bar{A}^{(i)}(t)$ to denote the player i's successfully pulled arm at time t. When the player fails to pull an arm successfully because of a lost conflict, we have $\bar{A}^{(i)}(t) = \emptyset$.

According to Algorithm 1, at each time step t each player p_i independently samples a biased Bernoulli random variable $D^{(i)}(t)$ with mean $\lambda \in [0, 1)$. When $D^{(i)}(t)$ comes up 1, the player chooses the same arm as they did at the previous time step. We will soon return to explain the rationale behind staying on the same arm as the previous time step with some probability. For now, let us focus on the case where $D^{(i)}(t)$ comes up 0.

When the Bernoulli random variable $D^{(i)}(t)$ comes up 0, the player constructs a *plausible set* of arms that includes all arms except those that the player would not have been able to pull successfully at the previous time step. In other words, the player p_i will consider an arm plausible, only if in the previous time step $t - 1$, the arm was not pulled by a player that the arm strictly prefers to p_i. Then, the player chooses the arm in the plausible set with the highest upper confidence bound, which is updated as in the single-player UCB method. We formally define the upper confidence bound in Equation 12 of Section 4.

We refer to the parameter λ as the *delay probability*. When $\lambda = 0$ the actions of the players that implement CA-UCB are deterministic functions of the history up to that point. This property has no impact on the algorithm's convergence when the players are globally ranked (i.e., all arms have the same preferences), as shown in Section 4. However, for more general preference structures, if all players implement CA-UCB with delay probability zero, they can enter into infinite loops. The following simple example showcases this failure mode.

Example 1 (2-player globally ranked arms) *Consider the following setting with two players and two arms:*

$$p_1 : a_1 \succ a_2 \qquad a_1 : p_1 \succ p_2$$
$$p_2 : a_1 \succ a_2 \qquad a_2 : p_2 \succ p_1.$$

In this case the unique stable matching is $(p_1, a_1), (p_2, a_2)$.

Suppose both players in Example 1 implement CA-UCB with zero probability of delay. Through a random initialization of CA-UCB it is possible that both players select arm a_1 at the first time step. Then, p_2 loses the conflict and at the next step will choose a_2, which is the only arm in their plausible set. On the other hand, the UCB of player p_1 for arm a_2 is positive infinity at this point because they have not pulled it yet. Hence, p_1 attempts to pull a_2 at the second time step. Since a_2 prefers p_2, p_1 loses the conflict and their UCB for arm a_2 remains infinite. The same argument shows that both players will keep choosing the same arm, alternating between a_1 and a_2. As long as they stay in this cycle, both players experience a constant stable regret. We showcase another example of when deterministic conflict-avoiding might fail in Appendix A.

To break such cycles CA-UCB incorporates randomness via the delay probability. As we will see, for arbitrary preferences and delay probability $\lambda \in (0, 1)$, the CA-UCB algorithm achieves $\mathcal{O}(\log(T)^2)$ regret, with the hidden constant depending on λ, the gap between mean rewards, and the number of players and arms. On the other hand, the size of the regret that we obtain depends exponentially on the number of players, regardless of the choice of λ. We can obtain stronger results by making additional assumptions on the structure of preferences. In particular, if the players are globally ranked, then we obtain a polynomial dependence on the number of players; moreover, we obtain $\mathcal{O}(\log(T))$ regret. We begin with this specialized setting in Section 4 and turn to the general case in Section 5.

4. Globally Ranked Players

In this section, we prove regret bounds for the CA-UCB algorithm, Algorithm 1, without random delays (i.e., with $\lambda = 0$). We assume all arms have the same preferences over players, whereas each player may have arbitrary preferences over arms. This preference structure is made precise in the following assumption.

Assumption A1 (Globally ranked players) *We assume the players are globally ranked: for any p_i, $p_{i'}$ where $i < i'$, and any arm a_j, we have $p_i \succ_{a_j} p_{i'}$.*

In other words, more preferred players have lower indices. Under this assumption, there is a unique stable matching in the market. By re-indexing the arms we can assume without loss of generality that the stable player-arm pairs are $\{(p_i, a_i)\}_{i=1}^N$. Under such an indexing, the following critical property holds: for any player p_i and any arm a_j with $j > i$, p_i must prefer a_i over a_j; that is, $a_i \succ_{p_i} a_j$. Also, since the stable matching is unique, there is a single notion of stable regret, that is, for any player p_k, we have $R_k(T) := \underline{R}_k(T) = \overline{R}_k(T)$.

Our goal in this section is to prove an upper bound on the stable regret of a player, taking into account their ranking in the market. We use the following notation to denote

the gaps in mean rewards of arms for players p_i, p_j:

$$\Delta_j^{(i)} := \mu_i^{(i)} - \mu_j^{(i)} \quad \text{and} \quad \Delta_\emptyset^{(i)} := \mu_i^{(i)}. \tag{2}$$

We use $\Delta^2 := \min_{i<j} |\Delta_j^{(i)}|^2$ to denote the minimum squared gap.

Theorem 2 (Stable regret under globally ranked players) *Suppose each player runs Algorithm 1 with $\lambda = 0$. The following regret bound holds for any player p_k and any horizon $T \geq 2$:*

$$R_k(T) \leq 6k^2 \left(\frac{\log T}{\Delta^2} + 1 \right) \cdot \left((L-k)\Delta_\emptyset^{(k)} + k \sum_{i:a_k \succ_{p_k} a_i} \Delta_i^{(k)} \right). \tag{3}$$

This result shows that the stable regret of any player in the market is logarithmic in the horizon T, matching the known lower bound for single-player stochastic bandits (Lai and Robbins, 1985). Moreover, the regret scales cubically with the rank of the player and linearly with the number of arms. It is useful to compare this result to the corresponding stable regret obtained by Liu et al. (2020) in the centralized setting, also under Assumption A1:

$$R_k(T) \leq 6k \sum_{l=k+1}^{L} \left(\Delta_l^{(k)} + \frac{\log T}{\Delta_l^{(k)}} \right). \tag{4}$$

We see that in the centralized setting, the dependence on the rank k is linear instead of cubic. Moreover, the dependence on the reward gap is reduced to $\sum_{i>k} 1/\Delta_i^{(k)}$, which matches the optimal dependence on the reward gaps in the classical single-player bandit problem (Lai and Robbins, 1985). In the decentralized setting where players are globally ranked, Sankararaman et al. (2020) showed a instance dependent lower bound suggesting that the dependence on $1/\Delta^2$ cannot be improved upon in general. We further discuss lower bounds in Section 9.

Before we proceed to the proof of Theorem 2, we introduce the following notation, and establish two technical lemmas.

- $A^{(k)}(t) \in [L]$ is the arm attempted by p_k at time t;

- $\bar{A}^{(k)}(t) \in [L] \cup \{\emptyset\}$ is outcome of p_k's attempt at time t;

- $T_i^{(k)}(t)$ is the total number of attempts by p_k of a_i up to time t;

- $\bar{T}_i^{(k)}(t)$ is the total number of successful attempts by p_k of a_i up to time t.

The following events are central to our analysis:

$$\Lambda_l^{(j)}[t] = \left\{ \bar{A}^{(j)}(t) = a_l, a_j \in S^{(j)}(t) \right\}. \tag{5}$$

In plain language, $\Lambda_l^{(j)}[t]$ denotes the event in which a player p_j chooses to pull an arm a_l over a stable matching arm a_j that belongs to the plausible set at time t.

The next lemma shows that if a player p_k pulls a suboptimal arm a_i (with $i > k$) at time t, then there must be some same or better-ranked player p_j (with $j \leq k$), who, though having its matching arm a_j in their plausible set, chose to pull a suboptimal arm a_l (with $l > j$) at some time t' between times $t - k$ and t.

Lemma 3 (Suboptimal pulls) *For any player p_k and arm a_i such that $a_k \succ_{p_k} a_i$,*

$$\left\{ \bar{A}^{(k)}(t) = a_i \right\} \subseteq \Lambda_i^{(k)}[t] \bigcup \left(\bigcup_{1 \leq j < l \leq k} \bigcup_{t-k \leq t' < t} \Lambda_l^{(j)}[t'] \right). \tag{6}$$

Proof The key to the proof is the following observation. Suppose the event $\left\{ \bar{A}^{(k)}(t) = a_i \right\}$ takes place. Then, one of the two things must happen:

- $a_k \in S^{(k)}(t)$, in which case the event $\Lambda_i^{(k)}[t]$ occurs by definition.

- $a_k \notin S^{(k)}(t)$, in which case some better-ranked player, say p_u with $u < k$, must have pulled the arm a_k at time $t-1$ according to the definition of Algorithm 1.

This observation translates to the following assertion: for any player p_k and arm a_i where $i \neq k$, we have

$$\left\{ \bar{A}^{(k)}(t) = a_i \right\} \subseteq \Lambda_i^{(k)}[t] \bigcup \left(\bigcup_{u<k} \left\{ \bar{A}^{(u)}(t-1) = a_k \right\} \right). \tag{7}$$

We can now prove the lemma by induction on k.

Base case $k = 1$: This is trivially true, due to the fact that the top-ranked player p_1 has all the arms in their plausible set at all times t, and thus, for any arm a_i,

$$\left\{ \bar{A}^{(1)}(t) = a_i \right\} = \left\{ \bar{A}^{(1)}(t) = a_i, a_i \in S^{(1)}(t) \right\} = \Lambda_i^{(1)}[t].$$

Induction step: We assume (6) for all $k < m$ and prove it also holds for $k = m$. Let arm a_i be such that $a_m \succ_{p_m} a_i$. By equation (7), we have

$$\left\{ \bar{A}^{(m)}(t) = a_i \right\} \subseteq \Lambda_i^{(m)}[t] \bigcup \left(\bigcup_{u<m} \left\{ \bar{A}^{(u)}(t-1) = a_m \right\} \right). \tag{8}$$

By our assumptions we know that $a_u \succ_{p_u} a_m$ when $u < m$. Consequently, we can apply the induction hypothesis for player p_u, with $u < m$, and arm a_m and time $t-1$, to obtain that

$$\left\{ \bar{A}^{(u)}(t-1) = a_m \right\} \subseteq \Lambda_m^{(u)}[t-1] \bigcup \left(\bigcup_{1 \leq j < l \leq u} \bigcup_{t-u \leq t' < t-1} \Lambda_l^{(j)}[t'] \right).$$

Taking the union over $u < m$ on both sides yields the inclusion

$$\bigcup_{u<m} \left\{ \bar{A}^{(u)}(t-1) = a_m \right\} \subseteq \bigcup_{1 \leq j < l \leq m} \bigcup_{t-m \leq t' < t} \Lambda_l^{(j)}[t']. \tag{9}$$

By substituting equation (9) into equation (8), we obtain the conclusion. ∎

The next lemma tells a similar story as Lemma 3; it shows that when p_k has a conflict, there must be some better player p_j, with $j < k$, who chooses to pull a suboptimal arm a_l at some time t' between times $t-k$ and t although they have the matching arm a_j in their plausible set.

Lemma 4 (Conflicts) *For any player p_k, we have the inclusion*

$$\left\{ \bar{A}^{(k)}(t) = \emptyset \right\} \subseteq \bigcup_{\substack{1 \leq j < k \\ j < l \leq L}} \bigcup_{t-k \leq t' \leq t} \Lambda_l^{(j)}[t']. \tag{10}$$

Proof Player p_k can have a conflict on any of the arms $a_1, a_2, \ldots, a_L$. We have

$$\left\{ \bar{A}^{(k)}(t) = \emptyset \right\} = \bigcup_{l=1}^{L} \left\{ \bar{A}^{(k)}(t) = \emptyset, A^{(k)}(t) = a_l \right\}.$$

For all $m \geq k$ we observe that p_k can have a conflict on a_l only if there is a player p_j with $j < k$ who successfully pulls arm a_m at time t. In this case we have

$$\left\{ \bar{A}^{(k)}(t) = \emptyset, A^{(k)}(t) = a_m \right\} \subseteq \bigcup_{j < k} \left\{ \bar{A}^{(j)}(t) = a_m \right\}.$$

We can then apply Lemma 3 to each event $\left\{ \bar{A}^{(j)}(t) = a_m \right\}$.

We now have to analyze the events $\left\{ \bar{A}^{(k)}(t) = \emptyset, A^{(k)}(t) = a_m \right\}$ with $m < k$. Since

$$\left\{ \bar{A}^{(k)}(t) = \emptyset, A^{(k)}(t) = a_m \right\} \subseteq \left\{ A^{(k)}(t) = a_m \right\},$$

it suffices to prove by induction that

$$\bigcup_{m=1}^{k-1} \left\{ A^{(k)}(t) = a_m \right\} \subseteq \bigcup_{\substack{1 \leq j < k \\ j < l \leq L}} \bigcup_{t-k \leq t' \leq t} \Lambda_l^{(j)}[t']. \tag{11}$$

The base case $k = 1$ is obvious since the left-hand side is the empty set. Now, we assume the induction hypothesis holds for all $k < k'$ and we prove it for $k = k'$. If $\left\{ A^{(k')}(t) = a_m \right\}$ holds, we know that p_m at time $t - 1$ did not attempt to pull a_m. They either attempted to pull an arm $a_{m'}$ with $m' > m$ or with $m' < m$. In the former case, the induction step follows from Lemma 3. In the latter case, we can apply our induction hypothesis. The result follows. ∎

The final ingredient we need to prove Theorem 2 is the UCB argument for a single player. This is given in the following display. For completeness, we provide an elementary proof in Appendix B.

Lemma 5 (UCB bound) *Suppose we use the following upper confidence bounds in Algorithm 1:*

$$UCB_j^{(i)}(t) = \begin{cases} \infty & \text{if } \bar{T}_j^{(i)}(t) = 0, \\ \widehat{\mu}_j^{(i)}(t) + \sqrt{\frac{3 \log t}{2 \bar{T}_j^{(i)}(t-1)}} & \text{otherwise.} \end{cases} \tag{12}$$

Then, for any player p_i, arms a_j, a_k, such that $a_j \prec_i a_k$, we have, for $T > 0$:

$$\sum_{t=1}^{T} \mathbb{P}\left(\left\{ UCB_j^{(i)}(t) > UCB_k^{(i)}(t) \right\} \cap \left\{ \bar{A}^{(i)}(t) = j \right\} \right) \leq \frac{6}{\Delta^2} \log(T) + 6.$$

Proof [Proof of Theorem 2] We bound the regret of player p_k. By definition, their regret is

$$R_k(T) \leq \Delta_\emptyset^{(k)} \cdot \mathbb{E}\left[\bar{T}_\emptyset^{(k)}(T)\right] + \sum_{i:a_k \succ_{p_k} a_i} \Delta_i^{(k)} \cdot \mathbb{E}\left[\bar{T}_i^{(k)}(T)\right], \tag{13}$$

where, because of our assumption on the indexing of arms, the last summation can also be written simply as a sum over all $i \in \{k+1, \ldots, L\}$.

Upper bounding $\mathbb{E}[\bar{T}_i^{(k)}(T)]$. By definition,

$$\mathbb{E}\left[\bar{T}_i^{(k)}(T)\right] = \sum_{t=1}^T \mathbb{P}\left(\bar{A}^{(k)}(t) = a_i\right). \tag{14}$$

We now bound the probability $\mathbb{P}\left(\bar{A}^{(k)}(t) = a_i\right)$ for each t. Lemma 3 yields

$$\mathbb{P}(\bar{A}^{(k)}(t) = a_i) \leq \mathbb{P}(\Lambda_i^{(k)}[t]) + \sum_{1 \leq j < l \leq k} \sum_{t-k \leq t' < t} \mathbb{P}(\Lambda_l^{(j)}[t']). \tag{15}$$

Summing from $t = 1$ to T, and using equation (14), we obtain the bound

$$\begin{aligned}
\mathbb{E}\left[\bar{T}_i^{(k)}(T)\right] &\leq \sum_{1 \leq t \leq T} \mathbb{P}(\Lambda_i^{(k)}[t]) + \sum_{1 \leq t \leq T} \sum_{1 \leq j < l \leq k} \sum_{t-k \leq t' \leq t} \mathbb{P}(\Lambda_l^{(j)}[t']) \\
&\leq \sum_{1 \leq t \leq T} \mathbb{P}(\Lambda_i^{(k)}[t]) + (k+1) \sum_{1 \leq j < l \leq k} \sum_{1 \leq t \leq T} \mathbb{P}(\Lambda_l^{(j)}(t)).
\end{aligned} \tag{16}$$

Recall that for all players p_j and arms a_l with $l > j$, and time $t > 0$,

$$\Lambda_l^{(j)}(t) \subseteq \left\{\text{UCB}_l^{(j)}(t) > \text{UCB}_i^{(j)}(t)\right\} \cap \left\{\bar{A}^{(j)}(t) = l\right\}.$$

Therefore, using Lemma 5, we can show that the following upper bound holds:

$$\sum_{1 \leq t' \leq T} \mathbb{P}(\Lambda_l^{(j)}(t')) \leq 6\left(\frac{\log T}{|\Delta_l^{(j)}|^2} + 1\right). \tag{17}$$

Substituting equation (17) into equation (16) yields the bound

$$\mathbb{E}\left[\bar{T}_i^{(k)}(T)\right] \leq 6\left(\frac{\log T}{|\Delta_i^{(k)}|^2} + 1\right) + 6(k+1) \sum_{1 \leq j < l \leq k} \left(\frac{\log T}{|\Delta_l^{(j)}|^2} + 1\right) \leq 6k^3\left(\frac{\log T}{\Delta^2} + 1\right). \tag{18}$$

Recall that $\Delta^2 = \min_{i \neq j} |\Delta_j^{(i)}|^2$.

Upper bounding $\mathbb{E}\left[\bar{T}_\emptyset^{(k)}(T)\right]$. By definition,

$$\mathbb{E}\left[\bar{T}_\emptyset^{(k)}(T)\right] = \sum_{t=1}^T \mathbb{P}\left(\bar{A}^{(k)}(t) = \emptyset\right). \tag{19}$$

11

Lemma 4 and a derivation mutatis mutandis to the argument from equation (15) to (18) yields

$$\mathbb{E}\left[\bar{T}_i^{(\emptyset)}(T)\right] \le 6(k+1) \sum_{\substack{1 \le j < k \\ j < l \le L}} \left(\frac{\log T}{|\Delta_l^{(j)}|^2} + 1\right) \le 6k^2(L-k)\left(\frac{\log T}{\Delta^2} + 1\right). \tag{20}$$

Substitute (18) and (20) into (13) to complete the proof of the theorem. ∎

5. Arbitrary Preferences on Both Sides of the Market

In this section, we analyze the convergence of Algorithm 1 under arbitrary preference lists for both sides of the market. Note that in this setting, the stable matching may not be unique. We consider throughout the randomized version of Algorithm 1, with delay probability $\lambda > 0$.

Without the assumption of shared preferences among the arms, the analysis of the convergence of Algorithm 1 becomes more challenging. In fact, it is not obvious that Algorithm 1, or any other algorithm, can achieve sublinear player regret against the pessimal stable matching for any set of preferences. As seen in Example 2 in Appendix A, decentralized coordination among players can be difficult even in small markets with only three players. In order to prove the regret bound in Section 4, we relied heavily on the structure conferred by the global ranking of players. Without this particular structure, we have to appeal to more general results about stable matching. This generality also comes at a cost: the regret bound we prove in this section is polylogarithmic in the horizon and has an exponential dependence on the number of players.

Before introducing the main result, we first present some essential notation. Recall that $\mathcal{N} = \{p_i\}_{i=1}^N$ denotes the set of players, and $\mathcal{K} = \{a_i\}_{i=1}^L$ denotes the set of arms. We denote the attempted actions (i.e., arms) at time t as

$$m_t : \mathcal{N} \mapsto \mathcal{K}, \text{ where } m_t(p_i) := A^{(i)}(t).$$

We note that m_t in general does not have to be a matching between players and arms, because two or more players may attempt to pull the same arm. However, whenever there are no conflicts, m_t is indeed a matching (an injective map) between players and arms, so we can distinguish the set of attempted actions that coincide with a stable matching. We thus refer to $m : \mathcal{N} \mapsto \mathcal{K}$ as *stable* if m indeed coincides with a stable matching between players and arms.

We denote the set of stable attempted actions as

$$M^* := \{M \mid M : \mathcal{N} \mapsto \mathcal{K}, M \text{ is stable}\}.$$

Let $\Delta = \min_{i,j,k} |\mu_j^{(i)} - \mu_k^{(i)}|$ denote the minimum reward gap between any two arms for any player. We also define the constant $\varepsilon := (1-\lambda)\lambda^{N-1}$, which depends on the delay probability λ.

Our goal in this section is to prove the following upper bound on the probability that the market is in an unstable configuration when running the algorithm. More formally,

we bound the sum, over t, of probabilities that the attempted actions at time t yield an unstable matching. Understanding how this quantity depends on the horizon and various problem parameters enables us to provide a general regret bound for Algorithm 1.

Theorem 6 (Convergence to stability of Algorithm 1 for arbitrary preferences)
Let $N, L \geq 2$, $T \geq 2$, and suppose we run Algorithm 1 with delay probability $\lambda \in (0, 1)$. Then,

$$\sum_{t=1}^{T} \mathbb{P}(m_t \neq M^*) \leq 24 \cdot \frac{N^5 L^2}{\varepsilon^{N^4+1}} \log(T) \left(\frac{1}{\Delta^2} \log(T) + 3 \right). \tag{21}$$

As a corollary of Theorem 6, we have the following upper bound on the pessimal stable regret of any player.

Corollary 7 (Pessimal stable regret of Algorithm 1 for arbitrary preferences) *The following inequality holds for the agent-pessimal regret of player p_k up to time T:*

$$\underline{R}_k(T) \leq 24 \cdot \max_{a_\ell \in \mathcal{K}} \underline{\Delta}_{k,\ell} \left(\frac{N^5 L^2}{\varepsilon^{N^4+1}} \log(T) \left(\frac{1}{\Delta^2} \log(T) + 3 \right) \right),$$

where $\underline{\Delta}_{k,\ell} = \max\{\mu_{\underline{m}(k)}^{(k)} - \mu_\ell^{(k)}, \mu_{\underline{m}(k)}^{(k)}\}$.

In short, we find that the stable regret of Algorithm 1 is $\mathcal{O}((\log T)^2)$. Unlike in previous sections where we derived player-specific stable regret bounds that depended on the ranking of the player, or the ranking of their stable arm, in the current setting the players have no particular ranking. Corollary 7 is derived from a general bound on the probabilities that the matching of the entire market is unstable.

Proof sketch We begin by sketching the main ideas in the proof of Theorem 6. There are two main technical ingredients that are new to the current section: the first is the observation that in the event that each player's UCB rankings of the arms in their plausible set are correct (colloquially we refer to this event as "no statistical mistakes"), and the previous matching was stable, then running one step of Algorithm 1 will preserve the stability of the matching with probability one. This is established in Lemma 8. Therefore, if the matching at time t is unstable, it must be either be that some player had incorrect UCB rankings, or there were no statistical ranking mistakes but the matching at time $t-1$ was unstable.

In Lemma 9, we generalize this statement to consider histories of arbitrary length. That is, if a matching at time t is unstable, it must either be that some player had incorrect UCB rankings over the last h time steps, or there were no ranking mistakes in all the last h time steps but the matchings reached were unstable.

As in Section 5, we know how to upper bound the probability that a player had incorrect UCB rankings when running Algorithm 1 with $\lambda > 1$. Recall that this entailed a simple adaptation of the single-player UCB argument (Lemma 5). The new problem we face is that of controlling the probability that there were no ranking mistakes but the matchings in all the last h time steps were unstable. It turns out that a classical result from the stable matching literature (Abeledo and Rothblum, 1995) gives us a way to argue that this probability is exponentially small in the length of the history considered (Lemma 14). Intuitively,

we are using the fact that Algorithm 1, when there are no ranking mistakes, is essentially resolving *blocking pairs*—pairs of players and arms that would prefer to be matched with each other over their current matches—in a randomized fashion, but following an order that is consistent with player preferences (Lemma 13). This is crucial for establishing that Algorithm 1 will always reach a stable matching with enough steps, as long as there are no ranking mistakes.

Finally, our analysis needs to balance the tradeoff inherent in the choice of the length of history considered, h. If we consider a longer history length, there can be many ranking mistakes made in this window, hence contributing to a higher probability of an unstable matching. On the other hand, a longer history length with no ranking mistakes means that there is a higher probability that a stable matching can be reached. By choosing h to depend on the time step t, we are able to achieve a $\log(T)^2$ dependence on the horizon T in the final bound (21).

Before presenting the technical lemmas, we first rigorously define the events of interest that were alluded to in the proof sketch.

1. Let E_t denote the event that, for every player, the arm that has the highest mean reward in their plausible set coincides with the arm with the highest UCB in their plausible set at time t:

$$E_t := \bigcap_{p_i \in \mathcal{N}} \left\{ \operatorname*{argmax}_{a_j \in S^{(i)}(t)} \mu_j^{(i)} = \operatorname*{argmax}_{a_j \in S^{(i)}(t)} \mathrm{UCB}_j^{(i)}(t) \right\}. \tag{22}$$

   Let E_t^c denote the complement of this event.

2. Let $F_{j,k}^{(i)}(t)$ denote the event that player p_i's UCB for arm a_j is greater than their UCB for arm a_k at time t:

$$F_{j,k}^{(i)}(t) = \left\{ \mathrm{UCB}_j^{(i)}(t) > \mathrm{UCB}_k^{(i)}(t) \right\}.$$

The following lemma shows that if the current matching is stable, then one step of Algorithm 1 under the event defined in (22) preserves the stability of the current matching.

Lemma 8 (Preservation of Stability) *Assume $m_t \in M^*$. Then $m_{t+1} \in M^*$ on the event E_{t+1}.*

Proof We show $m_{t+1} = m_t$ on event E_{t+1}. Let $m = m_t$. Assume E_{t+1} happens. Suppose, for a contradiction, that some player p attempts an arm $a \neq m(p)$ at time $t+1$. Let p' be the player that a is matched to at time t, that is, $M(a) = p'$, if a is matched at time t, and let $p' = \emptyset$, otherwise. Note that since p is matched with $m(p)$ at time t, $m(p)$ must belong to the plausible set of p at time $t+1$ by definition of the algorithm. Since p attempts $a \neq m(p)$ at $t+1$, this implies that (i) p truly prefers a over $m(p)$ by definition of E_{t+1} and (ii) a truly prefers p over p', since a must be in the plausible set of p at time $t+1$. Thus (p, a) are a blocking pair for the matching m, contradicting the assumption that $m \in M^*$. Thus we have shown $m_{t+1} = m_t = m \in M^*$. ∎

In the next lemma, we apply Lemma 8 repeatedly to show that the event that the current matching is unstable can be decomposed into prior events that occurred up to K steps in the past. Specifically, if the current matching is unstable, then either the UCB ranking of arms were wrong at some point in the history of length K (that is, (22) was false), or the matching was unstable for K consecutive steps even though (22) was true in all K steps.

Lemma 9 (Inclusion for unstable matching event) *We have the following inclusion that holds for any $0 \leq K < t - 1$:*

$$\{m_t \notin M^*\} \subseteq \left(\bigcup_{s=0}^{K} E_{t-s}^c \right) \bigcup \left(\bigcap_{s=0}^{K} (E_{t-s} \cap \{m_{t-s-1} \notin M^*\}) \right).$$

Proof This is an immediate consequence of Lemma 8. In fact, Lemma 8 shows

$$\{m_t \notin M^*\} \subseteq E_t^c \cup (E_t \cap \{m_{t-1} \notin M^*\}). \tag{23}$$

This shows Lemma 9 holds for $K = 0$. A simple induction argument shows that Lemma 9 holds for general $K > 0$, $K < t - 1$. ∎

Lemma 9 suggests that in order to derive an upper bound on the probability that m_t is unstable, we can separately bound the probabilities of the event that the UCB ranking of arms has an error, and the event that the matching was unstable for K consecutive steps even though UCB rankings were correct in all K steps. The following lemma addresses the former.

Lemma 10 (Probability of ranking error event) *The following inequality holds for any $t > 0$:*
$$\mathbb{P}(E_t^c) \leq \varepsilon^{-1} \cdot \sum_{(i,j,k),: a_j \prec_i a_k} \mathbb{P}(F_{j,k}^{(i)}(t) \cap \bar{A}^{(i)}(t) = j).$$

Proof The key is the following observation. That E_t^c happens implies the existence of some player p_i and arms a_j, a_k in their plausible set at time t, such that while the arm a_j achieves the highest UCB with respect to player p_i, the player truly prefers arm a_k over a_j, Hence, this implies

$$\mathbb{P}(E_t^c) \leq \sum_{(i,j,k),: a_j \prec_i a_k} \mathbb{P}(\mathrm{UCB}_j^{(i)}(t) > \mathrm{UCB}_k^{(i)}(t) \cap \{j = \operatorname*{argmax}_{j'} \mathrm{UCB}_{j'}^{(i)}(t)\}).$$

Recall $F_{j,k}^{(i)}(t) = \{\mathrm{UCB}_j^{(i)}(t) > \mathrm{UCB}_k^{(i)}(t)\}$. Lemma 10 now follows if we can show

$$\mathbb{P}(F_{j,k}^{(i)}(t) \cap \{j = \operatorname*{argmax}_{j'} \mathrm{UCB}_{j'}^{(i)}(t)\}) \leq \varepsilon^{-1} \cdot \mathbb{P}(F_{j,k}^{(i)}(t) \cap \bar{A}^{(i)}(t) = j).$$

To see this, note that the player p_i will successfully pull a_j if player p_i doesn't draw a random delay and all the rest of the players draw the random delay (meaning they all attempt the same arm as they attempted in the last round). By independence of the random draws, this event happens with probability at least $\varepsilon = (1 - \lambda)\lambda^{N-1}$. ∎

Having established this lemma, we can now easily apply the UCB argument as given in Lemma 5 to bound the relevant quantity, $\sum_{t=1}^{T} \mathbb{P}(E_t^c)$.

We proceed to analyze the probability of the event that the matching was unstable for K consecutive steps even though UCB rankings were correct in all K steps. Essentially, this requires us to establish how quickly the decentralized conflict-avoiding procedure converges to a stable matching when there are no statistical errors in the rankings of arms. To do so, we invoke a result from the stable matching literature (Abeledo and Rothblum, 1995). First, we introduce the notion of a blocking pair that is *player-consistent*.

Definition 11 (Player-consistent blocking pair) *A blocking pair (p_i, a_j) in a matching μ is* player-consistent *if*

$$a_j \succ_{p_i} a_k \quad \text{for any } k \text{ such that } (p_i, a_k) \text{ is a blocking pair in } \mu. \tag{24}$$

In other words, if player p_i most prefers the a_j out of all the arms that prefer p_i over the player that they are matched to in μ, then the blocking pair (p_i, a_j) is player-consistent. Notice that in Algorithm 1, at time t, if the UCB rankings are accurate, then each player p_i (who did not draw a random delay) will attempt precisely the arm a_j where (p_i, a_j) is a player-consistent blocking pair in the matching μ induced by the previous attempted actions m_{t-1}, by the definition of the plausible set.

We also require the following definition of *resolving* a blocking pair, in the context of running one step of Algorithm 1.

Definition 12 (Resolution of blocking pair) *Given attempted actions $m_t \notin M^*$ and a blocking pair (p_i, a_j) in the matching induced by m_t, we say that m_{t+1} is obtained by resolving (p_i, a_j), if $m_{t+1}(p_i) = a_j$ and $m_{t+1}(p) = m_t(p)$ for all $p \in \mathcal{N}, p \neq p_i$.*

We are ready to establish a key result—that there is a strictly positive probability that a single player-consistent blocking pair is resolved in one step of Algorithm 1.

Lemma 13 (Positive probability of resolving a single blocking pair) *Assume m_{t-1} is unstable. Let (p_i, a_j) be a blocking pair in m_{t-1} that is player-consistent. Condition on the event E_t. Then, with probability at least $\varepsilon = (1 - \lambda)\lambda^{N-1}$, (p_i, a_j) is the only blocking pair to be resolved at time t, i.e.,*

$$\mathbb{P}\left(m_t(p_i) = a_j, m_t(p) = m_{t-1}(p) \; \forall p \in \mathcal{N}, p \neq p_i \mid E_t\right) \geq \varepsilon. \tag{25}$$

Proof Assume E_t holds. Let (p_i, a_j) be any blocking pair that is player-consistent. First, we show p_i has probability at least λ of pulling the arm a_j conditioned on all of the other players attempting the same arm as they pulled at time t. Indeed, since (p_i, a_j) is a blocking pair of m_{t-1}, it means that a_j is in the plausible set of p_i at time t. As E_t occurs, and a_j is the top choice among all the arms in p_i's plausible set, the player p_i has probability at least λ of attempting a_j, and will be successful if all other players stay on the same arm as they pulled at time t. Second, independently, each of the rest of the $N - 1$ players have probability at least $(1 - \lambda)$ of attempting the same arm that they attempted at time $t - 1$. Together, this proves equation (25). ∎

Now we can finally show that the event that the matching was unstable for K consecutive steps even though UCB rankings were correct in all K steps happens with a probability that is exponentially small in K, as stated formally in the lemma below.

Lemma 14 (Probability of not reaching a stable matching) *For any $0 \le K < t-1$, the following inequality holds:*

$$\mathbb{P}\left(\bigcap_{s=0}^{K}\left(\{m_{t-s-1} \notin M^*\} \cap E_{t-s}\right)\right) \le (1-\varepsilon^{N^4})^{\lfloor K/N^4 \rfloor}. \tag{26}$$

Proof The result is a direct consequence of Lemma 13 and the theorem below.

Theorem 15 (Theorem 4.2 in Abeledo and Rothblum (1995)) *Given any unstable matching μ_0, there exists a sequence of blocking pairs of length at most N^4 such that resolving the sequence of blocking pairs reaches a stable matching. Moreover, this sequence of blocking pairs results from resolving blocking pairs in a* player-consistent *order, that is, any blocking pair (p_i, a_j) resolved in the current matching μ is player-consistent with respect to the matching μ.*

We now prove Lemma 14 using Lemma 13 and Theorem 15.

1. We first show Lemma 14 holds when $K = N^4$. Let $E = \cap_{s=0}^{K} E_{t-s}$. Condition on the event that E happens. Condition on the matching $\mu = m_{t-K-1}$. By Theorem 15 and Lemma 13, we know that with probability at least ε^{N^4}, a stable matching will be reached within N^4 steps of the algorithm. Since this holds for arbitrary $\mu = m_{t-K-1}$, we obtain

$$\mathbb{P}\left(\bigcap_{s=0}^{K}\{m_{t-s-1} \notin M^*\} \mid E\right) \le 1 - \varepsilon^{N^4}.$$

   Thus, we have

$$\mathbb{P}\left(\bigcap_{s=0}^{K}\left(\{m_{t-s-1} \notin M^*\} \cap E_{t-s}\right)\right) \le \mathbb{P}\left(\bigcap_{s=0}^{K}\{m_{t-s-1} \notin M^*\} \mid E\right) \le 1 - \varepsilon^{N^4}.$$

2. We next generalize the result to $K > N^4$. This is straightforward, as the random seeds x in Algorithm 1 are mutually independent for any non-overlapping blocks of N^4 steps.

$$\blacksquare$$

Note that in order for this bound to be meaningful, we require $K \gg \varepsilon^{-N^4} N^4$.

Finally, we are now fully equipped to prove the main result of this section.

Proof of Theorem 6 Let $0 \le h_t < t$ be a time window that we are free to choose in a way that depends on the time t. By Lemma 9 and the union bound, we have

$$\mathbb{P}\left(m_t \notin M^*\right) \le \mathbb{P}\left(\bigcap_{s=0}^{h_t}\left(E_{t-s} \cap \{m_{t-s-1} \notin M^*\}\right)\right) + \sum_{s=0}^{h_t}\mathbb{P}(E_{t-s}^c).$$

Let $g_t = \lfloor h_t/N^4 \rfloor$. Lemmas 10 and 14 immediately yield the following:

$$\mathbb{P}\left(m_t \notin M^*\right) \leq (1 - \varepsilon^{N^4})^{g_t} + \varepsilon^{-1} \sum_{s=0}^{h_t} \sum_{(i,j,k),:a_j \prec_i a_k} \mathbb{P}(F_{j,k}^{(i)}(t-s) \cap \bar{A}^{(i)}(t-s) = j).$$

Summing these inequalities over t up to T, we obtain

$$\sum_{t=1}^{T} \mathbb{P}\left(m_t \notin M^*\right) \leq \sum_{t=1}^{T}(1 - \varepsilon^{N^4})^{g_t} + \varepsilon^{-1} \sum_{t=1}^{T} \sum_{s=0}^{h_t} \sum_{(i,j,k),:a_j \prec_i a_k} \mathbb{P}(F_{j,k}^{(i)}(t-s) \cap \bar{A}^{(i)}(t-s) = j)$$

$$= \sum_{t=1}^{T}(1 - \varepsilon^{N^4})^{g_t} + \varepsilon^{-1} \sum_{(i,j,k),:a_j \prec_i a_k} \sum_{s=0}^{h_T} \sum_{\substack{t:s \leq h_t \\ 1 \leq t \leq T}} \mathbb{P}(F_{j,k}^{(i)}(t-s) \cap \bar{A}^{(i)}(t-s) = j)$$

$$(27)$$

We seek upper bounds for the terms on the right-hand side. Focus on the second term in equation (27). Recall the standard UCB Lemma (e.g., Lemma 5):

$$\sum_{\substack{t:s \leq h_t \\ 1 \leq t \leq T}} \mathbb{P}(F_{j,k}^{(i)}(t-s) \cap \bar{A}^{(i)}(t-s) = j) \leq 6 \cdot \left(\frac{1}{\Delta^2} \log(T) + 1\right), \quad \text{for each } s.$$

Substituting this bound into equation (27) yields

$$\sum_{t=1}^{T} \mathbb{P}\left(m_t \notin M^*\right) \leq \sum_{t=1}^{T}(1 - \varepsilon^{N^4})^{g_t} + 6\varepsilon^{-1}NL^2\left(h_T + 1\right)\left(\frac{1}{\Delta^2} \log(T) + 1\right), \quad (28)$$

where we have used the fact that there are at most NL^2 triplets (i, j, k) such that $a_j \prec_i a_k$. We now choose a specific sequence (h_t) to optimize the upper bound. Let $B \geq 1$ be determined later. Set $h_t = \min\{t, B\} - 1$. With this choice of h_t, and after some elementary computations, we can bound the first term in equation (28) by

$$\sum_{t=1}^{T}(1 - \varepsilon^{N^4})^{g_t} \leq 3 \cdot \sum_{t=1}^{T} \exp(-h_t \varepsilon^{N^4}/N^4) \leq 6 \cdot \left(T \exp\left(-\frac{B\varepsilon^{N^4}}{2N^4}\right) + \frac{N^4}{\varepsilon^{N^4}}\right).$$

The second term in equation (28) is bounded by $6\varepsilon^{-1}BNL^2\left(\frac{1}{\Delta^2} \log(T) + 1\right)$, since $h_T < B$ by definition. Consequently, these two bounds lead to the following (that holds for all B)

$$\sum_{t=1}^{T} \mathbb{P}\left(m_t \notin M^*\right) \leq 6 \cdot \left(T \exp\left(-\frac{B\varepsilon^{N^4}}{2N^4}\right) + \frac{N^4}{\varepsilon^{N^4}} + \frac{1}{\varepsilon}NL^2B\left(\frac{1}{\Delta^2} \log(T) + 1\right)\right).$$

By carefully setting $B = 2\left\lceil \frac{N^4}{\varepsilon^{N^4}} \log(T) \right\rceil$, we obtain the final bound as desired

$$\sum_{t=1}^{T} \mathbb{P}\left(m_t \notin M^*\right) \leq 24 \cdot \frac{N^5L^2}{\varepsilon^{N^4+1}} \log(T) \cdot \left(\frac{1}{\Delta^2} \log(T) + 3\right).$$

∎

6. Strategy and Incentive Compatibility

In this section, we examine the CA-UCB algorithm from the perspective of incentive compatibility.

Thus far we have given stable regret guarantees for each player, when all players follow the same algorithm, whether assuming a global ranking of players (Theorem 2), or without making assumptions on the market's preferences (Theorem 6). Given these results, a natural question to consider, in the decentralized setting, is whether the players are indeed incentivized to run the same algorithm as everyone else. In other words, could any single player benefit from running a different algorithm, when all other players are running Algorithm 1?

6.1 A positive result for globally ranked players

In the setting of Section 4, when players are globally ranked, we can show that the gains from deviating are limited. The following proposition gives an lower bound on the stable regret of the deviating player that scales logarithmically in the horizon T, for any algorithm that they run. This implies that the time-averaged gains from deviating must vanish quickly as learning progresses.

Proposition 16 (Incentive compatibility under globally ranked players) *Under Assumption A1, suppose that all players other than player p_k run Algorithm 1 with $\lambda = 0$, and p_k can run any algorithm. The following lower bound on player p_k's stable regret holds:*

$$R_k(T) \geq 6k^2(L - k)\left(\frac{\log T}{\Delta^2} + 1\right)\left(\min_{j:\Delta_j^{(k)}<0} \Delta_j^{(k)}\right). \tag{29}$$

This result follows from a simple application of the same arguments that we developed to prove Theorem 2. The key idea is as follows. A deviating player that is rank k in the market can successfully pull an arm a_i that they prefer to their stable arm, only if the better-ranked player p_i is not pulling their stable arm a_i in the same round. This can only happen if p_i or a better-ranked player had a mistake in their UCB rankings and pulled a suboptimal arm within the last k rounds, since all players other than p_k are indeed following the CA-UCB algorithm. The gains to deviating are limited for player p_k when all the arms have the same preferences, precisely because p_k cannot affect the actions of better ranked players. A complete proof can be found in Appendix C.

6.2 A negative result

Given that we have a general stable regret guarantee for arbitrary preferences, established in Section 5, one might ask if there also exists a general incentive compatibility result for Algorithm 1. Unfortunately, the answer is a negative one. The following proposition shows, by way of counterexample, that there can be no blanket incentive compatibility guarantee for Algorithm 1 without making additional assumptions, such as on the preference structure.

Proposition 17 *Consider the market of three players and three arms with preferences as given in Example 3. When two players p_1 and p_2 run Algorithm 1 with any $\lambda \in (0, 1/4)$,*

there exists a sequence of actions $\{A^{(3)}(t)\}_{t=1...T}$ for player p_3 such that p_3's stable regret can be upper bounded as:

$$R_3(T) \leq -C_1 \cdot T + C_2 \left(\frac{1}{\Delta^2} \log(T) + 1 \right),\tag{30}$$

where C_1 and C_2 are constants that depend only on $\lambda, \Delta_1^{(3)}, \Delta_\emptyset^{(3)}$. Moreover, there exists $\Delta_1^{(3)}, \Delta_\emptyset^{(3)}$ such that C_1 is strictly positive.

The above upper bound on the deviating player p_3's stable regret shows that there exists a set of preferences and arm reward gaps such that a player could make significant gains over their stable arm by not running Algorithm 1. We defer the full description of Example 3 and the proof of Proposition 17 to Appendix D. In this example, p_3 has stable arm a_3 but prefers a_1. Because the arms have idiosyncratic preferences (as opposed to shared preferences), p_3 could pull a suboptimal arm in order to 'trick' p_1 into not attempting a_1 two rounds later, by exploiting the conflict avoidance mechanism; p_3 can then successfully pull a_1 for one round, with some probability. As long as the reward for p_3 from a_1 is large enough, p_3 is guaranteed a strictly negative stable regret that is linear in the horizon T.

We have shown that Algorithm 1 is not incentive compatible in the fully general setting. It therefore remains an open question whether there exists an algorithm with low stable regret, under arbitrary preferences, that also has an incentive compatibility guarantee under the same.

7. Simulation experiments for random preferences

In our theoretical analysis we considered two cases: markets in which the players are globally ranked (i.e. all arms have the same preferences over players) and markets with arbitrary preferences. For the first case Theorem 2 we were able to prove a regret upper bound that resembles the guarantee derived by Liu et al. (2020) in the centralized case. However, in the case of general markets our guarantee (Theorem 6) has an exponential dependence on the size of the market.

In this section, through empirical evaluations we show that the true performance of our proposed method is likely better than our guarantee suggests for markets with randomly drawn preferences. More precisely, we perform two sets of simulations. In the first set, we investigate how the average regret and market stability depend on the size of the market in balanced markets—markets with an equal number of players and arms—with preferences drawn from a distribution that will be specified later. We find that empirically the algorithm converges more slowly for larger number of players as expected, though the dependence on the number of players, N, appears to be significantly better than the exponential dependence appearing in Theorem 6.

In the second set of experiments, we vary the heterogeneity of the players' preferences. We perform this experiment because one might expect that in markets in which different players have the same preferences there would be more conflicts (since different players have an incentive to attempt the same arms). Despite this intuition, our simulations show that CA-UCB performs equally well in markets with different level of heterogeneity. To sum

up, our simulations show that not only is Theorem 6 overly pessimistic, but that CA-UCB avoids conflicts equally well in different markets.

For all experiments we use Algorithm 1 with delay probability $\lambda = 0.1$. We now present the details of our simulations.

Varying the size of the market. We examine balanced markets of size $N \in \{5, 10, 15, 20\}$, and sample each player's and arm's ordinal preferences uniformly at random. For all players the reward gaps between consecutively ranked arms are chosen to be equal to $\Delta = 1$, regardless of the market size. The rewards are normally distributed with unit variance. We sampled ten markets as such, and run Algorithm 1 once on each market.

For each market size N, we plot the mean, over ten markets, of the following two quantities: (i) the maximum average regret among players, $\max_{k \in \mathcal{N}} R_k(T)$, and (ii) the probability of convergence to stability $\sum_{t=1}^{T} \mathbb{P}\left(m_t \notin M^*\right)$ for horizon T up to 5000. As can be seen in Figure 1, both the average regret and the market stability converge more slowly for larger markets. However, the dependence on N appears to be much better than exponential.

Varying the heterogeneity of the players' preferences. We examine balanced markets of size 10, and sample each arm's ordinal preferences uniformly at random. To sample the mean rewards $\mu_i^{(k)}$ of arm a_i for player p_k we rely on random utility model used by Ashlagi et al. (2017b), with a slight modification:

$$x_i \overset{i.i.d.}{\sim} \text{Uniform}([0, 1])$$

$$\varepsilon_{i,k} \overset{i.i.d.}{\sim} \text{Logistic}(0, 1)$$

$$\overline{\mu}_i^{(k)} = \beta x_i + \varepsilon_{i,k}$$

$$\mu_i^{(k)} = \#\{j : \overline{\mu}_j^{(k)} \leq \overline{\mu}_i^{(k)}\}$$

The intermediate utilities $\overline{\mu}_i^{(k)}$ are sampled according to random utility model used by Ashlagi et al. (2017b). We map these random utilities to $\mu_i^{(k)}$ so that the reward gaps between consecutively ranked arms are kept constant at $\Delta = 1$. The parameter $\beta > 0$ determines the degree of correlation between the players' preferences. As β increases the correlation between the players' preferences also increases.[5] In fact, in the limit as $\beta \to \infty$, all the players share the same preferences with probability 1.

As before, the rewards are normally distributed with unit variance. We sample ten markets for each β value, and plot the maximum average regret among players as well as the probability of convergence to stability for horizon T up to 5000 in Figure 2. As can be seen, there is no discernible difference in the convergence of Algorithm 1 in terms of regret or market stability, for markets with different levels of preference heterogeneity.

8. Related Work

There has been significant recent interest in stochastic multi-armed bandits problems with multiple, interacting players (Cesa-Bianchi et al., 2016; Shahrampour et al., 2017). In one

---

5. The Spearman's rank correlation matrices of the rankings produced for $\beta \in \{0, 10, 100, 1000\}$ have the following matrix means: $0.080, 0.189, 0.567, 0.999$.
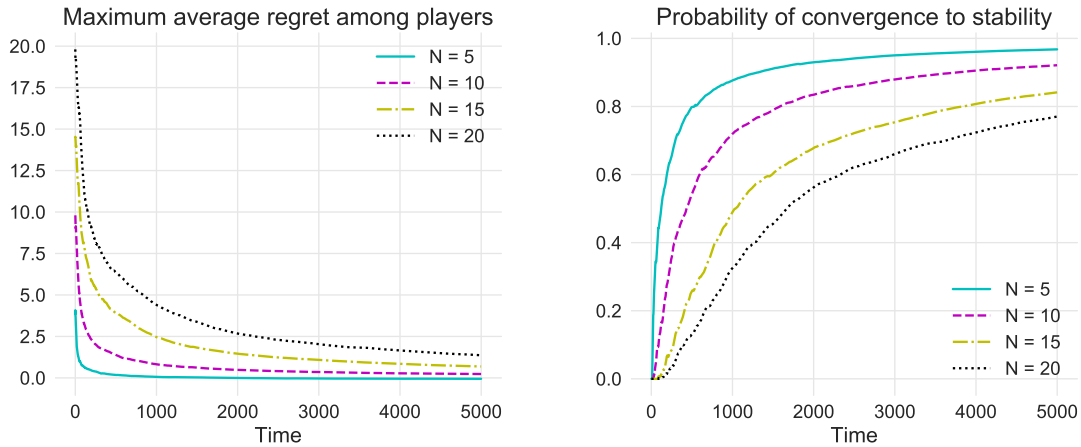
Figure 1: Varying the number of players. The plot on the left shows the maximum average regret among players and the plot on the right shows the probability of convergence to stability.
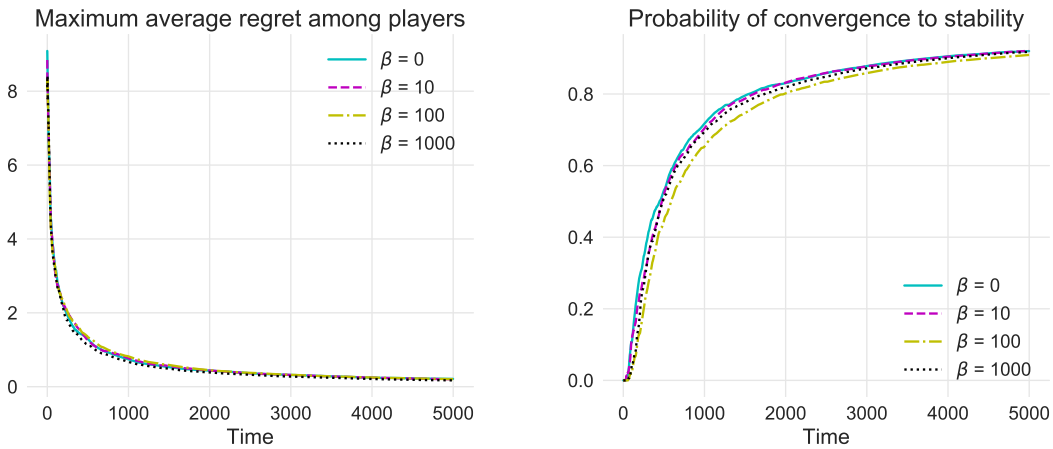

Figure 2: Varying the heterogeneity of the players' preferences. The plot on the left shows the maximum average regret among players and the plot on the right shows the probability of convergence to stability. The larger the β parameter, the more correlated the players' preferences are on average.

formulation, known as *bandits with collision*, multiple players choose from the same set of arms, and if two or more players choose the same arm, no reward is received by any player (Liu and Zhao, 2010; Anandkumar et al., 2011; Avner and Mannor, 2014; Bistritz

et al., 2020; Bubeck et al., 2020a,b; Kalathil et al., 2014; Rosenski et al., 2016; Lugosi and Mehrabian, 2018). In this setting, players are typically assumed to be cooperative, that is, their goal is to maximize the collective reward. Bistritz and Leshem (2018) and Boursier and Perchet (2020) consider the setting where agents have heterogeneous preferences over arms, and the latter work also analyzes the effect of selfish players whose goal is to maximize individual rewards. Avner and Mannor (2016) and Darak and Hanawal (2019) considered a "stable configuration" as a solution concept in the heterogeneous player preference setting; however, because the arms do not have preferences in their setting, their notion of "stability" is distinct from that of two-sided stable matching. Bubeck et al. (2020b) also delineated the optimal rates for the non-stochastic version of the cooperative problem.

Liu et al. (2020) introduced a multi-player stochastic multi-armed bandits problem motivated by two-sided matching markets, where arms also have preferences, and in case of collision only the most preferred player receives a reward. Unlike in the aforementioned line of work, where the natural goal is to find a maximum matching between players and arms, a more appropriate goal here is to find a stable matching. In the centralized setting, where a platform can coordinate the actions of players at each round, Liu et al. (2020)'s algorithm combining the upper confidence bound method and the deferred acceptance algorithm attains $\mathcal{O}(\log(T)/\Delta^2)$ stable regret, which is order-optimal. A suboptimal algorithm based on explore-then-commit was proposed for the decentralized setting. Follow-up work by Sankararaman et al. (2020) on the decentralized setting analyzed an order-optimal algorithm for globally ranked players. A more detailed discussion of this work is in Section 9.

The two-sided stable matching problem with preference learning has been studied in other dynamic settings under different assumptions. Given the large space of modeling choices, there has been a flowering of research on two-sided matching models that highlight different challenges introduced by uncertainty and decentralization. One modeling choice is to define arrival and departure processes for market participants, as opposed to analyzing a fixed set of players and arms. Johari et al. (2017) studied a sequential matching problem in which the market participants satisfy certain arrival processes, and the participants on the demand side of the market have a 'type' that is learned through bandit feedback.

Another choice is how one formulates the cost of preference learning. Ashlagi et al. (2017a), which studies the costs of communication and learning for stable matching, formulates preference learning as querying a costly but noiseless choice function. Different players can query their choice functions independently; thus there is no congestion in the preference learning process. Many models studied in the literature on information acquisition in two sided matching (see Lee and Schwarz, 2009; Immorlica et al., 2020, and references therein) also do not capture congestion in the information acquisition stage. In some markets, however, obtaining information about the other side of the market itself could lead to congestion and thus the need for strategic decisions. For example, Roth and Sotomayor (1990, chap. 10) note that graduating medical students go to interviews to ascertain their own preferences for hospitals, but the collection of interviews that a student can schedule is limited. In the model that is studied in the current work, congestion in preference learning is captured by conflicts when two or more players attempt to pull the same arm.

Other models of uncertainty in two-sided matching that do not explicitly consider preference learning have also been studied. In this setting, there has been much interest in decentralized models. For example, Niederle and Yariv (2009) studied a decentralized mar-

ket game in which firms make directed offers to workers, agents have aligned preferences, and equilibrium outcomes under preference uncertainty are analyzed. Arnosti et al. (2014) employed mean field modeling to analyze the welfare costs of not knowing the availability of agents, as opposed to preferences. Ashlagi et al. (2019) considered providing match recommendations to participants in markets for which both sides of the market propose with some probability, and a successful match occurs only in the case of a mutual proposal. Dai and Jordan (2020) study a single-stage matching problem with uncertain preferences where players learn from historical data and act in a decentralized manner.

Lastly, the empirical aspects of stable matching in decentralized settings have also garnered significant research interest (Das and Kamenica, 2005; Echenique and Yariv, 2012; Pais et al., 2012).

9. Discussion

In this section, we discuss the strengths and limitations of Algorithm 1, in the context of broader themes in decentralized matching and multiplayer bandit learning. We also suggest future research directions motivated by our current findings.

Single-phase algorithm   One advantage of Algorithm 1 is its simplicity, specifically the fact that it does not involve separate phases or subroutines. Recent work by Sankararaman et al. (2020) studied an algorithm ('UCB-D3') for decentralized matching bandits, assuming *globally ranked players*, that proceeds in phases of exponentially increasing length; each phase comprises of a learning stage, where players choose arms according to their own UCBs, followed by a communication subroutine, where players broadcast their preferred arms to other players. In contrast, our algorithm does not require players to keep track of which phase they are in, or when to begin a subroutine. Not having separate algorithmic phases is desirable because multiple phases requires players to synchronize their transition from one phase to the next. In 'more decentralized' situations this may not be possible. For example, players may enter the market at different times, or leave the market for a number of rounds only to return later (see e.g., Akbarpour et al., 2020). The CA-UCB algorithm can be run in such cases without modification and is still guaranteed to have small regret.

Dependence of stable regret on market size   While both UCB-D3 and our method are guaranteed to achieve $\mathcal{O}(\log(T)/\Delta^2)$ stable regret for globally ranked players, the regret guarantee for UCB-D3 has a better dependence on the number of arms (which upper bounds the number of players). In the worst case, the guarantee on the regret of UCB-D3 depends on the square of the number of arms while the guarantee on the regret of our method depends on the cube of the number of arms. The optimal order-dependence on the rank k and the number of arms L is still an open question, since the lower bound (e.g., Corollary 6 in Sankararaman et al., 2020) and upper bounds currently do not match. Another interesting question is whether UCB-D3's better regret guarantee under these assumptions translates to better performance in practice; an in-depth empirical comparison of UCB-D3 and CA-UCB will be needed and is beyond the scope of the current work.

Random delays   Another important feature of Algorithm 1 is the injection of additional randomness through each player's independently drawn random delays. Randomization is key for this algorithm to achieve a $\mathcal{O}(\log(T)^2)$ regret guarantee in the case of *arbitrary*

two-sided preferences. Intuitively, the added randomness allows players to escape conflict cycles, as illustrated in Examples 1 and 2. Technically, it allows us to leverage a result from Abeledo and Rothblum (1995)) to show that the players must converge to a stable matching (which may not be unique), in a low-regret sense. Nevertheless, repeating one's previous action with a constant probability at every step could be considered wasteful. Are there other, more efficient ways of utilizing randomness as an implicit coordination mechanism than random delays?

Improving the stable regret under arbitrary preferences While Algorithm 1 is the first method to provably achieve polylogarithmic regret in markets with arbitrary preferences, we believe there is a significant room for the development of better algorithms. In particular, for markets with arbitrary preferences, the regret guarantee for our method depends exponentially on the number of players. This dependence arises because our regret analysis hinges on a reduction to the convergence rate of the corresponding randomized decentralized matching dynamics under *known* preferences. As shown in Ackermann et al. (2008) and Hoffman et al. (2013), existing randomized dynamics for decentralized matching under known preferences have worst-case convergence time that is exponential in the number of market participants. While this may suggest that there is indeed a real computational barrier in the arbitrary preferences setting, it might be possible to improve upon the exponential dependence by considering sub-classes of two-sided preferences or randomly drawn preferences. For example, Algorithm 1 has improved rates if we assume that the players are globally ranked.

It is also not clear that the $\mathcal{O}((\log T)^2)$ dependence on the horizon is optimal in this setting, even though it is unavoidable given our analysis strategy and our algorithm. Obtaining a regret bound that depends polynomially on the number of players and arms *and* has an optimal order dependence on the horizon may require a new algorithm.

Information available to players A player that implements CA-UCB must observe the successful arm pulls of all other players. On one hand, by leveraging this information our algorithm ensures that players avoid conflicts most of the time. On the other hand, it is not clear that such information is absolutely necessary for achieving sublinear regret in general markets. For example, UCB-D3 (Sankararaman et al., 2020), which achieves sublinear regret in the setting of globally ranked players, does not require players to see the actions of other players. However, players must participate in a rank estimation routine, which relies on the assumption that the players are ranked globally.

Conclusion and open questions In this work we have made progress on the problem of stochastic bandits in decentralized matching markets. Still, many open questions remain. We conclude by highlighting the most intriguing directions for future inquiry:

1. *Better algorithms and matching lower bounds.* Even though algorithms such as UCB-D3 (Sankararaman et al., 2020) and CA-UCB have stable regret that is almost order-optimal in the setting of globally ranked players, there is still a lot of room for improvement in the setting of arbitrary preferences. Is there a large class of preferences for which one can show matching upper and lower regret bounds, in terms of the dependence on the horizon, the reward gap, and size of the market?

2. *Incentive compatibility in the decentralized setting.* Unlike in the centralized setting, where a single algorithm was shown to be incentive compatible given any set of preferences (Liu et al., 2020), decentralization appears to pose more challenges for incentive compatibility. As seen in Section 6, the randomized conflict avoidance mechanism of Algorithm 1 can be strategically exploited by a deviating player when arm preferences are uncorrelated. How fundamental is this difficulty to the decentralized setting, and can it be overcome by a better algorithm?

Acknowledgements

References

Atila Abdulkadiroğlu and Tayfun Sönmez. School choice: A mechanism design approach. *American economic review*, 93(3):729–747, 2003.

Atila Abdulkadiroğlu, Parag Pathak, Alvin E Roth, and Tayfun Sonmez. Changing the boston school choice mechanism. Technical report, National Bureau of Economic Research, 2006.

Hernan Abeledo and Uriel G. Rothblum. Paths to marriage stability. *Discrete Applied Mathematics*, 63:1–12, 10 1995.

Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking. Uncoordinated two-sided matching markets. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 256–263, 2008.

Mohammad Akbarpour, Shengwu Li, and Shayan Oveis Gharan. Thickness and information in dynamic matching markets. *Journal of Political Economy*, 128(3):783–815, 2020.

A. Anandkumar, N. Michael, A. K. Tang, and A. Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4):731–745, 2011.

Guy Aridor, Kevin Liu, Aleksandrs Slivkins, and Zhiwei Steven Wu. Competing bandits: The perils of exploration under competition. *The 20th ACM Conference on Economics and Computation*, 2019.

Nick Arnosti, Ramesh Johari, and Yash Kanoria. Managing congestion in decentralized matching markets. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 451–451, 2014.

Itai Ashlagi, Mark Braverman, Yash Kanoria, and Peng Shi. Communication requirements and informative signaling in matching markets. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, pages 263–263, 2017a.

Itai Ashlagi, Yash Kanoria, and Jacob D Leshno. Unbalanced random matching markets: The stark effect of competition. *Journal of Political Economy*, 125(1):69–98, 2017b.

Itai Ashlagi, Anilesh K. Krishnaswamy, Rahul M. Makhijani, Daniela Sabán, and Kirankumar Shiragur. Assortment planning for two-sided sequential matching markets. *CoRR*, abs/1907.04485, 2019.

O. Avner and S. Mannor. Multi-user lax communications: A multi-armed bandit approach. In *The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, 2016.

Orly Avner and Shie Mannor. Concurrent bandits and cognitive radio networks. In Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 66–81, 2014.

Ilai Bistritz and Amir Leshem. Distributed multi-player bandits—A game of thrones approach. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7222–7232, 2018.

Ilai Bistritz, Tavor Z. Baharav, A. Leshem, and N. Bambos. My fair bandit: Distributed learning of max-min fairness with multi-player bandits. In *Proceedings of The 37th International Conference on Machine Learning*, 2020.

Etienne Boursier and Vianney Perchet. Selfish robustness and equilibria in multi-player bandits. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of the 33rd Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 530–581, 2020.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

Sébastien Bubeck, Thomas Budzinski, and Mark Sellke. Cooperative and stochastic multi-player multi-armed bandit: Optimal regret with neither communication nor collisions. *arXiv preprint arXiv:2011.03896*, 2020a.

Sébastien Bubeck, Yuanzhi Li, Yuval Peres, and Mark Sellke. Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without. In *Proceedings of the 33rd Conference on Learning Theory*, pages 961–987, 2020b.

Sarah H Cen and Devavrat Shah. Regret, stability, and fairness in matching markets with bandit learners. *arXiv preprint arXiv:2102.06246*, 2021.

Nicolò Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora. Delay and cooperation in nonstochastic bandits. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 605–622, 23–26 Jun 2016.

Xiaowu Dai and Michael I. Jordan. Learning strategies in decentralized matching markets under uncertain preferences. *arXiv preprint arXiv:2011.00159*, 2020.

S. J. Darak and M. K. Hanawal. Multi-player multi-armed bandits for stable allocation in heterogeneous ad-hoc networks. *IEEE Journal on Selected Areas in Communications*, 37 (10):2350–2363, 2019.

Sanmay Das and Emir Kamenica. Two-sided bandits and the dating market. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 947–952, 2005.

Federico Echenique and Leeat Yariv. An experimental study of decentralized matching. 2012.

Drew Fudenberg and David K Levine. *The theory of learning in games*, volume 2. MIT press, 1998.

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

Moshe Hoffman, Daniel Moeller, and Ramamohan Paturi. Jealousy graphs: Structure and complexity of decentralized stable matching. In *Web and Internet Economics*, pages 263–276, 2013.

Junling Hu, Michael P Wellman, et al. Multiagent reinforcement learning: theoretical framework and an algorithm. In *ICML*, volume 98, pages 242–250. Citeseer, 1998.

Nicole Immorlica, Jacob Leshno, Irene Lo, and Brendan Lucier. Information acquisition in matching markets: The role of price discovery. *Available at SSRN*, 2020.

Ramesh Johari, Vijay Kamble, and Yash Kanoria. Matching while learning. In *ACM Conference on Economics and Computation*, pages 119–119, 2017.

D. Kalathil, N. Nayyar, and R. Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, 2014.

Donald E. Knuth. *Stable Marriage and its Relation to Other Combinatorial Problems*. American Mathematical Society, 1997.

T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4 – 22, 1985.

Robin S Lee and Michael Schwarz. Interviewing in two-sided matching markets. Technical report, National Bureau of Economic Research, 2009.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):5667–5681, 2010.

Lydia T. Liu, Horia Mania, and Michael Jordan. Competing bandits in matching markets. In *International Conference on Artificial Intelligence and Statistics*, volume 108, pages 1618–1628, 26–28 Aug 2020.

Gábor Lugosi and Abbas Mehrabian. Multiplayer bandits without observing collision information. *arXiv preprint arXiv:1808.08416*, 2018.

Yishay Mansour, Aleksandrs Slivkins, and Zhiwei Steven Wu. Competing bandits: Learning under competition. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 48:1–48:27, 2018.

Muriel Niederle and Leeat Yariv. Decentralized matching with aligned preferences. Technical report, National Bureau of Economic Research, 2009.

Joana Pais, Agnes Pintér, and Robert F Veszteg. Decentralized matching markets: a laboratory experiment. 2012.

Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits—A musical chairs approach. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 155–163, 2016.

Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Econometric Society Monographs. Cambridge University Press, 1990.

Alvin E. Roth and John H. Vande Vate. Random paths to stability in two-sided matching. *Econometrica*, 58(6):1475–1480, 1990.

Abishek Sankararaman, Soumya Basu, and Karthik Abinav Sankararaman. Dominate or delete: Decentralized competing bandits with uniform valuation. *arXiv preprint arXiv:2006.15166*, 2020.

S. Shahrampour, A. Rakhlin, and A. Jadbabaie. Multi-armed bandits in multi-agent networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2786–2790, 2017.

Appendix A. Example 2

In this section, we present a second counterexample in which CA-UCB without random delays (i.e., $\lambda = 0$) would fail to converge to a stable matching and the players can enter into a conflict cycle. In this example, neither the arms nor the players are globally ranked. In contrast to Example 1, the type of coordination failure seen in Example 2 is unrelated to the failure of the players to learn their rewards. In fact, they can enter into such a cycle even after they have acquired perfect information on all the arms.

Example 2 (3-player market with non-unique stable matching) *Let the set of players be $\mathcal{N} = \{p_1, p_2, p_3\}$ and the set of arms be $\mathcal{K} = \{a_1, a_2, a_3\}$, with true preferences given by:*

$$p_1 : a_3 \succ a_2 \succ a_1 \qquad\qquad a_1 : p_3 \succ p_2 \succ p_1$$
$$p_2 : a_1 \succ a_3 \succ a_2 \qquad\qquad a_2 : p_1 \succ p_3 \succ p_2$$
$$p_3 : a_2 \succ a_1 \succ a_3 \qquad\qquad a_3 : p_2 \succ p_1 \succ p_3.$$

Then the conflict-avoiding algorithm cycles even when the preferences of the players are known. Suppose the players are following Algorithm 1, and their UCB rankings for the arms always coincide with their true preferences. The cycle it enters is as follows:

- *Time t: p_1 and p_3 conflict on a_2, p_1 wins.*

  p_2 pulls a_1.

- *Time $t+1$: p_3 attempts a_1 because a_2 is not in its plausible set. p_2 and p_3 conflict on a_1, p_3 wins.*

  p_1 pulls a_3 because a_3 was not pulled by any player at time t.

- *Time $t+2$: p_2 attempts a_3 because a_1 is not in its plausible set. p_1 and p_2 conflict on a_3, p_2 wins.*

  p_3 pulls a_2 because a_2 was not pulled by any player at time $t+1$

At time $t+3$, the players attempt the same actions as they did at time t, entering into a cycle where there is a conflict at every round henceforth.

Previous work has found other examples where sequentially resolving blocking pairs in an unstable matching leads to cycling (Knuth, 1997; Roth and Vande Vate, 1990; Abeledo and Rothblum, 1995). Example 2 shows that players following the decentralized conflict-avoiding protocol (where more than one blocking pair may be resolved at every time step) can also enter into cycles.

These examples highlight the failure modes of decentralized conflict-avoiding algorithms. One way to escape these failure modes is by introducing randomness, such that the probability of coordination failures becomes exponentially small. This is the motivation for incorporating random delays into Algorithm 1.

Appendix B. Proof of Lemma 5

Proof Our proof is essentially identical to the single-agent UCB analysis in Section 2.2 of Bubeck and Cesa-Bianchi (2012). Assuming that the event

$$F_{j,k}^{(i)}(t) = \left\{ \text{UCB}_j^{(i)}(t) > \text{UCB}_k^{(i)}(t) \right\}$$

is true, then at least one of the three following events must occur:

$$\mathcal{E}_1(t) = \left\{ \widehat{\mu}_j^{(i)}(t) > \mu_j^{(i)} + \sqrt{\frac{3\log t}{2\bar{T}_j^{(i)}(t)}} \right\},$$

$$\mathcal{E}_2(t) = \left\{ \widehat{\mu}_k^{(i)}(t) + \sqrt{\frac{3\log t}{2\bar{T}_k^{(i)}(t)}} < \mu_k^{(i)} \right\},$$

$$\mathcal{E}_3(t) = \left\{ \bar{T}_j^{(i)}(t) \leq \frac{6}{\Delta^2}\log(t) \right\}.$$

To see this, suppose that none of three events $\mathcal{E}_1(t)$, $\mathcal{E}_2(t)$ and $\mathcal{E}_3(t)$ occur. Then,

$$\widehat{\mu}_k^{(i)}(t) + \sqrt{\frac{3\log(t)}{2\bar{T}_k^{(i)}(t)}} \geq \mu_k^{(i)} \geq \mu_j^{(i)} + \Delta \geq \mu_j^{(i)} + \sqrt{\frac{6\log(t)}{\bar{T}_j^{(i)}(t)}}$$

$$\geq \widehat{\mu}_j^{(i)}(t) + \sqrt{\frac{3\log(t)}{2\bar{T}_j^{(i)}(t)}}$$

which is a contradiction because the left-hand side equals $\text{UCB}_k^{(i)}(t)$ and the right-hand side equals $\text{UCB}_j^{(i)}(t)$.

Let $u > 0$ be some value to be chosen later. Then, we have

$$\sum_{t=1}^{T} \mathbf{1}\left\{ F_{j,k}^{(i)}(t) \cap \bar{A}^{(i)}(t) = j \right\} = \sum_{t=1}^{T} \mathbf{1}\left\{ F_{j,k}^{(i)}(t) \cap \bar{A}^{(i)}(t) = j \cap \bar{T}_j^{(i)}(t) \leq u \right\}$$

$$+ \sum_{t=1}^{T} \mathbf{1}\left\{ F_{j,k}^{(i)}(t) \cap \bar{A}^{(i)}(t) = j \cap \bar{T}_j^{(i)}(t) > u \right\}.$$

Therefore, if we choose $u = \frac{6}{\Delta^2}\log(t)$, we obtain

$$\sum_{t=1}^{T} \mathbf{1}\left\{ F_{j,k}^{(i)}(t) \cap \bar{A}^{(i)}(t) = j \right\} = u + \sum_{t=u+1}^{T} \mathbf{1}\left\{ F_{j,k}^{(i)}(t) \cap \bar{A}^{(i)}(t) = j \cap \bar{T}_j^{(i)}(t) > u \right\}$$

$$\leq u + \sum_{t=\lfloor u \rfloor + 1}^{T} \mathbf{1}\left\{ \mathcal{E}_1(t) \right\} + \sum_{t=\lfloor u \rfloor + 1}^{T} \mathbf{1}\left\{ \mathcal{E}_2(t) \right\}.$$

We are left to establish an upper bound on $\mathbb{P}(\mathcal{E}_1(t))$ and $\mathbb{P}(\mathcal{E}_2(t))$. We can do this by a simple application of a union bound and concentration:

$$
\begin{aligned}
\mathbb{P}(\mathcal{E}_1(t)) &\leq \mathbb{P}\left(\exists s \in \{1, 2, \dots t\} \colon \widehat{\mu}_j^{(i)}(s) + \sqrt{\frac{3 \log(t)}{2s}} \leq \mu_j^{(i)} \right) \\
&\leq \sum_{s=1}^{t} \mathbb{P}\left(\widehat{\mu}_j^{(i)}(s) + \sqrt{\frac{3 \log(t)}{2s}} \leq \mu_j^{(i)} \right) \\
&\leq \sum_{s=1}^{t} \frac{1}{t^3} = \frac{1}{t^2},
\end{aligned}
$$

where the last inequality follows by a standard concentration argument for independent sub-Gaussian random variables. The probability of $\mathcal{E}_2(t)$ occurring can be upper bounded similarly. Then, using $\sum_{t=1}^{\infty} t^2 = \frac{\pi^2}{6}$ yields the conclusion. ∎

Appendix C. Proof of Proposition 16

Proof By definition, player p_k's regret can be lower-bounded as follows:

$$
R_k(T) \geq \sum_{i : a_i \succ_{p_k} a_k} \Delta_i^{(k)} \cdot \mathbb{E}\left[\bar{T}_i^{(k)}(T) \right] \geq \left(\min_{i : \Delta_i^{(k)} < 0} \Delta_i^{(k)} \right) \cdot \sum_{i : a_i \succ_{p_k} a_k} \mathbb{E}\left[\bar{T}_i^{(k)}(T) \right]. \tag{31}
$$

Since $a_i \succ_{p_k} a_k$ implies that $i < k$, we may proceed to upper bound $\sum_{i : i < k} \mathbb{E}\left[\bar{T}_i^{(k)}(T) \right]$. We claim that the following inclusion is true:

$$
\bigcup_{i=1}^{k-1} \left\{ \bar{A}^{(k)}(t) = a_i \right\} \subseteq \bigcup_{\substack{1 \leq j < k \\ j < l \leq L}} \bigcup_{t-k \leq t' \leq t} \Lambda_l^{(j)}[t']. \tag{32}
$$

The argument is as follows. If $\left\{ \bar{A}^{(k)}(t) = a_i \right\}$ holds for some i, we know that p_i at time t did not attempt to pull a_i. They either attempted to pull an arm $a_{i'}$ with $i' > i$ or with $i' < i$. Since we know that p_i is running Algorithm 1, in the former case, we can apply Lemma 3 to player p_i; in the latter case, we can apply equation (11), also to player p_i. This establishes equation (32).

Since all players p_j with $j < k$ are running Algorithm 1, we may apply Lemma 5 to yield

$$
\sum_{i : i < k} \mathbb{E}\left[\bar{T}_i^{(k)}(T) \right] \leq 6(k+1) \sum_{\substack{1 \leq j < k \\ j < l \leq L}} \left(\frac{\log T}{|\Delta_l^{(j)}|^2} + 1 \right) \leq 6k^2(L-k)\left(\frac{\log T}{\Delta^2} + 1 \right). \tag{33}
$$

Substituting the above into (31) yields the desired lower bound. ∎

Appendix D. Proof of Proposition 17

Example 3 *Let the set of players be $\mathcal{N} = \{p_1, p_2, p_3\}$ and the set of arms be $\mathcal{K} = \{a_1, a_2, a_3\}$, with true preferences given by:*

$$p_1 : a_1 \succ a_3 \succ a_2 \qquad\qquad a_1 : p_2 \succ p_1 \succ p_3$$
$$p_2 : a_2 \succ a_1 \succ a_3 \qquad\qquad a_2 : p_3 \succ p_2 \succ p_1$$
$$p_3 : a_1 \succ a_3 \succ a_2 \qquad\qquad a_3 : p_3 \succ p_1 \succ p_2.$$

The unique stable matching in this case is $(p_1, a_1), (p_2, a_2), (p_3, a_3)$.

Proof Let $D^{(3)}(t) \overset{i.i.d.}{\sim} Ber(\lambda)$ for any t. The set of actions that player p_3 can play, for $t = 1, \cdots, T$, to get negative stable regret is as follows:

$$A^{(3)}(t) = \begin{cases} a_2 & \text{if } t = 3m - 2 \\ a_3 & \text{if } t = 3m - 1 \text{ and } D^{(3)}(t) = 0 \\ a_2 & \text{if } t = 3m - 1 \text{ and } D^{(3)}(t) = 1 \\ a_1 & \text{if } t = 3m \end{cases}, \quad \text{for } m \in \mathbb{N}. \tag{34}$$

By the definition of p_3's regret, and using the fact that $\Delta_\emptyset^{(k)} > \max\{\Delta_1^{(3)}, \Delta_2^{(3)}\} > 0$, we have:

$$R_3(T) \le \Delta_1^{(3)} \cdot \mathbb{E}[\bar{T}_1^{(3)}(T)] + \Delta_\emptyset^{(3)} \cdot \left(T - \mathbb{E}[\bar{T}_1^{(3)}(T)]\right). \tag{35}$$

Thus it suffices to lower bound the expected number of times that p_3 successfully attempts a_1.

Define the following events:

$$\Omega_t^1 := \{F_{3,1}^{(2)}(t) \cap A^{(2)}(t) = a_3\}^c,$$
$$\Omega_t^2 := \{F_{1,2}^{(2)}(t) \cap A^{(2)}(t) = a_1\}^c.$$

We first show the following inclusion, for any $m \in \mathbb{N}$:

$$\Omega_{3m-1}^1 \cap \Omega_{3m}^2 \cap \{D^{(2)}(3m-1) = D^{(3)}(3m-1) = D^{(2)}(3m) = D^{(1)}(3m) = 0\} \subseteq \{\bar{A}^{(3)}(3m) = a_1\}. \tag{36}$$

We can simply check that this holds:

- At time $3m - 2$, p_3 attempts and successfully pulls a_2.

- At time $3m - 1$, p_2 pulls a_1, since a_2 is not in its plausible set, $D^{(2)}(3m - 1) = 0$ and the event Ω_{3m-1}^1 holds. p_3 pulls a_3, since $D^{(3)}(3m - 1) = 0$.

- At time $3m$, p_1 does not pull a_1, since a_1 is not in its plausible set and $D^{(1)}(3m) = 0$. p_2 does not pull a_1, since a_2 is in its plausible set, $D^{(2)}(3m) = 0$ and the event Ω_{3m}^2 holds. Thus p_3 successfully pulls a_1.

Taking expectation of (36) and rearranging gives

$$\mathbb{P}\left(\bar{A}^{(3)}(3m) = a_1\right)$$

$$\geq 1 - \mathbb{P}\left(\left(\Omega^1_{3m-1} \cap \Omega^2_{3m} \cap \{D^{(2)}(3m-1) = D^{(3)}(3m-1) = D^{(2)}(3m) = D^{(1)}(3m) = 0\}\right)^c\right)$$

$$\geq 1 - \left(\mathbb{P}((\Omega^1_{3m-1})^c) + \mathbb{P}((\Omega^2_{3m})^c) + 4\lambda\right), \tag{37}$$

where the last inequality follows from a union bound.

It is useful to upper bound the following:

$$\sum_{m=1}^{\lfloor T/3 \rfloor} \mathbb{P}((\Omega^1_{3m-1})^c) + \mathbb{P}((\Omega^2_{3m})^c)$$

$$= \sum_{m=1}^{\lfloor T/3 \rfloor} \mathbb{P}\left(F^{(2)}_{3,1}(3m-1) \cap A^{(2)}(3m-1) = a_3\right) + \mathbb{P}\left(F^{(2)}_{1,2}(3m) \cap A^{(2)}(3m) = a_1\right)$$

$$\leq \frac{1}{\lambda(1-\lambda)} \sum_{m=1}^{\lfloor T/3 \rfloor} \mathbb{P}\left(F^{(2)}_{3,1}(3m-1) \cap \bar{A}^{(2)}(3m-1) = a_3\right) + \mathbb{P}\left(F^{(2)}_{1,2}(3m) \cap \bar{A}^{(2)}(3m) = a_1\right)$$

$$\leq \frac{1}{\lambda(1-\lambda)} \sum_{t=1}^{T} \mathbb{P}\left(F^{(2)}_{3,1}(t) \cap \bar{A}^{(2)}(t) = a_3\right) + \mathbb{P}\left(F^{(2)}_{1,2}(t) \cap \bar{A}^{(2)}(t) = a_1\right)$$

$$\leq \frac{1}{\lambda(1-\lambda)} \cdot 12 \cdot \left(\frac{1}{\Delta^2} \log(T) + 1\right), \tag{38}$$

where the last inequality follows from Lemma 5.

Now, we sum (37) over $m = 1, \cdots, \lfloor T/3 \rfloor$ to get:

$$\mathbb{E}[\bar{T}^{(3)}_1(T)] \geq \sum_{m=1}^{\lfloor T/3 \rfloor} \mathbb{P}\left(\bar{A}^{(3)}(3m) = a_1\right)$$

$$\geq (1 - 4\lambda) \cdot \lfloor T/3 \rfloor - \sum_{m=1}^{\lfloor T/3 \rfloor} \mathbb{P}((\Omega^1_{3m-1})^c) + \mathbb{P}((\Omega^2_{3m})^c)$$

$$\geq (1 - 4\lambda) \cdot \lfloor T/3 \rfloor - \frac{1}{\lambda(1-\lambda)} \cdot 12 \cdot \left(\frac{1}{\Delta^2} \log(T) + 1\right),$$

where the last two inequalities follow from Equation (37) and Equation (38).

Thus we have

$$R_3(T) \leq \Delta^{(3)}_1 \cdot \left((1 - 4\lambda) \cdot \lfloor T/3 \rfloor - \frac{1}{\lambda(1-\lambda)} \cdot 12 \cdot \left(\frac{1}{\Delta^2} \log(T) + 1\right)\right) + \Delta^{(3)}_\emptyset \cdot \left(\frac{2}{3}T\right). \tag{39}$$

Note that $\Delta^{(3)}_1 < 0$, and $1 - 4\lambda > 0$ by assumption. Upon rearranging terms, we get the desired result.

■