Competing Bandits in Time Varying Matching Markets

Deepan Muthirayan   DEEPAN.M@UCI.EDU
University of California, Irvine

Chinmay Maheshwari   CHINMAY_MAHESHWARI@BERKELEY.EDU
University of California, Berkeley

Pramod P. Khargonekar   PRAMOD.KHARGONEKAR@UCI.EDU
University of California, Irvine

Shankar Sastry   SASTRY@COE.BERKELEY.EDU
University of California, Berkeley

Abstract

We study the problem of online learning in two-sided non-stationary matching markets, where the objective is to converge to a stable match. In particular, we consider the setting where one side of the market, the arms, has fixed known set of preferences over the other side, the players. While this problem has been studied when the players have fixed but unknown preferences, in this work we study the problem of how to learn when the preferences of the players are time varying and unknown. Our contribution is a methodology that can handle any type of preference structure and variation scenario. We show that, with the proposed algorithm, each player receives a uniform sub-linear regret of $\widetilde{\mathcal{O}}(L_T^{1/2}T^{1/2})$ up to the number of changes in the underlying preferences of the agents, L_T. Therefore, we show that the optimal rates for single-agent learning can be achieved in spite of the competition up to a difference of a constant factor. We also discuss extensions of this algorithm to the case where the number of changes need not be known a priori.

Keywords: Matching Markets, Non-stationary multi-armed bandits.

1. INTRODUCTION

Matching market design has been influential and instrumental in many markets such as kidney markets, school exchange markets, labor markets etc. Matching markets are essentially two-sided markets where participants on either side of the market have preferences over the other side. The typical goal in a matching market is to match the players on one side to the players on the other side with desirable properties such as stability, maximality, efficiency, and strategy proofness etc. Several papers have been published exploring the various properties such as stability (Gale and Shapley (1962); Knuth (1997)), maximality (Edmonds (1965)), strategy proofness (Roth (1982)), and several applications such as housing allocation Roth (1984), content delivery Maggs and Sitaraman (2015), kidney exchanges Roth et al. (2005), control applications such as Transportation (Lowalekar et al., 2018, Ride Sharing Platforms), Smart Grids (Muthirayan et al., 2020, Peer-to-Peer Markets), etc. Thus, matching markets have significant relevance and importance.

In spite of the many advances in matching markets, majority of these studies ignore the fact that, in many practical circumstances, the participants of the markets may not know their underlying preferences. In addition, the participants may only be able to observe the outcome after a match

and that too with high statistical uncertainty. Therefore, the participants will have to learn their preferences with the added complexity that their exploration can be hampered by the other agents who are competing over the same resources. Thus, in such circumstances, the learning will have to be coordinated in a way that all the participants can uncover their underlying preferences and the market can reach its desired equilibrium.

Liu et al. (2020) introduced the competing bandits framework to study the interaction of learning and competition in the context of two-sided online matching markets settings. The competing bandits framework Liu et al. (2020) is inspired by many modern matching markets where the markets are two-sided and there are multiple rounds of interactions. In this framework, in each round, the participants submit their preferences and the market is cleared by the submitted preferences. After the market is cleared or the matching is made, the participants get to observe the final outcome or the reward from their matching, which can be prone to statistical uncertainty. Thus, the feedback the participants get is like a bandit type feedback. The players need to learn their preferences over the other side of the market from just the bandit feedback, with the added complexity that what they get to explore is also impacted by the other participants. Thus, the competing bandits allows us to study the interaction of learning, where multiple participants are simultaneously learning and competing. The goal is to design an algorithm that can enable the participants to learn their underlying preferences and enable the market to reach a stable match in spite of the complex interplay of competition and learning.

Contribution: We study an extension of the competing bandits problem where the underlying unknown preferences of the participants can be time varying. While Liu et al. (2020) study the competing bandits framework, they assume that the participants' preferences are fixed in time. This may not be realistic, given that the participants' preferences are not likely to be stationary. In this work, we present algorithms and guarantees for the scenario where the participants' preferences can be time variant. While Ghosh et al. (2022) study the same problem for a serial dictatorship preference structure and smoothly varying preferences, *the methodology we propose can handle any type of preference structure and any type of variation, and is also very efficient at the same time*. This is our key contribution to this line of work. We show that, with our approach, each agent can incur a regret of $\widetilde{O}(L_T^{1/2} T^{1/2})$, where T is the number of rounds and L_T is the number of changes in the preference ordering over the T rounds across all the participants, matching the optimal regret rates for single-agent bandit learning (Auer et al., 2019).

2. RELATED WORKS

This section discusses the literature on time-varying bandits and learning in matching markets.

Time Varying Multi-armed Bandits. Non-stationary Multi-Armed Bandits (MAB) problem was first studied by Auer et al. (2002). They proposed a variant of the EXP3 algorithm called EXP3.S and showed that EXP3.S can achieve Minimax optimal regret $\mathcal{O}(\sqrt{KLT})$ up to logarithmic factors, where K is the number of arms, L is the number of abrupt changes and T is the number of iterations. There were other studies such as Garivier and Moulines (2011) and Allesiardo et al. (2017) that demonstrated that a sliding window approach or a restart approach can also achieve the Minimax optimal regret. Recently, Auer et al. (2019) proposed a near-optimal regret algorithm for the same problem that does not need the knowledge of the number of changes. Besbes et al. (2019); Krishnamurthy and Gopalan (2021) studied the non-stationary MAB with slowly changing variations,

and showed that a regret of $\widetilde{O}\left(P_T^{1/3}T^{2/3}\right)$ can be achieved, where P_T denotes the total variation. The non-stationary linear bandits problem was studied by Cheung et al. (2019); Russac et al. (2019); Zhao et al. (2020). These works proposed approaches like sliding window least-squares, weighted UCB and restart strategy respectively and showed that a regret of $\widetilde{\mathcal{O}}\left(P_T^{1/4}T^{3/4}\right)$ can be achieved by all these approaches.

Learning in Two-sided Matching Markets. Learning in the context of two-sided matching markets is a new and active research area (Das and Kamenica, 2005; Liu et al., 2020, 2021; Jagadeesan et al., 2021; Maheshwari et al., 2022; Basu et al., 2021; Cen and Shah, 2022; Dai and Jordan, 2021). Das and Kamenica (2005) was the first work which employed multi-armed bandit algorithms to learn preferences in matching markets. However, it was only recently that Liu et al. (2020) formulated the problem more formally. One can classify the existing literature in this space into two classes. First is the setting where the true underlying preferences of the players over the firms are fixed (Liu et al., 2020; Jagadeesan et al., 2021; Cen and Shah, 2022; Basu et al., 2021; Sankararaman et al., 2021; Maheshwari et al., 2022; Kong et al., 2022), and second is the setting where the underlying preferences can change with time (Min et al., 2022; Ghosh et al., 2022). Interestingly, the literature on learning in time-varying matching markets is relatively sparse (Min et al., 2022; Ghosh et al., 2022). Min et al. (2022) consider the setting of *Markov Matching Market* where the preferences of agents depends on some underlying context which changes based on planner's policy in a Markovian manner. Meanwhile, in Ghosh et al. (2022), the authors consider the problem of learning in *smoothly varying* matching markets where the underlying preferences of agents are allowed to change by a certain threshold at every time step. Although the authors consider learning in a decentralized context, the preference structure they study is restricted to a serial dictatorship.

3. PROBLEM FORMULATION

Consider a two-sided matching market comprised of *players* and *arms*, to be referred respectively as the two sides, who gain some positive utility if matched with one another. More formally, we denote the set of N players by $\mathcal{N} = \{p_1, p_2, \ldots, p_N\}$ and the set of K arms by $\mathcal{K} = \{a_1, a_2, \ldots, a_K\}$. An important feature of two-sided matching market is that each side of the market has preference over the other side. Preference of any arm $a_j \in \mathcal{K}$ over players is captured by the utility $\pi_j \in \mathbb{R}_+^N$, where $\pi_j(i)$ is the utility derived by arm a_j on getting matched with player p_i. Analogously, preference of a player $p_i \in \mathcal{N}$ over arms is captured by utility $\mu_i \in \mathbb{R}_+^K$, where $\mu_i(k)$ denotes the utility derived by player p_i on getting matched with arm a_k.

For the sake of concise notation, if arm a_j prefers player p_i over $p_{i'}$ then we represent it as $p_{i'} \succ_j p_i$. Similarly, if player p_i prefers arm a_k over $a_{k'}$ then we represent it as $a_k \succ_i a_{k'}$.

In what follows, we first describe the framework of two-sided matching market and state relevant background in Section 3.1. Following which, in Section 3.2, we describe the problem formulation for learning in two-sided time-varying matching markets.

3.1. Preliminaries on Matching Markets

In order to formally present the setup in this paper, we recall some relevant concepts on the literature on two-sided matching markets. A matching $m : \mathcal{N} \to \mathcal{K}$ is an injective map such that $m(p) = a$ denotes that player $p \in \mathcal{N}$ is matched with arm $a \in \mathcal{K}$.

Definition 1 (Blocking Pair) *We say a tuple* $(p_i, a_j) \in \mathcal{N} \times \mathcal{K}$ *is a blocking pair for a matching* m *if player* p_i *is matched to an arm* $a_{j'} = m(p_i)$ *but* $a_j \succ_i a_{j'}$ *and* a_j *is either unmatched or* $p_i \succ_j m^{-1}(a_j)$. *We say that the triplet* $(p_i, a_j, a_{j'})$ *blocks the matching* m.

Having defined the notion of blocking pair, we are now ready to define stable matching.

Definition 2 (Stable matching) *A matching* m *is called* stable *if there is no blocking pair. Alternatively, a matching is called* unstable *if there exists atleast one blocking pair for it.*

Gale and Shapley 1962 proposed a polynomial time algorithm – referred as Deferred-Acceptance (DA) algorithm – to find a stable matching. Without enforcing any specific assumptions on the underlying preference structure, the stable matching is not unique. Therefore, we define the notion of valid partners of a player which captures the set of all arm to which it can match in some stable matching.

Definition 3 (Valid partner) *Given the full preference rankings of arms and players, we call arm* a_j *to be a valid partner of player* p_i *if there exists a stable matching* $\boldsymbol{m}$ *such that* $\boldsymbol{m}(p_i) = a_j$.

We now define two important types of matching which are very crucial for subsequent exposition.

Definition 4 (Optimal matching and pessimal matching) *We say a matching* $\overline{\boldsymbol{m}}$ *to be* optimal *matching if every player is matched to its most preferred valid partner. Similarly we say a matching* $\underline{\boldsymbol{m}}$ *to be* pessimal *matching if every player is matched to its least preferred valid partner.*

In order to find optimal and pessimal matching one can initialize the Deferred-Acceptance algorithm from players side and the arms side respectively. Note that if there is a unique stable matching then the optimal and pessimal matching coincide. Interestingly, in Karpov (2019) the authors provide necessary and sufficient conditions on underlying preference structure which ensure unique stable matching.

3.2. Learning in Time Varying Matching Markets

While a stable matching can be identified directly by employing Deferred-Acceptance algorithm when the players know their preferences, in many modern matching markets the players can be unaware of their preferences. Furthermore, the underlying preferences can be time-varying. In this paper, we study the problem of bandit learning in time varying matching markets, where the objective is to find a stable match by repeated interactions. As an example, consider the setting of online labor markets where the two sides of the market are employers (players) and freelancers (arms) respectively. Due to the scale of system, the employers do not know apriori the quality of work by a freelancer and has to repeatedly interact with it. Furthermore, the inherent quality of work by a freelancer could change in a nonstationary manner due to some health reasons, personal problems etc. Consequently the employers have to adapt to this nonstationary change while deciding its preference estimates.

We formulate the repeated market setting as follows. We assume that the preferences of arms are *fixed* and is *common knowledge*. However, the preferences of players are *unknown* and *time-varying*. The players and arms interact with each other for a total of T rounds, indexed by t. At any time t, the true underlying preference of any player $p_i \in \mathcal{N}$ over any arm $a_j \in \mathcal{K}$ is encapsulated by mean reward of its interaction, denoted by $\mu_{i,t}(j)$, which is unknown. The players repeatedly interact with arms through a platform to learn the underlying utilities.

At every round t, the players submit their *estimate* of their preferences over the arms to a platform based on past rounds of interaction. The platform then computes a stable matching m_t based on the submitted preferences and assigns the players to the arms accordingly. Upon being assigned or matched an arm $m_t(i)$, the player p_i pulls the arm $m_t(i)$ and receives a stochastic reward X_{i,m_t} sampled from a 1-subGaussian distribution with mean $\mu_{i,t}(m_t(i))$. The players use the observed reward to update their estimate of preferences over the arms.

In order to evaluate the performance of any online learning algorithm this setup, we introduce the relevant notion of *regret*. Unlike the single player learning setting, in a two-sided matching market, all of the players cannot be assigned their most preferred arm due to the misaligned preference structure of the different players and arms. With this consideration, Liu et al. (2020) proposed a regret metric for the stationary preference setting which we extend here to non-stationary setting. This notion of regret, termed as *arm-stable regret* with respect to a stable matching $\mathbf{m} = (\mathbf{m}_t)_{t\in[T]}$ corresponding to true underlying preferences, is given by

$$R_T^i(\mathbf{m}) = \sum_{t=1}^{T} \mu_{i,t}(\mathbf{m}_t(i)) - \sum_{t=1}^{T} \mu_{i,t}(m_t(i)), \tag{1}$$

where $\mu_{i,t}(\mathbf{m}_t(i))$ is the mean reward of the arm that would be assigned to a player p_i in a stable matching outcome if the players know the true underlying preference. In any two-sided matching market there are two important matchings: *player-optimal* and *player-pessimal* matching. Consequently, there are two associated regret metrics: *player-optimal regret*, which is the regret with respect to $\overline{\mathbf{m}}$ optimal matching, and the other notion is the *player-pessimal regret*, which is the regret with respect to $\underline{\mathbf{m}}$ the pessimal match. As pointed out in Liu et al. (2020), it is not possible to achieve sub-linear player-optimal regret even in static environment without imposing any assumptions on the underlying preference structure. Therefore, we adopt the *player-pessimal regret* as the metric of performance.

In order to quantify the non-stationarity in an environment, we introduce the notion of time-variation which capture the variability in the underlying true preferences. The time-variation in a single player bandit learning is typically specified by a quantity called the total variation Besbes et al. (2019) or the total number of changes Auer et al. (2019). The nature of the interdependence in the matching setting is such the change in other players' preferences can lead to lower reward than pessimal match often even sans changes to its own preferences. Such occurrences are not bounded by the extent of variation and are only dependent on occurrence of changes in the order of player's preferences. Therefore, we adopt the notion of number of changes as the variation measure for the competing bandits setting. Specifically, we define the variation measure as the total number of changes across all the players:

$$L_T = \sum_{t=2}^{T} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{K}} \mathbb{I}[\mu_{i,t}(j) \neq \mu_{i,t-1}(j)], \tag{2}$$

where $\mathbb{I}[\cdot]$ is the standard indicator function. Such a definition accounts for all the variations in the market and its impact uniformly across all the players. Naturally, if the player's preferences are fixed then $L_T = 0$.

In any realistic scenario the preferences does not change arbitrarily over time. Furthermore, it is required to assume certain structure on the underlying preference structure to get any mean-

ingful quantitative guarantees. Against this backdrop, we make the following assumption on the preferences of players.

Assumption 1 *We make the following assumptions on the mean reward of arms*

(i) The mean reward of arms are bounded. That is $\mu_{i,t}(k) \leq \bar{\mu}$ for all $t \in [T], i \in [N], k \in [K]$;

(ii) The gap between the arms for any player is always greater than Δ. That is,
$0 < \Delta = \min_t \min_{i,k,k'} |\mu_{i,t}(k) - \mu_{i,t}(k')|$

Some remarks about Assumption 1 are in order. Assumption 1-(i) and 1-(ii) are required to obtain any meaningful regret guarantees for competing bandits in non-stationary matching markets with no structure on preferences. Furthermore, we argue that these are practical assumptions on the reward obtained in any real-world application. Our focus here is on changing preferences on which we do not impose any restriction at all. We note that we do not require any assumption on the magnitude of instantaneous change on the preferences of any player, as it typically assumed in many prior works on non-stationary bandits Krishnamurthy and Gopalan (2021); Garivier and Moulines (2011). Furthermore, the change in underlying true preferences of players can occur asynchronously.

4. ALGORITHM AND RESULTS

There are three key challenges associated with designing effective algorithm that works in this setting: uncertainty, competition and non-stationarity. Overcoming each of these challenges individually has been studied extensively in the existing literature. However, the key challenge is to develop a provable effective algorithm that can overcome these challenges in a holistic manner. Particularly, any single player cannot explore independently in order learn its preferences because the arm it gets matched with is decided by submitted preferences of all players who are also exploring different arms simultaneously. Furthermore, this challenge gets exacerbated by inherent non-stationarity in the environment as the change in preference of any single player can effect the stable match. Interestingly, we show that the proposed algorithmic design, to be presented, handles this challenge while being able to achieve low player pessimal regret.

4.1. Algorithmic Description

Our algorithm is an extension of the bandit learning algorithm proposed by Liu et al. (2020) for the stationary setting, where players repeatedly interact with the market platform by submitting a preference order over the arms. After receiving the preferences, the platform assigns players to arms as per a stable matching computed using the DA algorithm on the set of preferences submitted by the players. Upon getting matched to an arm, the players receive a reward, which they then use to update their preferences before the next round. The key idea in Liu et al. (2020) is how the players compute their preferences, which enables them to efficiently explore the arms and let the market converge to the stable outcome in spite of the competition among the players. Specifically, each player submits the preference order as per the Upper Confidence Bound (UCB) for the mean reward of the arms, computed using the collected reward from previous time steps.

We extend this algorithm to the time varying setting by adopting a restart strategy. Restart strategy is a widely used strategy for time varying bandit learning (Zhao et al., 2020). The idea in

restart strategy is to restart a base algorithm (such as UCB algorithm) after a certain period H. If the number of changes L_T is known, then the restart period can be chosen optimally a priori. We adopt this simple strategy to handle the time variations in the competing bandits case. Specifically, in the algorithm we propose, *the market platform ensures that players restart the local UCB algorithm after every H number of rounds and between two restarts the players run their local UCB algorithms just as in the stationary case Liu et al. (2020)*. The complete algorithm is shown in Algorithm 1.

Algorithm 1: Restart Competing Bandits (RCB) Algorithm

Input: Common Restart Period: H
Set $k = 1$
for $t = \{1, \ldots, T\}$ **do**
    If $k = 1$, set $T_{i,t}(j) = 0$ for all $j \in [K], i \in [N]$
    Players update the UCB, $(u_{i,t}(j))_{j \in K}$ for their respective arms using (3).
    Players submit their rank ordering $\hat{r}_{i,t}$ computed according to $u_{i,t}(j)$ using (4).
    Players are matched to arms as per optimal stable matching m_t computed using $\hat{r}_{i,t}$ and π_j.
    For all $i \in [N]$, update $T_{i,t+1}(m_t(i)) = T_{i,t}(m_t(i)) + 1$
    If $k \leq H$ update $k = k + 1$, else update $k = 1$
end

We now introduce the algorithm more formally. Let $T_{i,t}(j)$ denote the number of times player p_i is matched with arm a_j by the platform after the latest restart. After every restart, $T_{i,t}(j)$ is assigned to zero, and $T_{i,t}(j)$ is incremented by one whenever player p_i is matched with arm a_j. Let $\hat{\mu}_{i,t}(j)$ denote the mean of the rewards observed whenever player p_i pulls arm a_j after a restart and till $t - 1$. Let S_t denote the latest restart before time step t. Then, $\hat{\mu}_{i,t}(j)$ is given by $\hat{\mu}_{i,t}(j) = \frac{\sum_{k=S_t}^{t-1} X_{i,m_k} \mathbb{I}[m_k(i)=j]}{T_{i,t-1}(j)}$. Then, for a player p_i, the upper confidence bound for arm a_j in the current period is given by

$$u_{i,t}(j) = \begin{cases} \infty & T_{i,t}(j) = 0 \\ \hat{\mu}_{i,t}(j) + \sqrt{\frac{3 \log(t - S_t)}{2 T_{i,t-1}(j)}} & \text{Otherwise.} \end{cases} \tag{3}$$

At the beginning of every round t, each player updates its UCB for the arms as in Eq. (3). The players then compute their respective rank ordering $\hat{r}_{i,t}$ according to the updated UCBs. Specifically, the rank ordering $\hat{r}_{i,t}$ is computed as follows: for any two arms $a_j, a_{j'}$

$$\hat{r}_{i,t}(j) > \hat{r}_{i,t}(j') \text{ if } u_{i,t}(j) > u_{i,t}(j'). \tag{4}$$

The platform receives these rank orderings from every player, computes a stable matching, and assigns the matched arms to the respective players. This completes a round.

Algorithm 1 has several important features: first, it is a simple algorithm which us computationally efficient and intuitive. Second, the algorithm does not require storing exact rewards obtained in past rounds at the end of both players and platforms. Third, the preferences of players are updated only using local information which is a crucial aspect to ensure privacy and reliability of the system.

4.2. Regret Result

Below, we characterize the regret accrued by a player while using the RCB algorithm (Algorithm 1) under time-varying preferences.

Theorem 5 *Suppose Assumption 1 holds. Under RCB algorithm (Algorithm 1), with the common restart period $H = L_T^{-1/2} T^{1/2}$, the pessimal regret for each player i is given as*

$$R_T^i = \widetilde{\mathcal{O}} \left(L_T^{1/2} T^{1/2} \left(1 + \frac{1}{\Delta^2} \right) \right).$$

Some comments about Theorem 5 are in order. First, we note that the constant in the above regret bound can be exponentially large in the size of the market, which is also the case with Liu et al. (2020) which considered stationary preferences of players. Second, the achievable regret bounds in a single agent MAB setting is $\widetilde{O}(\sqrt{K L_T T})$ Auer et al. (2002). Our regret bounds are similar to the single agent multi-arm bandit setting up to some constant, although under the assumption that the gap between the arm of any two players is greater than Δ at all the times. As pointed out in Liu et al. (2020), the dependence on Δ^2 cannot be improved in general because of the dependence of a player's regret on the gaps of other players in the competing bandit setting. For the same reason, it is not be possible to extend the results to the case where the arm gaps for a player can be arbitrarily close to each other. Third, we note that the regret R_T can be negative, which is not less desirable as player's can receive better utility than the pessimal matching.

A detailed proof of Theorem 1 can be found in the extended version of this paper Muthirayan et al. (2022). However, we provide a sketch of the proof below.

Proof Sketch: In the following analysis we only give the key steps. We start by analyzing the agent regret for a player p_i within an interval of length H from one restart to the next. The total regret can then be computed by summing the regret across the intervals of length H. Let us denote the start and the end of the ℓth such interval by t_s^ℓ and t_e^ℓ. Let us denote the agent regret within the ℓth interval by R_ℓ^i. Then, the player i's regret in interval ℓ is $R_\ell^i = \sum_{t=t_s^\ell}^{t_e^\ell} \left(\mu_{i,t}(\mathbf{\underline{m}}_t(i)) - \mu_{i,t}(m_t(i)) \right)$.

The critical part to bounding this is to account for the fact that (i) unlike in a single agent setting, given that all the players are exploring and learning, an unstable match with lower reward than its valid partners can be enforced on a player by the preferences submitted by the other players and (ii) the underlying preferences and consequently the set of stable matching themselves are time varying. While the first part was addressed by Liu et al. (2020) for the case where the underlying preferences of the players are stationary, our analysis addresses how to account for time variation of the underlying preferences and the further complexity this leads to on how the preferences submitted by other players can force an unstable match. The key to our analysis is the use of the minimal cover of an unstable match before the occurrence of the first change across the players within the interval. Below, we present a crucial lemma which provides the pessimal regret for any player between any two restarts. The proof of the lemma is deferred to the appendix.

Lemma 6 *Suppose Assumption 1 holds. Then, under the RCB algorithm (Algorithm 1), the pessimal regret for a player i between $(\ell - 1)$th restart and ℓth restart is given by*

$$R_\ell^i \le \mathcal{O}(K L_\ell H) + \mathcal{O} \left(K \left(1 + \frac{\log(H)}{\Delta^2} \right) \right).$$

Then, the agent-stable regret for all the rounds can be bound by summing the pessimal regrets for the individual intervals. Hence, $R_T^i \le \sum_{\ell=1}^{\lfloor T/H \rfloor} R_\ell^i \le \mathcal{O}(L_T H) + \widetilde{\mathcal{O}} \left(\frac{T}{H\Delta^2} \right).$ Substituting for $H = L_T^{-1/2} T^{1/2}$ gives us the final result.

Remark 7 (Relation to Liu et al. (2020)) *The bound we present in Lemma 6 clearly brings out the relation with respect to Liu et al. (2020). The lemma presents the regret incurred by a player within an interval after the restart. The constant accompanying the main term $\widetilde{\mathcal{O}}(1 + 1/\Delta^2)$ is equivalent to the constant in the time invariant case, which is the sum over the minimal cover induced by blocking pairs that block an unstable match. The difference in our case is that this minimal cover corresponds to the set of blocking pairs before the first change or variation within the interval under consideration, which reduces to the constant in Liu et al. (2020) in the absence of time variations. The first term in Lemma 6, $\mathcal{O}\left(L_\ell H\right)$, reflects the impact on the regret from the time variations. This term obviously vanishes in the absence of time variations.*

5. EXTENSION TO UNKNOWN TIME VARIATION

Algorithm 1 required knowledge of time variation L_T in order to compute the optimal restart period H. However, in a lot of real-world applications it is not known a priori. One approach to extend the RCB algorithm to the unknown time variation case is the *bandits over bandits* approach (Zhao et al., 2020). The key idea in this approach is to adaptively tune the restart period by exploring the restart periods from an ensemble that ranges from a very low restart period to a very high restart period. Specifically, a *meta-bandit algorithm* is used to periodically reset the restart period of the base restart algorithm by treating each period in the ensemble as an arm and employing a bandit algorithm to choose the restart periods while resetting. The bandit algorithm ensures that the exploration of the restart periods is efficient and that the process of exploring the restart period converges to the best restart period.

The same idea can be extended to learning in two-sided matching markets. The platform can employ a bandit meta-algorithm to explore and adaptively choose the best restart period for the base RCB algorithm. To ensure that the ensemble of restart periods is sufficiently rich, the following can be chosen as the ensemble: $\mathcal{H} = \{H = 2^{j-1}, j \in [1, N]\}$, where $N = \lceil 1/2 \log T \rceil + 1$. Lets call the restart period defined in Theorem 5 by H^*. Then, $H^* = \mathcal{O}(T^{1/2})$ when $L_T = \mathcal{O}(1)$ and $H^* = \mathcal{O}(1)$ when $L_T = \mathcal{O}(T)$. Therefore, the ensemble $\mathcal{H}$ as defined contains the H^*s corresponding to the full range of L_T and is thus sufficiently rich. To explore the restart periods, the meta-algorithm can divide the overall duration T into periods of duration $\Upsilon = \mathcal{O}(T^{1/2})$ and in each period it can explore a specific restart period. The meta-algorithm can employ a standard bandit algorithm such as the EXP3 (Auer et al., 2002) to explore the restart periods efficiently.

The key issue in the competing bandits setting is how to define the reward for the EXP3 meta-algorithm. We discuss this more elaborately below. Let the matching under RCB with the fixed restart period H^* be denoted by m_t^*. Then, the regret for each player can be split as

$$
\begin{aligned}
R_T^i &= \sum_{t=1}^{T} \mu_{i,t}(\mathbf{m}_t(i)) - \sum_{t=1}^{T} \mu_{i,t}(m_t(i)) \\
&= \underbrace{\left[\sum_{t=1}^{T} \mu_{i,t}(\mathbf{m}_t(i)) - \sum_{t=1}^{T} \mu_{i,t}(m_t^*(i))\right]}_{\text{Base Regret for } p_i} + \underbrace{\sum_{t=1}^{T} [\mu_{i,t}(m_t^*(i))] - \sum_{t=1}^{T} [\mu_{i,t}(m_t(i))]}_{\text{Meta-regret for } p_i}
\end{aligned}
$$

The first term is the regret incurred when following a fixed restart period. The second term is the difference between the rewards accumulated by following a fixed restart period and the meta-

algorithm that adaptively tunes the restart period by exploring from an ensemble of periods. We refer to this second term as the *meta-regret*.

The current proof technique to bound base regret requires that all agents restart simultaneously. Therefore, in this setting we consider that platform coordinates the restarting period uniformly across all players. Consequently, we can only obtain bounds on the joint regret of all players. Towards that goal, we define the reward for the meta-algorithm as the sum total of the accumulated rewards over the period Υ across the players. The term $\sum_{t=1}^{T} \mu_{i,t}(m_t^*(i))$ corresponds to the total returns with the restart period H^*. Since $H^* \in \mathcal{H}$, the total return must be less than or equal to the returns corresponding to the best restart period for the players. Therefore, the *meta-regret* term must be upper bounded by the actual regret of the meta-algorithm, which can be bounded just as in (Zhao et al., 2020, Theorem 3) by $\widetilde{O}(T^{3/4})$; see (Zhao et al., 2020, Theorem 3) for the details. Then, the joint regret when the number of changes are unknown can be bounded as $\sum_{i=1}^{\mathcal{N}} R_T^i = \widetilde{\mathcal{O}}\left(N L_T^{1/2} T^{1/2}\right) + \widetilde{O}(T^{3/4})$.

The exploration required for the restart period results in an additional regret on top of the regret in Theorem 5. Therefore, in the unknown L_T case, the achievable total regret is limited by the $\widetilde{O}(T^{3/4})$ term. It is possible that our regret bounds could be improved by a better exploration strategy. In addition, it remains an open problem to bound the individual regret of the players when the number of changes are unknown.

6. CONCLUSIONS AND FUTURE WORK

In this work, we study the framework of competing bandits in time varying matching markets. Specifically, we address the problem of how the players can learn the preferences amidst the competition so that the platform can converge to a stable matching outcome. While this specific challenge has been studied earlier, we address how the players can learn when the players preferences themselves can be time varying. For this problem, we propose a *Restart Competing Bandits* algorithm that can provably achieve sub-linear regret uniformly across the players up to sub-linear variations in the total number of changes across the players. The key insight is that the non-stationarity does not drastically impact learning in matching markets. Our result is *first-of-its-kind* for a general preference structure. We also discuss extension of our algorithm to the case where the number of changes need not be known a priori and discuss possible open challenges. There are several directions for future work. One direction is the meta-algorithm strategy that can achieve the same player pessimal regret for each player as the case where the total number of changes are known. Another major direction is the decentralized setting where there is no central platform to coordinate or communicate with the players. While this problem has been studied for the stationary case, its extension to non-stationary environment is non-trivial.

References

Robin Allesiardo, Raphaël Féraud, and Odalric-Ambrym Maillard. The non-stationary stochastic multi-armed bandit problem. *International Journal of Data Science and Analytics*, 3(4):267–283, 2017.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Peter Auer, Pratik Gajane, and Ronald Ortner. Adaptively tracking the best bandit arm with an unknown number of distribution changes. pages 138–158, 2019.

Soumya Basu, Karthik Abinav Sankararaman, and Abishek Sankararaman. Beyond log-squared regret for decentralized bandits in matching markets. *arXiv preprint arXiv:2103.07501*, 2021.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Optimal exploration-exploitation in multi-armed-bandit problems with non-stationary rewards. pages 319–337, 2019.

Sarah H Cen and Devavrat Shah. Regret, stability & fairness in matching markets with bandit learners. In *International Conference on Artificial Intelligence and Statistics*, pages 8938–8968. PMLR, 2022.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Learning to optimize under non-stationarity. pages 1079–1087, 2019.

Xiaowu Dai and Michael Jordan. Learning in multi-stage decentralized matching markets. *Advances in Neural Information Processing Systems*, 34:12798–12809, 2021.

Sanmay Das and Emir Kamenica. Two-sided bandits and the dating market. In *IJCAI*, volume 5, page 19. Citeseer, 2005.

Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of research of the National Bureau of Standards B*, 69(125-130):55–56, 1965.

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

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. pages 174–188, 2011.

Avishek Ghosh, Abishek Sankararaman, Kannan Ramchandran, Tara Javidi, and Arya Mazumdar. Decentralized competing bandits in non-stationary matching markets. *arXiv preprint arXiv:2206.00120*, 2022.

Meena Jagadeesan, Alexander Wei, Yixin Wang, Michael Jordan, and Jacob Steinhardt. Learning equilibria in matching markets from bandit feedback. *Advances in Neural Information Processing Systems*, 34, 2021.

Alexander Karpov. A necessary and sufficient condition for uniqueness consistency in the stable marriage matching problem. *Economics Letters*, 178:63–65, 2019.

Donald Ervin Knuth. *Stable marriage and its relation to other combinatorial problems: An introduction to the mathematical analysis of algorithms*, volume 10. American Mathematical Soc., 1997.

Fang Kong, Junming Yin, and Shuai Li. Thompson sampling for bandit learning in matching markets. *arXiv preprint arXiv:2204.12048*, 2022.

Ramakrishnan Krishnamurthy and Aditya Gopalan. On slowly-varying non-stationary bandits. *arXiv preprint arXiv:2110.12916*, 2021.

Lydia T Liu, Horia Mania, and Michael Jordan. Competing bandits in matching markets. pages 1618–1628, 2020.

Lydia T Liu, Feng Ruan, Horia Mania, and Michael I Jordan. Bandit learning in decentralized matching markets. *Journal of Machine Learning Research*, 22(211):1–34, 2021.

Meghna Lowalekar, Pradeep Varakantham, and Patrick Jaillet. Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence*, 261:71–112, 2018.

Bruce M Maggs and Ramesh K Sitaraman. Algorithmic nuggets in content delivery. *ACM SIG-COMM Computer Communication Review*, 45(3):52–66, 2015.

Chinmay Maheshwari, Eric Mazumdar, and Shankar Sastry. Decentralized, communication-and coordination-free learning in structured matching markets. *arXiv preprint arXiv:2206.02344*, 2022.

Yifei Min, Tianhao Wang, Ruitu Xu, Zhaoran Wang, Michael I Jordan, and Zhuoran Yang. Learn to match with no regret: Reinforcement learning in markov matching markets. *arXiv preprint arXiv:2203.03684*, 2022.

Deepan Muthirayan, Masood Parvania, and Pramod P Khargonekar. Online algorithms for dynamic matching markets in power distribution systems. *IEEE Control Systems Letters*, 5(3):995–1000, 2020.

Deepan Muthirayan, Chinmay Maheshwari, Pramod P Khargonekar, and Shankar Sastry. Competing bandits in time varying matching markets. *arXiv preprint arXiv:2210.11692*, 2022.

Alvin E Roth. The economics of matching: Stability and incentives. *Mathematics of operations research*, 7(4):617–628, 1982.

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

Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. Pairwise kidney exchange. *Journal of Economic theory*, 125(2):151–188, 2005.

Yoan Russac, Claire Vernade, and Olivier Cappé. Weighted linear bandits for non-stationary environments. *Advances in Neural Information Processing Systems*, 32, 2019.

Abishek Sankararaman, Soumya Basu, and Karthik Abinav Sankararaman. Dominate or delete: Decentralized competing bandits in serial dictatorship. In *International Conference on Artificial Intelligence and Statistics*, pages 1252–1260. PMLR, 2021.

Peng Zhao, Lijun Zhang, Yuan Jiang, and Zhi-Hua Zhou. A simple approach for non-stationary linear bandits. pages 746–755, 2020.