
Dominate or Delete: Decentralized Competing Bandits in Serial Dictatorship

Abishek Sankararaman*
AWS AI, Palo Alto, USA¹

Soumya Basu*
Google, Mountain View, USA²

Karthik Abinav Sankararaman
Facebook, Menlo Park, USA

Abstract

Online learning in a two-sided matching market, with demand side agents continuously competing to be matched with supply side (arms), abstracts the complex interactions under partial information on matching platforms (e.g. UpWork, TaskRabbit). We study the decentralized serial dictatorship setting, a two-sided matching market where the demand side agents have unknown and heterogeneous valuation over the supply side (arms), while the arms have known uniform preference over the demand side (agents). We design the first decentralized algorithm – UCB with Decentralized Dominant-arm Deletion (UCB-D3), for the agents, that does not require any knowledge of reward gaps or time horizon. UCB-D3 works in phases, where in each phase, agents delete *dominated arms* – the arms preferred by higher ranked agents, and play only from the non-dominated arms according to the UCB. At the end of the phase, agents broadcast in a decentralized fashion, their estimated preferred arms through *pure exploitation*. We prove both, a new regret lower bound for the decentralized serial dictatorship model, and that UCB-D3 is order optimal.

1 INTRODUCTION

Online matching markets (e.g. UpWork and Mechanical Turk) are economic platforms that connect de-

mand side, (e.g. businesses in Upwork or Mechanical Turk), to the supply side (e.g. freelancers in Upwork, or crowdworkers in Mechanical Turk). These platforms enable the demand side agents (a.k.a. agents) to make repeated decisions to match with the supply side agent (a.k.a. arms) of their preference. On these platforms, supply side agents when faced with multiple offers, chooses the demand side agent of her choice. With resource limited supply side, the agents thus compete for arms, while navigating uncertainty on the quality of each arm. Uncertainty in matching markets have recently been studied in various disciplines, *e.g.*, [Das and Kamenica, 2005] uses an empirical approach and [Johari et al., 2016] uses an economic approach, and [Liu et al., 2020] formalizes the problem from a learning perspective. In particular, Liu. et.al [Liu et al., 2020] introduced the centralized multi-agent matching bandit model, where at each time, every agent (demand side) shares her learned preference *truthfully over all the arms* (supply side agents) with a central arbiter, who then makes an allocation.

In this paper, we initiate the systematic study of the *decentralized paradigm* of this model, where no central arbiter exists, the knowledge of gaps and time horizon is unknown, and each agent relies only on her own observations to make arm choices. The decentralized paradigm, without central control, is of utmost importance in multi-agent setting, as sharing observation with a central arbiter is prone to privacy breach, lacks transparency of the arbiter, and is susceptible to untruthful inputs from agents [Larsson, 2018, Rees-Jones and Skowronek, 2018]. Further, in such systems any global information (e.g. minimum gap between arm rewards for agents, time horizon) is unavailable a priori.³

We focus on the *Serial Dictatorship* model (made precise in the sequel), a subclass of the

*Equal Contribution

¹Part of work done while affiliated with UC Berkeley

²Part of work done while affiliated with UT Austin

³In [Liu et al., 2019] the decentralized setting was briefly mentioned for the matching bandit model, but no solution was proposed when global information is unavailable.

matching bandits model which is well studied in economics, game-theory and matching markets [Abdulkadiroğlu and Sönmez, 1998, Bogomolnaia and Moulin, 2001, Bade, 2020, Aziz et al., 2013]. In particular, this setting has attracted a lot of interest, both in theory and practice since this is the only known mechanism that is both truthful (to elicit incentives) and Pareto optimal (to make allocations) in a two sided market [Zhou, 1990].

Model Overview. A serial dictatorship model consists of N agents and $K \geq N$ arms. For each agent the arms (supply side agents) are ranked *heterogeneously*, in increasing order of arm-means which is different for each agent-arm pair. The agents are ranked *homogeneously* across all arms (uniform valuation). Agents do not know the arm-means (and thus their preference) and have to learn them over time. All agents choose an arm each simultaneously in each round (a decentralized system). Thus, in every round, each arm is chosen by any number (0 to N) of agents. An arm matches only with the highest ranked proposing agent (if any), while blocking the remaining agents (if any). Therefore, each agent is either *matched* to her arm of choice and receives a corresponding stochastic reward, or is *blocked* by her arm of choice (which they are notified of) and receive a deterministic 0 reward.

Objective. The serial dictatorship system is said to be in *equilibrium*, when the matching between arms and the agents is the *unique stable matching* [Clark, 2006]. Here, a matching is stable, if there exists no agent-arm pair, who would mutually prefer each other as opposed to their current partners in the matching [Gale and Shapley, 1962]. The *regret* of an algorithm is defined as the expected difference of cumulative reward attained by the algorithm, and the cumulative reward in equilibrium when agents and arms always match according to the unique stable matching. The objective is to design a decentralized algorithm for the agents that has *regret* sub-linear (preferably logarithmic) in time horizon.

Applications. Online learning in a serial dictatorship setting finds its application in pareto-optimal allocation of resources among agents, where the objective is to ensure that the satisfaction of an agent cannot be increased without decreasing satisfaction of another agent. A canonical use case is that of scheduling jobs to servers in an online marketplace, *e.g.*, scheduling in datacenters (AWS, Azure) [Dickerson et al., 2019, Even et al., 2009, Brucker, 1999], crowdsourcing platforms (Upwork and TaskRabbit) [Massoulié and Xu, 2016, Basu et al., 2019], question answering platforms (Quora, Stack Overflow) [Shah et al., 2020]. In such marketplaces, the rewards are often stochastic and only observable upon com-

pletion. Moreover, these platforms solve thousands of such instances every hour through the numerous repeated interactions and thus would like to optimize the cumulative reward over a long time horizon. Finally, for scalability, truthfulness, and privacy-concerns decentralized solutions are preferred, where jobs and servers interact without a central authority.

As a concrete example, let us consider a matching platform with N workers and K tasks (or more specifically task types), where workers need to be matched with realized tasks (drawn from the task types). This can be captured by the repeated matching model proposed in this paper. Each worker requests their top-choice task (*e.g.*, based on the monetary return for the task), and using a serial dictatorship algorithm (*e.g.*, a universal rating of workers) the allocation is made to the workers. In particular, among all the offers received for a task the highest rated worker is assigned the task. The reward of a worker and task pair can only be truly realized after making the assignment. In this platform, each worker aims for maximizing their own reward.

Why are Decentralized Algorithms hard? One natural attempt to design a decentralized algorithm for this problem is for each agent to play the UCB1 algorithm [Auer et al., 2002] independently (without coordination). When an agent gets blocked, it treats the realized reward as 0. However, this approach fails for many canonical examples. Consider a setting with two agents and two arms. Let the mean rewards for agent 1 be 1 and $1/2$ on arm 1 and arm 2, respectively. For agent 2, let the mean rewards be 1 and ϵ for arms 1 and 2, respectively. Arm 1 and 2 both prefers agent 1 over agent 2. Note that the reward gap for agent 1 is $\Delta_1 = 1/2$ and agent 2 is $\Delta_2 = 1 - \epsilon$. In this case, agent 1 plays arm 1 for $(t - \mathcal{O}(\log(t)))$ time in t steps, and thus agent 2 perceives the arm 1 as having mean 0. Therefore, an application of the instance dependent regret lower bound for bandits (Theorem 16.2 in [Lattimore and Szepesvári, 2018]) shows the regret for such naive decentralized UCB1 algorithm is $\Omega\left(\frac{\log(T)}{\epsilon}\right)$. For $\epsilon \approx 0$, this bound is arbitrarily worse than the $\mathcal{O}(\log(T))$ regret upper bound of our proposed algorithm (see, Section 4). The key challenge in this problem is that the agents need to have a *coordinated back-off mechanism without explicit communication*. The main contribution in this paper is a novel procedure that provably achieves optimal regret (order-wise) in a N agent, K arm system for $K \geq N \geq 1$.

1. Algorithmic Contributions.

(i) Non-Monotone Arm-elimination: We introduce the idea of non-monotone phase based arm-elimination in decentralized matching bandits problem. Specifically, in each phase, each agent deletes the

arms that are estimated to be the stable match partner of higher ranked agents, a.k.a. *dominated arms*. The intuition is as follows – under an algorithm with low regret, agents will play its stable match partner arm, for a large fraction of time. Thus, any agent through this deletion avoids collisions, at an arm that is the true stable match partner of a higher ranked arm. Furthermore, unlike all classical algorithms such as Successive Rejects [Audibert and Bubeck, 2010], the arm-deletion is *non-monotone*. Namely, the set of deleted arm in the past is not a subset of the set of deleted arms in the future. We show if an agent incorrectly deletes a non-dominated arm, it will, eventually with probability 1 under our algorithm, reverse the deletion and play this arm. Thus, non-monotonicity ensures linear regret is avoided.

(ii) Decentralized Dominated Arm Detection: We introduce a decentralized communication scheme that ensures in finite time each agent with high probability deletes all its dominated arms (if any) in a serial dictatorship. After deleting arms at the beginning of a phase, agents play according to the standard UCB in a phase by ignoring collisions. At the end of the phase, every agent sets the arm to which it was matched the most in the current phase, as its estimated stable match partner. Agents then in a decentralized fashion, communicate this through a simple pure exploitation strategy. The arms communicated will be deleted by agents in the next phase.

2. Technical Contributions.

(i) Order Optimal Regret Bounds: We prove in Section 4, that, the regret of any agent ranked $j \geq 2$ over a time horizon T , scales as $O(\log(T)/\Delta^2)$, where Δ is the smallest arm-gap across all agents. A precise regret bound involving all arm means is given in Theorem 1. In the decentralized setting, the dependence of our regret upper bound on Δ and T matches with the centralized setting as given in Theorem 1 of [Liu et al., 2020]. Our results rely on a new *inductive dominated-arm locking* technique mentioned above.

We further show in Section 6, through a matching lower bound, that a regret scaling $\Omega(\log(T)/\Delta^2)$ is unavoidable. This is the first $\Omega(\log(T)/\Delta^2)$ regret lower bound in decentralized multi-agent bandits, to the best of our knowledge. In our decentralized and heterogeneous setting, agents can not collaboratively learn the arm-means of other agents. This necessitates $\Omega(\log(T)/\Delta^2)$ exploration for any agent with a sub-optimality gap Δ , which in turn may lead to $\Omega(\log(T)/\Delta^2)$ collisions with the optimal arm for some lower ranked agent. Formalizing this intuition, we prove our instance dependent regret lower bound, and establish an order optimal regret for our algorithm.

(ii) Inductive Dominated-arm Locking: We develop a new analysis idea for the decentralized serial dictatorship, that relies on inductive blindness of the agents (from highest to lowest rank). By inductive blindness we mean, for any $i \geq 1$, the agent ranked i is unaffected by any other agent ranked $i + 1$ or higher. Such blindness sets us apart from the literature of decentralized bandits [Chakraborty et al., 2017, Wang et al., 2020, Boursier and Perchet, 2019, Rosenski et al., 2016] where all agents can affect any other agents through collisions. In our algorithm, the errors in an agent’s estimation of its stable match partner, *leaks to the other agents through the communication block*, when an agent signals its estimated stable matched arm. As agents delete their perceived dominated arms, an error by the higher ranked agent, gets amplified as it goes downstream. Thus any error in estimating a stable matched arm can create a domino effect that propagates across agents and phases. We prove that, if agents use the most matched arm in a phase as the estimate for its stable match partner, with probability 1, all such erroneous cascades die, and every agent eventually, in all phases, correctly identify its stable match partner arms. Our technical contribution lies in identifying an inductive structure in the algorithm, where, with probability 1, agents (in the order of their rank), stop spreading the wrong arm (see also Fig. 1).

2 PROBLEM SETTING

We consider N agents, and $K \geq N$ arms. The agents are ranked where the rank of any agent $j \in [N]$ is j (which is unknown to the agents). This is without loss of generality, as we can relabel the agents to obtain this. At each time, all agents choose one of the K arms simultaneously, to play and collect a reward. An agent is *matched* to the arm of its choice in a given round, if and only if it is the highest ranked agent playing that arm at that time.⁴ Otherwise, it is *blocked*. If agent $j \in [N]$ is matched with arm $k \in [K]$ at any time, then, agent j receives a stochastic $[0, 1]$ valued reward with mean $\mu_{jk} \in (0, 1)$ independent of all other rewards.⁵ If an agent is blocked, then it is notified and receives a deterministic reward of 0.

The arm means $(\mu_{jk})_{j \in [N], k \in [K]}$ are heterogeneous across the agents, and are not known to the agents apriori. Furthermore, for every agent j , the set of K arm-means $(\mu_{jk})_{k \in [K]}$ are all distinct. In the sequel, we denote by $I^{(j)}(t) \in [K]$, to be the arm played by

⁴We use the notion that rank l' is higher than rank l if and only if $l < l'$, throughout the paper.

⁵This is done for convenience. Our analysis can be easily adapted to any sub-gaussian reward.

agent j in round t . For each arm $k \in [K]$ and time $t \geq 1$, denote by $M_k(t) = \min\{j : I^{(j)}(t) = k\}$, to be the agent with which it is matched, where the minimum of an empty set is defined to be infinity.

Decentralized algorithms. We consider decentralized algorithms, namely, at each time t , the choice of arm to choose by any agent, must only depend on the events (past arm choices, rewards and blocking) observed by the agent. At the beginning, the agents are allowed to form a protocol to follow while playing.

Unique stable matching and regret. In our setting, each agent *prefers* arms in the increasing order of arm-means, and each arm *prefers* agents according to the uniform agent ranking. A matching of agents to arms is stable, if there is no agent-arm pair, unmatched in the current matching, that mutually prefer each other than their respective current matches [Gale and Shapley, 1962]. In our system, as the agents have uniform rank across all arms there exists a unique *stable matching* (which is not true when agents are ranked non-uniformly across arms.) Indeed, in any stable match agent j must match with its most preferred arm which is not matched with any agent with rank $(j - 1)$ or higher.⁶

We compare the performance of any decentralized online learning strategy to an oracle, in which the agents and arms are matched according to this unique *stable matching*. Stable matching is the appropriate benchmark, as it captures the equilibrium of the allocation of arms to agents, when agents are myopic and want to maximize individual rewards as in our setup. (See also in [Liu et al., 2020]). Agent ranked 1 prefers the arm $k_*^{(1)} = \arg \max_{k \in [K]} \mu_{1k}$ the most, therefore $(1, k_*^{(1)})$ forms a stable match as arm $k_*^{(1)}$ also prefers agent 1 the most. Now, for any agent ranked $2 \leq j \leq N$, the stable match denoted by $k_*^{(j)}$, is defined inductively as $k_*^{(j)} = \arg \max_{k \in [K] \setminus \{k_*^{(1)}, \dots, k_*^{(j-1)}\}} \mu_{jk}$. In words, the stable match for agent ranked j is the best arm from among all arms that do not form stable match to agents ranked 1 through to $j - 1$. Recall that $I^{(j)}(t) \in [K]$ denotes the arm chosen by agent j in time t . The regret of any agent $j \in [N]$, after T time steps is $R_T^{(j)} = \sum_{t=1}^T \mathbb{E}[\mu_{j k_*^{(j)}} - \mu_{j I^{(j)}(t)} \mathbf{1}_{M_{I^{(j)}(t)}(t)=j}]$.

3 UCBD3 Algorithm

Algorithm Overview. UCB-D3; proceeds in phases, where, in each phase, each agent only plays from a subset of arms, we call *active arms*. The set of active arms are fixed for an agent in a particular phase. However, the active arm sets are different across agents,

and are also *non-monotone*, for the same agent across time. Each phase $i \in \mathbb{N}$ is split into two blocks, (i) a regret minimization block lasting 2^{i-1} rounds and a (ii) communication block lasting a constant $(N - 1)K$ rounds. During regret minimization, all agents play from among their active set using the standard UCB algorithm [Auer et al., 2002] and ignoring collisions. During the communication block agents communicate their estimated stable match partners through collisions. An agent's estimate of its stable match partner arm is the *one with which it was matched the most number of times in a phase*. The active arms for an agent in the next phase is all the arms except those that were estimated to be stable-match partner to other higher ranked agents *in that phase*. UCB-D3 is *non-monotone*; even if an agent deletes an arm in a given phase, it can potentially be active in a future phase, if in the future phase, no higher ranked agent estimates this arm to be their stable match partner.

Notation. For an agent with rank $j \in [N]$, arm $k \in [K]$ and time slot $t \in \mathbb{N}$, denote by $N_k^{(j)}(t) \in \mathbb{N} \cup \{0\}$, to be the number of times agent with rank j was matched to arm k , upto and including time t . If $N_k^{(j)}(t) > 0$, denote by $\hat{\mu}_k^{(j)}(t)$ to be the empirical observed mean of arm k by agent j using all the samples upto and including time t .

The first $(N - 1)$ time slots are used to estimate agent's ranks and subsequently UCB-D3 proceeds in phases. From time slots N onwards, the algorithm proceeds in phases, numbered $i \in \mathbb{N}$, with an agent's phase being non-decreasing with time. The algorithm is synchronous, i.e., at each time t , all agents are in the same phase. The pseudocode is given in Algorithm 1 and also described below.

• **Rank estimation.** Rank estimation occurs for the first $(N - 1)$ time slots as follows. In the first time slot, all agents will pull arm 1. In subsequent time slots $t \in [2, N - 1]$, all those agents that have never been matched in time slots $[1, t - 1]$ will pull arm indexed t . Agents that were matched to some $t' \in [1, t - 1]$, will play arm t' (the index of the first match) at time t . The estimated rank of an agent is the first time slot when it was matched. If an agent is unmatched in the first $N - 1$ time slots, its rank is N . One can observe using an inductive reasoning that, the estimated rank of an agent is equal to its true rank. Thus, all agents are aware of their rank after this phase. The pseudocode **RANK-ESTIMATION()** is given in Algorithm 2 in Appendix A.

• **Phase i.** We now describe the algorithm by fixing a particular phase $i \in \mathbb{N}$ and an agent with rank j . The phase i starts at time $S_i := (2^{i-1} + (i - 1)(N - 1))K + N - 1$. It is divided into two blocks: the *first*

⁶Agents ranked 1 through $j - 1$

block $[S_i, S_i + 2^{i-1} - 1]$ is the regret-minimization block lasting 2^{i-1} time-slots and the *second block* is $[S_i + 2^{i-1} - 1, S_{i+1} - 1]$ is the communication block lasting $(N - 1)K$ time slots. At the beginning of phase $i \in \mathbb{N}$, associated with rank $j \in [N]$ is an active set of arms $\mathcal{A}_i^{(j)} \subseteq [K]$ with cardinality $|\mathcal{A}_i^{(j)}| \geq K + 1 - j$. (Observe that $\mathcal{A}_i^{(1)} = [K]$). In the beginning of phase 1, we initialize $\mathcal{A}_1^{(j)} = [K]$ for all $j \in [N]$.

◊ **Regret-Minimization (RM) block.** In the RM block of phase $i \in \mathbb{N}$, an agent with rank j , will play from among the arms in $\mathcal{A}_i^{(j)}$ according to the standard UCB- α rule [Auer et al., 2002], where $\alpha \geq 2$ is a hyper-parameter. Thus, the arm played (but not necessarily matched) at time t in the RM block of phase i is $I^{(j)}(t) \in \arg \max_{k \in \mathcal{A}_i^{(j)}} \left(\hat{\mu}_k^{(j)}(t-1) + \sqrt{\frac{2\alpha \ln(t)}{N_k^{(j)}(t)}} \right)$. Ties are broken arbitrarily.

At the end of the RM block of phase i , agent with rank j , creates an estimate of its best arm denoted by $\mathcal{O}_i^{(j)} \in \mathcal{A}_i^{(j)}$, as the arm that it matched with the most number of times in the RM block of phase i . Formally, $\mathcal{O}_i^{(j)} \in \arg \max_{k \in \mathcal{A}_i^{(j)}} (N_k^{(j)}[i] - N_k^{(j)}[i-1])$, where, for any agent $j \in [N]$, phase $i \in \mathbb{N}$ and arm $k \in [K]$, $N_k^{(j)}[i]$ is the number of times arm k was matched to agent j in all the RM blocks up to and including phase i , with the convention $N_k^{(j)}[-1] = 0$.

◊ **Communication block.** In the communication block of phase $i \in \mathbb{N}$, agent with rank $j \in [N]$, communicates its estimated best arm $\mathcal{O}_i^{(j)}$ to every other agent with ranks $(j + 1)$ through N . This is achieved by arranging arm collisions in a specific way. The communication block, which is of duration $(N - 1)K$ time slots, is further sub-divided into $N - 1$ sub-blocks, each of K contiguous time slots. In the sub-block $l \in [N - 1]$, agent with rank $l + 1$ will pull all the K arms once each in a round robin fashion. All other agents $l' \neq l$ will play their estimated match, i.e., agent $l' \neq l$, will play arm $\mathcal{O}_i^{(l')}$ in all the K time slots, of the l th communication sub-block. Observe that agent ranked 1 will play its estimated best arm $\mathcal{O}_i^{(1)}$ in all the $(N - 1)K$ communication blocks of phase i .

For agent ranked $j \geq 2$, denote by $\mathcal{D}_i^{(j)} \subseteq [K]$, the set of arms with which agent j collides in the $j - 1$ th sub-block of the communication block in phase i . Observe from the communication protocol that $\mathcal{D}_i^{(j)} := \{\mathcal{O}_i^{(1)}, \dots, \mathcal{O}_i^{(j-1)}\}$. The pseudo-code of the communication block, namely DOMINATED-ARM-DETECTION(), is provided in Algorithm 3 in Appendix A.

◊ **Update active arms.** At the end of the communication block of a phase $i \in \mathbb{N}$, every agent $j \geq 2$ will update its active set of arms for phase $i + 1$, by deleting

the arms $\mathcal{D}_i^{(j)}$, i.e. $\mathcal{A}_{i+1}^{(j)} := [K] \setminus \mathcal{D}_i^{(j)}$. Agent ranked 1 will have all the K arms active, ($\mathcal{A}_{i+1}^{(1)} = [K]$).

4 REGRET UPPER BOUND

Dominated arms and Gaps. For each agent $j \in [N]$, denote by the set of dominated arms for agent j as $\mathcal{D}_*^{(j)} := \{k_*^{(1)}, \dots, k_*^{(j-1)}\}$ and the non-dominated arm as $\mathcal{A}_*^{(j)} := [K] \setminus \{k_*^{(1)}, \dots, k_*^{(j-1)}\}$, where $k_*^{(j)}$ is the stable matched arm for agent j .

For any agent $j \in [N]$ and arm $k \neq k_*^{(j)}$, denote by $\Delta_k^{(j)} = (\mu_{j k_*^{(j)}} - \mu_{jk})$ the gap in reward of the k -th arm and the stable matching arm $k_*^{(j)}$. Thus, by definition of $k_*^{(j)}$ and by the uniqueness assumption of arm-means, for all non dominated arms $k \in \mathcal{A}_*^{(j)} \setminus \{k_*^{(j)}\}$, we have $\Delta_k^{(j)} > 0$. Let $\Delta := \min_{j \in [N]} \min_{k \in \mathcal{A}_*^{(j+1)}} \Delta_k^{(j)}$ the smallest gap of non-dominated arms across all agents. Since arm-means are unique for all agents, $\Delta > 0$.

Theorem 1. Suppose every agent runs UCB-D3 with $\alpha \geq 2$, then the regret of any agent with rank j satisfies

$$\begin{aligned} \mathbb{E}[R_T^{(j)}] &\leq \underbrace{2(j-1)(\log_2(T) + 1)}_{\text{Regret due to Communication}} + \\ &\underbrace{\sum_{j'=1}^{j-1} \sum_{k \in \mathcal{A}_{j'}^*} \left(\frac{9\alpha \log(T)}{(\Delta_k^{(j')})^2} + 1 + \frac{178T^{-4\alpha}}{(\Delta_k^{(j')})^2} + T^{2-4\alpha} \right) \mu_{jk_*^{(j)}}}_{\text{Regret due to Collision}} \\ &+ \underbrace{\sum_{k \in \mathcal{A}_j^* \setminus \{k_*^{(j)}\}} \left(\frac{9\alpha \log(T)}{(\Delta_k^{(j)})^2} + 1 + \frac{178T^{-4\alpha}}{(\Delta_k^{(j)})^2} + T^{2-4\alpha} \right) \Delta_k^{(j)}}_{\text{Regret due to Sub-optimal arm pull}} \\ &+ N + j(2^{i^*} + i^* + 4(NK)^2 + 3NK) + \frac{2NK}{T^{2\alpha-4}}, \quad (1) \end{aligned}$$

where $i^* := \min \{i \in \mathbb{N} : \frac{20NK\alpha i}{\Delta^2} \leq 2^{i-1}\}$.

The following corollary highlights the dependencies of the regret on different model parameters.

Corollary 2. Suppose every agent runs UCB-D3 with parameter $\alpha \geq 2$, then the regret of any agent with rank j after T time steps satisfies

$$\begin{aligned} \mathbb{E}[R_T^{(j)}] &\leq 9\alpha \log(T) \left(\frac{(j-1)(K+1-j)}{\Delta^2} + \frac{[K-1-j]_+}{\Delta} \right) \\ &+ O((NK)^2 + \frac{NK}{\Delta^2} \log(\frac{NK}{\Delta^2})), \end{aligned}$$

where for any $x \in \mathbb{R}$, $[x]_+ := \max(x, 0)$.

Corollary 2 implies that the best ranked agent (i.e., agent 1) will experience no collisions, and thus its regret is at most $O(\log(T)/\Delta)$. For any other agent ranked $2 \leq j \leq N$, the regret is at most $O((K+1-j)(j-1)/\Delta^2 \log(T)) +$

Algorithm 1 UCB with Decentralized Dominant-arm Deletion (UCB-D3) (at Agent $j \in [N]$)

```

1:  $\text{Rank}^{(j)} \leftarrow \text{RANK-ESTIMATION}()$  ▷ First  $N - 1$  arm pulls to estimate rank using Algorithm 2
2:  $\mathcal{C}_0 = \emptyset$  ▷ Set of arms blocked in the beginning of phase 1
3: for  $i \in \{1, 2, \dots\}$  do ▷ For each Phase  $i$ 
4:    $\mathcal{A}_i^{(j)} \leftarrow [K] \setminus \mathcal{C}_{i-1}$  ▷ Set of Active arms in phase  $i$ 
5:   for  $S_i \leq t \leq S_i + 2^{i-1}$  do ▷ The first  $2^{i-1}$  times (RM block) of phase  $i$ 
6:     | Play an arm  $I^{(j)}(t) \in \arg \max_{k \in \mathcal{A}_i^{(j)}} \left( \hat{\mu}_k^{(j)}(t-1) + \sqrt{\frac{2\alpha \log(t)}{N_k^{(j)}(t-1)}} \right)$ 
7:   end for
8:    $\mathcal{O}_i^{(j)} \leftarrow$  the most matched arm from  $\mathcal{A}_i^{(j)}$  in the first  $2^{i-1}$  time slots of phase  $i$ .
9:    $\mathcal{C}_i \leftarrow \text{DOMINATED-ARM-DETECTION}(\mathcal{O}_i^{(j)}, i, \text{Rank}^{(j)})$  ▷ The last  $(N - 1)K$  times of phase  $i$  given in Algorithm 3
10: end for
    
```

$O([K - 1 - j]_+ / \Delta \log(T))$. However, this is only an *upper bound* on regret. In particular, under certain circumstances the regret under our algorithm can be negative —if an agent is matched to an arm with higher mean compared to the mean of its stable match partner arm. Moreover, we observe in simulations (Figure 2 and Appendix E) that, UCB-D3 outperforms the only prior decentralized algorithm, the ETC from [Liu et al., 2020] and is comparable to the centralized UCB in [Liu et al., 2020].

Proof sketch of Theorem 1. Using the rank estimation protocol each agent learns its own rank correctly in a decentralized way. We may assume agent j knows it is ranked j henceforth. As agent 1 experiences no collisions, we show that for all phases after a random finite number of phases $\tau^{(1)}$, agent 1 identifies the best arm in the stable match (see also Figure 1). Thus, from phase $\tau^{(1)}$ on-wards, agent 2 will eliminate the correct arm, arm $k_*^{(1)}$. Moreover, as agent 1 is playing UCB algorithm, the number of times it plays a sub-optimal arm is small and thus, the total number of collisions experienced by agent 2 is small. Subsequently, we show that for all phases after a random phase $\tau^{(2)} \geq \tau^{(1)}$, agent 2 always identifies its correct arm $k_*^{(2)}$ as the best arm. Similarly, we show that for all $j \in [2, N]$, for all phases after a random phase $\tau^{(j)}$, all agents $j' \leq j$ always identifies the set of dominated arms $\mathcal{D}_*^{(j)}$ (hence the non-dominated arms $\mathcal{A}_*^{(j)}$), correctly by eliminating the most played arm of the higher ranked agents in the previous phase. Furthermore, from $\mathcal{A}_*^{(j)}$ it identifies its stable match $k_*^{(j')}$ as its best arm due to UCB dynamics, and thus, regret of all agents ranked j or lower is well-behaved (see also Figure 1). The regret bound is proved by establishing upper bounds on the expectations of $\tau^{(j)}$, $\forall j \in [K]$. We do so by establishing high probability upper bound on the number of times a sub-optimal arm is played by the UCB algorithm, even though the

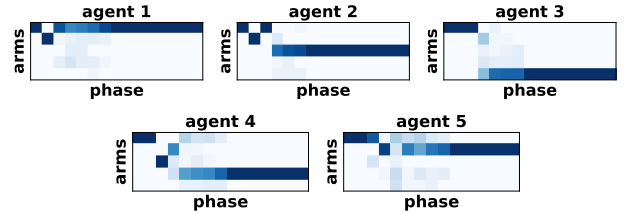


Figure 1: A system of 5 agents and 5 arms with arm-means chosen i.i.d. uniform in $[0, 1]$. Over 100 parallel runs with 13 phases each, this heatmap counts the arm communicated by each agent at the end of each phase across runs. The color intensity increases with the number of runs where the arm is communicated. This figure demonstrates the freezing of dominated-arms empirically; all agents communicate its stable match (depicted as a contiguous black strip) partner arm after a certain phase number.

set of active arms varies across phases for each agent (except agent 1) (Lemma 12 in appendix).

5 Incentive Compatibility and Robustness to Selfish Players

In this section, we discuss robustness properties when any agent deviates from executing the UCB-D3 algorithm in order to maximize the collected reward. We consider a desirable robustness property called ε Nash Equilibrium for multi-agent algorithms, recently proposed in [Boursier and Perchet, 2020]. Roughly, this property guarantees that no agent can significantly increase (by at-most additive ε) its rewards by unilaterally deviating from the UCB-D3; protocol. Although this is a weaker concept compared to the classical Nash-Equilibrium used in the theory of repeated games [Myerson, 2013], is nevertheless a useful property for practical algorithms in multi-agent bandits to possess [Boursier and Perchet, 2020].

We set some notations to define this concept. For any

agent $j \in [N]$, time horizon T and *algorithm profile* $s := (s_1, \dots, s_N)$ executed by the N agents respectively, denote by $\text{Rew}_T^{(j)}(s)$ to be the reward collected by agent j in T time slots, when agent $j \in [N]$ executes algorithm s_i . For an algorithm profile s , an algorithm s' and any agent $j \in [N]$, denote by $\text{Rew}_T^{(j)}(s_{-j}, s')$ to be the reward obtained by agent j when it executes algorithm s' and the other agents play according to the profile s .

Definition 3. An algorithm profile s is an $\varepsilon := (\varepsilon_j)_{j=1}^N$ Nash Equilibrium if, for every agent $j \in [N]$ and algorithm s' , $\mathbb{E}[\text{Rew}_T^{(j)}(s_{-j}, s')] \leq \mathbb{E}[\text{Rew}_T^{(j)}(s)] + \varepsilon_j$.

Proposition 4. The UCB-D3 algorithm profile is $\varepsilon : (\varepsilon_j)_{j=1}^N$ stable where, for all $j \in [N]$, $\varepsilon_j = \sum_{l=1}^{j-1} \mathbf{1}_{(\mu_{jl} > \mu_{jk_*}^{(j)})} \frac{\mu_{jl}}{\mu_{(l)}} \mathbb{E}[R_T^{(l)}] + \mathbb{E}[R_T^{(j)}]$, where for all $j' \in [N]$, $\mathbb{E}[R_T^{(j')}]$ is given in Equation (1).

The proof is given in Appendix C. This proposition gives that for agent ranked j , $\varepsilon_j = O\left(j \frac{\mu_{\max}^{(j)}}{\mu_{\min}^{(j)}} \mathbb{E}[R_T^{(j)}]\right)$, where $\mu_{\max}^{(j)}$ ($\mu_{\min}^{(j)}$) is the maximum (minimum) arm-mean for agent j . Roughly speaking, this proposition guarantees that even if an agent unilaterally deviates from playing the UCB-D3 algorithm, it must still incur $O(\log(T)/\Delta^2)$ regret.

6 REGRET LOWER BOUND

We provide a regret lower bound by adapting the approach in [Auer et al., 2002] to our multi-agent setup. Let $R_T(\nu, \pi)$ denote the cumulative (sum over all agents and time horizon) expected regret of a policy π on the instance with arm distributions $\nu = \{\nu_{jk} : j \in [N], k \in [K]\}$ for a horizon of length T . Also, denote by \mathcal{P} the set of all probability distributions with bounded support $[0, 1]$.⁷ We define $D_{\inf}(\nu, x, \mathcal{P}) = \inf_{\nu' \in \mathcal{P}} \{D(\nu, \nu') : \mu(\nu') > x\}$ for any distribution $\nu \in \mathcal{P}$. Here, $\mu : \mathcal{P} \rightarrow \mathbb{R}$ is the operator mapping a distribution in \mathcal{P} to its mean and $D(\cdot, \cdot)$ is the KL divergence.

Definition 5 (Uniformly Consistent Policies). A policy π is uniformly consistent if and only if for all $\nu \in \mathcal{P}$, all $\alpha \in (0, 1)$, the regret $\limsup_{T \rightarrow \infty} \frac{R_T(\nu, \pi)}{T^\alpha} = 0$.

This notion is used to eliminate tuning a policy to the current instance while admitting large regret in other instances (c.f. [Auer et al., 2002, Lattimore and Szepesvári, 2018]).

Optimally stable bandits. Our regret lower bounds hold over a sub-class of bandits where the stable

matching is optimal. Let us consider the class of bandit instances where dominated arms are bad, i.e. for any instance ν in this class, for all agents $j \in [N]$, $\mu_{jk} < \mu_{jk_*}^{(j)}$ for all arms $k \in [K] \setminus \{k_*^{(j)}\}$. We call this class of instances Optimally Stable Bandits (OSB), as each agent is matched with its optimal arm in the stable matching. Let for all $j \in [N]$, $\Delta_{\min}^{(j)} = \min_{k \neq k_*^{(j)}} \Delta_k^{(j)}$, which is always non-negative for an OSB instance.

Lemma 6 (Regret Decomposition for OSB). For a OSB instance $\nu = \{\nu_{jk} : j \in [N], k \in [K]\}$, and any uniformly consistent policy π , agent $j \in [N]$ the following holds

$$R_T^{(j)}(\nu, \pi) \geq \max \left\{ \sum_{j'=1}^{j-1} \Delta_{\min}^{(j)} \mathbb{E}_{\nu, \pi}[N_{k_*^{(j)}}^{(j')}(T)], \sum_{k \notin \mathcal{A}_*^{(j)} \setminus k_*^{(j)}} \Delta_k^{(j)} \mathbb{E}_{\nu, \pi}[N_k^{(j)}(T)] \right\}.$$

In a OSB instance, if an agent j 's stable match partner $k_*^{(j)}$ gets matched to an agent ranked higher than j , then the next best case is agent j gets matched to its second best (among all K arms). Even though this may happen rarely, it suffices to bound the first term in the above expression. Similar observations were utilized in providing a minimax (not instance dependent) lower bounds in [Liu et al., 2020].

Theorem 7. For any agent $j \in [N]$, under any decentralized universally consistent algorithm π on a OSB instance ν satisfies

$$\liminf_{T \rightarrow \infty} \frac{R_T^{(j)}(\nu, \pi)}{\log T} \geq \max \left\{ \sum_{j'=1}^{j-1} \frac{\Delta_{\min}^{(j)}}{D_{\inf}(\nu_{j'k_*^{(j)}}, \mu_{j'k_*^{(j')}}(\mathcal{P}))}, \sum_{k \notin \mathcal{A}_j \setminus k_*^{(j)}} \frac{\Delta_k^{(j)}}{D_{\inf}(\nu_{jk}, \mu_{jk_*^{(j)}}(\mathcal{P}))} \right\}$$

In the following corollary, we show that the dependence of $\log T/\Delta^2$ in the regret upper bound (Theorem 1) is tight up to $O(K)$. A detailed discussion on arm-gap dependence is provided in Appendix D. For all agents other than the best ranked agent, the regret scales as $\Theta(\log(T)/\Delta^2)$ in our model.

Corollary 8. There exists a OSB bandit instance with Bernoulli rewards, where the regret of agent j is lower bounded as $\Omega\left(\max\left\{\frac{(j-1)\log T}{\Delta^2}, \frac{K \log T}{\Delta}\right\}\right)$.

As the reward is heterogeneous the highest ranked agents with sub-optimality gap Δ is forced to explore $\Omega(\log(T)/\Delta^2)$ time each. Whereas, all other lower ranked agents are forced to compromise a lot whenever it's sub-optimality gap is large ($\Delta_{\min}^{(j)} = \Omega(1)$). The proof is built on the above observation.

⁷Please refer to the supplementary material for a formal definition of the policy and the environment.

6.1 Comment on the Gap between the Upper and Lower Regret Bounds

In order to improve our bounds, we hit on a fundamental roadblock, that we describe here. In the proof of Theorem 7, when an agent 1 to $j-1$ plays the stable arm (which is also optimal in a OSB instance) arm for agent j , the following needs to be understood.

1. Whether agent j is able to “anticipate the collision” and move to a sub-optimal arm that is collision free. If such anticipation is missing the agent accrues $\mu_{jk_*}^{(j)}$ regret (not $\Delta_{\min}^{(j)}$).
2. Given that the agent j avoids a collision, it is unclear whether the agent successfully plays the second-best arm. It is possible that it accrues $\Delta_{\max}^{(j)}$ regret instead in the worst case.

The above scenarios lead to the gap in the upper versus lower regret bounds. For closing such gaps it becomes essential to formalize anticipatory behavior among multi-agent decentralized bandits. This is beyond the scope of the current paper. Transfer of knowledge from lower to higher rank agents also is not possible here which makes mimicking a centralized algorithm difficult in our setting.

7 SIMULATIONS

In this section, we compare our algorithm to the centralized UCB [Liu et al., 2020] and the ETC based decentralized algorithm [Liu et al., 2019]. We simulate the three algorithms on a OSB instance with 5 agents and 5 arms, and report the regret from these algorithms in Figure 2. For this instance, we first fix a random permutation σ , where the arm-mean of arm $\sigma(i)$ for agent i was set to 0.9 for all i . All other arm-means was chosen randomly and uniformly in $[0, 0.8]$. We estimate the mean regret from 30 independent runs. Our results clearly indicate that, our algorithm outperforms the decentralized ETC algorithm, and is comparable to the centralized UCB algorithm. In the appendix, we present more simulation results, and related discussions. In particular, we consider 4 additional systems – the first two systems are the OSB systems with 10 agents and 10 arms (Fig. 4), and 10 agents, 15 arms (Fig. 5). We also consider two non OSB systems with 5 agents and 7 arms (Fig. 3), and 10 agents and 15 arms (Fig. 6).

8 MODEL EXTENSIONS

We now highlight that our algorithmic paradigm is quite general, and can be potentially extended well beyond the serial dictatorship setting considered in the

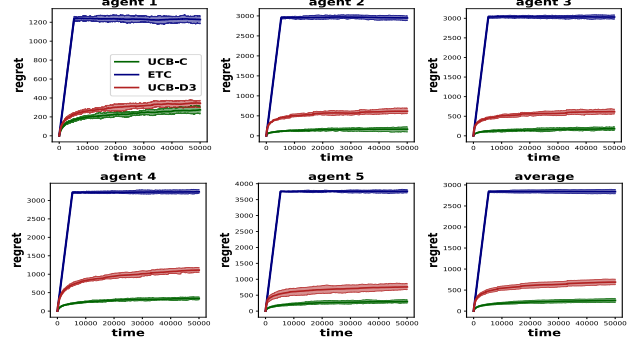


Figure 2: Simulations with 5 agents, and 5 arms. We observe UCB-D3 outperforms ETC by large margin.

present paper. A well known sufficient condition for a bipartite graph to admit a unique stable matching is known as the Sequential Preference Condition (SPC) condition [Clark, 2006], [Karpov, 2019]. Until a recent breakthrough in [Karpov, 2019], SPC was the weakest known sufficient condition for unique matching. The ideas introduced in this paper, can potentially be extended to provide a novel decentralized algorithm that obtains low (logarithmic) regret in the SPC setting.

Consider a marketplace with N agents (indexed $1, 2, \dots, N$) and K arms, with $N \leq K$. Under the SPC condition, there exists a permutation of the arms σ , and the agents σ' such that for all $i \in [N]$, (i) agent $\sigma'(i)$ prefers the arm $\sigma(i)$ the most among $[K] \setminus \{\sigma(l) : l < i\}$, and (ii) arm $\sigma(i)$ prefers the agent $\sigma'(i)$ the most among $[N] \setminus \{\sigma'(l) : l < i\}$. The above condition ensures, the inductive matching of agent $\sigma'(i)$ with arm $\sigma(i)$. Note that, the serial dictatorship studied in this paper is a special case of the SPC model with agent permutation σ' as identity, and $\sigma(i)$ as the stable matched arm for agent i , for all $i \in [N]$.

UCB-D3 can be adapted to this general setting as follows. The current algorithm does not work as there is no dictator (an agent which is preferred by all ‘unmatched’ arms) at any point. However, this can be resolved by adding a local-deletion step, where agents further discard arms, that have collided a certain number of times in a phase (hence local-deletion, since deletion is in a phase). The key intuition is that under an algorithm with appropriately tuned local-deletion, agents can “lock-in” to the equilibrium in an inductive fashion and incur negligible collisions given that the phase lengths are sufficiently large. For the base case, observe that as the agent $\sigma'(1)$ tries out an arm other than $\sigma(1)$, it either is the preferred agent for the arm and discards the arm as it is inferior to $\sigma(1)$ (through UCB); otherwise it is blocked often at the arm and discards it through local deletion. Thus agent $\sigma'(1)$ locks in to $\sigma(1)$. Similar arguments will ensure lock in for all other agents. The appropriate tuning of

local-deletion to achieve logarithmic regret in the SPC setting remains open for future work.

Finally, obtaining good decentralized algorithms remain open beyond SPC condition. This includes unique stable matching when SPC condition fails, and the general matching setup with multiple stable matching. We believe these are out of reach of the current tools developed in this paper, as the phenomenon of “lock-in” into a stable equilibrium is missing.

9 RELATED WORK

MAB are widely studied owing to a multitude of applications ([Lattimore and Szepesvári, 2018, Bubeck et al., 2012]). In recent times, as the scale of applications increases, multi-agent MAB problems have come into focus. The paper [Liu et al., 2020] is the closest to ours, as it introduced the matching bandit model in more generality. However, the algorithms in that paper were either (i) centralized and required agents at all points of time to submit a ranked list of arms to a centralized scheduler (platform) or (ii) decentralized but needed information on minimum arm-gap across all agents and the time horizon. For the important special case of Serial dictatorship, we give a decentralized algorithm (and a lower bound), that does not need knowledge of arm-gaps or time horizon. Thus, we partially resolve an open question in [Liu et al., 2020] on decentralized algorithms for matching bandits.

The literature on multi-agent MAB can be classified into two —*competitive* where different agents compete for limited resources (as in this paper) or *collaborative*, where agents jointly accomplish a shared objective. The canonical model of competitive multi-agent bandits is one wherein if multiple agents play the same arm, they all are blocked and receive no reward (colliding bandit model) [Lai et al., 2008, Kalathil et al., 2014, Avner and Mannor, 2014, Bistritz and Leshem, 2018, Boursier and Perchet, 2019, Mehrabian et al., 2020, Rosenski et al., 2016, Alatur et al., 2020]. Such models are motivated from applications in wireless networks [Avner and Mannor, 2016]. However the symmetry in the problem, where if multiple agents choose the same arm, they are all blocked, are crucially used in all decentralized algorithms for that model and are hence not applicable directly to our setup. See more details in Section 9.1. The collaborative models, consists of settings where if multiple agents play the same arm simultaneously, then they all receive independent rewards [Sankararaman et al., 2019, Chakraborty et al., 2017, Buccapatnam et al., 2015, Kolla et al., 2018,

Martínez-Rubio et al., 2019, Landgren et al., 2016]. Such models have primarily been motivated by applications such as internet advertising [Chawla et al., 2020]. However, algorithms there rely on agents collecting independent samples for arms, which are not applicable to our setting.

9.1 Comparison with Regret Bounds for Related Models

The regret bound in our problem is in contrast with the performance of a related and a widely studied model known as the multi-agent *colliding bandits model* [Boursier and Perchet, 2019], [Wang et al., 2020] where, if two or more agents pull the same arm, then all agents are blocked. The regret in this case is measured with respect to an optimal allocation of agents to arms by an oracle that knows all the arm means. The per-agent regret for the colliding bandit model, for both the centralized and decentralized algorithms scale as $O\left(\frac{\log(T)}{\Delta}\right)$. The gap in performance between our model and the colliding bandits model arises due to the *asymmetric collisions*; when multiple agents choose the same arm, not all of them experience a collision in our model. Thus, agents cannot infer if its actions cause a collision to other higher ranked agents and hence the regret must scales as $\Omega\left(\frac{\log(T)}{\Delta^2}\right)$. In contrast, the colliding bandit model is symmetric; if multiple agents pull the same arm simultaneously, then all of them get blocked. Thus, the agents can coordinate in a decentralized way [Wang et al., 2020] to eliminate blocking and obtain a regret of $O\left(\frac{\log(T)}{\Delta}\right)$.

10 CONCLUSION

We considered the heterogeneous multi agent matching bandit problem and proposed UCB-D3 a novel decentralized algorithm with optimal regret guarantees. This proceeds in phases, and in each phase, an agent ‘eliminates’ arms it will likely collide. The main insight from our algorithm was that if agents delete their dominated arms (arm optimal for higher ranked agents), then they will incur low regret. An interesting avenue for future work is to devise decentralized algorithms in the non-serial dictatorship setting.

References

- [Abdulkadiroğlu and Sönmez, 1998] Abdulkadiroğlu, A. and Sönmez, T. (1998). Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701.

- [Alatur et al., 2020] Alatur, P., Levy, K. Y., and Krause, A. (2020). Multi-player bandits: The adversarial case. *Journal of Machine Learning Research*, 21(77):1–23.
- [Audibert and Bubeck, 2010] Audibert, J.-Y. and Bubeck, S. (2010). Best arm identification in multi-armed bandits.
- [Auer et al., 2002] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multi-armed bandit problem. *Machine learning*, 47(2-3):235–256.
- [Avner and Mannor, 2014] Avner, O. and Mannor, S. (2014). Concurrent bandits and cognitive radio networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 66–81. Springer.
- [Avner and Mannor, 2016] Avner, O. and Mannor, S. (2016). Multi-user lax communications: a multi-armed bandit approach. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE.
- [Aziz et al., 2013] Aziz, H., Brandt, F., and Brill, M. (2013). The computational complexity of random serial dictatorship. *Economics Letters*, 121(3):341–345.
- [Bade, 2020] Bade, S. (2020). Random serial dictatorship: the one and only. *Mathematics of Operations Research*, 45(1):353–368.
- [Basu et al., 2019] Basu, S., Gutstein, S., Lance, B., and Shakkottai, S. (2019). Pareto optimal streaming unsupervised classification. In *International Conference on Machine Learning*, pages 505–514.
- [Besson and Kaufmann, 2017] Besson, L. and Kaufmann, E. (2017). Multi-player bandits revisited. *arXiv preprint arXiv:1711.02317*.
- [Bistritz and Leshem, 2018] Bistritz, I. and Leshem, A. (2018). Distributed multi-player bandits—a game of thrones approach. In *Advances in Neural Information Processing Systems*, pages 7222–7232.
- [Bogomolnaia and Moulin, 2001] Bogomolnaia, A. and Moulin, H. (2001). A new solution to the random assignment problem. *Journal of Economic theory*, 100(2):295–328.
- [Boursier and Perchet, 2019] Boursier, E. and Perchet, V. (2019). Sic-mmab: synchronisation involves communication in multiplayer multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 12048–12057.
- [Boursier and Perchet, 2020] Boursier, E. and Perchet, V. (2020). Selfish robustness and equilibria in multi-player bandits. *arXiv preprint arXiv:2002.01197*.
- [Brucker, 1999] Brucker, P. (1999). Scheduling algorithms. *Journal-Operational Research Society*, 50:774–774.
- [Bubeck et al., 2012] Bubeck, S., Cesa-Bianchi, N., et al. (2012). Regret analysis of stochastic and non-stochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122.
- [Buccapatnam et al., 2015] Buccapatnam, S., Tan, J., and Zhang, L. (2015). Information sharing in distributed stochastic bandits. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2605–2613. IEEE.
- [Chakraborty et al., 2017] Chakraborty, M., Chua, K. Y. P., Das, S., and Juba, B. (2017). Coordinated versus decentralized exploration in multi-agent multi-armed bandits. In *IJCAI*, pages 164–170.
- [Chawla et al., 2020] Chawla, R., Sankararaman, A., Ganesh, A., and Shakkottai, S. (2020). The gossiping insert-eliminate algorithm for multi-agent bandits. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3471–3481. PMLR.
- [Clark, 2006] Clark, S. (2006). The uniqueness of stable matchings. *The BE Journal of Theoretical Economics*, 6(1).
- [Das and Kamenica, 2005] Das, S. and Kamenica, E. (2005). Two-sided bandits and the dating market. In *IJCAI*, volume 5, page 19.
- [Dickerson et al., 2019] Dickerson, J., Sankararaman, K., Sarpatwar, K., Srinivasan, A., Wu, K.-L., and Xu, P. (2019). Online resource allocation with matching constraints. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- [Even et al., 2009] Even, G., Halldórsson, M. M., Kaplan, L., and Ron, D. (2009). Scheduling with conflicts: online and offline algorithms. *Journal of scheduling*, 12(2):199–224.
- [Gale and Shapley, 1962] Gale, D. and Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15.

- [Johari et al., 2016] Johari, R., Kamble, V., and Kanoria, Y. (2016). Matching while learning. *arXiv preprint arXiv:1603.04549*.
- [Kalathil et al., 2014] Kalathil, D., Nayyar, N., and Jain, R. (2014). Decentralized learning for multi-player multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345.
- [Karpov, 2019] Karpov, A. (2019). A necessary and sufficient condition for uniqueness consistency in the stable marriage matching problem. *Economics Letters*, 178:63–65.
- [Kolla et al., 2018] Kolla, R. K., Jagannathan, K., and Gopalan, A. (2018). Collaborative learning of stochastic bandits over a social network. *IEEE/ACM Transactions on Networking*, 26(4):1782–1795.
- [Lai et al., 2008] Lai, L., Jiang, H., and Poor, H. V. (2008). Medium access in cognitive radio networks: A competitive multi-armed bandit framework. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pages 98–102. IEEE.
- [Landgren et al., 2016] Landgren, P., Srivastava, V., and Leonard, N. E. (2016). Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 167–172. IEEE.
- [Larsson, 2018] Larsson, S. (2018). Law, society and digital platforms: Normative aspects of large-scale data-driven tech companies. In *The RCSL-SDJ Lisbon Meeting 2018” Law and Citizenship Beyond The States”*.
- [Lattimore and Szepesvári, 2018] Lattimore, T. and Szepesvári, C. (2018). Bandit algorithms. *preprint*, page 28.
- [Liu et al., 2020] Liu, L. T., Mania, H., and Jordan, M. (2020). Competing bandits in matching markets. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1618–1628. PMLR.
- [Liu et al., 2019] Liu, L. T., Mania, H., and Jordan, M. I. (2019). Competing bandits in matching markets. *arXiv preprint arXiv:1906.05363*.
- [Martínez-Rubio et al., 2019] Martínez-Rubio, D., Kanade, V., and Rebeschini, P. (2019). Decentralized cooperative stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 4531–4542.
- [Massoulié and Xu, 2016] Massoulié, L. and Xu, K. (2016). On the capacity of information processing systems. In *Conference on Learning Theory*, pages 1292–1297.
- [Mehrabian et al., 2020] Mehrabian, A., Boursier, E., Kaufmann, E., and Perchet, V. (2020). A practical algorithm for multiplayer bandits when arm means vary among players. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1211–1221. PMLR.
- [Myerson, 2013] Myerson, R. B. (2013). *Game theory*. Harvard university press.
- [Rees-Jones and Skowronek, 2018] Rees-Jones, A. and Skowronek, S. (2018). An experimental investigation of preference misrepresentation in the residency match. *Proceedings of the National Academy of Sciences*, 115(45):11471–11476.
- [Rosenski et al., 2016] Rosenski, J., Shamir, O., and Szlak, L. (2016). Multi-player bandits—a musical chairs approach. In *International Conference on Machine Learning*, pages 155–163.
- [Sankararaman et al., 2019] Sankararaman, A., Ganesh, A., and Shakkottai, S. (2019). Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–35.
- [Shah et al., 2020] Shah, V., Gulikers, L., Massoulié, L., and Vojnović, M. (2020). Adaptive matching for expert systems with uncertain task types. *Operations Research*.
- [Wang et al., 2020] Wang, P.-A., Proutiere, A., Ariu, K., Jedra, Y., and Russo, A. (2020). Optimal algorithms for multiplayer multi-armed bandits. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4120–4129. PMLR.
- [Zhou, 1990] Zhou, L. (1990). On a conjecture by gale about one-sided matching problems. *Journal of Economic Theory*, 52(1):123–135.