# Bandit Learning in Many-to-one Matching Markets with Uniqueness Conditions

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

An emerging line of research is dedicated to the problem of one-to-one matching markets with bandits, where the preference of one side is unknown and thus we need to match while learning the preference through multiple rounds of interaction. However, in many real-world applications such as online recruitment platform for short-term workers, one side of the market can select more than one participant from the other side, which motivates the study of the many-to-one matching problem. Moreover, the existence of a unique stable matching is crucial to the competitive equilibrium of the market. In this paper, we first introduce a more general new $\tilde{\alpha}$-condition to guarantee the uniqueness of stable matching in many-to-one matching problems, which generalizes some established uniqueness conditions such as *SPC* and *Serial Dictatorship*, and recovers the known $\alpha$-condition if the problem is reduced to one-to-one matching. Under this new condition, we design an MO-UCB-D4 algorithm with $O\left(\frac{NK\log(T)}{\Delta^2}\right)$ regret bound, where $T$ is the time horizon, $N$ is the number of agents, $K$ is the number of arms, and $\Delta$ is the minimum reward gap. Extensive experiments show that our algorithm achieves uniform good performances under different uniqueness conditions.

## 1 Introduction

The rise of platforms for the online matching market has led to an emergence of opportunities for companies to participate in personalized decision-making [13, 17]. Companies (like Thumbtack and Taskrabbit and Upwork platforms) use online platforms to address short-term needs or seasonal spikes in production demands, accommodate workers who are voluntarily looking for more flexible work arrangements or probation period before permanent employment. The supply and demand sides in two-sided markets make policies on the basis of their diversified needs, which is abstracted as a matching market with the agent side and arm side, and each side has a preference profile over the opposite side. They choose from the other side according to preference and perform a matching. The stability of the matching result is a key property of the market [29, 1, 25].

The preferences in the online labor market may be unknown to one side in advance, thus matching while learning the preferences is necessary. The multi-armed bandit (MAB) [33, 12, 4] is an important tool for $N$ independent agents in matching market simultaneously selecting arms adaptively from received rewards each round. The idea of applying MAB to one-to-one matching problems, introduced by [20], assumes that there is a central platform to make decisions for all agents. Following this, other works [21, 31, 6] consider a more general decentralized setting where there is no central platform to arrange matchings, and our work is also based on this setting.

However, it is not enough to just study the one-to-one setting. Take online short-term worker employment as an example, it is an online platform design with an iterative matching, where

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

employers have numerous similar short-term tasks or internships to be recruited. Workers can only choose one task according to the company's needs at a time while one company can accept more than one employee. Each company makes a fixed ranking for candidates according to its own requirements but workers have no knowledge of companies' preferences. The reward for workers is a comprehensive consideration of salary and job environment. Since tasks are short-term, each candidate can try many times in different companies to choose the most suitable job. We abstract companies as arms and workers as agents. Each arm has a $capacity$ $q$ which is the maximum number of agents this arm can accommodate. When an arm faces multiple choices, it accepts its most $q$ preferred agents. Agents thus compete for arms and may receive zero reward if losing the conflict. It is worth mentioning that arms with capacity $q$ in many-to-one matching can not just be replaced by $q$ independent individuals with the same preference since there would be implicit competition among different replicates of this arm, not equal treatment. In addition, when multiple agents select one arm at a time, there may be no collision, which will hinder the communication among different agents under the decentralized assumption. They cannot distinguish who is more preferred by this arm in one round as it can accept more than one agent while this can be done in one-to-one. Communication here lets each agent learn more about the preferences of arms and other agents, so as to formulate better policies to reduce collisions and learn fast about their stable results.

This work focuses on a many-to-one market under uniqueness conditions. Previous work [9, 14] emphasize the importance of constructing a unique stable matching for the equilibrium of matching problems and some existing uniqueness conditions are studied in many-to-one, such as *Sequential Preference Condition (SPC)*, *Serial Dictatorship*, and *Acyclicity* [24, 2]. Our work is motivated by [6], but the unique one-to-one mapping between arms and agents in their study which gives a surrogate threshold for arm elimination does not work in the many-to-one setting. And the uniqueness conditions in many-to-one matching are not well-studied, which also brings a challenge to identify and leverage the relationship between the resulting stable matching and preferences of two sides in the design of bandit algorithms. We propose an $\tilde{\alpha}$-condition that can guarantee a unique stable matching and recover $\alpha$-condition [18] if reduced to the one-to-one setting. We establish the relationships between our new $\tilde{\alpha}$-condition and existing uniqueness conditions in many-to-one setting.

In this paper, we study the bandit algorithm for a decentralized many-to-one matching market with uniqueness conditions. Under our newly introduced $\tilde{\alpha}$*-condition*, we design an MO-UCB-D4 algorithm with arm elimination and the regret can be upper bounded by $O\left(\frac{NK\log(T)}{\Delta^2}\right)$, where $N$ is the number of agents, $K$ is the number of arms, and $\Delta$ is the minimum reward gap. Finally, we conduct a series of experiments to simulate our algorithm under various conditions of *Serial dictatorship*, *SPC* and $\tilde{\alpha}$*-condition* to study the stability and regret of the algorithm.

**Related Work** The study of matching markets has a long history in economics and operation research [7, 5, 29] with real applications like school enrollment, labor employment, hospital resource allocation, and so on [1, 22, 28, 16]. A salient feature of market matching is making decisions for competing players on both sides [33, 11]. MAB is an important tool to study matching problems under uncertainty to obtain a maximum reward, and upper confidence bound algorithm (UCB) [4] is a typical algorithm, which sets a confidence interval to represent uncertainty. Matching market with MAB is studied in both centralized and decentralized setting [20, 21]. Following these, Abishek Sankararaman et al. [31] propose a phased UCB algorithm under a uniqueness condition, *Serial Dictatorship*, to manage collisions. They solve the problem of the decentralized market without knowing arm-gaps or time horizon, and reduce the probability of linear regret through non-monotonic arm elimination. The introduction of the uniqueness condition plays an important role in the equilibrium of matching results [14, 6]. Under a more general and robust condition, Uniqueness Consistency [18], Soumya Basu et.al [6] apply MAB to online matching and obtain robust results that the subset of stable matchings being separated from the system does not affect other stable matchings.

We discuss many-to-one problems such as online short-term employment and MOOC [13, 23, 17] as the one-to-one setting has limitations in practice. Somouaoga Bonkoungo [8] runs a student-proposing deferred acceptance algorithm (DA) [11] to study decentralized college admission. Ahmet Altinok [3] considers dynamic matching in many-to-one that can be solved as if it is static many-to-one or dynamic one-to-one under certain assumptions. As the existence and uniqueness of competitive equilibrium and core are important to allocations, the unique stable results need to be considered [25]. Similar to conditions for unique stable matching in one-to-one, some uniqueness conditions of stable results in the many-to-one setting also are studied [15, 26, 14, 2, 25].

## 2 Setting

This paper considers a matching market as a triplet $\mathcal{M} = (\mathcal{K}, \mathcal{J}, \mathcal{P})$. $\mathcal{K} = [K]$ denotes the finite set of arms, $\mathcal{J} = [N]$ denotes the finite set of agents. $\mathcal{P}$ is the set of preference order of agents and arms. In the many-to-one matching market, each arm $k$ has a capacity $q_k \geq 1$. At each time $t$, a random reward $X_{j,k}(t) \in [0, 1]$ is independently drawn from a fixed distribution with mean $\mu_{j,k}$ for every agent $j$ and arm $k$. For each agent, the preference order over arms is ranked by the mean reward. Denote $k \succ_j k'$ as agent $j$ prefers arm $k$ over $k'$, which also means $\mu_{j,k} > \mu_{j,k'}$. We assume that $\mu_{j,k} \neq \mu_{j,k'}$ if $k \neq k'$. Every arm has a total preference order over agents, which is denoted as $\succ_k$ for arm $k$. $j \succ_k j'$ means that arm $k$ prefers agent $j$ over $j'$. Throughout, we focus on markets where all agent-arm pairs are *mutually acceptable*, that is $j \succ_k \emptyset$ and $k \succ_j \emptyset$ for all $k \in [K]$ and $j \in [N]$.

We denote the mapping $m$ to be a matching result. $m_t(j)$ is the matched arm for agent $j$ at time $t$, and $\gamma_t(k)$ is the set of agents that are matched with arm $k$[1]. Each agent $j$ selects an arm $I_t(j)$ to match. If multiple agents select arm $k$ at the same time, only top $q_k$ agents can successfully match. The matched agent $j$ will observe the reward $X_{j,m_t(j)}(t)$ while the unmatched agent will only she has a *collision*. We use $M_t(j)$ to denote whether agent $j$ is successfully matched with her selected arm, and $M_t(j) = 1$ if agent $j$ is matched with $I_t(j)$. Then the reward obtained by agent $j$ is $X_{j,I_t(j)}(t)\ M_t(j)$.

An agent $j$ and an arm $k$ form a *blocking pair* for a matching if they are not matched but prefer each other over their assignments, i.e. $k \succ_j m(j)$ and $\exists j' \in m^{-1}(k), j \succ_k j'$. We say a matching satisfies individually rationality (IR), if $a_j \succ_{p_i} \emptyset$ and $p_i \succ_{a_j} \emptyset$ for all $i \in [N]$ and $j \in [K]$, that is, every worker prefers to find a job rather than do nothing, and every company also wants to recruit workers rather than not recruit anyone. Under the IR condition, a matching in the many-to-one setting is *stable* if there does not exist a blocking pair [30, 32].

This paper considers the matching markets under the uniqueness condition. Thus the overall goal is to find the unique stable matching between the agent side and arm side through iterations. Let $m^*(j)$ be the stable matched arm for agent $j$ under the stable matching $m^*$. The reward obtained by agent $j$ is compared against the reward received by matching with $m^*(j)$ at each time. We aim to minimize the expected stable regret for agent $j$ over time horizon $T$, which is defined as

$$R_j(T) = T\mu_{j,m^*(j)} - \mathbb{E}\left[\sum_{t=1}^{T} M_t(j) X_{j,I_t(j)}(t)\right].$$

## 3 Algorithm

In this section, we introduce our MO-UCB-D4 Algorithm (Many-to-one UCB with Decentralized Dominated arms Deletion and Local Deletion Algorithm) (Algorithm 1) for the decentralized many-to-one market, where there is no platform to arrange actions for agents, which leads to conflicts among agents. The MO-UCB-D4 algorithm for each agent $j$ first takes the agent set $\mathcal{J}$ and arm set $\mathcal{K}$ as input and chooses a parameter $\theta \in (0, 1/K)$ (discussed in Section C). It set multiple phases, where the duration of each phase is $2^{i-1}, i = 1, 2, \cdots$, and each phase mainly includes regret minimization block (line 5 - 12) and communication block (line 13 - 16).

The algorithm adding arm elimination to reduce potential conflicts mainly contains global deletion and local deletion. The former deletes the arms most preferred by agents who have obvious competitive advantages over agent $j$ and the latter is to delete the arms that still have many conflicts with agent $j$ after global deletion. We set a collision counter $C_{jk}[i]$ which records the number of collisions for agent $j$ pulling arm $k$ in phase $i$.

In regret minimization phase, we use $L_j[i] = \{k : C_{jk}[i] \geq \lceil \theta 2^i \rceil\}$ to represent the arms that collide too many times when matching with agent $j$ in phase $i$, and the threshold for collisions times is set to $\lceil \theta 2^i \rceil$. The algorithm first locally deletes the elements in $L_j[i]$ to reduce potential collisions for $j$. Then for each agent $j$, she selects an optimal action $I_t(j)$ from the active set $\texttt{Ch}_j[i]$ in phase $i$ according to UCB index, which is computed by $\hat{\mu}_{j,k}(t-1) + \sqrt{\frac{2\alpha \log(t)}{N_{j,k}(t-1)}}$ (line 7), where $N_{j,k}(t-1)$ is the number agent $j$ and arm $k$ have been matched. If the selected arm is successfully matched

[1]The mapping $m$ is not reversible as it is not a injective, hence wo donot use $m_t^{-1}(k)$.

**Algorithm 1** MO-UCB-D4 algorithm (for agent $j$)

**Input:**
$\theta \in (0, 1/K)$, $\alpha > 1$.
1: Set global dominated set $G_j[0] = \phi$
2: **for** phase $i = 1, 2, ...$ **do**
3: Reset the collision set $C_{jk}[i] = 0, \forall k \in [K]$;
4: Reset active arms set $\texttt{Ch}_j[i] = [K] \backslash G_j[i-1]$;
5: **if** $t < 2^i + NK(i-1)$ **then**
6: Local deletion $L_j[i] = \{k : C_{jk}[i] \geq \lceil \theta 2^i \rceil\}$;
7: Play arm $I_t(j) \in \underset{k \in \texttt{Ch}_j[i] \backslash L_j[i]}{\arg\max} \left( \hat{\mu}_{j,k}(t-1) + \sqrt{\frac{2\alpha \log(t)}{N_{j,k}(t-1)}} \right)$;
8: **if** $k = I_t(j)$ is successfully matched with agent $j$, i.e. $m_t(j) = k$ **then**
9: Update estimate $\hat{\mu}_{j,k}(t)$ and matching count $N_{j,k}(t)$;
10: **else**
11: $C_{jk}[i] = C_{jk}[i] + 1$;
12: **end if**
13: **else if** $t = 2^i + NK(i-1)$ **then**
14: $\mathcal{O}_j[i] \leftarrow$ most matched arm in phase $i$;
15: $G_j[i] \leftarrow$ *COMMUNICATION*$(i, \mathcal{O}_j[i])$;
16: **end if**
17: **end for**

with agent $j$, then the algorithm updates estimated reward $\hat{\mu}_{j,k}(t) = \frac{1}{N_{j,k}(t)} \sum_{s=1}^{t} 1\{I_s(j) = k \text{ and } M_s(j) = 1\} X_{j,k}(t)$ and $N_{j,k}(t)$ (line 9). Otherwise, it is regarded as the collision happens (line 11) and $j$ receives zero reward. The regret minimization block identifies the most played arm $\mathcal{O}_j[i]$ for agent $j$ in phase $i$. The intuition of regret minimization is to estimate the best arm for agent $j$ $\mathcal{O}_j[i] \in \texttt{Ch}_j[i]$ as the most matched arm and to make decisions to take action and minimize regret.

**Algorithm 2** COMMUNICATION

**Input:**
Phase number $i$, and most played arms $\mathcal{O}_j[i]$ for agent $j$, $\forall j \in [N]$ .
Set $\mathcal{C} = \emptyset$;
**for** $t = 1, 2, \cdots, NK - 1$ **do**
**if** $K(j-1) \leq t \leq Kj - 1$ **then**
Agent $j$ plays arm $I_t(j) = (t \mod K) + 1$;
**if** Collision Occurs **then**
$\mathcal{C} = \mathcal{C} \cup \{I_t(j)\}$;
**end if**
**else**
Play arm $I_t(j) = \mathcal{O}_j[i]$;
**end if**
**end for**
RETURN $\mathcal{C}$;

In the communication block (see Algorithm 2), there are $N$ sub-blocks, and the duration of each sub-block is $K$. In the $\ell - th$ sub-block, only the agent $\ell$ pulls arm 1, arm 2, $\cdots$, arm $K$ in round-robin while the rest of the agents pull their most preferred arm estimated as the most played one. This block aims to detect globally dominated arms for agent $j$: $\mathcal{C} \subset \{\mathcal{O}_{j'}[i] : j' \succ_{\mathcal{O}_{j'}[i]} j\}$. Under stable matching $m^*$, the globally dominated arms set for agent $j$ is denoted as $G_j^*$. After the communication block for phase $i$, each agent $j$ updates its active arms set $\texttt{Ch}_j[i+1]$ for phase $i+1$, by globally deleting arms set $G_j[i]$, and enters into the next phase (line 4 in Algorithm 1).

Hence, we set multi-phases to guarantee that the active set in different phases has no inclusion relationship so that if an agent deletes an arm in a certain phase, this arm can still be selected in the later rounds. This ensures that each agent will not permanently eliminate its stable matched arm, and when the agent mistakenly deletes an arm, it will not lead to linear regret.

# 4 Results

## 4.1 $\tilde{\alpha}$-condition

Constructing a unique stable matching plays an important role in market equilibrium [14] and some uniqueness conditions have been studied for many-to-one market [24, 2, 26]. In this section, we propose a new $\tilde{\alpha}$-condition to guarantee a unique stable result. First, we introduce *uniqueness consistency (Unqc)* [18], which guarantees robust and stable results in MO-UCB-D4 algorithm with arm deletion.

**Definition 1.** *A preference profile satisfies uniqueness consistency if and only if*

*(i) there exists a unique stable matching $m^*$;*

*(ii) for any subset of arms or agents, the restriction of the preference profile on this subset with their stable-matched pair has a unique stable matching.*

It guarantees that even if an arbitrary subset of agents are deleted out of the system with their respective stable matched arms, there still exists a unique stable matching among the remaining agents and arms. This condition allows any algorithm to identify at least one stable pair in the unique stable matching system and guide the system to the global unique stable matching in an iterative manner. To obtain robust stable matching results in our setting, we propose a new $\tilde{\alpha}$*-condition* in the many-to-one market to guarantee a unique stable matching with consistency, which is a sufficient and necessary condition for Unqc (proved in Appendix B).

Suppose that each agent has a ranking for arms, or to say that each arm corresponds to a sequence number, which is the *left order*. In the *left order*, the agents have their stable matched arm as the best arm once the stable matched arms for higher order agents are removed. Similarly, each arm will sort the agents, and in the *right order* the agents have their stable matched agents as most preferred agents within capacities once the stable matched agents set for higher ranked arms are removed. We considers a finite set of arms $[K] = \{1, 2, \cdots, K\}$ and a finite set of agents $j \in [N] = \{1, 2, \cdots, N\}$ with preference profile $\mathcal{P}$. Assume that $[N]_r$=$\{A_1, A_2, \cdots, A_N\}$ is a permutation of $\{1, 2, \cdots, N\}$ and $[K]_r$=$\{c_1, c_2, \cdots, c_K\}$ is a permutation of $\{1, 2, \cdots, K\}$, where the $k$-th arm in $[K]_r$ is the $c_k$-th arm in $[K]$, and the $j$-th agent in $[N]_r$ is the $A_j$-th agent in $[N]$. The main idea from one-to-one to many-to-one analysis is to replace individuals with sets. Considering arm capacity, we denote $\gamma^*(c_k)$ (right order) as the stable matched agents set for arm $c_k$.

**Definition 2.** *A many-to-one matching market satisfies the $\tilde{\alpha}$-condition if,*

*(i) The left order of agents and arms satisfies*

$$\forall j \in [N], \forall k > j, k \in [K], \mu_{j,m^*(j)} > \mu_{j,k}$$

*and $m^*(j)$ is agent $j$'s stable matched arm;*

*(ii) The right order of agents and arms satisfies*

$$\forall k < k' \leq K, c_k \in [K]_r, A_{k'} \subset [N]_r, \gamma^*(c_k) \succ_{c_k} A_{\sum_{i=1}^{k'-1}+1},$$

*where the set $\gamma^*(c_k)$ is more preferred than $A_{\sum_{i=1}^{k'-1}+1}$ means that the least preferred agent in $\gamma^*(c_k)$ for $c_k$ is better than $A_{\sum_{i=1}^{k'-1}+1}$ for $c_k$.*

Similar to $\alpha$*-condition* in marriage problem [18], the $\tilde{\alpha}$*-condition* detection contains two steps: First run the Gale–Shapley algorithm in many-to-one setting [11] and find a stable matching $m$. Next to find two orders of arms and agents by sequential elimination of agents with the most preferred matching pairs. If the algorithm fails to find two orders, then the $\tilde{\alpha}$*-condition* is violated. This condition is more general than existing uniqueness conditions like *SPC* [26] and is can recover the known $\alpha$-condition in one-to-one matching market [18]. The relationship between the existing uniqueness conditions and our proposed conditions will be analyzed in detail later in Section 4.3.

We show that under our $\tilde{\alpha}$*-condition*, when arm 1 to arm $k-1$ are removed, agents $\left(\sum_{i=1}^{k-1} q_i + 1\right)$ to $\left(\sum_{i=1}^{k} q_i\right)$ prefer $k$ most, and the *right order* satisfies that when $A_1$ to agents $A_{\sum_{i=1}^{k-1} q_i}$ are removed, arm $k$ prefers agents $\mathcal{A}_k = \{A_{\sum_{i=1}^{k-1} q_i+1}, A_{\sum_{i=1}^{k-1} q_i+2}, \cdots, A_{\sum_{i=1}^{k} q_i}\}$, which gives the concrete correspondence of *left order* and *right order*. In other words, $\mathcal{A}_k$ is the agents set that are most $q_k$ preferred by arm $k$ among those who have not been matched by arm $1, 2, \cdots, k-1$.

**Theorem 1.** *If a market $\mathcal{M} = (\mathcal{K}, \mathcal{J}, \mathcal{P})$ satisfies $\tilde{\alpha}$- condition, then we have $m^*(\sum_{i=1}^{j-1} q_i + 1) = m^*(\sum_{i=1}^{j-1} q_i + 2) = \cdots = m^*(\sum_{i=1}^{j} q_i) = j$ (left order), $\gamma^*(c_k) = \mathcal{A}_k$ and $m^*(\mathcal{A}_j) = c_j$ (right order).*

Under $\tilde{\alpha}$-condition, the stable matched arm may not be the most preferred one for agent $j$, $j \in [N]$. In other words, we do not have $m^*(j)$ to be dominated only by the agent 1 to agent $j-1$, i.e. there may exist $j' > j$, s.t. $j' \succ_{m^*(j)} j$. To connect two sides of the market, we define a mapping *lr* of agents in left order to agents in right order under $\tilde{\alpha}$-*condition* as the choice of the right order from arms to agents is a set. From the Theorem 1, the stable matched set for arm $k$ is its first $q_k$ favorite agents $\gamma^*(c_k) = \mathcal{A}_k$. Assume that the preference is strict and the first $q_k$ agents are ranked as $\mathcal{A}_k^{(1)} \succ \mathcal{A}_k^{(2)} \succ \cdots \mathcal{A}_k^{(q_k)}$. Then the rule of the mapping $lr$ we set is as follows: In the right order, the mapping for arm $k' \in [K]$ is defined as the least preferred one among its most preferred $q_{k'}$ agents, that is $A_{lr(k')} = \mathcal{A}_{k'}^{(q_{k'})}$. A concrete example to illustrate this definition is that suppose an agent $s$ is in the first $q_1$ favorite agents set of arm 1. Then in this mapping, arm 1 corresponds to $A_{q_1} = \mathcal{A}_1^{(q_1)}$ instead of other agents like $s$ in $\mathcal{A}_1$. Note that this mapping is not an injective, i.e. $\exists j, j'$, s.t. agent $j = A_{lr(j)} = A_{lr(j')}$. An intuitive representation can be seen in Figure 4 of Appendix A.1.

### 4.2 Theoretical Results of Regret

We then provide the theoretical results of MO-UCB-D4 algorithm under our $\tilde{\alpha}$-condition. Recall that $G_j^*$ is the globally dominated arms for agent $j$ under stable matching $m^*$. For each arm $k \notin G_j^*$, we give the definition of the *blocking agents* for arm $k$ and agent $j$: $\mathcal{B}_{jk} = \{j' : j' \succ_k j, k \notin G_j^*\}$, which contains agents more preferred by arm $k$ than $j$. The *hidden arms* for agent $j$ is $\mathcal{H}_j = \{k : k \notin G_j^*\} \cap \{k : \mathcal{B}_{jk} \neq \emptyset\}$. The reward gap for agent $j$ and arm $k$ is defined as $\Delta_{jk} = |\mu_{j,m^*(j)} - \mu_{j,k}|$ and the minimum reward gap across all arms and agents is $\Delta = \min\{|\mu_{j,k} - \mu_{j,k'}| : j \in [N], k, k' \in [K]\}$. We assume that the reward is different for each agent, thus $\Delta_{j,k} > 0$ for every agent $j$ and arm $k$.

**Theorem 2.** *(Upper bound of regret) Following MO-UCB-D4 algorithm with horizon $T$, the regret of a stable matching under $\tilde{\alpha}$-condition (Definition 2) for the agent $j \in [N]$, is upper bounded by*

$$\mathbb{E}\left[R_j(T)\right] \leq \sum_{k \notin G_j^* \cup m^*(j)} \frac{8\alpha}{\Delta_{jk}} \left(\log(T) + \sqrt{\frac{\pi}{\alpha}\log(T)}\right) + \sum_{k \notin G_j^*} \sum_{j' \in \mathcal{B}_{jk}: k \notin G_{j'}^*} \frac{8\alpha\mu_{j,m^*(j)}}{\Delta_{j'k}^2} \left(\log(T) + \sqrt{\frac{\pi}{\alpha}\log(T)}\right)$$
$$+ c_j \log_2(T) + O\left(\frac{N^2K^2}{\Delta^2} + \left(\min(1, \theta|\mathcal{H}_j|) f_\alpha(J_{\max}(j)\right) + f_\alpha(j) - 1)2^{i^*} + N^2 K i^*\right),$$

*where $i^* = \max(8, i_1, i_2)$ (then $i^* \leq 8$ and $i_1, i_2$ are defined in equation* (1)*), $f_\alpha(j) = j + lr_{\max}(j)$ and $lr_{\max}(j) = \{lr(j') : 1 \leq j' \leq j\}$.*

From the Theorem 2, the scale of the regret upper bound under $\tilde{\alpha}$-*condition* is $O\left(\frac{NK\log(T)}{\Delta^2}\right)$ and the proof is in Section 2.

**Proof Sketch of Theorem 2.** Under the guarantee of the unique stable condition, we only need to discuss the regret of the unique result. We construct a *good phase* (in appendix A.2) and denote that the time point of agent $j$ reaching her *good phase* by $\tau_j$. After $\tau_j$, agent $j$ could identify its best arm and matches with this stable pair. Thus, from phase $\tau_j$ on-wards, agent $j+1$ will find the set of globally dominated arms $G_{j+1}^*$ and will eliminate the arm $m^*(j)$ if $m^*(j)$ brings collision in communication block according to Algorithm 1. Then when agent $j$ enters into regret minimization block next phase, the number of times she plays a sub-optimal arm is small which leads to a small total number of collisions experienced by agent $j+1$. Then the process of each agent after *good phase* is divided into two stages: before $\tau_j$ and after $\tau_j$. Phases before $\tau_j$ can be bounded by induction. After $\tau_j$, according to the causes of regret, it is divided into four blocks: collision, local deletion, communication, and sub-optimal play. The regret decomposition is bound by the following.

**Lemma 1.** *(Regret Decomposition) For a stable matching under $\tilde{\alpha}$-condition, the upper bound of regret for the agent $j \in [N]$ under our algorithm can be decomposed by:*

$$\mathbb{E}\left[R_j(T)\right] \leq \underbrace{\mathbb{E}\left[S_{F_{\alpha j}}\right]}_{\textit{(Regret before phase } F_{\alpha j})} + \underbrace{\min(\theta|\mathcal{H}_j|, 1)\mathbb{E}\left[S_{V_{\alpha j}}\right]}_{(\textit{Local deletion})} + \underbrace{\left((K - 1 + |\mathcal{B}_{jm^*(j)}|)\log_2(T) + NK\mathbb{E}\left[V_{\alpha j}\right]\right)}_{(\textit{Communication})}$$

$$+\underbrace{\sum_{k\notin G_j^*}\sum_{j'\in\mathcal{B}_{jk}:k\notin G_{j'}^*}\frac{8\alpha\mu_{j,m^*(j)}}{\Delta_{j'k}^2}\left(\log(T)+\sqrt{\frac{\pi}{\alpha}\log(T)}\right)}_{(\textit{Collision})}$$

$$+\underbrace{\sum_{k\notin G_j^*\cup m^*(j)}\frac{8\alpha}{\Delta_{jk}}(\log(T)+\sqrt{\frac{\pi}{\alpha}\log(T)})}_{(\textit{Sub-optimal play})}+NK\left(1+(\phi(\alpha)+1)\frac{8\alpha}{\Delta^2}\right),$$

*where $F_{\alpha j}$, $V_{\alpha j}$ are the time points when agent $j$ enters into $\tilde{\alpha}$-Good phase and $\tilde{\alpha}$-Low Collision phase respectively, mentioned as "good phase" above, are defined in Appendix A.2.*

### 4.3 Conditions for Unique Stable matching

Uniqueness consistency (Unqc) leads the stable matching to a robust one which is a desirable property in large dynamic markets with individual departure and heterogeneous preferences [6]. A precondition of Unqc is to ensure global unique stability, hence finding uniqueness conditions is essential.

The existing unique stable conditions are well established in one-to-one setting ( analysis can be found in Appendix B), and in this section, we focus on uniqueness conditions in many-to-one market, such as *SPC*, [26], *Aligned Preference*, *Serial Dictatorship Top-top match* and *Acyclicity* [24, 2, 26] (Definition 9, 7, 8, 10 in Appendix B.2), and the relationship between different existing conditions and our new $\tilde{\alpha}$ condition can be intuitively expressed in figure 1 and will be proved in Appendix B.2.

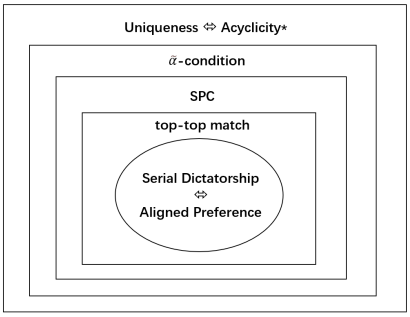


Figure 1: Relations of Uniqueness Conditions in Many-to-one.

**Lemma 2.** *In a many-to-one matching market $\mathcal{M}=(\mathcal{K},\mathcal{J},\mathcal{P})$, both Serial Dictatorship and Aligned Preference can produce a unique stable matching in many-to-one market and they are equivalent.*

Takashi Akahoshi [2] proposes a necessary and sufficient condition for uniqueness of stable matching in many-to-one matching where unacceptable agents and arms may exist on both sides. We denote their condition as *Acyclicity*$^*$. Under our setting, both two sides are acceptable, and we first give the proof of that *Acyclicity*$^*$ is a necessary and sufficient condition for uniqueness in our setting (see Section B.2.3 in Appendix B). We then give the relationships between our newly $\tilde{\alpha}$-condition and other uniqueness conditions, and we give proof for this section in Appendix B.

**Theorem 3.** *Under the preference profile of market $(\mathcal{K},\mathcal{J},\mathcal{P})$, our $\tilde{\alpha}$-condition satisfies:*

*(i) SPC is a sufficient condition to $\tilde{\alpha}$-condition;*

*(ii) $\tilde{\alpha}$-condition is a necessary and sufficient condition to Unqc.*

## 5 Difficulties and Solutions

### 5.1 Unique stable matching under $\tilde{\alpha}$-condition

While putting forward our $\tilde{\alpha}$-condition in the many-to-one setting, many new problems need to be taken into account. We first consider the problems encountered in the transition from *SPC* to

$\tilde{\alpha}$-condition. The new uniqueness condition, $\tilde{\alpha}$-condition, is more general than existing *SPC*. When we use the events decomposition for regret minimization block to prove the bound inequality of the number of times agent $j$ is pulled (Lemma 6), it requires that $m^*(j)$ always exit and will not be deleted. Under *SPC* condition, $m^*(j)$ always exits as the stable matched partner is the most preferred one among the remaining market for the certain agent while $\tilde{\alpha}$-condition cannot guarantee this property. Hence, it is important to find conditions or a certain phase with good properties to guarantee that $m^*(j)$ will not be globally deleted or locally deleted. And we consider $F_{\alpha_j}$ and $V_{\alpha_j}$ in Lemma 3 (in Appendix A.2) to solve this problem. And since the stable matched pair is not top-top match in the remaining system under $\tilde{\alpha}$-condition while the answer is true under *SPC*, we introduce a new mapping (Figure 4) to describe the corresponding relationships of stable pairs. In addition, as shown in Figure 1, $Acyclicity^*$ is the weakest condition to ensure uniqueness up to now, and Bettina Klaus and Flip Klijn [19] point that acyclicity has a tight connection with consistency. Hence, whether we can further weaken $\tilde{\alpha}$-condition and propose a new algorithm remains to study.

### 5.2 From one-to-one setting to many-to-one setting.

We first focus on some difficulties of this extension. First, in the many-to-one matching, vacant seats within arm capacity $q$ are not $q$ independent individuals with the same preference since competitions among different seats of this arm exist. And an arm can reject the previously accepted agents when it faces a more preferred agent. Secondly, in communication block, when two agents select one arm at the same time, as an arm can accept more than one agent at a time, these two cannot distinguish who is more preferred by it, while it can be done in one-to-one markets. Thus it is difficult to identify arm preferences. Bounding the regret bound is bounding the time that an agent identifies its stable pair, thus capacities hindering communication among agents will bring more regret. Third, when describing the stable agents set that the arm matched with agent $j$ finally matches with, it is not a simple case as the one-to-one setting. We simply use arrows to indicate the left order and the right order. In the one-to-one setting, agent $j_1 \rightleftharpoons k_1$ implies that $j_1 \to k_1 \to j_1$, while $j_1 \rightleftharpoons k_1$ in many-to-one means that $j_1 \to k_1 \to$ an agent set that contains $j_1$. We need to define a mapping that connects the matched pairs. Another question is how capacity influences algorithm and stable regret.

In order to solve these problems, we illustrate from the following aspects: First, we take the capacity into account in the algorithm design, and for communication block, we introduce an arms set $G_j^*$ to represent the arms that will block agent $j$ globally, which will be deleted before each phase. Secondly, the idea from one-to-one to many-to-one is the transformation from individual to set. It is natural to think about how to split sets or how to correspond sets to individuals. In analysis, we give a new mapping (defined in Figure 4) by mapping the arms to the least preferred one of its stable matched agents in the right order, thus giving the matching relationship between individuals and individuals. After defining this mapping, we construct $lr_{\max}(j)$ for agent $j$ as the mapping of agent $j$ in the left order to agents in the right order under $\tilde{\alpha}$-condition. $lr_{\max}(j)$ in regret bound (Theorem 2) is implicitly related to capacity, since regret bound is essentially bounding the time each agent finds its stable matched arm. The larger the capacity, the smaller the index $k$ of its stable matched arm $k$, the smaller the $lr_{\max}(j)$, and then the regret changes.

## 6 Experiments

In this section, we verify the results of our MO-UCB-D4 algorithm (Algorithm 1) for decentralized many-to-one matching markets through experiments. The preference is sampled under certain conditions. We set the reward value towards the least preferred arm to be $1/N$ and the most preferred one as 1 for each agent, then the reward gap between any adjacently ranked arms is $\Delta = 1/N$. The reward for agent $j$ matches with arm $k$ at time $t$ $X_{j,k}(t)$ is sampled from $\mathrm{Ber}(\mu_{j,k})$. The capacity $q$ is set as $N/K$. We investigate how the cumulative regret and cumulative market unstability depend on the size of the market and the number of arms under three different unique stability conditions: *Serial Dictatorship*, *SPC*, $\tilde{\alpha}$*-condition*. The former cumulative regret is the total mean reward gap between the stable matching result and the simulated result, and the latter cumulative unstability is defined as the number of unstable matchings in round $t$. In our experiments, all results are averaged over 10 independent runs, hence the error bars are computed as standard deviations divided by $\sqrt{10}$.

**Varying the market size** To test effect on cumulative regret and cumulative stability, we first varying $N$ with fixed $K$ with market size of $N \in \{10, 20, 30, 40\}$ agents and $K = 5$ arms. The

number of rounds is set to be $100,000$. The cumulative regret in Figure 2(a)(c)(e) show an increasing trend with convergence as the number of agents increases under the three conditions. When the number of agents increases, there is a high probability of collisions among different agents, resulting in the increase of cumulative regret. Similar results for cumulative unstability are shown in Figure 2(b)(d)(f). When $N$ keeps increasing, the number of unstable pairs becomes more. With the increase of the number of rounds, both the cumulative regret and the cumulative unstability increase first and then tends to be stable and the jumping points are caused by multi-phases of the MO-UCB-D4 Algorithm.

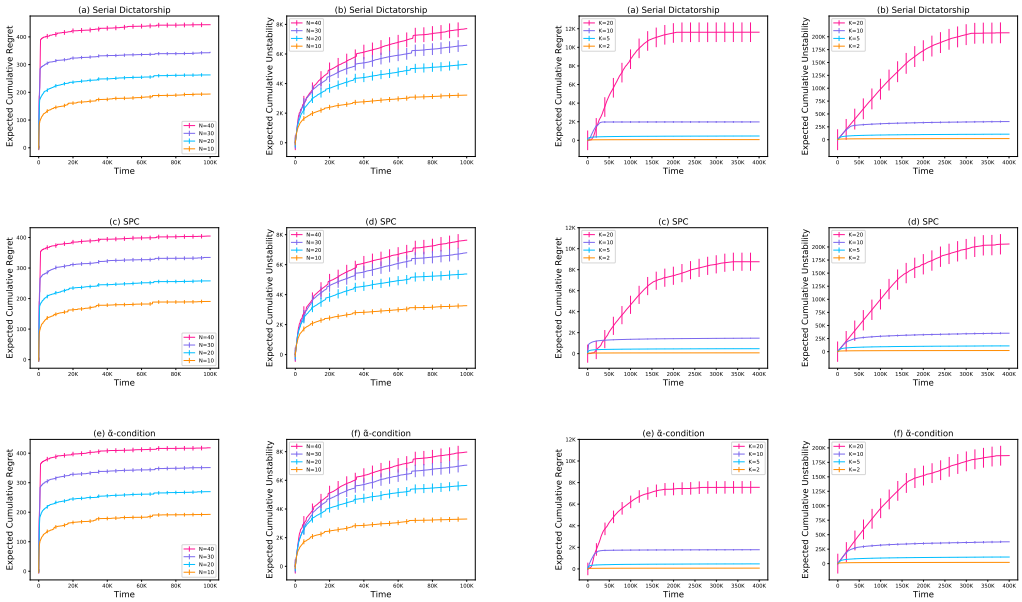


Figure 2: Cumulative regret and cumulative unstability of MO-UCB-D4 of size with $N \in \{10, 20, 30, 40\}$ and the number of arms $K = 5$ under *Serial Dictatorship*, *SPC*, $\tilde{\alpha}$-*condition*.




Figure 3: Cumulative regret and cumulative unstability of MO-UCB-D4 of size with $K \in \{2, 5, 10, 20\}$ under *Serial Dictatorship*, *SPC*, $\tilde{\alpha}$-*condition*.

**Varying arm capacity** The number of arms $K$ is chosen by $K \in \{2, 5, 10, 20\}$, with $N = 20$ and $q = N/K$. The number of rounds we set is $400,000$. With the increase of $K$, both the cumulative regret in Figure 3(a)(c)(e) and the cumulative unstability in Figure 3(b)(d)(f) increase monotonously. When $K$ increases, the capacity $q$ for each arm decreases, and then the number of collisions will increase, which leads to the increase of cumulative regret. And it also leads to more unstable pairs, which needs more communication phases to converge to the stable matching. Under these three conditions, the performances of the algorithm are similar.

## 7 Conclusion

We are the first to study the bandit algorithm for the many-to-one matching market under the unique stable matching. This work focuses on a decentralized market. A new $\tilde{\alpha}$-*condition* is proposed to guarantee a unique stable outcome in many-to-one market, which is more general than existing uniqueness conditions like *SPC*, *Serial Dictatorship* and could recover the usual $\alpha$-*condition* in one-to-one setting. We propose a phase-based algorithm of MO-UCB-D4 with arm-elimination, which obtains $O\left(\frac{NK\log(T)}{\Delta^2}\right)$ stable regret under $\tilde{\alpha}$-*condition*. By carefully defining a mapping from arms to the least preferred agent in its stable matched set, we could effectively correspond arms and agents by individual-to-individual. A series of experiments under two environments of varying the market size and varying arm capacity are conducted. The results show that our algorithm performs well under *Serial Dictatorship*, *SPC* and $\tilde{\alpha}$-*condition* respectively.

# References


[1] Azar Abizada. Stability and incentives for college admissions with budget constraints. *Theoretical Economics*, 11(2):735–756, 2016.

[2] Takashi Akahoshi. Singleton core in many-to-one matching problems. *Mathematical Social Sciences*, 72:7–13, 2014.

[3] Ahmet Altinok. Dynamic many-to-one matching. *Available at SSRN 3526522*, 2019.

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

[5] Sophie Bade. Random serial dictatorship: the one and only. *Mathematics of Operations Research*, 45(1):353–368, 2020.

[6] Soumya Basu, Karthik Abinav Sankararaman, and Abishek Sankararaman. Beyond $log^2(t)$ regret for decentralized bandits in matching markets. In *International Conference on Machine Learning*, pages 705–715, 2021.

[7] Anna Bogomolnaia and Hervé Moulin. A new solution to the random assignment problem. *Journal of Economic theory*, 100(2):295–328, 2001.

[8] Somouaoga Bonkoungou. Decentralized college admissions under single application. *Review of Economic Design*, 25(1):65–91, 2021.

[9] Simon Clark. The uniqueness of stable matchings. *Contributions in Theoretical Economics*, 6(1), 2006.

[10] Jan Eeckhout. On the uniqueness of stable marriage matchings. *Economics Letters*, 69(1):1–8, 2000.

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

[12] Aurélien Garivier, Tor Lattimore, and Emilie Kaufmann. On explore-then-commit strategies. *Advances in Neural Information Processing Systems*, 29:784–792, 2016.

[13] Virginia Gunn, Bertina Kreshpaj, Nuria Matilla-Santander, Emilia F Vignola, David H Wegman, Christer Hogstedt, Emily Q Ahonen, Theo Bodin, Cecilia Orellana, Sherry Baron, et al. Initiatives addressing precarious employment and its effects on workers' health and well-being: A systematic review. *International Journal of Environmental Research and Public Health*, 19(4):2232, 2022.

[14] Gregory Z Gutin, Philip R Neary, and Anders Yeo. Unique stable matchings. *arXiv preprint arXiv:2106.12977*, 2021.

[15] Guillaume Haeringer and Flip Klijn. Constrained school choice. *Journal of Economic theory*, 144(5):1921–1947, 2009.

[16] John William Hatfield, Fuhito Kojima, and Scott Duke Kominers. Investment incentives in labor market matching. *American Economic Review*, 104(5):436–41, 2014.

[17] Ramesh Johari, Vijay Kamble, and Yash Kanoria. Matching while learning. *Operations Research*, 69(2):655–681, 2021.

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

[19] Bettina Klaus and Flip Klijn. Local and global consistency properties for student placement. *Journal of Mathematical Economics*, 49(3):222–229, 2013.

[20] Lydia T Liu, Horia Mania, and Michael Jordan. Competing bandits in matching markets. In *International Conference on Artificial Intelligence and Statistics*, pages 1618–1628. PMLR, 2020.

[21] Lydia T Liu, Feng Ruan, Horia Mania, and Michael I Jordan. Bandit learning in decentralized matching markets. *arXiv preprint arXiv:2012.07348*, 2020.

[22] Jinpeng Ma. The singleton core in the college admissions problem and its application to the national resident matching program (nrmp). *Games and Economic Behavior*, 69(1):150–164, 2010.

[23] Onkar Malgonde, He Zhang, Balaji Padmanabhan, and Moez Limayem. Taming complexity in search matching: Two-sided recommender systems on digital platforms. *Mis Quarterly*, 44(1), 2020.

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

[25] Jaeok Park. Competitive equilibrium and singleton cores in generalized matching problems. *International Journal of Game Theory*, 46(2):487–509, 2017.

[26] Philip J Reny. A simple sufficient condition for a unique and student-efficient stable matching in the college admissions problem. *Economic Theory Bulletin*, 9(1):7–9, 2021.

[27] Antonio Romero-Medina and Matteo Triossi. Acyclicity and singleton cores in matching markets. *Economics Letters*, 118(1):237–239, 2013.

[28] Alvin E Roth. On the allocation of residents to rural hospitals: a general property of two-sided matching markets. *Econometrica: Journal of the Econometric Society*, pages 425–427, 1986.

[29] Alvin E Roth and Marilda Sotomayor. Two-sided matching. *Handbook of game theory with economic applications*, 1:485–541, 1992.

[30] Hannu Salonen and Mikko AA Salonen. Mutually best matches. *Mathematical Social Sciences*, 91:42–50, 2018.

[31] 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.

[32] Jay Sethuraman, Chung-Piaw Teo, Liwen Qian, et al. Many-to-one stable matching: Geometry and fairness. *Mathematics of Operations Research*, 31(3):581–596, 2006.

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

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Please see Abstract and Section 1.
   (b) Did you describe the limitations of your work? [Yes] Please see Section C.3.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This work mainly focuses on the online learning theory, which does not have any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Please see Section 2.
   (b) Did you include complete proofs of all theoretical results? [Yes] Please see Appendix.
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Please see supplemental material.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section 6 and supplemental material.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Please see Section 6.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]