Bandit Learning in Many-to-One Matching Markets

Zilong Wang
wangzilong@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Liya Guo
19020182203584@stu.xmu.edu.cn
Xiamen University
Xiamen, China

Junming Yin
junmingy@cmu.edu
Tepper School of Business, Carnegie Mellon University
Pittsburgh, United States

Shuai Li*
shuaili8@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

ABSTRACT

The problem of two-sided matching markets is well-studied in social science and economics. Some recent works study how to match while learning the unknown preferences of agents in one-to-one matching markets. However, in many cases like the online recruitment platform for short-term workers, a company can select more than one agent while an agent can only select one company at a time. These short-term workers try many times in different companies to find the most suitable jobs for them. Thus we consider a more general bandit learning problem in many-to-one matching markets where each arm has a fixed capacity and agents make choices with multiple rounds of iterations. We develop algorithms in both centralized and decentralized settings and prove regret bounds of order $O(\log T)$ and $O(\log^2 T)$ respectively. Extensive experiments show the convergence and effectiveness of our algorithms.

CCS CONCEPTS

• **Theory of computation** → **Regret bounds**; **Online learning algorithms**; • **Applied computing** → **Economics**.

KEYWORDS

matching markets, multi-armed bandit, many-to-one setting

ACM Reference Format:
Zilong Wang, Liya Guo, Junming Yin, and Shuai Li. 2022. Bandit Learning in Many-to-One Matching Markets. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/101145/3511808.3557248

1 INTRODUCTION

Recent growth of online communication has resulted in an expansion of opportunities for companies to participate in personalized decision-making. 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 as an extensive probation period before moving workers into permanent employment. The rise of matching for supply and demand in bilateral markets makes policy-making be customized on the basis of diversified needs for both sides of the market. A basic framework of the two-sided matching market is that there are two sides, and each side has a preference profile over the agents on the opposite side. A key feature in the design of matching markets is obtaining stable outcomes [30, 33] since any unstable matching result might do damage to the whole system. We say a matching is *stable* if there exists no blocking agents pair, who would mutually prefer each other to their current matched pair. It is worth mentioning that the stable matching is typically not unique. Gale–Shapley algorithm [11] is a common method to find a stable outcome that is optimal for one side.

Motivated by online short-term labor markets where the preferences from one side may be unknown in advance, we study how to match while learning the preferences. The goal of this problem is to find an adaptive policy for choosing arms with unknown reward distributions due to the uncertainty of preferences. As a known learning framework, the multi-armed bandit (MAB) model [39] has been widely studied in many problems recently [9, 19, 20, 23, 40], and it is one of the important tools for matching market. The combination of the bandit algorithm with two-sided matching markets, originally proposed by Das and Kamenica [8], can be typically viewed as a stylized abstraction of a platform where N agents are matched to K arms with limited resources and unknown reward distributions. Agents learn the distribution iteratively through receiving rewards in order to maximize their expected reward or minimize their regret. The one-to-one matching market with bandits has been studied for a period of time, which describes a market where each agent can choose one arm to pull, and each arm can select only one agent. In the one-to-one matching, two common settings divided by whether there is a central platform to arrange matchings are centralized and decentralized markets, and previous work develop bandit models for these two respectively [3, 7, 19, 25, 26].

However, it is limited to consider only the one-to-one setting since such a model is not applicable in many scenarios like online labor market and Internet-based educational platforms [13, 14, 27]. In our work, we study many-to-one matching markets in which one arm can match more than one agent. The maximum number that one arm can accommodate is called its *capacity*. Among all agents

*Corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/101145/3511808.3557248

who select arm a_j, the most preferred agents within its capacity can successfully receive rewards and other agents are regarded as to have collisions. In the short-term online recruitment platform with numerous similar short-term tasks or internships, companies are abstracted as arms, which can accept multiple workers, while workers are regarded as agents, and each can only choose one company to submit resumes at a time. Each company scores each recruited employee according to the needs of the company and the score is assumed to be known and fixed. Workers have no knowledge of the preference of companies and the reward for them is a comprehensive consideration of salary and company environment. Since the work task is short-term, each worker can try many times in different companies to choose the most suitable job and competes with other candidates during application. This motivates us to study the many-to-one matching market with multiple rounds of iterations. As mentioned in [32], the many-to-one matching problem is not equivalent to the stable one-to-one marriage problem and we will discuss the difference in the later analysis.

Our work introduces a novel model for many-to-one matching problems with bandit feedback in both centralized and decentralized settings. The decentralized here assumes that agents can only observe their reward and successfully matched pairs of other agents each round. Our model has two salient features, namely, from one-to-one to many-to-one, and the application of MAB in market matching. First, there need more considerations from one-to-one to many-to-one setting. As the capacity of each arm is more than one, agents may not collide when selecting the same arm thus their reward will be non-zero, which will hinder them to learn more about the agents set each arm is more preferred to accept, hence hinder to formulate better policies to reduce collisions in the later rounds. Second, to demonstrate the versatility of our methodology, we design three new algorithms: centralized ETC, centralized UCB, and MOCA-UCB based on MAB policies respectively in centralized and decentralized settings. Both the agent-optimal regret for the centralized ETC algorithm and the agent-pessimal regret for the centralized UCB algorithm attain a $O(\log(T))$ regret upper bound, which are the same as the one-to-one setting, and our MOCA-UCB achieves a $O(\log^2(T))$ agent-pessimal regret. Extensive experiments show that our algorithm achieves uniform good performance.

2 RELATED WORK

MAB has received a lot of attention since Thompson [39] puts forward related concepts, and ETC, UCB are two typical bandit algorithms [2, 12]. It is an important tool for the online matching platform, which additionally adds the preference to the arms side compared with the general bandit problem. Market matching problems are often based on two main settings: centralized and decentralized where the difference between them is whether there is a medium to collect information and arrange agents' actions.

Centralized. So far the literature on centralized matching [10, 34] has a relatively complete theory where the platform can effectively reduce the collision. Lee [22] studies the manipulability of centralized stable matching mechanisms for both one-to-one and many-to-one settings with utilities. In the one-to-one setting, Liu et al. [25] apply ETC and UCB algorithm to stable matching and receive a $O(\log(T))$ regret upper bound. In the many-to-one matching market, Johari et al. [15] apply bandit algorithm in the dynamic market with known rewards, while the reward is unknown in our work and needs to be learned. Moreover, it aims to maximize the steady-state rate of payoff accumulation without considering stability, while our goal is to form stable matching and minimize the stable regret.

Decentralized. Under the decentralized assumption where there is no platform for agents to formulate policies, collisions among agents are inevitable and agents will receive zero rewards when collisions happen [24, 38]. In the one-to-one matching market with unknown preferences, constructing stable matchings is a key feature [8, 16, 29]. The work [4] then designs the first algorithm that achieves a poly-logarithmic regret $O\left(\log^2 T\right)$ in the fully distributed setting without communication among agents. Following these, two typical bandit algorithms, ETC and UCB, are combined with matching respectively to make decisions for agents under one-to-one decentralization setting [25, 26]. Thompson sampling (TS) is another popular bandit algorithm due to its good empirical performances, and it is also studied combined with one-to-one matching [18]. Under uniqueness condition, Sankararaman et al. [36] devise a phased UCB-D3 algorithm. Improving the previous works, a phased UCB-D4 algorithm with arm elimination is proposed under some more general uniqueness conditions [3]. The previous works mostly focus on one-to-one setting, and formulating policies in many-to-one is a natural extension. Nguyen et al. [30] develops an iterative rounding algorithm that relaxes capacity constraints in order to find a pairwise stable result in the many-to-one fractional matching with general preferences. Both sides of the market may continue to have new individuals join and Johari et al. [15] give a many-to-one optimization strategy under dynamic matching.

Motivated by [25, 26], we consider a general bandit model for many-to-one setting both in centralized and decentralized setting.

3 PROBLEM SETTING

3.1 Preliminaries

Suppose there are N agents and K arms. Like previous work [25], each arm a_j is assumed to have a fixed known preference ranking π_j over agents, and $\pi_j(i)$ is the rank of agent p_i in the arm a_j's preference. For example, each short-time worker's personal ability is different, and the job fields that each candidate is good at are also different. Each company will make a ranking for each worker according to the company's own needs, representing its preference over workers. And this ranking will remain unchanged for a period of time since the workers' personal abilities and companies' interests will not change too fast. Denote $p_i \succ_{a_j} p_{i'}$ if arm a_j prefers agent p_i over $p_{i'}$, and similarly denote $a_j \succ_{p_i} a_{j'}$ if agent p_i prefers arm a_j over $a_{j'}$. Here for the many-to-one setting, each arm a_j has a fixed *capacity* $c_j \geq 1$. For simplicity, we assume that $N \leq \sum_{j=1}^{K} c_j$, similar to previous work [3, 25, 36] in one-to-one setting.

Recall that a matching is *stable* in the one-to-one matching market if there is no pair of an agent and an arm, or so-called *blocking pair*, that prefer each other over the current matched partner [34]. Dislike one-to-one markets where stable matchings always exist [11], empty sets might appear in the stable matching under many-to-one setting [34]. We incorporate a reasonable assumption of

individually rationality (IR) that can refrain from this unexpected situation [34]. This assumption states that $a_j \succ_{p_i} \emptyset$ 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 of an agent and an arm that the agent prefers this arm over the matched one and the arm prefers this agent over one of its matched agents [35, 37].

Among all stable matchings, the one that is the most preferred by all agents is called agent-optimal stable matching. Similarly, the one that is the least preferred by all agents is called the agent-pessimal stable matching. And the agent-pessimal stable matching is the most preferred by arms among all stable matchings [28, 31, 34]. A commonly adopted algorithm of student-applying deferred acceptance (SADA) could produce an agent-optimal stable matching for many-to-one setting [11][1]. Like the Gale-Shapley algorithm in the one-to-one setting, all agents first select their favorite arms. An arm with capacity c accepts her favorite c applicants, or all applicants if there are less than c, and rejects the rest agents. Rejected agents then select their next favorite arms in the next round while arms always keep their top choices and reject the rest. Accepted agents remain their choices in the next round. Such a procedure terminates when all agents are matched with their choices successfully.

3.2 Online Setting

Our goal is to learn stable matchings through online interactions between agents and arms.

At each time t, a random reward $X_{t,i,j} \in [0, 1]$ is independently drawn from a fixed distribution with mean $\mu_{i,j}$ for every agent p_i and arm a_j. Note that this distribution is determined by the features of agent p_i and arm a_j, such as the payment of a job, thus it is independent from other distribution and fixed. Each agent p_i selects an arm $m_t(i)$ to match. If multiple agents select arm a_j at the same time, only top c_j agents can successfully match. The matched agent p_i will observe the reward $X_{t,i,m_t(i)}$ while the unmatched agent will only receive the notification of *collision*. We use $I_t(i)$ to denote whether agent p_i is successfully matched with her selected arm. Then the reward obtained by agent p_i is $X_{t,i,m_t(i)}\ I_t(i)$.

We consider both centralized and decentralized matching markets similar to previous work [25, 26]. In the centralized setting, there is a central platform to synchronize all agents such that collisions can be avoided. In the decentralized setting, each agent can only observe its own reward and the resulting matching of all agents by the end of each time, while she cannot see others' rewards. Collisions between agents would frequently occur as there is no platform to arrange all agents.

We denote the agent-optimal stable matching by $\bar{m}$, a function mapping from agents to arms; similarly we denote the agent-pessimal stable matching by $\underline{m}$. Similar to previous work [25, 26], we consider both agent-optimal stable regret and agent-pessimal stable regret for every agent p_i, that is the difference of expected reward from the reward of the corresponding stable matching,

$$\bar{R}_{T,i} := T\mu_{i,\bar{m}(i)} - \sum_{t=1}^{T} \mathbb{E}[X_{t,i,m_t(i)}\ I_t(i)]\,,$$

$$\underline{R}_{T,i} := T\mu_{i,\underline{m}(i)} - \sum_{t=1}^{T} \mathbb{E}[X_{t,i,m_t(i)}\ I_t(i)]\,.$$

When the true preferences are known, the SADA algorithm [11] could output an agent-optimal stable matching, which might not be achieved [25, 26] in online setting. Thus we only provide guarantees for the agent-pessimal stable regret for our online setting.

4 CENTRALIZED MARKET

This section focuses on the centralized market where there is a platform to arrange the matching for all agents at each time. We apply two commonly used methods of explore-then-commit algorithm (ETC) and upper-confidence-bound algorithm (UCB) to this setting and study their effectiveness.

4.1 Centralized ETC Algorithm

We first introduce the centralized ETC (cenETC) algorithm (Algorithm 1). The platform first arranges an exploration phase where matchings are arranged in a round-and-robin manner. Note that each arm a_j has a capacity c_j, thus formulating $\sum_{j=1}^{K} c_j =: C$ budgets in total. For each index $c \in [C]$, form C budgets as C seats in a circle, let agent p_1 take the c-th seat, and other agents follow to take seat one-by-one (when some agent takes the C-th seat, next one would take the 1-st seat). Thus every index c will correspond to a different matching and when c traverses $[C]$, both agents and arms would be matched with the other side uniformly. The platform arranges hC rounds for exploration (line 3 - 4) where each agent is matched to a_j arm hc_j times. After exploration, the estimated reward of agent p_i for arm a_j can be computed as

$$\hat{\mu}_{i,j} = \frac{1}{hc_j} \sum_{t=1}^{hC} 1\{m_t(i) = j\}\ X_{t,i,j}\,. \tag{1}$$

Note that with this central design, there is no collision, i.e. $I_t(i) \equiv 1$. According to the estimated rewards, the empirical ranking $\hat{r}_i$ for each agent i is the decreasing order based on $\{\hat{\mu}_{i,j} : j \in [K]\}$ among arms, that is for any two different arms j and j',

$$\hat{r}_i(j) < \hat{r}_i(j') \Leftrightarrow \hat{\mu}_{i,j} > \hat{\mu}_{i,j'}\,. \tag{2}$$

Then the platform sticks to this agent-optimal matching for these empirical rankings (line 9), which can be computed by the SADA algorithm (line 7).

To analyze the regret of the centralized ETC algorithm, we first introduce two useful lemmas. Similar to previous work [25], we say the empirical ranking $\hat{r}_i$ of agent p_i is *valid* if an arm a_j is ranked higher than $\bar{m}(i)$, i.e. $\hat{r}_{i,j} < \hat{r}_{i,\bar{m}(i)}$, it follows that $\mu_{i,j} > \mu_{i,\bar{m}(i)}$. This valid ranking is a sufficient condition to ensure the matching result is no worse than $\bar{m}$ (and thus no incurred regret) under SADA algorithm, which will be proved to happen with high probability.

Lemma 4.1. *If rankings of all agents are valid, then the SADA algorithm finds a matching m that performs at least as good as $\bar{m}$ under the true rankings also, i.e., $\mu_{i,m(i)} \geq \mu_{i,\bar{m}(i)}$.*

[1] Note that we only focus on the deferred acceptance algorithm proposed by agents even though proposal by the arm side can also produce agent-pessimal stable matching [17].

Algorithm 1 Centralized ETC

Input: The preference ranking π_j and the capacity c_j for each arm a_j, exploration parameter h, and the horizon T.
for $t = 1, 2, ..., T$ **do**
  if $t \le hC$ **then**
    $m_t(i) \leftarrow a_j$ where j satisfies

$$\sum_{j'=1}^{j-1} c_{j'} < ((t+i-2) \mod C) + 1 \le \sum_{j'=1}^{j} c_{j'} ;$$

  else if $t = hC + 1$ **then**
    Calculate the empirical reward $\hat{\mu}_{i,j}$ by (1) and ranking $\hat{r}_i$ by (2) for every agent p_i ;
    Run SADA algorithm to compute the agent-optimal stable matching m_t with respect to $\{\hat{r}_i\}_i$ and $\{\pi_j\}_j$;
  else
    $m_t \leftarrow m_{hC+1}$;
  end if
  Perform matching m_t and receive reward $X_{t,i,m_t(i)}$ for each agent p_i .
end for

Proof. Since the SADA algorithm outputs the agent-optimal stable matching, it only needs to show that $\bar{m}$ is a stable matching under valid rankings. Let a_j be an arm that satisfies $\hat{r}_i(j) < \hat{r}_i(\bar{m}(i))$ for agent p_i under the valid ranking $\hat{r}_i$. Since $\hat{r}_i$ is a valid ranking, a_j also ranks higher under the true preference, i.e., $\mu_{i,j} > \mu_{i,\bar{m}(i)}$. Since $\bar{m}$ is stable under true preference, arm a_j prefers every matched agent in $\bar{m}$ over p_i. Thus agent p_i and arm a_j can not form a blocking pair under ranking $\{\hat{r}_i\}_i$. Such a conclusion holds for any agent, thus $\bar{m}$ is stable under $\hat{r}$, which concludes the proof. □

Lemma 4.2. *For agent p_i, let $\bar{\Delta}_{i,j} = \mu_{i,\bar{m}(i)} - \mu_{i,j}$ and $\bar{\Delta}_{i,\min} = \min_{j:\bar{\Delta}_{i,j}>0} \bar{\Delta}_{i,j}$. After hC rounds of exploration, there is*

$$\mathbb{P}\left(\hat{r}_i \text{ is invalid}\right) \le \sum_{j=1}^{K} \exp(-hc_j\bar{\Delta}_{i,\min}^2/4) .$$

Proof. If the ranking is invalid, there must exist some arm a_j such that $\mu_{i,\bar{m}(i)} > \mu_{i,j}$ but $\hat{r}_i(j) < \hat{r}_i(\bar{m}(i))$, or equivalently $\hat{\mu}_{i,j} > \hat{\mu}_{i,\bar{m}(i)}$. Then there is

$$\begin{aligned}
&\mathbb{P}\left(\hat{r}_i \text{ is invalid}\right)\\
&\le \sum_{j=1}^{K} \mathbb{P}\left(\hat{\mu}_{i,j} \ge \hat{\mu}_{i,\bar{m}(i)}, \mu_{i,j} < \mu_{i,\bar{m}(i)}\right) = \sum_{j:\bar{\Delta}_{i,j}>0} \mathbb{P}\left(\hat{\mu}_{i,j} \ge \hat{\mu}_{i,\bar{m}(i)}\right)\\
&\le \sum_{j:\bar{\Delta}_{i,j}>0} \mathbb{P}\left(\left(\hat{\mu}_{i,\bar{m}(i)} - \mu_{i,\bar{m}(i)}\right) - \left(\hat{\mu}_{i,j} - \mu_{i,j}\right) \le \mu_{i,j} - \mu_{i,\bar{m}(i)}\right)\\
&\le \sum_{j:\bar{\Delta}_{i,j}>0} \mathbb{P}\left(\left(\hat{\mu}_{i,\bar{m}(i)} - \mu_{i,\bar{m}(i)}\right) - \left(\hat{\mu}_{i,j} - \mu_{i,j}\right) \le -\bar{\Delta}_{i,\min}\right)\\
&\le \sum_{j:\bar{\Delta}_{i,j}>0} \exp(-hc_j\bar{\Delta}_{i,\min}^2/4) \le \sum_{j=1}^{K} \exp(-hc_j\bar{\Delta}_{i,\min}^2/4) .
\end{aligned}$$

The penultimate inequality is due to that $\left(\hat{\mu}_{i,\bar{m}(i)} - \mu_{i,\bar{m}(i)}\right) - \left(\hat{\mu}_{i,j} - \mu_{i,j}\right)$ is a $\sqrt{\frac{2}{hc_j}}$-subgaussian random variable and Chernoff concentration inequality [6]. □

When there is enough exploration, the agent-optimal regret bound of centralized ETC could be guaranteed.

Theorem 4.3. *Let $\bar{\Delta}_{i,j} = \mu_{i,\bar{m}(i)} - \mu_{i,j}$, $\bar{\Delta}_{i,\min} = \min_{j:\bar{\Delta}_{i,j}>0} \bar{\Delta}_{i,j}$ and $\Delta = \min_i \bar{\Delta}_{i,\min}$. Then the expected agent-optimal regret for agent p_i satisfies*

$$\bar{R}_{T,i} \le h\sum_{j=1}^{K} c_j\bar{\Delta}_{i,j} + (T - hC)N\sum_{j=1}^{K} \exp\left(-\frac{hc_j\Delta^2}{4}\right) . \tag{3}$$

When $h = \frac{4}{c_{\min}\Delta^2}\log\left(1 + \frac{TN\Delta^2}{4}\right)$ and $c_{\min} = \min_j c_j$, the regret becomes

$$\begin{aligned}
\bar{R}_{T,i} \le& \frac{4}{c_{\min}\Delta^2}\log\left(1 + \frac{TN\Delta^2}{4}\right)\sum_{j=1}^{K} c_j\bar{\Delta}_{i,j} + \frac{4K}{\Delta^2}\log\left(1 + \frac{TN\Delta^2}{4}\right)\\
=& O\left(\frac{K}{\Delta^2}\log(TN)\right) .
\end{aligned}$$

Proof. The agent p_i pulls a_j arm hc_j times during the exploring stage, and incurs $\bar{\Delta}_{i,j}$ in the expected regret each time. Thus the regret during the exploring stage is the first term in (3). For the later exploitation part, there is regret only when some agent's empirical ranking is invalid, whose probability is at most

$$\sum_{i=1}^{N} \mathbb{P}\left(\hat{r}_i \text{ is invalid}\right) \le N\sum_{j=1}^{K} \exp(-hc_j\Delta^2/4)$$

due to Lemma 4.1. Also since the reward per round lies in $[0, 1]$, the regret in the exploitation is at most, if there is, $T - hC$, thus giving the second term in (3).

A proper h can well balance the two terms in (3). Note that

$$\bar{R}_{T,i} \le h\sum_{j=1}^{K} c_j\bar{\Delta}_{i,j} + (T - hC)N\sum_{j=1}^{K} \exp\left(-\frac{hc_{\min}\Delta^2}{4}\right) . \tag{4}$$

Choose $h = \frac{4}{c_{\min}\Delta^2}\log\left(1 + \frac{TN\Delta^2}{4}\right)$ to ensure the two terms in the regret to have the same order, that is

$$\begin{aligned}
\bar{R}_{T,i} \le& \frac{4}{c_{\min}\Delta^2}\log\left(1 + \frac{TN\Delta^2}{4}\right)\sum_{j=1}^{K} c_j\bar{\Delta}_{i,j} + \frac{4K}{\Delta^2}\log\left(1 + \frac{TN\Delta^2}{4}\right)\\
=& O\left(\frac{K}{\Delta^2}\log(TN)\right) .
\end{aligned}$$

□

4.2 Centralized UCB Algorithm

Although the ETC algorithm can effectively find stable matching in the centralized setting, it needs to know the horizon T and the minimum reward gap Δ in advance, which may not be satisfied in practice. To refrain from such a constraint, we further introduce a centralized UCB algorithm (cenUCB) (Algorithm 2) which is adaptive and refrains such constraint.

Algorithm 2 Centralized UCB

1: Input: The preference ranking π_j and the capacity c_j for each arm a_j, and the horizon T.
2: **for** $t = 1, 2, ..., T$ **do**
3: Compute UCB index $u_{t,i,j}$ for each agent p_i and arm a_j as in (5);
4: Rank all arms for p_i as $\hat{r}_i$ according to the decreasing order of $\{u_{t,i,j} : j \in [K]\}$;
5: Run SADA algorithm to compute the agent-optimal stable matching m_t with respect to $\{\hat{r}_{t,i}\}_i$ and $\{\pi_j\}_j$;
6: Perform matching m_t, receive reward $X_{t,i,m_t(i)}$ for each agent p_i.
7: **end for**

At each time t, the platform first computes UCB index $u_{t,i,j}$ (whose formula will be given later in (5)) for every agent p_i and every arm a_j which serves as an upper confidence bound of the associated reward (line 3). Then the platform forms the ranking $\hat{r}_i$ as the decreasing order of $\{u_{t,i,j} : j \in [K]\}$ for every agent p_i similar to (2) (line 4). After running SADA algorithm on $\hat{r}_{t,i}$ and π_j, every agent p_i selects arm $m_t(i)$ based on this output matching m_t (line 5) and receive the corresponding reward $X_{t,i,m_t(i)}$ (line 6). Note that in this case there is also no collision and thus $I_t(i) = 1$ for all agents. Specifically, the UCB index is computed as [21]

$$u_{t,i,j} = \hat{\mu}_{t,i,j} + \sqrt{\frac{3\log t}{2T_{t-1,i,j}}}, \tag{5}$$

where $T_{t,i,j} = \sum_{s=1}^{t} 1\{m_s(i) = j\}$ is the times arm a_j has successfully been matched with agent p_i at time $t-1$ and $\hat{\mu}_{t,i,j} = \frac{1}{T_{t,i,j}} \sum_{s=1}^{t} 1\{m_s(i) = j\}\, X_{s,i,j}$ the estimated reward for agent p_i who is matched to arm a_j.

Since there is no collision in centralized setting, stable regret is incurred from wrong matchings based on wrong estimated preferences. Note that an example in [25] shows that it is hard to approach agent-optimal stable matching with such an algorithm in one-to-one setting, which is a special case of our setting. Thus we only prove the agent-pessimal stable regret for this algorithm.

Recall that the agent-pessimal stable matching is the least preferred by agents among all stable matchings. Then if the performed matching is a stable matching under the true preferences, there would be no pessimal stable regret. We formally define that a matching is *truly stable* if it is stable under the true rankings and it is *achievable* if it appears as some m_t in the running of the algorithm. If m_t is non-truly stable, then there must be a blocking pair (i, j) under the true preferences. To be specific, agent p_i prefers arm a_j over $m_t(i)$ at time t (note that agents are always matched in this case) and arm a_j prefers agent p_i over some of its matched agents $m_t^{-1}(j)$ or a_j is unmatched.

For a certain triplet $(p_\ell, a_k, a_{k'})$, we denote $B_{\ell,k,k'}$ as the set of all matchings blocked by this triplet.. Given a set S of matching, we say a set Q of triplets $(p_\ell, a_k, a_{k'})$ is a *cover* of S if

$$\bigcup_{(p_\ell,a_k,a_{k'})\in Q} B_{\ell,k,k'} \supseteq S,$$

and $C(S)$ is the set of all covers of S. Then for non-truly stable matchings for agent p_i and arm a_j, which is denoted as $O_{i,j}$, hence $C(O_{i,j})$ is the set containing all blocking triplets of (p_i, a_j).

Theorem 4.4. *For each agent p_i, the agent-pessimal stable regret of our centralized UCB algorithm up to time T satisfies*

$$\underline{R}_{T,i} \le \sum_{j:\underline{\Delta}_{i,j}>0} \underline{\Delta}_{i,j} \left[\min_{Q\in C(O_{i,j})} \sum_{(p_\ell,a_k,a'_k)\in Q} \left(5 + \frac{6\log(T)}{\Delta^2_{\ell,k,k'}}\right)\right] = O(\frac{NK^3}{\Delta^2}\log(TN)),$$

where $O_{i,j}$ is the set of matchings that agent p_i and arm a_j is not truly-stable.

Theorem 4.4 guarantees an $O(\log(T))$ upper bound of agent-pessimal stable regret for each agent p_i.

By the definition of the blocking triplet, for agent p_ℓ, a_k is a better choice for her rather than $m_t(\ell) = a_{k'}$ according to true preference, i.e. $\mu_{t,\ell,k} > \mu_{t,\ell,k'}$. If p_ℓ pulls $a_{k'}$ successfully rather than a_k when $(p_\ell, a_k, a_{k'})$ is blocking, there must be a higher upper confidence bound for $a_{k'}$ than for a_k. In other words, in order to prove Theorem 4.4, we are trying to upper bound the expected number of times that $u_{t,\ell,k'}$ in (5) is higher than $u_{t,\ell,k}$ when $a_{k'}$ is successfully matched.

Lemma 4.5. *Under UCB policy, for agent $p_\ell, \ell \in [N]$, the expected number of times that the UCB index of $a_{k'}$ is higher than that of the better arm a_k and $a_{k'}$ is successfully pulled is at most $5 + \frac{6\log(t)}{\Delta^2_{\ell,k,k'}}$ by time t.*

Proof. By Chernoff concentration inequality [6], for any k, ℓ, t we have,

$$\mu_{\ell,k} - \sqrt{\frac{3\log(t)}{2T_{t-1,\ell,k}}} < \hat{\mu}_{t,\ell,k} < \mu_{\ell,k} + \sqrt{\frac{3\log(t)}{2T_{t-1,\ell,k}}}. \tag{6a}$$

Recall that the UCB index is:

$$u_{t,\ell,k} = \hat{\mu}_{t,\ell,k} + \sqrt{\frac{3\log(t)}{2T_{t-1,\ell,k}}}. \tag{6b}$$

The event arm $a_{k'}$ is successfully selected for agent p_j rather than the better arm a_k at time t implies that

$$u_{t,\ell,k'} > u_{t,\ell,k}. \tag{6c}$$

Hence,

$$\begin{aligned}
\mu_{\ell,k'} + 2\sqrt{\frac{3\log(t)}{T_{t-1,\ell,k'}}} &\overset{(6a)}{>} \hat{\mu}_{t,\ell,k'} + \sqrt{\frac{3\log(t)}{T_{t-1,\ell,k'}}} \\
&\overset{(6c)}{>} \hat{\mu}_{t,\ell,k} + \sqrt{\frac{3\log(t)}{T_{t-1,\ell,k}}} \\
&> \mu_{\ell,k} - \sqrt{\frac{3\log(t)}{T_{t-1,\ell,k}}} + \sqrt{\frac{3\log(t)}{T_{t-1,\ell,k}}} \\
&= \mu_{\ell,k},
\end{aligned}$$

which leads to

$$T_{t,\ell,k'} < \frac{6\log(t)}{\Delta^2_{\ell,k,k'}},$$

where $\Delta_{\ell,k,k'}$ is the reward difference between the $\mu_{\ell,k'}$ and $\mu_{\ell,k}$.

Note that the inequality $T_{t,\ell,k'} < \frac{6\log(t)}{\Delta^2_{t,\ell,k,k'}}$ holds true with high probability. By Theorem 2.1 in [5] and $T_{t,\ell,k'} < \frac{6\log(t)}{\Delta^2_{t,\ell,k,k'}}$ at time t, $\mathbb{E}\left[T_{t,\ell,k'}\right] \le 5 + \frac{6\log(t)}{\Delta^2_{t,\ell,k,k'}}$ can be obtained. □

Proof of Theorem 4.4. Let $N_m(t)$ be the number of times matching m has been played at time t and denote $O_{i,j}$ as the set of matchings that the matched pair (p_i, a_j) is not truly-stable. Since regret comes from the untrue preferences reported by the agents, we can bound the regret by

$$\underline{R}_{T,i} \le \sum_{j:\underline{\Delta}_{i,j}>0} \underline{\Delta}_{i,j} \left[\sum_{m\in O_{i,j}} \mathbb{E}\left[N_m(T)\right]\right].$$

Let $L_{t,(\ell,k,k')}$ be the number of times agent p_ℓ pulls arm $a_{k'}$ when $(p_\ell, a_k, a_{k'})$ is a blocking triplet of the matching m_t. Note that a non-truly stable matching means that there exists a tuple (ℓ, k, k') such that agent p_ℓ pulls arm a'_k but prefers arm a_k under the true preferences. Then we have,

$$L_{T,(\ell,k,k')} = \sum_{m\in B_{(\ell,k,k')}} N_m(T).$$

The blocking triplet $(p_\ell, a_k, a_{k'})$ exists at time t when p_ℓ pulls $a_{k'}$ successfully only if $\mu_{\ell,k} > \mu_{\ell,k'}$, but $u_{t,\ell,k'} > u_{t,\ell,k}$. By Lemma 4.5, we have

$$\mathbb{E}\left[L_{T,(\ell,k,k')}\right] \le \mathbb{E}\left[T_{T,\ell,a_{k'}}\right] \le 5 + \frac{6\log(T)}{\Delta^2_{\ell,k,k'}}.$$

Thus we can bound the regret

$$\begin{aligned}\underline{R}_{T,i} &\le \sum_{j:\underline{\Delta}_{i,j}>0} \underline{\Delta}_{i,j} \left[\min_{Q\in C(O_{i,j})} \sum_{(p_\ell,a_k,a'_k)\in Q} \left(5 + \frac{6\log(T)}{\Delta^2_{\ell,k,k'}}\right)\right] \\ &= O\left(\frac{NK}{\Delta^2}\log(TN)\right).\end{aligned}$$

□

5 DECENTRALIZED CASE

In real applications, agents usually have no chance to observe others' rewards, like the online recruitment platform. Such a case cannot be covered by the centralized market. Thus the decentralized market is more general and realistic. In this section, we propose a decentralized conflict-avoiding upper-confidence-bound (MOCA-UCB) algorithm in many-to-one matching markets (Algorithm 3).

5.1 Algorithm

Each agent p_i independently runs the MOCA-UCB algorithm. At the beginning, our algorithm sets the UCB index as in (5) for all arms as ∞ (line 5), the same as the centralized UCB algorithm.

At each time t, agent p_i independently samples a random variable q with distribution Bernoulli(λ), where $\lambda \in [0, 1)$ is a hyperparameter (line 7). Then p_i constructs a plausible arm set $S^{(i)}(t)$ (line 9), from which it selects the arm with highest UCB index (line 10). Specifically, an arm a_j is in the plausible set if agent p_i is matched with a_j at time $t-1$ or a_j prefers p_i than one of its matched agents last time. Note that this is different from the CA-UCB [26] in the one-to-one setting since the capacity of each arm is considered. Intuitively, the plausible set contains arms that may accept p_i at time t.

Algorithm 3 MOCA-UCB for agent p_i

1: Input: Capacity c_j of each arm a_j, parameter $\lambda \in (0, 1)$.
2: **for** $t = 1, ..., T$ **do**
3: **if** $t = 1$ **then**
4: Set upper confidence bound to ∞ for all arms;
5: Sample an index $j \sim 1, ..., K$ uniformly at random. Set $m_t(i) \leftarrow a_j$;
6: **else**
7: Draw $q \sim \text{Ber}(\lambda)$;
8: **if** $q = 0$ **then**
9: Update plausible set $S^{(i)}(t)$ for agent p_i: $S^{(i)}(t) := \{a_j : p_i$ is matched with a_j or a_j prefers p_i than some of its matched agent at time $t-1\}$;
10: Pull $a_j \in S^{(i)}(t)$ with maximum $u_{t,i,j}$ and set $m_t(i) \leftarrow a_j$;
11: **else**
12: Pull $m_{t-1}(i)$. Set $m_t(i) \leftarrow m_{t-1}(i)$;
13: **end if**
14: **end if**
15: **if** p_i wins the conflict **then**
16: Update $u_{t,i,j}$ for arm $m_t(i)$ as in (5).
17: **end if**
18: **end for**

Instead of pulling the arm with the highest UCB index in the plausible set, agent p_i may still keep pulling its last choice with probability λ. It is a key step to ensure our algorithm could reduce conflicts. Intuitively, if all other agents hold their last choices then p_i will not collide when selecting an arm from its plausible set.

If p_i is accepted by the selected arm $m_t(i)$ it selects, it will update the UCB index for this arm (line 16), where the estimated reward is calculated by $\hat{\mu}_{t,i,j} = \frac{1}{T_{t,i,j}} \sum_{s=1}^{t} 1\{m_s(i) = j \text{ and } I_s(i) = 1\}\, X_{s,i,j}$. Otherwise, it will be collided and receive zero reward.

The delay parameter λ mentioned above is the probability that the agent can maintain its last-time choice. It helps to avoid conflict cycles and converge to a stable matching. However, higher λ seems to be wasteful since the probability of choosing the arm with the highest upper confidence bound is reduced. Thus there is a trade-off of choosing the λ.

5.2 Regret Analysis

As shown in Example 1 of [26], it is hard to guarantee that an UCB-type algorithm can converge to the agent-optimal stable matching in the one-to-one matching markets. Such challenges still exist when we study the more general decentralized many-to-one setting. Thus we focus on the agent-pessimal stable regret bound in this section.

Theorem 5.1. *The agent-pessimal regret for agent p_i of MOCA-UCB algorithm is upper bounded by*

$$\underline{R}_{T,i} \leq O\left(7\underline{\Delta}_i \frac{N^5K^2}{\kappa^{N^4}} \frac{\log^2(T)}{\Delta^2}\right),$$

where $\kappa = (1-\lambda)\lambda^{N-1}$ and λ is the random delay variable.

Recall that m_t is the matching results at time t and $\underline{m}$ is the agent-pessimal stable matching. Denote M_s as the set of all stable matchings. The agent-pessimal regret can be upper bounded by

$$\underline{R}_{T,i} \leq \underline{\Delta}_i \sum_{t=1}^{T} \mathbb{P}\left(m_t \notin M_s\right), \tag{7}$$

where $\underline{\Delta}_i = \max_j\{\mu_{i,\underline{m}(i)} - \mu_{i,j}\}$ is the maximum reward gap. To bound $\sum_{t=1}^{T}\mathbb{P}\left(m_t \notin M_s\right)$, we need to find out what case the matching is unstable. The following lemma illustrates the condition to guarantee m_{t+1} to be stable.

Lemma 5.2. *When the arm with highest UCB index in $S^{(i)}(t)$ at time t is also the arm with the highest mean in $S^{(i)}(t)$ for all agents, i.e., $\forall i$, $\operatorname{argmax}_{a_j\in S^{(i)}(t)} u_{t,i,j} \subseteq \operatorname{argmax}_{a_j\in S^{(i)}(t)} \mu_{i,j}$, then m_{t+1} is also stable if m_t is stable.*

Proof. We prove this lemma by contradiction. Suppose $m_{t+1} \neq m_t$, then there must exist an agent p_i who selects another arm $a \neq m_t(i)$ at time $t+1$. This implies arm a has the highest UCB index in the plausible set $S^{(i)}(t+1)$. From the assumption, we know that arm a is also the arm with the highest mean in $S^{(i)}(t+1)$. Thus $a \succ_{p_i} m_t(i)$. In addition, since $a \in S^{(i)}(t+1)$, we can also conclude that arm a prefers p_i than one of its stable matched agents at time t. Thus arm a and agent p_i can form a blocking pair, which contradicts the assumption that m_t is stable. So m_{t+1} is stable under our assumptions, further we can also conclude that $m_{t+1} = m_t$ if m_t is stable. □

Denote the aforementioned event that the arm with the highest UCB in the plausible set at time t is also the arm which has the highest mean reward in the plausible set as G_t, which can be mathematically expressed as

$$G_t = \bigcap_{p_i}\left\{\operatorname*{argmax}_{a_j\in S^{(i)}(t)} u_{t,i,j} \subseteq \operatorname*{argmax}_{a_j\in S^{(i)}(t)} \mu_{i,j}\right\}.$$

According to Lemma 5.2, if m_{t+1} is unstable, then one of the following two events must happen: there are some UCB ranking errors such that $\operatorname{argmax}_{a_j\in S^{(i)}(t)} u_{t,i,j} \nsubseteq \operatorname{argmax}_{a_j\in S^{(i)}(t)} \mu_{i,j}$ for some agent p_i; or there is no ranking error but the matching m_t is not stable. By induction, we can further conclude that if m_{t+1} is unstable, there must be some UCB ranking errors in the last L rounds or no ranking error but the matching is unstable in the last L rounds, where $0 < L < t$. Recall that M_s is the set of all stable matchings, then the event $\{m_t \notin M_s\}$ implies that

$$\underbrace{\left(\bigcap_{t'=t-L}^{t}\left(G_{t'} \cap \{m_{t'-1} \notin M_s\}\right)\right)}_{(a)} \bigcup \underbrace{\left(\bigcup_{t'=t-L}^{t} G_{t'}^{c}\right)}_{(b)}. \tag{8}$$

We then bound the probability of event $\{m_t \notin M_s\}$ by bounding above two terms (a) and (b) separately.

The bound of the probability of (a). Given a matching m_t and a blocking pair (p_i, a_j) in m_t, we say that m_{t+1} is obtained by *resolving* (p_i, a_j), if $m_{t+1}(i) = a_j$ and other agents remain their choices in m_t. Abeledo and Rothblum [1] point out that for the one-to-one setting, given any unstable matching m, there exists a sequence of blocking pairs with length at most N^4 such that *resolving* blocking pair in sequence order reaches a stable matching. Here we extend this result to the many-to-one matching.

Lemma 5.3. *For the many-to-one matching markets, given any unstable matching m, there exists a sequence of blocking pairs with length at most N^4 such that resolving this blocking pair sequence can reach a stable matching.*

Proof. For any many-to-one matching market M, we can construct a related one-to-one model M'. Specifically, the arm in M with capacity c is broken into c independent arms in M', each with capacity 1. And those broken arms in M' share the same preference as the original arm in M. Agents and their preferences in M' are the same as that in M. Thus for the many-to-one matching m in M, we have the related one-to-one matching m' in M'. Since agents and arms in M and M' share the same preference, the blocking pair in m' can also block m and vice versa. Thus we can further conclude that m' is stable if and only if m is stable.

For the related one-to-one matching m', previous works show that there exists a sequence of blocking pairs whose length is at most N^4 (Theorem 4.2 in [1]) such that we can get a stable matching from m' by resolving this blocking pair sequence. Since the blocking pairs and stable matching of m' are consistent with those of m, this sequence of blocking pairs is also the resolving route for m in the original many-to-one model M to achieve stable matching. Thus the proof is completed. □

Recall that term (a) denotes the event that there is no UCB ranking error but the matching is unstable from time $t-L$ to t, where L is the window length.

Lemma 5.4. *Let $\kappa = (1-\lambda)\lambda^{N-1}$, where λ is the delay parameter in Algorithm MOCA-UCB. For any window length L such that $N^4 \leq L < t-1$, the probability of event (a) can be upper bounded as follows,*

$$\mathbb{P}\left(\bigcap_{\tau=t-L}^{t}\left(G_\tau \cap \{m_{\tau-1} \notin M_s\}\right)\right) \leq \left(1-\kappa^{N^4}\right)^{\left\lfloor \frac{L}{N^4}\right\rfloor}.$$

Proof. Note that event G_τ means that $\forall i \in [N]$, with $t-L \leq \tau \leq t$, the arm with the largest UCB in $S^{(i)}(\tau)$, denoted as a_j, is also the arm preferred most by p_i in $S^{(i)}(\tau)$. Thus under event G_τ, if $m_{\tau-1}(i) \neq a_j$, we know that (p_i, a_j) forms a blocking pair for matching $m_{\tau-1}$ since agent p_i and arm a_j both prefer each other than their matched result in $m_{\tau-1}$. Note that at time τ, p_i turns to pull the arm a_j with the highest UCB in $S^{(i)}(\tau)$ with probability $1-\lambda$, and all of other agents hold their choices at $\tau-1$ with probability λ^{N-1} bt line 7-12 of MOCA algorithm. Thus when G_τ holds, m_τ is obtained by resolving (p_i, a_j) with probability $\kappa = (1-\lambda)\lambda^{N-1}$. From lemma 5.3, we know that there exists a blocking pair sequence with length no more than N^4 such that we can get a stable matching from $m_{\tau-1}$ by resolving the blocking pairs on the sequence. Thus $m_{\tau-1}$ can reach stable within N^4 steps if it resolves the blocking pairs in the order of that sequence, which happens with probability

at least κ^{N^4}. When the window length $L = N^4$, then the probability of (a) can be bounded by $1 - \kappa^{N^4}$. When $L > N^4$, it is independent for any non-overlapping blocks of N^4 steps, thus the probability of (a) can be bounded by

$$\mathbb{P}\left(\bigcap_{\tau=t-L}^{t} (G_\tau \cap \{m_{\tau-1} \notin M_s\})\right) \leq \left(1-\kappa^{N^4}\right)^{\left\lfloor \frac{L}{N^4} \right\rfloor}.$$

Thus the proof is completed. □

The bound of the probability of (b). Then we turn to focus on term (b). Recall that G_t means $\text{argmax}_{a_j \in S^{(i)}(t)} u_{t,i,j} \subseteq \text{argmax}_{a_j \in S^{(i)}(t)} \mu_{i,j}$ for each agent p_i. When G_t does not happen, there must exist at least one triplet (i, j, k) such that arm a_j ranks the highest UCB index for agent p_i while p_i truly prefers another $a_k \in S^{(i)}(t)$ over a_j. Then we have

$$\mathbb{P}\left(G_t^c\right) \leq \sum_{(i,j,k):a_k \succ_i a_j} \mathbb{P}\left(m_t(i) = a_j, u_{t,i,k} < u_{t,i,j}\right).$$

As a_j has the highest UCB index for p_i, then with probability at least $\kappa = \lambda^{N-1}(1-\lambda)$, p_i could successfully pull arm a_j. κ is the probability that p_i attempts to pull a_j at time t, and other agents keep pulling the arms they chose at time $t-1$. Recall that $I_t(i)$ indicates whether agent p_i is successfully matched, then we can further bound the probability that G_t does not happen as follows:

$$\mathbb{P}\left(G_t^c\right) \leq \kappa^{-1} \sum_{(i,j,k):a_k \succ_i a_j} \mathbb{P}\left(m_t(i) = a_j, I_t(i) = 1, u_{t,i,k} < u_{t,i,j}\right).$$

The term $\sum_{t=1}^{T} G_t^c$ can be then bounded by standard UCB analysis and by Lemma 5 in [26]:

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{P}\left(G_t^c\right) &\leq \sum_{t=1}^{T} \kappa^{-1} \sum_{(i,j,k):a_k \succ_i a_j} \mathbb{P}\left(m_t(i) = a_j, I_t(i) = 1, u_{t,i,k} < u_{t,i,j}\right) \\ &\leq \kappa^{-1} \frac{6}{\Delta^2} \log(T) + 6.\end{aligned}$$

Choose the window $L > N^4$ and sum up the bound for term (a) and (b) above (equation (8)), we have

$$\begin{aligned}&\sum_{t=1}^{T} \mathbb{P}\left(m_t \notin M_s\right) \\ \leq &\sum_{t=1}^{T} \mathbb{P}\left(\bigcap_{t'=t-L}^{t} (G_{t'} \cap \{m_{t'-1} \notin M_s\})\right) + \sum_{t=1}^{T} \mathbb{P}\left(\bigcup_{t'=t-L}^{t} G_{t'}^c\right) \\ \leq &T \exp\left(\frac{-L\kappa^4}{N^4}\right) + (L+1)NK^2\left(\kappa^{-1}\frac{6}{\Delta^2}\log(T) + 6\right).\end{aligned}$$

Then we can balance the two terms by choosing a proper time window length $L = \lceil \frac{N^4}{\kappa} \log(T) \rceil$. The final bound would be

$$\sum_{t=1}^{T} \mathbb{P}\left(m_t \notin M_s\right) \leq O\left(7\frac{N^5K^2}{\kappa^{N^4}\Delta^2}\log^2(T)\right).$$

Thus the agent-pessimal regret bound in Theorem 5.1 is obtained. The regret upper bound with MOCA-UCB is exponential with the number of agents N and polynomial with the number of arms K. In particular, when it is reduced to the one-to-one setting, it has similar performances to [26] and this regret upper bound can recover the CA-UCB algorithm. The difference between the above two results is that the number of arms K could be very small. An interesting question is whether the regret is dependent on the total capacity of all arms C and each arm's capacity $c_j, j \in [K]$.

6 EXPERIMENTS

In this section, we show the empirical performances of our many-to-one cenETC (Algorithm 1), cenUCB (Algorithm 2), and decentralized MOCA-UCB algorithm (Algorithm 3). For all experiments, the rankings of all agents and arms are generated uniformly. We set the mean of the reward value towards the least preferred arm to be $1/N$ and the most preferred one as 1 for each agent. And the reward gap between any adjacently ranked arms is $\Delta = 1/N$. The random reward of agent p_i for arm a_j $X_{t,i,j}$ is sampled from $\text{Ber}(\mu_{i,j})$ when p_i successfully pulls arm a_j at time t. The horizon T is set to be 100,000. The capacity $c = \frac{N}{K}$ is equally distributed to each arm. In our experiments, all results are averaged over 10 independent runs, and the error bars are calculated as standard deviations divided by $\sqrt{10}$. Since our work is the first one to study the many-to-one setting, there are indeed no comparable baselines.

In order to show the performances of the algorithm, we test the cumulative regret and the averaged unstability for each experiment, where the latter is defined as the number of unstable matchings over t rounds divided by t.

6.1 Varying the Market Size

In this experiment, we investigate how performances of the cenETC, cenUCB and MOCA-UCB algorithms are influenced by market sizes. The number of agents is set to be $N \in \{5, 10, 15, 20\}$ and the number of arms is $K = \lfloor N/2 \rfloor$. For cenETC algorithm, the exploration h is set to be 50, 100, 200, 300 separately.

We first investigate the centralized setting. It can be seen from Figure 1 (a)(b) that our cenETC and cenUCB algorithm show the same trend that both the cumulative regret and averaged unstability increase when the market size becomes larger. Such phenomenon is also verified by theoretical analysis (Theorem 4.3 and Theorem 4.4). And when N is smaller, the convergence rate is faster. Though cenETC requires knowledge of both the time horizon and the minimum gap Δ, it performs the best when changing approximate parameters. CA-UCB performs a bit worse since it requires less information.

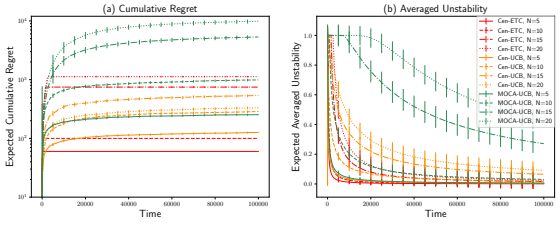

Figure 1: Cumulative regret and average unstability of centralized ETC, centralized UCB and MOCA-UCB algorithm of size with $N \in \{5, 10, 15, 20\}$ and the number of arms $K = \lfloor N/2 \rfloor$.

We then examine the MOCA-UCB algorithm when changing the market size. It can be seen from Figure 1 that the cumulative regret increases with the increase of the market size, and so does

the averaged unstability. Regret is caused by unstable matchings from the analysis of MOCA-UCB (Equation (7)), which explains that unstability and cumulative regret show the same growth trend. From the analysis we know that the regret upper bound of MOCA-UCB is also positively correlated with the number of agents N (Theorem 5.1).

6.2 Varying Arm's Capacity

In this section, we test how the performances depend on the arm's capacity. The market size is fixed with $N = C = 15$ and the number of arms K is chosen by $K \in \{2, 5, 10, 15\}$. The average capacity of each arm is N/K, which is in inverse proportion to K. For cenETC algorithm, the exploring time h is set to be 50, 100, 200, 300 respectively when K is chosen as above.

In the centralized setting, it can be seen from Figure 2 that for both centralized ETC and UCB algorithms, the reduction of capacity will increase the averaged unstability and cumulative regret. As proved in Section 4, regret is resulted from inaccurate estimate rankings. When an agent p_i incorrectly ranks arms, the smaller the capacity of each arm, the larger the probability of producing unstable pairs. Without collision, the cenUCB performs more unstability and less regret than ETC. The difference between the two metrics in the two algorithms is caused by the selection of h. When K, N and T are fixed, cenUCB keeps approaching the optimal matching, and its regret can be bounded by a fixed upper bound. The regret of cenETC comes from the exploration of the previous hC round, which is closely related to the selection of h.

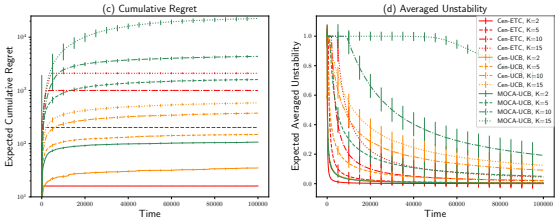

Figure 2: Cumulative regret and averaged unstability of centralized ETC, centralized UCB and MOCA-UCB algorithm of size with $K \in \{2, 5, 10, 15\}$ and fixed $N = C = 15$.

The MOCA-UCB algorithm is designed for decentralized matching where conflicts are unavoidable. It can be seen from Figure 2 that with the increase of the number of arms K, both the cumulative regret and the averaged unstability increase as the competitions become more intense.

If we compare the MOCA-UCB algorithm with the other two centralized algorithms, it can be seen from Figure 1 and 2 that MOCA-UCB suffers more regret and averaged unstability than centralized ETC and UCB when N or K is fixed. This is reasonable since there is no platform for MOCA-UCB to assign the matching at each time. Thus the collision is unavoidable and MOCA-UCB has to find the stable matching while decreasing the collision.

7 CONCLUSIONS

7.1 Contributions

This paper analyzes bandit models in the many-to-one matching market with online short-term worker employment problem as an example. We are the first to study both ETC algorithm and UCB algorithm in the centralized many-to-one market, both of which achieve an optimal regret of $O(\log(T))$. In the decentralized setting, we propose the MOCA-UCB algorithm in the many-to-one setting which achieves an agent-pessimal stable matching regret bound of $O(\log^2(T))$. Regrets of our algorithms in these two settings can recover the algorithms in the one-to-one setting [25, 26] when each arm's capacity is exactly 1. Extensive experiments show the good performances of the above algorithms by testing the cumulative regret and averaged market unstability.

7.2 Many-to-One Matching is Different from One-to-One Setting

As Roth [32] mentioned, the many-to-one problem is not equivalent to the one-to-one problem and the analysis in the many-to-one matching markets is more difficult. First, the capacity of an arm can be regarded as multiple seats accepting agents, and there is an order in the vacant seats according to the arm's preference. It is worth noting that the arm with capacity c cannot be broken into c independent individuals sharing the same preference due to the implicit competition among the vacant seats.

To better solve these problems, we analyze them from two main aspects. Firstly, we add the capacity limit to the algorithm design. For the construction of plausible sets, we add the limitation of the arm's capacity. An arm is plausible for agent p_i if they have been matched at time $t-1$, or the arm prefers p_i to some of its matched agents at last time step. Note that each arm matches a set of agents, we need to consider the arm's capacity as this arm may still have a vacant seat. We then prove the feasibility of the algorithm from two aspects of theory and numerical experiments. Secondly, in theoretical analysis, bounding the length of the sequence to get a stable matching by resolving the blocking pairs is a key point in the many-to-one decentralized markets. Considering that the capacity is usually more than one, we show that the length of the sequence of resolving blocking pairs is no more than N^4 (in Lemma 5.3) by using the set splitting idea and generalizing the existing one-to-one theory, which guarantees that our MOCA-UCB algorithm can converge to a stable matching.

In terms of the comparisons of the results, our work in the many-to-one setting obtains similar results to the one-to-one matching markets [25, 26]. For the centralized market, the agent-optimal regret with the centralized-ETC algorithm has a $O(\log(T))$ upper bound and the agent-pessimal stable regret of the centralized-UCB algorithm is also upper bounded by $O(\log(T))$, both of them obtain the same scale compared with algorithms in [25] when reduced to one-to-one matching. As for the decentralized setting, our MOCA-UCB algorithm achieves an $O(\log^2(T))$ agent-pessimal stable regret, which is similar to the CA-UCB in the one-to-one setting [26]. Specifically, when it is reduced to the one-to-one setting, i.e., $|c_j| = 1$ for all $j \in [K]$, the regret of our algorithms is the same as the previous work.

REFERENCES

[1] Hernan Abeledo and Uriel G Rothblum. 1995. Paths to marriage stability. *Discrete Applied Mathematics* 63, 1 (1995), 1–12.

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

[3] Soumya Basu, Karthik Abinav Sankararaman, and Abishek Sankararaman. 2021. Beyond $\log^2(T)$ Regret for Decentralized Bandits in Matching Markets. In *Proceedings of the 38th International Conference on Machine Learning*.

[4] Ilai Bistritz and Amir Leshem. 2018. Distributed Multi-Player Bandits-a Game of Thrones Approach. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*.

[5] Sébastien Bubeck and Nicolo Cesa-Bianchi. 2012. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Machine Learning* 5, 1 (2012), 1–122.

[6] Herman Chernoff. 1952. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics* (1952), 493–507.

[7] Xiaowu Dai and Michael I Jordan. 2021. Learning strategies in decentralized matching markets under uncertain preferences. *Journal of Machine Learning Research* 22, 260 (2021), 1–50.

[8] Sanmay Das and Emir Kamenica. 2005. Two-sided bandits and the dating market. In *Proceedings of the 19th international joint conference on Artificial intelligence*. 947–952.

[9] Jing Dong, Ke Li, Shuai Li, and Baoxiang Wang. 2022. Combinatorial Bandits under Strategic Manipulations. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 219–229.

[10] Marcelo A Fernandez, Kirill Rudov, and Leeat Yariv. 2021. *Centralized matching with incomplete information.* Technical Report. National Bureau of Economic Research.

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

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

[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. 2022. 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 (2022), 2232.

[14] Ramesh Johari, Vijay Kamble, and Yash Kanoria. 2017. Matching while Learning. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. 119–119.

[15] Ramesh Johari, Vijay Kamble, and Yash Kanoria. 2021. Matching While Learning. *Operations Research* 69, 2 (2021), 655–681.

[16] Dileep Kalathil, Naumaan Nayyar, and Rahul Jain. 2014. Decentralized Learning for Multiplayer Multiarmed Bandits. *IEEE Transactions on Information Theory* 4, 60 (2014), 2331–2345.

[17] Alexander S Kelso Jr and Vincent P Crawford. 1982. Job matching, coalition formation, and gross substitutes. *Econometrica: Journal of the Econometric Society* (1982), 1483–1504.

[18] Fang Kong, Yueran Yang, Wei Chen, and Shuai Li. 2021. The Hardness Analysis of Thompson Sampling for Combinatorial Semi-bandits with Greedy Oracle. *Advances in Neural Information Processing Systems* 34 (2021), 26701–26713.

[19] Fang Kong, Junming Yin, and Shuai Li. 2022. Thompson Sampling for Bandit Learning in Matching Markets. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. 3164–3170. Main Track.

[20] Fang Kong, Yichi Zhou, and Shuai Li. 2022. Simultaneously Learning Stochastic and Adversarial Bandits with General Graph Feedback. In *International Conference on Machine Learning*. PMLR, 11473–11482.

[21] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[22] SangMok Lee. 2016. Incentive compatibility of large centralized matching markets. *The Review of Economic Studies* 84, 1 (2016), 444–463.

[23] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. 2016. Contextual combinatorial cascading bandits. In *International conference on machine learning*. PMLR, 1245–1253.

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

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

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

[27] Onkar Malgonde, He Zhang, Balaji Padmanabhan, and Moez Limayem. 2020. Taming the Complexity in Search Matching: Two-Sided Recommender Systems on Digital Platforms. *Mis Quarterly* 44, 1 (2020).

[28] Jie-Ping Mo. 1988. Entry and structures of interest groups in assignment games. *Journal of Economic Theory* 46, 1 (1988), 66–96.

[29] N. Nayyar, D. Kalathil, and R. Jain. 2018. On Regret-Optimal Learning in Decentralized Multiplayer Multiarmed Bandits. *IEEE Transactions on Control of Network Systems* 5 (2018), 597–606.

[30] Hai Nguyen, Thành Nguyen, and Alexander Teytelboym. 2021. Stability in matching markets with complex constraints. *Management Science* 67, 12 (2021), 7438–7454.

[31] Alvin E Roth. 1984. Stability and polarization of interests in job matching. *Econometrica: Journal of the Econometric Society* (1984), 47–57.

[32] Alvin E Roth. 1985. The college admissions problem is not equivalent to the marriage problem. *Journal of Economic Theory* 36, 2 (1985), 277–288.

[33] Alvin E Roth. 2002. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica* 70, 4 (2002), 1341–1378.

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

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

[36] Abishek Sankararaman, Soumya Basu, and Karthik Abinav Sankararaman. 2021. Dominate or Delete: Decentralized Competing Bandits in Serial Dictatorship. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1252–1260.

[37] Jay Sethuraman, Chung-Piaw Teo, Liwen Qian, et al. 2006. Many-to-One Stable Matching: Geometry and Fairness. *Mathematics of Operations Research* 31, 3 (2006), 581–596.

[38] Shahin Shahrampour, Alexander Rakhlin, and Ali Jadbabaie. 2017. Multi-armed bandits in multi-agent networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2786–2790.

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

[40] Canzhe Zhao, Tong Yu, Zhihui Xie, and Shuai Li. 2022. Knowledge-aware Conversational Preference Elicitation with Bandit Feedback. In *Proceedings of the ACM Web Conference 2022*. 483–492.