Improved Analysis for Bandit Learning in Matching Markets

Fang Kong
Southern University of Science and Technology
kongf@sustech.edu.cn

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

Shuai Li[*]
Shanghai Jiao Tong University
shuaili8@sjtu.edu.cn

Abstract

A rich line of works study the bandit learning problem in two-sided matching markets, where one side of market participants (players) are uncertain about their preferences and hope to find a stable matching during iterative matchings with the other side (arms). The state-of-the-art analysis shows that the player-optimal stable regret is of order $O(K \log T/\Delta^2)$ where K is the number of arms, T is the horizon and Δ is the players' minimum preference gap. However, this result may be far from the lower bound $\Omega(\max\{N \log T/\Delta^2, K \log T/\Delta\})$ since the number K of arms (workers, publisher slots) may be much larger than that N of players (employers in labor markets, advertisers in online advertising, respectively). In this paper, we propose a new algorithm and show that the regret can be upper bounded by $O(N^2 \log T/\Delta^2 + K \log T/\Delta)$. This result removes the dependence on K in the main order term and improves the state-of-the-art guarantee in common cases where N is much smaller than K. Such an advantage is also verified in experiments. In addition, we provide a refined analysis for the existing centralized UCB algorithm and show that, under α-condition, it achieves an improved $O(N \log T/\Delta^2 + K \log T/\Delta)$ regret.

1 Introduction

The two-sided matching market problem has been extensively studied in the literature due to its wide range of applications like labor market, school admission, house allocation, and online advertising [25, 9, 1]. There are two sides of participants in the market, such as the employers and workers in the labor market, advertisers and publishers in online advertising. Each participant on the one side has a preference ranking over the other side. The concept of stability, which characterizes the equilibrium state of the market where no participant wants to break up the current matching relationship and find another partner, has attracted great interest from researchers [25]. Achieving stability is critical for ensuring the long-term viability of the market.

A rich line of works [9, 15, 24] study how to find a stable matching in the market. Most of them assume the preference ranking of each market participant is known beforehand, which we refer to as the *offline* setting. However, in real applications, the knowledge of the preferences may be uncertain. For example, in the labor market, employers usually do not know the working abilities of workers before being matched, and advertisers also do not know the exact conversion rate of placing the advertisement in a publisher slot. This makes the traditional algorithms unavailable to

[*] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

find an exact stable matching. With the emergence of online market platforms such as the online labor market UpWork and TaskRabbit as well as online advertising platforms where employers or advertisers have many similar tasks, market participants are able to learn their unknown preferences during iterative matchings with the other side of agents.

Multi-armed bandit (MAB) is a classic framework that characterizes the learning process during iterative interactions [3, 18]. It considers the setting with single player on one side and multiple arms on the other side. At each round, the player selects an arm and receives a reward. The player has unknown preferences over arms and would learn this knowledge based on the collected rewards. To accumulate as many rewards as possible, the player faces the dilemma of exploration and exploitation. The former selects arms with less observations while the latter focuses on arms with better historical performances. How to balance the exploration and exploitation trade-off is the key of bandit algorithm design. The upper confidence bound (UCB) [3], Thompson sampling (TS) [14, 2], and explore-then-commit (ETC) [10] are common strategies in MAB to achieve this objective.

Liu *et al.* [19] introduce the bandit learning problem in matching markets and try to provide theoretical guarantees. Two sides of agents in the market can be modeled as players and arms. Without loss of generality, denote N and K as the number of players and arms, respectively. It is worth noting that this work and all of the following works assume $N \leq K$ to ensure each player has a chance to be matched. In this problem, the objective is to find a stable matching and minimize the stable regret for each player, which is defined as the difference between the reward of the stable arm and that the player receives during the horizon. Since there may be more than one stable matching, they mainly focus on the players' most preferred one corresponding to the player-optimal stable matching and the least preferred one corresponding to the player-pessimal stable matching. Note that players receive more rewards in the player-optimal stable matching and thus the former objective is the most desirable. Liu *et al.* [19] first study a centralized setting where a central platform would compute allocations for players to avoid conflicts. Both ETC and UCB-type algorithms are proposed for this setting. The former achieves a player-optimal stable regret guarantee with prior knowledge of players' minimum preference gap Δ and the latter can only ensure to reach the player-pessimal stable matching. Motivated by real applications where the central platform may not always exist, a rich line of works then study the decentralized case where no platform coordinates players' behavior [20, 27, 4, 17, 21]. This line of works again only achieve guarantees for player-pessimal stable regret [20, 17, 27, 4, 21]. Table 1 compares settings and regrets among these works. Until recently, Zhang *et al.* [30] and Kong and Li [16] independently derive algorithms that have polynomial player-optimal stable regret and show the upper bound is $O(K \log T/\Delta^2)$. However, this result may be still far from the lower bound $\Omega(\max\{N \log T/\Delta^2, K \log T/\Delta\})$ [27] since K is usually much larger than N such as that the number of workers (publisher slots) is usually much larger than that of employers in labor markets (advertisers in online advertising, respectively).

In this paper, we try to provide more efficient algorithms and improve the results over existing works. The detailed contribution can be summarized as follows:

- We propose an adaptive online Gale-Shapley (AOGS) algorithm for general markets. State-of-the-art works [30, 16] explicitly separate the exploration and exploitation processes, which can lead to unnecessary regret, as exploring certain preference rankings may not contribute to the exploitation process. To avoid excessive exploration, our AOGS algorithm integrates the players' learning process into the GS steps. Players explore only the necessary number of candidate arms and adaptively switch between exploration and exploitation.
- We prove that the player-optimal stable regret of AOGS can be upper bounded by $O(N^2 \log T/\Delta^2 + K \log T/\Delta)$. This is the first result that removes the dependence on K in the main regret order term and improves existing works in common cases where N is much smaller than K. We also conduct experiments to show the advantages of the algorithm.
- We refine the analysis of the centralized UCB algorithm in Liu *et al.* [19] for markets satisfying the α-condition. By investigating the preference hierarchy structure of the α-condition, we demonstrate that the stable matching converges sequentially from player 1 to player N. Through inductive analysis over players, we establish an $O(N \log T/\Delta^2 + K \log T/\Delta)$ regret upper bound, which improves the original result for this algorithm in this specific market.

	Regret bound	Setting
Liu *et al.* [19]	$O\left(K\log T/\Delta^2\right) * \#$ $O\left(NK\log T/\Delta^2\right) \#$	known Δ, gap_1 gap_2
Liu *et al.* [20]	$O\left(\frac{N^5K^2\log^2 T}{\varepsilon^{N^4}\Delta^2}\right)$	gap_2
Sankararaman *et al.* [27]	$O\left(NK\log T/\Delta^2\right)$ $\Omega\left(\max\left\{N\log T/\Delta^2, K\log T/\Delta\right\}\right)$	serial dictatorship, gap_1
Basu *et al.* [4]	$O\left(K\log^{1+\varepsilon} T + 2^{\left(\frac{1}{\Delta^2}\right)^{\frac{1}{\varepsilon}}}\right) *$ $O\left(NK\log T/\Delta^2\right)$	gap_2 α-condition, gap_1
Maheshwari *et al.* [21]	$O\left(CNK\log T/\Delta^2\right)$	α-reducible condition, gap_1
Kong *et al.* [17]	$O\left(\frac{N^5K^2\log^2 T}{\varepsilon^{N^4}\Delta^2}\right)$	gap_2
Zhang *et al.* [30]	$O\left(K\log T/\Delta^2\right) *$	gap_2
Kong and Li [16]	$O\left(K\log T/\Delta^2\right) *$	gap_3
Ours	$O\left(N^2\log T/\Delta^2 + K\log T/\Delta\right) *$ $O\left(N\log T/\Delta^2 + K\log T/\Delta\right) \#$	gap_4 α-condition, gap_3

Table 1: Comparisons of settings and regret bounds with most related works, $*$ represents the player-optimal stable regret and bounds without labeling $*$ are for player-pessimal stable regret, $\#$ represents the centralized setting. N and K are the number of players and arms with $N \le K$, T is the total horizon, Δ corresponds to some preference gap, ε depends on the hyper-parameter of algorithms, and C is related to the unique stable matching condition which can grow exponentially in N. The definition of Δ in different works requires particular care. We use $\text{gap}_1, \text{gap}_2, \text{gap}_3, \text{gap}_4$ represent the minimum preference gap between the (player-optimal) stable arm and the next arm after the stable arm in the preference ranking among all players, the minimum preference gap between any different arms among all players, the minimum preference gap between the first $N+1$ ranked arms among all players, and the minimum preference gap between arms that are more preferred than the next of the player-optimal stable arm among all players, respectively. Based on the property that the player-optimal stable arm of each player must be its first N-ranked (shown in Appendix), there would be $\text{gap}_1 \ge \text{gap}_4 \ge \text{gap}_3 \ge \text{gap}_2$. So our dependence on Δ is better than the state-of-the-art works [30, 16] for general markets.

2 Related Work

The problem of bandit learning in matching markets is first introduced by Das and Kamenica [8]. They study the special case where both sides of agents have the same preferences and propose some empirical methods to solve the problem. Liu *et al.* [19] first theoretically formulate this problem and provide an upper bound for the stable regret of players. They propose a centralized explore-then-commit (ETC) algorithm and upper confidence bound (UCB) algorithm, which obtain an $O\left(K\log T/\Delta^2\right)$ player-optimal stable regret and $O\left(NK\log T/\Delta^2\right)$ player-pessimal stable regret, respectively. It is worth noting that the former ETC algorithm requires knowledge about Δ to ensure the algorithmic operation. Due to the generality, the following works focus on the decentralized setting. Liu *et al.* [20] and Kong *et al.* [17] propose the UCB and TS-type algorithm for general decentralized markets, respectively. Such a setting is much more challenging and both of them only achieve $O\left(\exp(N^4)N^5K^2\log^2(T)/\Delta^2\right)$ upper bound for the player-pessimal stable regret.

To improve the stable regret guarantee, a line of research studies some special markets with unique stable matching in which case the player-optimal stable matching is equivalent to the player-pessimal one. Sankararaman *et al.* [27] propose the UCB-D3 algorithm based on the assumption of serial dictatorship, i.e., all arms share the same preferences, and obtain an $O\left(NK\log T/\Delta^2\right)$

regret upper bound. To investigate the problem hardness, they also derive a lower bound $\Omega\left(\max\left\{N\log T/\Delta^2, K\log T/\Delta\right\}\right)$ under this assumption. Basu *et al.* [4] consider more general α-condition setting for unique stable matching. They propose the UCB-D4 algorithm and also achieve the $O\left(NK\log T/\Delta^2\right)$ regret bound. Later, Maheshwari *et al.* [21] study the market satisfying α-reducible condition and proposes a communication-free algorithm. Their regret bound has an exponential dependence on the number of market participants. Recently, researchers have developed algorithms that can achieve player-optimal stable regret guarantees without assuming unique stable matching. Both Zhang *et al.* [30] and Kong and Li [16] propose ETC-type algorithms that achieve $O\left(K\log T/\Delta^2\right)$ player-optimal stable regret. Wang and Li [28] studies the matching markets with serial dictatorship and obtains the $O\left(N\log T/\Delta^2 + K\log T/\Delta\right)$ regret. Table 1, compares our proposed algorithm with these related works in terms of their corresponding settings and theoretical guarantees.

There are also other works considering unknown preferences in matching markets. Wang *et al.* [29] study a many-to-one market where an arm can accept multiple players. Jagadeesan *et al.* [12] consider online matching markets with monetary transfers. Min *et al.* [22] investigate Markov matching markets where state transitions occur during the matching process and players' rewards depend on the current state. Other studies have focused on non-stationary rewards, such as Muthirayan *et al.* [23], Ghosh *et al.* [11], who propose robust algorithms to mitigate the impact of reward disturbances. Additionally, several studies have explored the problem of offline matching market learning. Dai and Jordan [6, 7] propose approaches that leverage historical data to design optimal matching or recommend participants on both sides.

3 Setting

This paper considers the problem of bandit learning in two-sided matching markets. Denote $\mathcal{N} = \{p_1, p_2, \dots, p_N\}$ as the player set and $\mathcal{K} = \{a_1, a_2, \dots, a_K\}$ as the arm set. Let N and K be the number of players and arms, respectively. To ensure that each player can be matched with an arm, we follow previous works and assume $N \le K$ [19, 20, 27, 4, 17, 30, 16].

For each player $p_i \in \mathcal{N}$, its preference towards arm a_j can be portrayed by an absolute utility $\mu_{i,j} \in (0,1]$. For any pair of arms a_j and $a_{j'}$, $\mu_{i,j} > \mu_{i,j'}$ indicates that player p_i prefers arm a_j over $a_{j'}$. Following previous works for matching markets [9, 19, 20, 27, 4, 17, 30, 16], players are assumed to have distinct preferences over different arms, i.e., $\mu_{i,j} \neq \mu_{i,j'}$ for any $a_j \neq a_{j'}$. In practice, players' preferences which correspond to workers' abilities and the publisher's conversion rates are typically unknown and can be learned through the interactive matching process. On the other side, each arm a_j also has a fixed and distinct preference utility $\pi_{j,i}$ over each player $p_i \in \mathcal{N}$, and $\pi_{j,i} > \pi_{j,i'}$ means that arm a_j prefers player p_i over $p_{i'}$. As in labor markets where workers usually know their preferences over employers based on the payments and task types, the preferences of arms are assumed to be known beforehand [19, 20, 17, 27, 4, 30, 16].

At each round $t = 1, 2, \dots$, each player p_i proposes to an arm $A_i(t)$. For each arm a_j, denote $A_j^{-1}(t) = \{p_i : A_i(t) = a_j\}$ as the set of players who selects arm a_j at round t. When more than one player selects a_j, it accepts its most-preferred one in $A_j^{-1}(t)$, i.e. a_j will match with $p_i \in \arg\max_{p_i \in A_j^{-1}(t)} \pi_{j,i}$. If a player p_i is successfully matched with arm $A_i(t)$, it will receive a random reward $X_i(t)$ characterizing its matching experience, which we assume is a 1-subgaussian random variable with expectation $\mu_{i,A_i(t)}$. Otherwise, p_i is rejected by its proposed arm and only gets reward $X_i(t) = 0$. Denote $\bar{A}_i(t)$ as the final matched arm of player p_i at round t. Then $\bar{A}_i(t) = A_i(t)$ if p_i is accepted by the arm $A_i(t)$ and we simply set $\bar{A}_i(t) = \emptyset$ if p_i is rejected.

Stability is a key property of a matching in two-sided markets [9, 26, 24]. A matching $\bar{A}(t) = \{(i, \bar{A}_i(t)) : i \in [N]\}$ is stable if no market participant wants to break up its current matching relationship and find a new partner. Formally speaking, there is no player-arm pair (p_i, a_j) such that $\mu_{i,j} > \mu_{i,\bar{A}_i(t)}$ and $\pi_{j,i} > \pi_{j,\bar{A}_j^{-1}(t)}$. It is worth noting that there may be multiple stable matchings in the market. Denoted $M = \{m : m \text{ is stable}\}$ as the set of all stable matchings. It is shown that there exists a stable matching $m^* \in M$ such that all players are matched with their most preferred stable arm [9], i.e., $\mu_{i,m_i^*} \ge \mu_{i,m_i}$ for any $m \in M, i \in [N]$. Given a specified horizon T, the learning objective is to minimize the player-optimal stable regret for each player p_i which is defined as the difference between the cumulative reward received by being matched with m_i^* and the cumulative reward received by p_i over T rounds:

$$Reg_i(T) = \mathbb{E}\left[\sum_{t=1}^{T}\left(\mu_{i,m_i^*} - X_i(t)\right)\right] .$$

Here, the expectation is taken over by the randomness of the reward generation and the randomness inherent in the player's strategy.

For completeness, we introduce the procedure of the offline Gale-Shapley (GS) algorithm, which would be useful when describing the algorithmic details. The offline GS algorithm is a classic algorithm to find the player-optimal stable matching when both sides of the market participants know their exact preference rankings. Following offline GS, each player proposes to the arm one by one based on its preference ranking. Until no rejection happens, the final matching is exactly the player-optimal stable matching [9]. Specifically, at the first step, all players propose to their most preferred arm. Arms would accept their most preferred player among those who propose to it and reject others. Then players who are rejected at previous steps would then propose to their next preferred arm. And arms still reject the players who propose to it except for their most preferred one. Such a process continues until no rejection happens.

For convenience, we also define some useful notations that quantify the hardness of the learning problem in matching markets and are used in the later analysis.

Definition 3.1. For each player p_i, denote σ_i as p_i's preference ranking and let $\sigma_{i,k}$ as p_i's the k-th preferred arm in its ranking. With a little abuse of notation, let $\sigma_i(a_j)$ represent the rank of arm a_j in p_i's preference. For each player p_i and arm $a_j \neq a_{j'}$, let $\Delta_{i,j,j'} = |\mu_{i,j} - \mu_{i,j'}|$ be the preference gap of p_i between a_j and $a_{j'}$. Define $\Delta = \min_{i,k\in[\sigma_i(m_i^*)]} \Delta_{i,\sigma_{i,k},\sigma_{i,k+1}}$ as the minimum preference gap between the arm ranked the first $(\sigma_i(m_i^*)+1)$-th among all players. Further, define $\Delta_N = \min_{i,k\in[N]} \Delta_{i,\sigma_{i,k},\sigma_{i,k+1}}$ as the minimum preference gap between the arm ranked the first $(N+1)$-th among all players.

4 Adaptive Online Gale-Shapley Algorithm for General Markets

In this section, we propose an algorithm called adaptive online Gale-Shapley (AOGS). For simplicity, we present the centralized version of the algorithm in Algorithm 1 from view of player p_i. The discussion on how to extend it to a decentralized version is deferred to later subsections.

In general, AOGS is an online version of the GS algorithm. Since players do not know their preference rankings, they need to learn this knowledge by exploring arms (Line 4). To reduce the regret during exploration, players would adaptively eliminate sub-optimal arms (Line 6-8). Until they find their most preferred arm among available arms, they will stop exploration and focus on this arm (Line 9-11). And once the player finds this arm is occupied by a more preferred player, it would re-start exploration to find the next preferred arm (Line 12-19).

Specifically, each player p_i still maintains $\hat{\mu}_{i,j}(t)$ and $T_{i,j}(t)$ to represent the empirical mean and the number of observations on each arm a_j at the end of round t. To determine whether an arm is more preferred than another, it maintains a confidence interval for each arm a_j with upper bound $\mathrm{UCB}_{i,j}(t) := \hat{\mu}_{i,j}(t) + \sqrt{6\log T/T_{i,j}(t)}$ and lower bound $\mathrm{LCB}_{i,j}(t) := \hat{\mu}_{i,j} - \sqrt{6\log T/T_{i,j}(t)}$. When $T_{i,j} = 0$, they will be initialized as $+\infty$ and $-\infty$, respectively. And once the confidence intervals of the two arms are disjoint, it can regard the arm with a higher empirical mean to be more preferred (Line 1). To be consistent with the offline GS, each player maintains $\mathcal{D}_i$ to represent the set of arms that have rejected p_i during previous steps. In the beginning, it is initialized as an empty set. And we use $\mathcal{A}_i$ to represent the available arms with the potential to be the stable arm of p_i, which is initialized as $\mathcal{K} \setminus \mathcal{D}_i$. For convenience, denote E_i as the exploration status of player p_i. $\mathrm{E}_i = \mathrm{True}$ means that p_i still needs to explore arms in $\mathcal{A}_i$ to determine its most preferred arm. And $\mathrm{E}_i = \mathrm{False}$ means that p_i already finds its most preferred arm and now focuses on this arm (Line 2).

To reduce the regret suffered during exploration, players would update $\mathcal{A}_i$ and eliminate sub-optimal arms in real-time (Line 6-8). Here to avoid collision during round-robin exploration, we would maintain $\mathcal{A}_i$ such that it contains no less than N arms if p_i still has not determined its most preferred one. Thus the union of the available arm set over all players with $\mathrm{E}_i = \mathrm{True}$ contains more than N arms and the round-robin exploration over $\mathcal{A}_i$ for each such player p_i can be carried out without

Algorithm 1 adaptive online Gale-Shapley (from the view of player p_i)

```
Input: $N, K, T$.
 1: Initialize: $\hat{\mu}_{i,j}(0) = 0, T_{i,j}(0) = 0, \mathrm{UCB}_{i,j}(0) = \infty, \mathrm{LCB}_{i,j}(0) = -\infty, \forall j \in [K]$;
 2: Initialize: $\mathcal{D}_i = \emptyset, \mathcal{A}_i = \mathcal{K}, \mathrm{E}_i = \mathrm{True}$;
 3: for round $t = 1, 2, \cdots, T$ do
 4:   Select $A_i(t) \in \mathcal{A}_i$ in a round-robin manner;
 5:   Update $\hat{\mu}_{i,j}(t), T_{i,j}(t)$ as Algorithm 2; compute $\mathrm{UCB}_{i,j}(t), \mathrm{LCB}_{i,j}(t)$ for any $j \in [K]$;
 6:   if $|\mathcal{A}_i| > N$ and $\exists j \in \mathcal{A}_i$ s.t. $\mathrm{UCB}_{i,j}(t) < \max_{j' \in \mathcal{A}_i} \mathrm{LCB}_{i,j'}(t)$ then
 7:     $\mathcal{A}_i = \mathcal{A}_i \backslash \{j\}$;
 8:   end if
 9:   if $\exists j \in \mathcal{A}_i$ s.t. $\mathrm{LCB}_{i,j}(t) > \mathrm{UCB}_{i,j'}(t)$ for any $j' \in \mathcal{A}_i$ and $j' \neq j$ then
10:     $\mathcal{A}_i = \{j\}, \mathrm{E}_i = \mathrm{False}, A_i = j$;
11:   end if
12:   for other player $p_{i'}$ with $\mathrm{E}_{i'} = \mathrm{False}, A_{i'} \in \mathcal{A}_i$ do
13:     if $\pi_{A_{i'},i'} > \pi_{A_{i'},i}$ then
14:       $\mathcal{D}_i = \mathcal{D}_i \cup \{A_{i'}\}, \mathcal{A}_i = \mathcal{K} \backslash \mathcal{D}_i$;
15:       if $\mathrm{E}_i = \mathrm{False}$ and $A_i \in \mathcal{D}_i$ then
16:         $\mathrm{E}_i = \mathrm{True}$;
17:       end if
18:     end if
19:   end for
20: end for
```

collisions. For completeness, we defer how to arrange players' explorations in later discussions. And once there exists an arm in $\mathcal{A}_i$ that can be regarded to be optimal, p_i will set the exploration status E_i to be False and update the exploration arm set $\mathcal{A}_i$ to only contain this optimal arm. For convenience, denote A_i as this arm (Line 9-11).

The update of the available arm set should not only depend on p_i's own observations but also on the other market participants. Specifically, if a player $p_{i'}$ determines $A_{i'}$ as its most preferred arm, then the final stable player of $A_{i'}$ would be the same as or more preferred than $p_{i'}$. So if $A_{i'}$ prefers $p_{i'}$ than p_i, then $A_{i'}$ would not be the stable arm of p_i and there is no need for p_i to explore $A_{i'}$ anymore. In this case, p_i deletes arm $A_{i'}$ from its available set and update $\mathcal{A}_i$ (Line 12-19). It is worth noting that this operation may incorporate the eliminated arms again in $\mathcal{A}_i$. This is reasonable as the previously eliminated arm may be more preferred than the current arms in $\mathcal{A}_i$ after the deletion operation. And if the deleted arm is p_i's current most preferred arm, it will mark E_i as True and restart exploration to find the next most preferred one (Line 15-17).

4.1 Theoretical Results.

Theorem 4.1. *Following Algorithm 1, the player-optimal stable regret for each player p_i satisfies*

$$Reg_i(T) \leq O\left(N^2 \log T/\Delta^2 + K \log T/\Delta\right) .$$

Due to the space limit, the proof of Theorem 4.1 is deferred to Appendix A. The following are discussions on the detailed implementation as well as the novelty and significance of the result.

Arrangement of the round-robin exploration process. Recall that the number of available arms of each player is always larger than N based on Line 6 and we assume players can explore their available arms in a round-robin manner without conflict. We now propose an arrangement by letting players explore the available arms in units of N to guarantee this property. Specifically, in every $2N$ rounds, each player selects the N available arms with the fewest observations (randomly breaks ties) and explores them in a round-robin way. It can be shown by contradiction that there exists an assignment such that each player can successfully match with their respective N arms once during these $2N$ rounds (Lemma B.2 in Appendix), which only doubles the original regret without influencing the regret order. This guarantees that after every $2N$ rounds, the observation count difference among all available arms is at most 1. Players would perform arm elimination and optimal arm identification (Line 6 and 9) in the end of each $2N$ rounds. Therefore, compared to the timely eliminating/deleting of arms, this approach ensures that each player will select each arm at most one additional time during each exploration cycle before the player finds the optimal one. Since each player may restart exploration (Line 12-13) up to N^2 times (each of N players can focus on N arms), this scenario leads to an additional $O(N^2K)$ constant regret and does not influence the final regret order.

Extension to the decentralized setting. For simplicity, we present the AOGS algorithm in a centralized manner. It can also be extended to the decentralized version where no central platform coordinates players' selections. Specifically, we divide the total horizon into several phases and the length of each grows exponentially, i.e., the lengths of phases are $2, 4, 8, \cdots$. At the end round of each phase, players would decide whether to eliminate sub-optimal arms as Line 6-8, whether to determine one arm as the most preferred one and update the exploration status as Line 9-11. After the end of each phase, players would communicate their current exploration status, update their deletion set and available arm set, re-update the exploration status as Line 12-19, and then communicate their updated available arm set to determine the round-robin exploration process in the next phase. If there exists a player whose exploration status becomes True from False during communication, the phase length would re-start from 2 and grows exponentially since new arms may be required for exploration. If the final exploration status of all players is False and their optimal arms are different, the next phase would continues until the end of the interaction. The detailed implementation of communication is deferred to the next paragraph. Based on the communication, players with $\mathrm{E}_i = \mathrm{True}$ can have a pre-agreed protocol to explore arms in their available arm set in a round-robin manner without collision as discussed in the last paragraph. Within each phase, they just round-robin explore arms and collect observations but do not make any decision on arms' optimality. If L observations on a sub-optimal arm are enough to decide its sub-optimality in the centralized version, then this arm would be eliminated at the end of the corresponding phase with the selected time to be at most $2L$ due to the exponentially increasing phase length. So the regret in this decentralized version is at most two times as that suffered in the centralized version.

This paragraph describes the implementation of the communication procedure. Recall that players need to communicate their exploration status and available arm set (calculated by subtracting the deletion and eliminating set from $\mathcal{K}$) at the end of each phase. For the phase length, recall that it grows exponentially until a player's exploration status becomes True from False and a player updates E_i from False as True only when its most preferred arm is occupied by a higher-priority player (Line 15). As shown by Lemma A.3, each player may occupy N arms, so such event happens at most N^2 times. And when all players find their unique optimal arm which requires $O(N^2 \log T/\Delta^2)$ times, the phase would continue until the end of the interaction. Above all, the total number of phases is of order $O\left(N^2 \log\left(N^2 \log T/\Delta^2\right)\right)$. For the detailed communication procedure, as phase 1 in Kong and Li [16], players can first estimate their unique indices and we assume the matching results are public as [16, 17, 20]. During the communication block of each phase, players sequentially transmit their data based on their indices, received by others through matching outcomes. Specifically, in the corresponding round, player p_i selects the focused arm if $\mathrm{E}_i = \mathrm{False}$ and nothing otherwise, incurring an $O\left(N^3 \log\left(N^2 \log T/\Delta^2\right)\right)$ cost for status communication in all phases. For deletion (eliminating) sets, p_i first selects the arm with index k to indicate it will transmit k arms and then sequentially selects these k arms. The communication cost on the arm set size is $O\left(N^3 \log\left(N^2 \log T/\Delta^2\right)\right)$. Recall that players delete arms only when a higher-priority player focuses on this arm, so N players focus on at most N arms before reaching stability and each player deletes up to N arms. Also, each player can eliminate up to $K-N$ arms during each exploration and would re-start exploration for at most N^2 times. Thus the communication cost on the deletion (eliminating) arms is $O(N^3K)$ and the total communication cost is $O\left(N^3 \log\left(N^2 \log T/\Delta^2\right) + N^3K\right)$, which is not the main order of the regret.

Novelty and significance. Balancing the exploration-exploitation trade-off is the key to achieving low regret. Previous efforts were devoted to addressing pessimal stable regret [19, 20, 17] and uniqueness assumptions [27, 4] using classic UCB and TS strategies. Until recently, Zhang *et al.* [30] and Kong and Li [16] show that ETC-type strategies better fit this problem. Specifically, players first uniformly explore arms to learn the complete preference ranking of the top N arms, and then use the GS procedure for exploitation to find the player-optimal stable matching. However, such a method may over-explore and cause unnecessary regret. The reason is that to learn the first N-ranked arms, each sub-optimal arm a_j must be selected $O(\log T/\Delta^2_{i,\sigma_{i,N},j})$ times to be distinguished from the N-ranked arm. And each time selecting this arm, the player pays $\Delta_{i,m_i^*,j}$ regret. The mismatch between the paid regret and the difference to be figured out results in $O(K\log T/\Delta_N^2)$ regret.

In contrast to the existing approach, we present a more adaptive perspective that integrates the learning process into each GS step. To avoid additional regret, players do not need to estimate their complete preference. Instead, they would start exploitation once the optimal available arm is identified. And if this arm is occupied by a higher-priority player, this player would restart exploration to find the next preferred one. To avoid the additional cost while exploring the optimal arm, we also design a more efficient way to let players promptly eliminate $K-N$ sub-optimal arms and only maintain the remaining N arms to guarantee no collision. For the eliminated arm a_j, the reward difference to be figured out is at least $\Delta_{i,m_i^*,j}$, which matches the regret when selecting this arm. So the regret caused by these eliminated arms is $O(K\log T/\Delta)$, avoiding the dependence on K in the main order term. This is a more effective way to balance exploration and exploitation in matching market scenarios. Our result also shows a significant improvement over Zhang *et al.* [30] and Kong and Li [16] in common applications such as labor markets and online advertising where the number N of players (employers, advertisers) is far smaller than that K of arms (workers, publisher slots).

Discussion on the definition of gaps. Recall that our Δ is defined as the minimum preference difference among arms that ranked in the first $(\sigma_i(m_i^*)+1)$-th positions (gap_4), while the lower bound in Sankararaman *et al.* [27] for markets with serial dictatorship depends on Δ that is defined as the minimum preference difference between the arm ranked $\sigma_i(m_i^*)$ and the arm ranked $\sigma_i(m_i^*)+1$ (gap_1 in Table 1, respectively). It is an open problem whether the lower bound should depend on our gap_4 in general markets. Here we would like to discuss that the knowledge of gap_4 is necessary to learn the true stable matching. Consider a market with 4 players and 5 arms. The preference rankings of players are $p_1 : a_1 > a_2 > a_3 > a_4 > a_5; p_2 : a_2 > a_3 > a_1 > a_4 > a_5; p_3 : a_3 > a_1 > a_2 > a_4 > a_5; p_4 : a_1 > a_2 > a_3 > a_4 > a_5$ and the preference rankings of arms are $a_1 : p_2 > p_3 > p_4 > p_1; a_2 : p_3 > p_4 > p_1 > p_2; a_3 : p_4 > p_1 > p_2 > p_3; a_4 : p_1 > p_2 > p_3 > p_4; a_5 : p_1 > p_2 > p_3 > p_4$. In this market, the player-optimal stable matching is $\{(p_1,a_4),(p_2,a_1),(p_3,a_2),(p_4,a_3)\}$. However, if player p_1 has collected enough observations to identify gap_1 but not collected enough observations to identify gap_4 and wrongly estimate the first $\sigma_i(m_i^*)$ ranked arms, i.e., p_1 wrongly estimate the preference ranking as $p_1 : a_1 > a_2 > a_4 > a_3 > a_5$. Then the computed player-optimal stable matching under this preference ranking is $\{(p_1,a_4),(p_2,a_3),(p_3,a_1),(p_4,a_2)\}$, which is not stable in the original market as player p_1 and arm a_3 form a blocking pair. This example shows that player p_1 must identify the gap among the first $\sigma_i(m_i^*)$ ranked arms to find a stable matching in the market, which further illustrates the crucial role of gap_4 in finding the real stable matching. We leave the lower bound in general markets as an important future direction.

5 Experiments

In this section, we compare our Algorithm 1 with baselines ETGS [16], ML-ETC [30] and Phased ETC [4] which also enjoy guarantees for player-optimal stable regret in general decentralized one-to-one markets. To better illustrate the advantages of our algorithm, especially when N is much smaller than K, we set $N=3$ and $K=10$. The preference rankings for both players and arms are generated as random permutations. The preference gap between any adjacent ranked arms is set as 0.1. The feedback $X_{i,j}(t)$ for player p_i on arm a_j at time t is drawn independently from the Gaussian distribution with mean $\mu_{i,j}$ and variance 1. We report the maximum cumulative player-optimal stable regret among all players and the cumulative player-optimal instability in Figure 1 (a) and (b), respectively. Here the cumulative player-optimal unstability is defined as the number of matchings that are not the player-optimal stable one. All algorithms run for $T=100k$ rounds and all results are averaged over 50 independent runs. The error bars represent standard errors, which are computed as standard deviations divided by $\sqrt{50}$.

As shown in the figure, our AOGS algorithm, which only conducts necessary explorations over unknown preferences and promptly eliminates sub-optimal arms, achieves the least cumulative regret and cumulative player-optimal unstability among all baselines. The ML-ETC and ETGS algorithms need to sufficiently explore K arms to estimate the full preference ranking, requiring more exploration time to find the player-optimal stable matching. The PhasedETC algorithm has not yet converged within the displayed rounds due to the cold start problem.

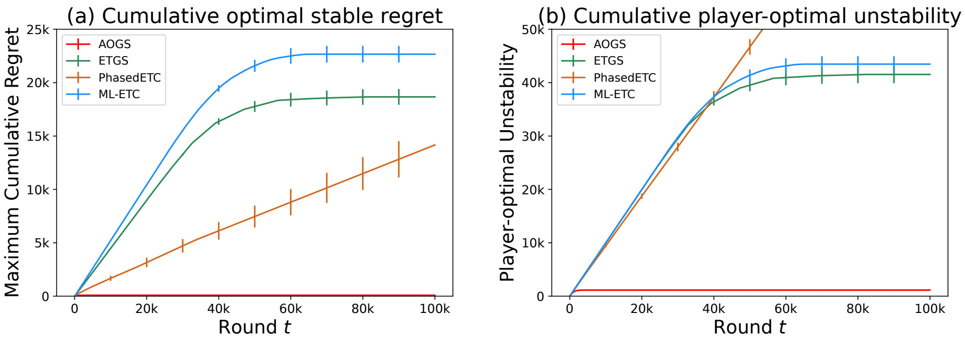

Figure 1: Experimental comparisons of our AOGS with ETGS, ML-ETC and Phased ETC in one-to-one decentralized markets with $N = 3$ players and $K = 10$ arms.

6 Centralized UCB Algorithm for α-condition

In this section, we provide a new analysis for the centralized UCB algorithm in markets satisfying α-condition. The algorithm is first introduced by Liu *et al.* [19]. For completeness, we present the full algorithm in Algorithm 2. At each round, players submit their UCB rankings to the centralized platform (Line 3). The platform runs the GS algorithm (based on players' submitted rankings) and returns the partner to each player (Line 4).

Algorithm 2 centralized UCB

```
Input: $N, K$.
1: Initialize: $\hat{\mu}_{i,j}(0) = 0, T_{i,j}(0) = 0, \mathrm{UCB}_{i,j}(1) = \infty, \forall i \in [N], j \in [K]$.
2: for round $t = 1, 2, \ldots, T$ do
3:     Receive rankings $\hat{\sigma} := \{\hat{\sigma}_i\}_{i \in [N]}$ according to the decreasing order of $\{\mathrm{UCB}_{i,j}(t)\}_{j \in [K]}, \forall i \in [N]$;
4:     $A_i(t) \leftarrow$ Gale-Shapley $(\hat{\sigma}, \pi)$ for each player $p_i$;
5:     Observe $X_i(t)$, and update $\hat{\mu}_{i,j}(t), T_{i,j}(t), \mathrm{UCB}_{i,j}(t+1)$ for each $p_i, a_j$;
6: end for
```

In the following, we introduce the α-condition. Conditions guaranteeing the unique stable matching have been widely studied in the offline setting [5, 13] and also the online setting to improve the learning efficiency [27, 4, 21]. Among these conditions, the α-condition is shown to be the weakest sufficient one [13] and incorporate the conditions studied in existing works [27, 4, 21].

Let β denote a pair of permutations of $[N]$ and $[K]$. Then $[N]_\beta = \{Q_1^{(\beta)}, \ldots, Q_N^{(\beta)}\}$ and $[K]_\beta = \{q_1^{(\beta)}, \ldots, q_K^{(\beta)}\}$ denote permutations of the ordered sets $[N]$ and $[K]$, respectively. The j-th player in $[N]_\beta$ is the $Q_j^{(\beta)}$-th player in $[N]$, and the k-th arm in $[K]_\beta$ is the $q_k^{(\beta)}$-th arm in $[K]$. Then we can define the α-condition below.

Definition 6.1. The α-condition is satisfied if there is a stable matching $(\mathbf{j}^*, \mathbf{i}^*)$, a left-order of players and arms s.t. $\forall i \in [N]_l, \forall j > i, j \in [K]_l : \mu_{i,j_i^*} > \mu_{i,j}$ where j_i^* is the partner of player p_i in stable matching $(\mathbf{j}^*, \mathbf{i}^*)$, and a (possibly different) right-order of players and arms s.t. $\forall j < i \le N, q_j \in [K]_r, Q_i \in [N]_r : \pi_{q_j, Q_{i^*_{q_j}}} > \pi_{q_j, Q_i}$. Here similarly, $i^*_{q_j}$ is the partner of arm a_{q_j} in stable matching $(\mathbf{j}^*, \mathbf{i}^*)$.

Without loss of generality, we consider the identity of players and arms is just the left order, i.e., $[N] = [N]_l$ and $[K] = [K]_l$. Thus we only deal with player order $Q_i^{(r)} = Q_i$ and arm order $q_j^{(r)} = q_j$, for $i \in [N], j \in [K]$ in the rest of the paper. Under α-condition, it is easy to inductively verify that for any $i \in [N]$, the player p_i is matched with arm a_i, and the player p_{Q_i} is matched with the arm a_{q_i} in the unique stable matching [4].

6.1 Theoretical Results

We analyze the regret for the centralized UCB algorithm under α-condition.

Theorem 6.2. *When preferences of participants satisfy α-condition, following Algorithm 2, the stable regret for each player p_i satisfies*

$$Reg_i(T) \leq O\left(N \log T/\Delta_N^2 + K \log T/\Delta\right) \, .$$

The centralized UCB algorithm is proposed by Liu *et al.* [19] and shown to have $O(NK \log T/\Delta^2)$ player-pessimal stable regret for general markets. We provide a new analysis for markets satisfying α-condition which removes the dependence of K in the regret. Due to the space limit, the detailed proof is deferred to Appendix C.

Key idea of the proof. We investigate the preference structure of α-condition to obtain the improved analysis. For player p_i, its regret is due to selecting sub-optimal arm a_k with $\mu_{i,k} < \mu_{i,i}$. Arm a_k will be selected by p_i when its UCB value is higher than p_i's stable matched arm a_i, which time is bounded by $O(\log T/\Delta_{i,i,k}^2)$, and one $\Delta_{i,i,k}$ on the denominator can be eliminated when multiplying $\Delta_{i,i,k}$ to compute regret. This contributes $O\left(K \log T/\Delta\right)$ regret since there are at most $K-1$ sub-optimal arms. It is worth noting that arm a_k will also be selected by p_i if p_i is rejected by a_i in the GS algorithm. Recall that under α-condition, there is a right order $Q_{i'} = i \in [N]_r$ for player p_i, such that $\forall i'' > i', Q_{i''} \in [N]_r : \pi_{q_{i'},Q_{i'}} > \pi_{q_{i'},Q_{i''}}$, which means arm $a_{q_{i'}} = a_i$ can only prefer players $p_{Q_1}, p_{Q_2}, \cdots, p_{Q_{i'-1}}$ than player $p_{Q_{i'}} = p_i$. Thus p_i is rejected by a_i only when these players select a_i, and a_i is sub-optimal for those players. To bound the regret of p_i when being rejected, we just need to bound the exploration times of these players $p_{Q_1}, p_{Q_2}, \cdots, p_{Q_{i'-1}}$ on arm a_i. However, the exploration time of a single player p_{Q_ℓ} with $1 \leq \ell \leq i'-1$ on a_i can not be trivially bounded by $O(\log T/\Delta_N^2)$ since p_{Q_ℓ} may have to select arm a_i after rejected by its stable arm a_{q_ℓ} in offline GS, where a_{q_ℓ} might be selected by $p_{Q_1}, \cdots, p_{Q_{\ell-1}}$. This leads to a recursion form. We control this term using the fact that when a player is rejected by its stable matched arm in the GS, it can date back to a higher right-order player wrongly over-estimate its preference for a sub-optimal arm. This key observation and the definition of Δ make it possible to derive the final $O\left(N \log T/\Delta_N^2\right)$ bound.

7 Conclusion

In this paper, we investigate the problem of whether a tighter bound can be derived for the bandit learning problem in two-sided matching markets. For the general one-to-one matching markets, we try to improve the learning efficiency of the existing algorithms. By integrating the offline GS procedure into the online learning process and carefully designing the elimination strategy, we show that the player-optimal stable regret can be upper bounded by $O(N^2 \log T/\Delta^2 + K \log T/\Delta)$. This result removes the dependence on K in the main order term of existing works and improves the state-of-the-art result [30, 16] in common cases where the number of players is much smaller than that of arms. An experiment is conducted to verify its advantage over other baselines in such markets. We also present a novel analysis for the centralized UCB algorithm in markets satisfying α-condition and derive an improved $O(N \log T/\Delta_N^2 + K \log T/\Delta)$ regret upper bound.

One significant future direction is to investigate the optimality of algorithms. Although the dependence on N, K, T in Theorem 6.2 matches the lower bound, the definition of Δ differs. It remains unclear how the upper bound changes with the same Δ. Furthermore, since the lower bound provided by [27] applies only to special markets, and the learning problem in general markets is more challenging due to the complex preference structure, determining whether an algorithm can perform better in general markets is still an open problem.

Acknowledgments and Disclosure of Funding

The corresponding author Shuai Li is supported by National Key Research and Development Program of China (2022ZD0114804) and National Natural Science Foundation of China (62376154).

We thank Yuhao Zhang and Wenqian Wang for valuable discussions and suggestions on the proof of Lemma B.2.

References

[1] Atila Abdulkadiroğlu and Tayfun Sönmez. House allocation with existing tenants. *Journal of Economic Theory*, 88(2):233–260, 1999.

[2] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, pages 99–107, 2013.

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

[4] 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.

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

[6] Xiaowu Dai and Michael Jordan. Learning in multi-stage decentralized matching markets. In *Advances in Neural Information Processing Systems*, volume 34, pages 12798–12809, 2021.

[7] Xiaowu Dai and Michael I Jordan. Learning strategies in decentralized matching markets under uncertain preferences. *Journal of Machine Learning Research*, 22(1):11806–11855, 2021.

[8] Sanmay Das and Emir Kamenica. Two-sided bandits and the dating market. In *International Joint Conference on Artificial Intelligence*, pages 947–952, 2005.

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

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

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

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

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

[14] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

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

[16] Fang Kong and Shuai Li. Player-optimal stable regret for bandit learning in matching markets. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2023.

[17] Fang Kong, Junming Yin, and Shuai Li. Thompson sampling for bandit learning in matching markets. In *International Joint Conference on Artificial Intelligence*, 2022.

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

[19] 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.

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

[21] Chinmay Maheshwari, Eric Mazumdar, and Shankar Sastry. Decentralized, communication- and coordination-free learning in structured matching markets. In *Advances in Neural Information Processing Systems*, 2022.

[22] Yifei Min, Tianhao Wang, Ruitu Xu, Zhaoran Wang, Michael Jordan, and Zhuoran Yang. Learn to match with no regret: Reinforcement learning in markov matching markets. In *Advances in Neural Information Processing Systems*, volume 35, pages 19956–19970, 2022.

[23] Deepan Muthirayan, Chinmay Maheshwari, Pramod Khargonekar, and Shankar Sastry. Competing bandits in time varying matching markets. In *Learning for Dynamics and Control Conference*, pages 1020–1031. PMLR, 2023.

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

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

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

[27] 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.

[28] Zilong Wang and Shuai Li. Optimal analysis for bandit learning in matching markets with serial dictatorship. *Theoretical Computer Science*, 1010:114703, 2024.

[29] Zilong Wang, Liya Guo, Junming Yin, and Shuai Li. Bandit learning in many-to-one matching markets. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2088–2097, 2022.

[30] Yirui Zhang, Siwei Wang, and Zhixuan Fang. Matching in multi-arm bandit with collision. In *Advances in Neural Information Processing Systems*, 2022.

A Proof of Theorem 4.1

Define $\mathcal{F} = \left\{\exists 1 \le t \le T, i \in [N], j \in [K] : |\hat{\mu}_{i,j}(t) - \mu_{i,j}| > \sqrt{\frac{6\log T}{T_{i,j}(t)}}\right\}$ as the failure event that the estimated reward is far from the expected reward at some time and some player-arm pair. The regret can be upper bounded as follows.

$$
\begin{aligned}
Reg_i(t) =& \mathbb{E}\left[\sum_{t=1}^{T}\left(\mu_{i,m_i^*} - X_i(t)\right)\right] \\
\le& \mathbb{E}\left[\sum_{t=1}^{T}\left(\mu_{i,m_i^*} - X_i(t)\right) |\neg\mathcal{F}\right] + \mathbb{P}\left(\mathcal{F}\right)\cdot T \cdot \mu_{i,m_i^*} \\
\le& \mathbb{E}\left[\sum_{t=1}^{T}\sum_{a_j}\mathbb{1}\left\{\bar{A}_i(t) = a_j\right\}\cdot\Delta_{i,m_i^*,j} |\neg\mathcal{F}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left\{\bar{A}_i(t) = \emptyset\right\}\cdot\mu_{i,m_i^*} |\neg\mathcal{F}\right] \\
&+ \mathbb{P}\left(\mathcal{F}\right)\cdot T\cdot\mu_{i,m_i^*} \\
\le& 96N^2\log T/\Delta^2 + 96K\log T/\Delta + 192N^2\log T/\Delta^2 + 2NK \qquad (1)\\
=& O\left(N^2\log T/\Delta^2 + K\log T/\Delta\right) \,.
\end{aligned}
$$

where Eq. (1) holds based on Lemma A.1, A.2, and A.4.

Lemma A.1.

$$
\mathbb{E}\left[\sum_{t=1}^{T}\sum_{a_j}\mathbb{1}\left\{\bar{A}_i(t) = a_j\right\}\cdot\Delta_{i,m_i^*,j} |\neg\mathcal{F}\right] \le 96N^2\log T/\Delta^2 + 96K\log T/\Delta \,.
$$

Proof. Recall that player p_i would update the available set $\mathcal{A}_i$ when other players $p_{i'}$ sets $\mathrm{E}_{i'} =$ False and $\pi_{A_{i'},i'} > \pi_{A_{i'},i}$ as Line 14. Denote t_s as the round index when this operation happens for the s-th time. Without loss of generality, let $t_0 = 1$.

Recall that at a high level, each time another player $p_{i'}$ sets $\mathrm{E}_{i'}$ as False, it means that $p_{i'}$ learns its most preferred arm in current available set. Combined with $\mathcal{F}$ and Lemma A.6, the determined arm of players during each exploration would be truly their most preferred one. Thus the AOGS algorithm is an online version of GS and the s'-th time player $p_{i'}$ sets $\mathrm{E}_{i'}$ as False is equivalent to that $p_{i'}$ proposes its s'-th most preferred arm in the offline GS. According to Lemma A.3, at most $N-1$ arms are proposed by all players before reaching stability. Thus for player p_i, the operation in Line 14 would happen for at most $N-1$ times.

Recall that for each s, during time t_s to t_{s+1}, player p_i would explore all available arms in a round-robin manner, eliminate sub-optimal arms until N arms are in the set, and focuses on the best one among these N when it is identified. For convenience, denote R_s as the set of the remaining N arms that p_i explored in $\mathcal{A}_i$ in a round-robin manner until condition Line 9 is satisfied, D_s as the set of arms that p_i eliminated due to condition Line 6, and j_s as the arm that p_i focuses from the time it sets E_i as False to time $t_{s+1}-1$. Then it holds that

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{a_j}\mathbb{1}\left\{\bar{A}_i(t) = a_j\right\}\cdot\Delta_{i,m_i^*,j} |\neg\mathcal{F}\right] \\
\le& \mathbb{E}\left[\sum_{s=0}^{N-1}\sum_{t=t_s}^{t_{s+1}-1}\sum_{a_j}\mathbb{1}\left\{\bar{A}_i(t) = a_j\right\}\cdot\Delta_{i,m_i^*,j} |\neg\mathcal{F}\right] \\
\le& \mathbb{E}\left[\sum_{s=0}^{N-1}\sum_{t=t_s}^{t_{s+1}-1}\left(\sum_{a_j\in R_s}\mathbb{1}\left\{\bar{A}_i(t) = a_j\right\}\cdot\Delta_{i,m_i^*,j} + \sum_{a_j\in D_s}\mathbb{1}\left\{\bar{A}_i(t) = a_j\right\}\cdot\Delta_{i,m_i^*,j}\right.\right. \\
&\left.\left.+\mathbb{1}\left\{\bar{A}_i(t) = a_{j_s}\right\}\cdot\Delta_{i,m_i^*,j_s}\right) |\neg\mathcal{F}\right]
\end{aligned}
$$

$$\leq \mathbb{E}\left[\sum_{s=0}^{N-1}\sum_{t=t_s}^{t_{s+1}-1}\left(\sum_{a_j\in R_s}\mathbb{1}\left\{\bar{A}_i(t)=a_j\right\}\cdot\Delta_{i,m_i^*,j}+\sum_{a_j\in D_s}\mathbb{1}\left\{\bar{A}_i(t)=a_j\right\}\cdot\Delta_{i,m_i^*,j}\right)\mid\neg\mathcal{F}\right], \tag{2}$$

where Eq. (2) is due to that, based on Lemma A.6 and the offline GS algorithm, the arm j_s before GS stops would be better than m_i^*, thus the regret caused by selecting these arms is less than 0.

For the first term in Eq. (2), we have

$$\mathbb{E}\left[\sum_{s=0}^{N-1}\sum_{t=t_s}^{t_{s+1}-1}\sum_{a_j\in R_s}\mathbb{1}\left\{\bar{A}_i(t)=a_j\right\}\cdot\Delta_{i,m_i^*,j}\mid\neg\mathcal{F}\right] \tag{3}$$

$$=\mathbb{E}\left[\sum_{s=0}^{N-1}\sum_{a_j\in R_s}\sum_{t=t_s}^{t_{s+1}-1}\mathbb{1}\left\{\bar{A}_i(t)=a_j\right\}\cdot\Delta_{i,m_i^*,j}\mid\neg\mathcal{F}\right]$$

$$\leq\mathbb{E}\left[\sum_{s=0}^{N-1}\sum_{a_j\in R_s}\frac{96\log T}{\Delta^2_{i,j_s,j_{s+1}}}\cdot\Delta_{i,m_i^*,j}\mid\neg\mathcal{F}\right] \tag{4}$$

$$\leq\frac{96N^2\log T}{\Delta^2}. \tag{5}$$

where Eq. (4) is due to Lemma A.5, Eq. (5) holds since R_s contains no more than N arms according to the elimination condition (Line 6).

We now analyze the second term in Eq. (2). For any arm a_j and $s\in\{0,...,N-1\}$, denote $T_{i,j,s}$ as the value of $T_{i,j}$ at the end of the round $t_{s+1}-1$. For $s\geq 1$ and arm $a_j\in D_s$, if $T_{i,j,s-1}\leq 96\log T/\Delta^2_{i,j_{s-1},j}$, it must hold that $T_{i,j,s}:=\sum_{s'\leq s}(T_{i,j,s'}-T_{i,j,s'-1})\leq 96\log T/\Delta^2_{i,j_s,j}$ to ensure arm a_j is eliminated from $\mathcal{A}_i$ at step s based on Lemma A.5. On the other hand, if $T_{i,j,s-1}>96\log T/\Delta^2_{i,j_{s-1},j}$, based on Lemma A.5, it holds that $T_{i,j,s}-T_{i,j,s-1}\leq 96\log T/\Delta^2_{i,j_s,j}-96\log T/\Delta^2_{i,j_{s-1},j}$ when a_j is eliminated. For any arm a_j, denote $s_{j,1}:=\max_{0\leq s\leq N-1}\left\{s:T_{i,j,s}\leq 96\log T/\Delta^2_{i,j_s,j}\right\}$ as the last step when the number of observation times on a_j is less than that threshold. Then the second term in Eq. (2) satisfies

$$\mathbb{E}\left[\sum_{s=0}^{N-1}\sum_{t=t_s}^{t_{s+1}-1}\sum_{a_j\in D_s}\mathbb{1}\left\{\bar{A}_i(t)=a_j\right\}\cdot\Delta_{i,m_i^*,j}\mid\neg\mathcal{F}\right]$$

$$=\mathbb{E}\left[\sum_{a_j\in\mathcal{K}}\sum_{s:a_j\in D_s}\sum_{t=t_s}^{t_{s+1}-1}\mathbb{1}\left\{\bar{A}_i(t)=a_j\right\}\cdot\Delta_{i,m_i^*,j}\mid\neg\mathcal{F}\right]$$

$$=\mathbb{E}\left[\sum_{a_j\in\mathcal{K}}\sum_{s:a_j\in D_s}(T_{i,j,s}-T_{i,j,s-1})\cdot\Delta_{i,m_i^*,j}\mid\neg\mathcal{F}\right]$$

$$=\mathbb{E}\left[\sum_{a_j\in\mathcal{K}}\left(\sum_{s:a_j\in D_s,s\leq s_{j,1}}(T_{i,j,s}-T_{i,j,s-1})+\sum_{s:a_j\in D_s,s>s_{j,1}}(T_{i,j,s}-T_{i,j,s-1})\right)\cdot\Delta_{i,m_i^*,j}\mid\neg\mathcal{F}\right]$$

$$\leq\mathbb{E}\left[\sum_{a_j\in\mathcal{K}}\left(T_{i,j,s_{j,1}}+\sum_{s:a_j\in D_s,s>s_{j,1}}(T_{i,j,s}-T_{i,j,s-1})\right)\cdot\Delta_{i,m_i^*,j}\mid\neg\mathcal{F}\right]$$

$$\leq\sum_{a_j\in\mathcal{K}}\left(\frac{96\log T}{\Delta^2_{i,j_{s_{j,1}},j}}+\sum_{s:a_j\in D_s,s>s_{j,1}}\left(96\log T/\Delta^2_{i,j_s,j}-96\log T/\Delta^2_{i,j_{s-1},j}\right)\right)\cdot\Delta_{i,m_i^*,j}$$

$$\le \sum_{a_j \in \mathcal{K}} \frac{96 \log T}{\Delta^2_{i,m_i^*,j}} \cdot \Delta_{i,m_i^*,j} \le 96K \log T/\Delta\,,$$

where the second last line is due to the definition of $s_{j,1}$ and the above analysis.

Above all,

$$\mathbb{E}\left[\sum_{t=1}^{T} \sum_{a_j} \mathbb{1}\left\{\bar{A}_i(t) = a_j\right\} \cdot \Delta_{i,m_i^*,j} \mid \neg\mathcal{F}\right] \le \text{Eq. (2)} \le 96N^2 \log T/\Delta^2 + 96K \log T/\Delta\,.$$

□

Lemma A.2.

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\bar{A}_i(t) = \emptyset\right\} \cdot \mu_{i,m_i^*} \mid \neg\mathcal{F}\right] \le 192N^2 \log T/\Delta^2\,.$$

Proof. Based on the AOGS algorithm, when $\mathrm{E}_i = \mathrm{True}$, the central platform would assign arms in $\mathcal{A}_i$ to player p_i in a round-robin manner. Since the number of arms $\lvert \cup_{i:\mathrm{E}_i=\mathrm{True}} \mathcal{A}_i \rvert$ to be explored is larger than the number $\sum_i \mathbb{1}\{\mathrm{E}_i = \mathrm{True}\}$ of players with $\mathrm{E}_i = \mathrm{True}$ based on the elimination condition in Line 6, we can assume that there is no collision in the exploration phase as discussed in Section 4. So the regret caused by collision only occurs during time with $\mathrm{E}_i = \mathrm{False}$.

Denote $\underline{t}_s$ and $\bar{t}_s$ as the round index when p_i sets E_i as False for the s-th time and as True for the $s+1$-th time, respectively. Recall that when $\mathrm{E}_i = \mathrm{False}$, p_i will always select arm A_i. Here we use j_s to represent the arm that is selected by p_i from time $\underline{t}_s$ to $\bar{t}_s$.

Further, recall that in the AOGS algorithm, each time an arm is added into $\mathcal{D}_i$ (Line 13), the eliminated arms may be contained into $\mathcal{A}_i$ again. And only when other players focus on their currently most preferred arm, such operation of adding arms to $\mathcal{D}$ happens. Based on Lemma A.3, such an operation happens for at most N times. For any player $p_{i'}$, denote $t_{i',r}$ as the round index when $p_{i'}$ adds arms to $\mathcal{D}_{i'}$ (Line 13) for r-th time. Then $\left\{t_{i',r}\right\}_{r\in[N]}$ further divide $\left\{[\underline{t}_s, \bar{t}_s]\right\}_{s\in[N]}$ into at most $2N$ slices. We use $\underline{t'}_s, \overline{t'}_s$ to represent the start round and end round index of the s-th slice, where $s \in [2N]$. Based on Lemma A.5, $p_{i'}$ and p_i would select the same arm for at most $96 \log T/\Delta^2$ times within each slice.

Then the regret satisfies

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\bar{A}_i(t) = \emptyset\right\} \cdot \mu_{i,m_i^*} \mid \neg\mathcal{F}\right] &\le \mathbb{E}\left[\sum_{s=1}^{N} \sum_{t=\underline{t}_s}^{\bar{t}_s} \mathbb{1}\left\{\bar{A}_i(t) = \emptyset\right\} \cdot \mu_{i,m_i^*} \mid \neg\mathcal{F}\right] \\
&= \mathbb{E}\left[\sum_{s=1}^{N} \sum_{t=\underline{t}_s}^{\bar{t}_s} \mathbb{1}\left\{\bar{A}_i(t) = \emptyset, A_i(t) = j_s\right\} \cdot \mu_{i,m_i^*} \mid \neg\mathcal{F}\right] \\
&\le \mathbb{E}\left[\sum_{i'\neq i} \sum_{s=1}^{N} \sum_{t=\underline{t}_s}^{\bar{t}_s} \mathbb{1}\{A_i(t) = A_{i'}(t) = j_s\} \cdot \mu_{i,m_i^*} \mid \neg\mathcal{F}\right] \\
&\le \mathbb{E}\left[\sum_{i'\neq i} \sum_{s=1}^{2N} \sum_{t=\underline{t'}_s}^{\overline{t'}_s} \mathbb{1}\{A_i(t) = A_{i'}(t)\} \cdot \mu_{i,m_i^*} \mid \neg\mathcal{F}\right] \\
&\le \sum_{i'\neq i} \sum_{s=1}^{2N} 96 \log T/\Delta^2 \cdot \mu_{i,m_i^*} \\
&\le 192N^2 \log T/\Delta^2\,.
\end{aligned}$$

□

Lemma A.3. *In the offline GS algorithm, at most $N-1$ arms have been proposed by players before the algorithm stops.*

Proof. Based on the offline GS algorithm, once an arm is proposed, it has a temporary player. By contradiction, once N arms have been proposed, it means that N players are occupied. In this case, each player has a partner and the algorithm stops. □

Lemma A.4.

$$\mathbb{P}\left(\mathcal{F}\right) \le 2NK/T\,.$$

Proof.

$$\begin{aligned}
\mathbb{P}\left(\mathcal{F}\right) &= \mathbb{P}\left(\exists 1 \le t \le T, i \in [N], j \in [K] : \left|\hat{\mu}_{i,j}(t) - \mu_{i,j}\right| > \sqrt{\frac{6\log T}{T_{i,j}(t)}}\right)\\
&\le \sum_{t=1}^{T}\sum_{i\in[N]}\sum_{j\in[K]} \mathbb{P}\left(\left|\hat{\mu}_{i,j}(t) - \mu_{i,j}\right| > \sqrt{\frac{6\log T}{T_{i,j}(t)}}\right)\\
&\le \sum_{t=1}^{T}\sum_{i\in[N]}\sum_{j\in[K]}\sum_{s=1}^{t} \mathbb{P}\left(T_{i,j}(t) = s, \left|\hat{\mu}_{i,j}(t) - \mu_{i,j}\right| > \sqrt{\frac{6\log T}{s}}\right)\\
&\le \sum_{t=1}^{T}\sum_{i\in[N]}\sum_{j\in[K]} t\cdot 2\exp(-3\ln T)\\
&\le 2NK/T\,,
\end{aligned}$$

where the second last inequality is due to Lemma B.1. □

Lemma A.5. *For any player p_i, let $\bar{T}_i = 96\log T/\Delta^2$. For any two arms j, j' with $\mu_{i,j} > \mu_{i,j'}$ and $\sigma_i(a_j) \in [1, \sigma_i(m_i^*)]$, if $T_i(t) := \min\left\{T_{i,j}(t), T_{i,j'}(t)\right\} > \bar{T}_i$, we have $\mathrm{UCB}_{i,j'}(t) < \mathrm{LCB}_{i,j}(t)$ conditioned on $\neg\mathcal{F}$.*

Proof. By contradiction, suppose $\mathrm{UCB}_{i,j'}(t) \ge \mathrm{LCB}_{i,j}(t)$. According to $\neg\mathcal{F}$ and the definition of LCB and UCB, we have

$$\mu_{i,j} - 2\sqrt{\frac{6\log T}{T_i(t)}} \le \mathrm{LCB}_{i,j}(t) \le \mathrm{UCB}_{i,j'}(t) \le \mu_{i,j'} + 2\sqrt{\frac{6\log T}{T_i(t)}}\,.$$

We can then conclude $\Delta_{i,j,j'} = \mu_{i,j} - \mu_{i,j'} \le 4\sqrt{\frac{6\log T}{T_i(t)}}$, which implies that $T_i(t) \le \frac{96\log T}{\Delta^2_{i,j,j'}} \le \frac{96\log T}{\Delta^2}$. This contradicts the fact that $T_i(t) > \bar{T}_i$. □

Lemma A.6. *Conditioned on $\neg\mathcal{F}$, at any time t, $\mathrm{UCB}_{i,j}(t) < \mathrm{LCB}_{i,j'}(t)$ implies $\mu_{i,j} < \mu_{i,j'}$.*

Proof. According to the definition of LCB and UCB, we have

$$\mathrm{LCB}_{i,j}(t) = \hat{\mu}_{i,j}(t) - \sqrt{\frac{6\log T}{T_{i,j}(t)}} \le \mu_{i,j} \le \hat{\mu}_{i,j}(t) + \sqrt{\frac{6\log T}{T_{i,j}(t)}} = \mathrm{UCB}_{i,j}(t)\,,$$

where two inequalities comes from $\neg\mathcal{F}$. Thus if $\mathrm{UCB}_{i,j}(t) < \mathrm{LCB}_{i,j'}(t)$, there would be

$$\mu_{i,j} \le \mathrm{UCB}_{i,j}(t) < \mathrm{LCB}_{i,j'}(t) \le \mu_{i,j'}\,.$$

The lemma can thus be proved. □

B Technical Lemmas

Lemma B.1. *(Corollary 5.5 in Lattimore and Szepesvári [18]) Assume that $X_1, X_2, \ldots, X_n$ are independent, σ-subgaussian random variables centered around μ. Then for any $\varepsilon > 0$,*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i \ge \mu + \varepsilon\right) \le \exp\left(-\frac{n\varepsilon^2}{2\sigma^2}\right)\,, \quad \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i \le \mu - \varepsilon\right) \le \exp\left(-\frac{n\varepsilon^2}{2\sigma^2}\right)\,.$$

Lemma B.2. *(Arrangement of players' round-robin exploration) Suppose there are N players who need to explore their respective N arms. There exists an assignment such that during $2N$ rounds, each player can match with each of its arm for once.*

Proof. Without loss of generality, let's assume that players assign their respective N arms in $2N$ rounds one by one, based on the players' and arms' indices, aiming to ensure that no arm is assigned to more than one player at the same round. By contradiction, suppose when player p_i assigns its j-th arm, there is no available round to make this assignment due to conflicting constraints. Given that player p_i is currently assigning the j-th arm, it implies that there are $2N-j+1$ rounds where no arm is assigned to player p_i. Since none of these rounds satisfy the conflict constraint, it means that the previous $i-1$ players assigned arm j in these $2N-j+1$ rounds. This creates a contradiction since the first $i-1$ players can only occupy $i-1$ rounds when selecting arm j, where $i-1 < 2N-j+1$ with $i \leq N, j \leq N$. □

C Proof of Theorem 6.2

In this section, we analyze the regret of the centralized-UCB algorithm under α-condition. Recall that $\mathcal{F} = \left\{\exists 1 \leq t \leq T, i \in [N], j \in [K] : |\hat{\mu}_{i,j}(t) - \mu_{i,j}| > \sqrt{\frac{6\log T}{T_{i,j}(t)}}\right\}$ is the failure event that the estimated reward is far from the expected reward at some time and some player-arm pair.

For any player p_i with $i \in [N]$, we know that its stable arm is a_i under α-condition. Thus its regret can be decomposed as

$$Reg_i(T) \leq \mathbb{E}\left[\sum_{k:\mu_{i,k}<\mu_{i,i}} \Delta_{i,i,k} \sum_{t=1}^{T} \mathbb{1}\{\bar{A}_i(t) = k, \neg\mathcal{F}\}\right] + T \cdot \mathbb{P}(\mathcal{F}) \ .$$

The first term is the number of selections for sub-optimal arms. The second term is the regret caused by the bad events.

For the arm a_k such that it is sub-optimal for player p_i, i.e., $\mu_{i,k} < \mu_{i,i}$, it will be selected because the preference for arm a_k of player p_i is estimated higher than its stable matched arm a_i, or player p_i is rejected by arm a_k in the GS algorithm. Note that under α-condition, there is a right order $Q_{i'} = i \in [N]_r$ for player p_i, such that $\forall i' < i'' \leq N, Q_{i''} \in [N]_r$:$\pi_{q_{i'},Q_{i'}} > \pi_{q_{i'},Q_{i''}}$, which means arm $a_{q_{i'}} = a_i$ can only prefer players $p_{Q_1}, p_{Q_2}, \cdots, p_{Q_{i'-1}}$ than player $p_{Q_{i'}} = p_i$. Denote The right-order mapping for α-condition for player p_i is $lr(i)$ so that $Q_{lr(i)} = i$ with Q_i defined in Definition 6.1, and $lr(i) \leq N$. For player p_i, denote $\mathcal{G}_{t,i} := \{\forall 1 \leq i' \leq lr(i) - 1, \bar{A}_{Q_{i'}}(t) \neq i\}$ as the event all players preferred by arm a_i do not select p_i at time t. Then the number of selections for sub-optimal arm a_k can be decomposed as

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\bar{A}_i(t) = k, \neg\mathcal{F}\}\right] \\
=&\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\bar{A}_i(t) = k, \mathcal{G}_{t,i}, \neg\mathcal{F}\}\right] + \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\bar{A}_i(t) = k, \neg\mathcal{G}_{t,i}, \neg\mathcal{F}\}\right] \\
\leq&\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\bar{A}_i(t) = k, \mathrm{UCB}_{i,k}(t) > \mathrm{UCB}_{i,i}(t), \neg\mathcal{F}\}\right] + \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\bar{A}_i(t) = k, \neg\mathcal{G}_{t,i}, \neg\mathcal{F}\}\right] \\
\leq&\frac{24\log T}{\Delta_{i,i,k}^2} + \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\bar{A}_i(t) = k, \neg\mathcal{G}_{t,i}, \neg\mathcal{F}\}\right] \ .
\end{aligned}$$

The last inequality is from Lemma C.1.

For the second term in the RHS of the last inequality, $\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\bar{A}_i(t) = k, \neg\mathcal{G}_{t,i}, \neg\mathcal{F}\}\right]$, we can sum over all sub-optimal arms and it turns out to be

$$\mathbb{E}\left[\sum_{k}\sum_{t=1}^{T} \mathbb{1}\{\bar{A}_i(t) = k, \neg\mathcal{G}_{t,i}, \neg\mathcal{F}\}\right]$$

$$
\begin{aligned}
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\neg \mathcal{G}_{t,i}, \neg \mathcal{F}\}\right] \\
&\leq \mathbb{E}\left[\sum_{i'=1}^{lr(i)-1} \sum_{t=1}^{T} \mathbb{1}\left\{\bar{A}_{Q_{i'}}(t) = i, \neg \mathcal{F}\right\}\right] \\
&\leq \sum_{u'=1}^{lr(i)-1} \sum_{u''=u'+1}^{lr(i)} \frac{24 \log T}{\Delta^2_{Q_{u'}, q_{u'}, q_{u''}}} \\
&\leq \sum_{u'=1}^{lr(i)-1} \sum_{k=1}^{lr(i)-u'} \frac{24 \log T}{(k\Delta_N)^2} \\
&\leq (lr(i) - 1) \left(\sum_{k=1}^{lr(i)-u'} \frac{1}{k^2}\right) \frac{24 \log T}{\Delta_N^2} \\
&\leq (lr(i) - 1) \frac{5\pi^2 \log T}{\Delta_N^2},
\end{aligned}
$$

where the third inequity is from the Lemma C.2. The fourth inequality is from the definition of Δ_N.

Above all, the stable regret of player i can be bounded by

$$
\begin{aligned}
Reg_i(T) \leq& \mathbb{E}\left[\sum_{k:\mu_{i,k}<\mu_{i,i}} \Delta_{i,i,k} \sum_{t=1}^{T} \mathbb{1}\left\{\bar{A}_i(t) = k, \neg \mathcal{F}\right\}\right] + T \cdot \mathbb{P}\left(\mathcal{F}\right) \\
\leq& \sum_{k:\mu_{i,k}<\mu_{i,i}} \Delta_{i,i,k} \frac{24 \log T}{\Delta^2_{i,i,k}} + \Delta_{i,i,k} \left(lr(i) - 1\right) \frac{5\pi^2 \log T}{\Delta_N^2} + 2NK \\
\leq& \frac{24K \log T}{\Delta} + \left(lr(i) - 1\right) \frac{5\pi^2 \log T}{\Delta_N^2} + 2NK \\
\leq& O\left(\frac{K \log T}{\Delta} + \frac{N \log T}{\Delta_N^2}\right),
\end{aligned}
$$

where the second inequality is based on Lemma A.4.

Lemma C.1. *Conditioned on $\neg\mathcal{F}$, under the traditional single-player UCB algorithm with single player p_i, the expected number of times at which the UCB index of arm $a_{j'}$ exceeds that of the better arm a_j, is at most $24\log(T)/\Delta^2_{i,j,j'}$ by round T.*

Proof. Conditioned on $\neg\mathcal{F}$, for any i, j, t we have,

$$
\mu_{i,j} - \sqrt{\frac{6\log(T)}{T_{i,j}(t-1)}} < \hat{\mu}_{i,j}(t-1) < \mu_{i,j} + \sqrt{\frac{6\log(T)}{T_{i,j}(t-1)}}\,. \tag{6a}
$$

Recall that the UCB index is:

$$
\mathrm{UCB}_{i,j}(t) = \hat{\mu}_{i,j}(t-1) + \sqrt{\frac{6\log(T)}{T_{i,j}(t-1)}}\,. \tag{6b}
$$

The event that arm $a_{j'}$ is successfully selected for player p_i rather than the better arm a_j at time t implies that

$$
\mathrm{UCB}_{i,j'}(t) > \mathrm{UCB}_{i,j}(t)\,. \tag{6c}
$$

Hence,

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

which leads to

$$
T_{i,j'}(t-1) < \frac{24\log(T)}{\Delta^2_{i,j,j'}}\,,
$$

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

□

Lemma C.2. *For any player p_i with right order $Q_{lr(i)}$, the following inequality holds:*

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{i'=1}^{lr(i)-1}\sum_{t=1}^{T}\mathbb{1}\{\bar{A}_{Q_{i'}}(t) = i, \neg\mathcal{F}\}\right] \\
\leq &\mathbb{E}\left[\sum_{u'=1}^{lr(i)-1}\sum_{u''=u'+1}^{lr(i)}\sum_{t=1}^{T}\mathbb{1}\{\bar{A}_{Q_{u'}}(t) = q_{u''}, \mathcal{G}_{t,u'}, \neg\mathcal{F}\}\right] \\
\leq &\sum_{u'=1}^{lr(i)-1}\sum_{u''=u'+1}^{lr(i)}\frac{24\log T}{\Delta^2_{Q_{u'},q_{u'},q_{u''}}}\,.
\end{aligned}
$$

Proof. For player p_i with right order $Q_{lr(i)}$, from α-condition we have that its stable matched arm a_i may prefer $p_{Q_1}, p_{Q_2}, \cdots, p_{Q_{lr(i)-1}}$ than player $p_{Q_{lr(i)}}$. For any $i' < lr(i)$, we know that the number of times player $p_{Q_{i'}}$ selects arm a_i is decomposed as by

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\bar{A}_{Q_{i'}}(t) = i, \neg\mathcal{F}\}\right] \\
=&\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\bar{A}_{Q_{i'}}(t) = i, \mathcal{G}_{t,i'}, \neg\mathcal{F}\}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\bar{A}_{Q_{i'}}(t) = i, \neg\mathcal{G}_{t,i'}, \neg\mathcal{F}\}\right]\,.
\end{aligned}
$$

The event $\neg\mathcal{G}_{t,i'}$ implies that there exists another player $p_{Q_{i''}}$ with $i'' < i'$ that selects the stable arm of $p_{Q_{i'}}$. This leads to a recursion form. But it is easy to verify that every event $\neg\mathcal{G}_{t,i'}$ happens only when there exists two players $p_{Q_{u'}}, p_{Q_{u''}}$ with $u' < u'' \leq lr(i')$, such that player $p_{Q_{u'}}$ explores the stable matched arm $a_{q_{u''}}$ of $p_{Q_{u''}}$, i.e., $p_{Q_{u'}}$ selects $a_{q_{u''}}$ conditioned on $\mathcal{G}_{t,u'}$. And thus it holds that

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{i'=1}^{lr(i)-1}\sum_{t=1}^{T}\mathbb{1}\{\bar{A}_{Q_{i'}}(t) = i, \neg\mathcal{F}\}\right] \\
\leq &\mathbb{E}\left[\sum_{u'=1}^{lr(i)-1}\sum_{u''=u'+1}^{lr(i)}\sum_{t=1}^{T}\mathbb{1}\{\bar{A}_{Q_{u'}}(t) = q_{u''}, \mathcal{G}_{t,u'}, \neg\mathcal{F}\}\right] \\
\leq &\sum_{u'=1}^{lr(i)-1}\sum_{u''=u'+1}^{lr(i)}\frac{24\log T}{\Delta^2_{Q_{u'},q_{u'},q_{u''}}}\,.
\end{aligned}
$$

□

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction clearly describe the scope and outline our main contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses the limitation of this paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide techniques clearly in the paper, and their detailed proofs are in Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 provides the settings and results of our experiments carefully.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The codes are uploaded in supplementary material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 provides the settings and results of our experiments carefully.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: We provide the error bar in the results of experiments in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer ”Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We do not provide specific information on the computer resources used, as most experiments on multi-armed bandits are lightweight.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn’t make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper complies with NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: The paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.