Title

Statistical Matching Model in Centralized Two-sided Markets With Contexts, Constraints, and Incentive Compatibility Consideration

Author

Li, Yuantong

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Statistical Matching Model in Centralized Two-sided Markets With Contexts, Constraints, and Incentive Compatibility Consideration

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Yuantong Li

2024

ABSTRACT OF THE DISSERTATION

Statistical Matching Model in Centralized Two-sided Markets
With Contexts, Constraints, and Incentive Compatibility Consideration

by

Yuantong Li
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2024
Professor Guang Cheng, Co-Chair
Professor Xiaowu Dai, Co-Chair

Two-sided online matching is a crucial aspect of optimizing social welfare sequentially within economic frameworks, achieved through pairing participants via third-party platforms. These platforms are utilized across various marketplaces such as college admissions, ride-sharing, doctor placement, dating, and job applications. Typically, these markets allocate indivisible "good" to multiple agents based on mutual compatibility, with preferences often being unknown due to the large participant volume, making it explicitly challenging. Moreover, matching markets inherently involve scarcity due to limited resources on both sides. This dissertation presents significant advances in statistical sequential modeling for two-sided online matching markets, considering dynamic markets, quota constraints, and participants' incentive compatibility. Situated at the intersection of sequential decision-making algorithm design and economics, this work introduces new algorithms, theories, and insights with applications spanning economics, statistics, and machine learning.

Part I establishes foundational concepts of statistical sequential decision making and

relevant economic terminology. Chapter 1 explores bandit algorithms, probability theory, and concentration inequalities, while Chapter 2 elucidates essential concepts of two-sided matching markets from an economic perspective, laying the groundwork for subsequent applications.

Part II presents a theoretical framework for multi-agent competitive two-sided matching markets, crucial for online recommendation systems in job markets. The first project, detailed in Chapter 3, introduces an online statistical ridge estimation method for the dynamic matching problem (DMP) with its application in the LinkedIn text data. The second project, discussed in Chapter 4, presents an online statistical sequential decision-making method for the competing matching under complementary preferences recommendation problem (CM-CPR), along with a novel algorithm addressing both complementary preferences and quota constraints simultaneously.

The dissertation of Yuantong Li is approved.

Arash Ali. Amini

Will Wei Sun

Xiaowu Dai, Committee Co-Chair

Guang Cheng, Committee Co-Chair

University of California, Los Angeles

2024

To my mother and father.

To my beloved wife.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I want to express my heartfelt thanks to my PhD Advisor Prof. Guang Cheng, Prof. Xiaowu Dai from UCLA, and previous PhD Advisor Prof. Will Wei Sun from Purdue University for their support and inspiration on my Ph.D. research on the two-sided online matching market over the past five years. After five years study, I gradually become an independent researcher with capability of critical thinking, independent academic writing and presentation, and collaboration with peers and senior researchers. Secondly, I would like to thank my committee members: Prof. Arash A. Amini for his insight, critique and suggestions toward better status of this dissertation. Thirdly, I would like to thank Prof. Mahtash Esfandiari for her kindness support of my job and academic application during last year. It is my great pleasure to complete my adventure in Department of Statistics at UCLA.

I wish to express my profound gratitude to my esteemed collaborators who have significantly contributed to my research journey. Firstly, I extend my deepest appreciation to Prof. Rui Feng, my mentor during the 2017 summer research at the University of Pennsylvania. Prof. Feng's unwavering support during my PhD application process, coupled with her invaluable guidance in both academic and personal matters to my growth. Secondly, I am honored to acknowledge Prof. Fei Wang, with whom I had the privilege to work as a research assistant at Weill Cornell University in 2019. Collaborating with Prof. Wang provided me with an invaluable opportunity to immerse myself in the research of computer science. Thirdly, I extend my thanks to Dr. Chi-Hua Wang. Our collaboration on projects exploring two-sided matching and bandit algorithms has been a defining experience of my PhD journey. Dr. Wang's insightful feedback about bandit and unwavering patience have not only enriched my research endeavors but have also contributed significantly to my development as a independent researcher.

I also want to give a shoutout to my PhD cohort: Mr. Shuang Wu, Mr. Mingxuan

Zhang, and Dr. Yiran Jiang, Mr. Tian Xia. We bonded big time during our PhD days at Purdue, from playing Warcraft III till mid-night to reviewing for qualifying exams back in the crazy summer of 2020 at HAAS 271. Huge thanks also go to my college colleagues: Dr. Boyuan Pan, Yu Lan, and Dr. Sheng Zhang. They've always been by my side throughout this wild ride of life, offering tons of advice and just being great pals all around. I would also like to thank to my collaborators and friends, Dr. Qiyu Han, Mr. Jingyuan Chou, Dr. Zhanyu Wang, Dr. Wenjie Li, Dr. Yitao Li, Mr. Lin Gan, Prof. Yue Xing, Dr. Botao Hao, Dr. Shirong Xu, Prof. Chengchun Shi, Prof. Rong Ma, Prof. Jie Xu, Prof. Sendong Zhao, Prof. Chang Su, Prof. Zhenxing Xu, Dr. Qi Ma, Dr. Weilian Zhou, Dr. Haoyu Chen, Dr. Lili Wu, Dr. Tengyu Xu, Dr. Yuan Feng, Dr. Shiyu Wang, Dr. Bing Xue, Mr. Chenzhao Guo, Prof. Yunke Li, Dr. He Guo, Ms. Wenli Zhang, Mr. Hang Ren, Mr. Xin Shi, Dr. Weiwei Han, Dr. Qi Zhao, Mr. Zijiang Wang, Dr. Shen Wang, and Dr. Xiaokai Wei, Prof. Zhuoran Yang, Prof. Zhaoran Wang, and Prof. Sujit Ghosh.

Finally, I want to express my heartfelt gratitude to my parents and wife. Their endless support and love have been the guiding light of my PhD's journey. Through every challenge and triumph, their unwavering belief in me has been a source of strength. Their constant encouragement and unwavering presence have lifted me up during my darkest moments and celebrated with me during my achievements. I am profoundly thankful for their enduring love, which has shaped me into the person I am today.

VITA

2022–2024	Ph.D. Candidate in statistics, Department of Statistics, UCLA.
2023	Applied Scientist Intern, Amazon M5 Search AI.
2021, 2022	Applied Scientist Intern, Amazon AWS AI Lab.
2019–2022	Ph.D. student, Department of Statistics, Purdue University.
2019	RA, Department of Healthcare Policy and Research, Cornell University.
2018–2019	RA, Department of Computer Science and Engineering, OSU.
2017	RA, Department of Biostatistics, University of Pennsylvania.
2016–2018	M.S. in Statistics, Department of Statistics, NCSU
2015	Summer Research Program, NCSU.
2012–2016	B.S. in Math, Department of Mathematics, Zhejiang University.

PUBLICATIONS

Y. Li, Guang Cheng, Xiaowu Dai. Two-sided Competing Matching Recommendation Markets With Quota and Complementary Preferences Constraints, *Proceedings of the 41th International Conference on Machine Learning*, (**ICML 2024**) [paper].

Y. Li†, J. Li† , X. Dai. Discussion on "Estimating Means of Bounded Random Variables by Betting" by W. Smith and A. Ramdas, *Journal of the Royal Statistical Society: Series B*, (**JRSSB 2023**)[paper].

P. Ramprasad, **Y. Li**, Z. Yang, Z. Wang, W. Sun, and G. Cheng. Online Bootstrap Inference For Policy Evaluation in Reinforcement Learning, *Journal of the American Statistical Association*, **(JASA 2023)**[paper].

S. Wu, M. Zhang, **Y. Li**, P. Li. When Federated Learning Meets Graph Neural Network, *The 1st International Workshop on Federated Learning with Graph Data,* **(CIKM 2022 Workshop)**[paper].

Y. Li, X. Wei, S. Wang, Z. Wang, G. Cheng, A. Arnold, Debiasing Neural Retrieval via In-batch balancing regularization, *2022 North American Chapter of the Association for Computational Linguistics*, **(NACCL 2022 Workshop)**[paper].

S. Wu, C. Wang, **Y. Li**, G. Cheng. Residual Bootstrap Exploration for Stochastic Linear Bandit, *Uncertainty in Artificial Intelligence*, **(UAI 2022)**[paper].

Y. Li, C. Wang, and G. Cheng. Online Forgetting Process for Linear Regression Models, *In Proc of the 24nd International Conference on Artificial Intelligence and Statistics*, **(AISTATS 2021)**[paper].

Y. Li, F. Wang, M. Yang, F. Yang, E. Cantu, H. Rao, and R. Feng. (2021). Peel Learning for Pathway-Related Outcome Prediction, **(Bioinformatics 2021)**[paper].

S. Zhao, Y. Huang, C. Su, **Y. Li** and F. Wang. Interactive Attention Networks for Semantic Text Matching, *2020 IEEE International Conference on Data Mining*, **(ICDM 2020)**[paper].

Y. Li, Q. Ma, and S. Ghosh. Determining the Number of Mixture Components of Heavy-Tailed Densities, *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2020*, (**KDD 2020**)[paper].

Part I

Foundation of Sequential Decision Making and Two-Sided Matching Markets

CHAPTER 1

Probability, Concentration, and Bandits

1.1 Probability

Proposition 1.1 (Connection between expectation and tail probability)**.** *If $X > 0$ is a non-negative random variable, then*

$$\mathbb{E}[X] = \int_0^\infty f(X > x)dx \tag{1.1}$$

Definition 1.1. (Subgaussian Noise). The noise ϵ's are drawn independently from a σ-subgaussian distribution. That is, for every $\alpha \in \mathbb{R}$, it is satisfied that

$$\mathbb{E}[\exp(\alpha\epsilon)] \leq \exp(\alpha^2\sigma^2/2) \tag{1.2}$$

Proposition 1.2 (Tails of Normal distribution)**.** *Let $g \sim N(0,1)$. Then for all $t > 0$, we have*

$$(\frac{1}{t} - \frac{1}{t^3})\frac{1}{\sqrt{2\pi}}e^{(-t^2/2)} \leq \mathbb{P}(g \geq t) \leq \frac{1}{t}\frac{1}{\sqrt{2\pi}}e^{(-t^2/2)}. \tag{1.3}$$

Definition 1.2 (Martingale)**.** A $\mathcal{F}$-adapted sequence of random variables $\{X_t\}_{t\in N_+}$ is a $\mathcal{F}$-adapted martingale if

(*a*) $\mathbb{E}[X_t|\mathcal{F}_{t-1}] = X_{t-1}$, almostly surely for all $t \in \{2, 3, \cdots\}$; and

(*b*) X_t is integrable.

If the equality is replaced with a less-than (greater-than), then we call $(X_t)_t$ a **supermartingale** (respectively, a **submartingale**).

1.2 Concentration Inequaltiy

Proposition 1.3 (Hoeffding Inequaltiy)**.** *Suppose that the variables $X_k, k = 1, \cdots, n$, are independent, and X_k has mean μ_k and sub-Gaussian parameter σ_k. Then for all $t \geq 0$, we have*

$$Pr[\sum_{k=1}^{n}(X_k - \mu_k) \geq t] \leq \exp\{-\frac{t^2}{2\sum_{k=1}^{n}\sigma_k^2}\} \tag{1.4}$$

Definition 1.3 (Martingale Difference Sequence)**.** An adapted sequence $\{(D_k, \mathcal{F}_k)\}_{k=1}^{\infty}$, such that, for all $k \geq 1$, then

$$\mathbb{E}[|D_{k+1}|] < \infty \text{ and } \mathbb{E}[D_{k+1}|\mathcal{F}_k] = 0 \tag{1.5}$$

As suggested by their name, such difference sequences arise in a natural way from martingales. In particular, given a martingale $\{(X_k, \mathcal{F}_k)\}_{k=0}^{\infty}$, let us define $D_k = X_k - X_{k1}$ for $k \geq 1$. We then have

$$\begin{aligned}\mathbb{E}[D_{k+1}|\mathcal{F}_k] &= \mathbb{E}[X_{k+1} - X_k|\mathcal{F}_k] \\ &= \mathbb{E}[X_{k+1}|\mathcal{F}_k] - X_k \\ &= 0,\end{aligned}$$

using the definition of martingale and the fact that X_k is measurable with respect to $\mathcal{F}_k$. Thus, for any martingale sequence $\{X_k\}_{k=0}^{\infty}$, we have the telescoping decomposition

$$X_n - X_1 = \sum_{k=0}^{n} D_k$$

where $\{D_k\}_{k=1}^{\infty}$ is a martingale difference sequence. This decomposition plays an important role in the following concentration inequalities.

Proposition 1.4 (Bernstein Concentration). *Let* $\{D_k, \mathcal{F}_k\}_{k=1}^{\infty}$ *be a martingale difference, and suppose that* D_k *is a* σ*-subgaussian in an adapted sense, i.e., for all* $\alpha \in \mathbb{R}$. $\mathbb{E}[e^{\alpha D_k}|\mathcal{F}_{k-1}] \leq e^{\frac{\alpha^2\sigma^2}{2}}$ *almost surely. Then, for all* $t \geq 0$,

$$\mathbb{P}[|\sum_{k=1}^{n} D_k| \geq t] \leq 2e^{-\frac{t^2}{2n\sigma^2}}. \tag{1.6}$$

Proposition 1.4 is from Theorem 2.3 of Wainwright (2019) (Wai19) when $\alpha_* = \alpha_k = 0$ and $\nu_k = \sigma$ for all k.

1.3 Bandit Algorithms

The bandit problem is a classic sequential decision making problem. In the simplest form of the bandit problem, there are a fixed number of arms, each with an unknown probability distribution of yielding rewards when played. The objective is to maximize the total reward accumulated over a series of plays.

The challenge lies in balancing the exploration of different arms (trying out different options to learn their rewards) and the exploitation of the information gathered so far (favoring the arms that appear to yield the highest rewards based on past experience).

There are various strategies and algorithms to solve the sequential decision making problem (BC12; Sli19; Mai19; LS20), such as the ϵ-greedy algorithm (ACF02; CLS21c; CLS21a; HSZ22; SZL22), explore-then-commit algorithm (Rob52; AAS09; LWC22), upper confidence bound (UCB) algorithms (LR85; Aue02; LWC21; WWS23), Thompson sampling (Tho33; RV14; RVK18; LCD23), boostrap sampling algorithm (KSV19; WYH20; WWL22; RLY23), information directed sampling methods (RV14; HLQ22; HL22), and betting methods (WWR22; LLD24). These algorithms employ different trade-offs between exploration and exploitation to achieve optimal or near-optimal rewards over time.

The contextual bandit problem extends the classic bandit problem by introducing contex-

tual information or features associated with each bandit. In this setup, each arm is associated with a context or a set of features that provide additional information about the environment or the state of the system. The objective in the contextual bandit problem remains the same: to maximize the total reward accumulated over a series of plays. However, now the reward that a arm yields may depend not only on the bandit itself but also on the context or features associated with it. For example, consider a scenario where you have multiple ads to display to users on a website. Each ad (arm) has its own click-through rate (CTR), but the CTR may vary depending on factors like the user's demographic information, browsing history, or current session context. In this case, the contextual bandit problem arises in deciding which ad to display to a user based on their context, with the goal of maximizing the total number of clicks or some other relevant metric.

Solving the contextual bandit problem requires learning a policy that maps contexts to actions (arms) in a way that maximizes the expected cumulative reward. This typically involves using statistical or machine learning method to model the relationship between contexts, actions, and rewards, and updating the policy based on observed data over time. Algorithms like contextual bandit algorithms and reinforcement learning methods are commonly used to address this problem.

CHAPTER 2

Two-sided Matching Market

2.1 Centralized Two-sided Matching Market

The Centralized Two-sided Matching Market problem, also known as the "stable marriage problem", is a variation of the classical stable marriage problem introduced by (GS62) and summarized in (Rot08). In this problem, there are two groups of participants, traditionally referred to as "men" and "women", though the problem can be applied to any two-sided matching scenario.

In the centralized two-sided matching problem, each participant in one group (e.g., men) has preferences over the participants in the other group (e.g., women), and vice versa. The goal is to find a stable matching where there are no two participants who prefer each other over their current partners.

The deferred acceptance (DA) algorithm is a solution to this problem. Here's how it works:

1. Initialization: Initially, all participants are free and unmatched.

2. Proposal Phase: In each round, each unmatched participant (e.g., man) proposes to the most preferred unmatched participant (e.g., woman) on his list whom he has not yet proposed to.

3. Acceptance Phase: Each unmatched participant (e.g., woman) who receives proposals holds on to the best proposal she has received so far and rejects the rest. If a participant receives multiple proposals, she rejects all but the most preferred one.

4. Iteration: The proposal and acceptance phases continue until all participants are matched.

5. Termination: The algorithm terminates when no unmatched participants remain.

The resulting matching is *stable* because, by design, no participant prefers any other participant over their current partner. If there were a blocking pair where a man and a woman both prefer each other over their current partners, they would have already been matched during the algorithm's execution.

Overall, the DA algorithm provides a stable and efficient solution to the centralized two-sided matching problem by ensuring that each participant ends up with a partner they find acceptable and that no unstable pairings exist.

2.2 Decentralized Two-sided Matching Market

In the decentralized two-sided matching problem, participants on both sides of the market (e.g., buyers and sellers, employers and job seekers) have preferences over potential matches, but there is no central authority coordinating the matching process. Instead, participants have to make their own decisions about whom to match with based on the information available to them.

This decentralized setup often arises in real-world scenarios where agents made their own decisions and can negotiate directly with potential matches without centralized control. In this matching process, agents typically engage in a process of searching, evaluating, and negotiating potential matches based on their preferences and constraints. The goal for each agent is to find a satisfactory match that maximizes their utility or meets their specific criteria.

Decentralized two-sided matching markets can be complex and challenging due to several factors:

1. Lack of Information Sharing: Participants may have incomplete information about potential matches, leading to uncertainty and the need for strategies to gather information effectively.

2. Dynamic Environment: The availability and preferences of participants may change over time, requiring adaptive strategies to respond to changing conditions.

3. Negotiation and Bargaining: Participants may engage in negotiation and bargaining to reach mutually acceptable matches, introducing additional complexity and uncertainty into the matching process.

4. Potential for Suboptimal Matches: Without centralized coordination, there is a risk of suboptimal matches or inefficiencies arising from participants' decentralized decision-making processes.

Addressing the decentralized two-sided matching problem often involves developing algorithms, protocols, or mechanisms to facilitate efficient and stable matches while respecting the autonomy and preferences of individual participants. Game theory, mechanism design, and distributed optimization are some of the theoretical frameworks used to study and address decentralized matching problems.

Part II

Statistical Matching Models for Centralized Two-sided Online Markets

CHAPTER 3

Dynamic Matching For Two-Sided Online Market

Two-sided online matching platforms are employed in various markets. However, agents' preferences in the current market are usually implicit and unknown, thus needing to be learned from data. With the growing availability of dynamic side information involved in the decision process, modern online matching methodology demands the capability to track shifting preferences for agents based on contextual information. This motivates us to propose a novel framework for this dynamic online matching problem with contextual information, which allows for dynamic preferences in matching decisions. Existing works focus on online matching with static preferences, but this is insufficient: the two-sided preference changes as soon as one side's contextual information updates, resulting in non-static matching. In this paper, we propose a dynamic matching algorithm to adapt to this dynamic online matching problem. The key component of the proposed dynamic matching algorithm is an online estimation of the preference ranking with a statistical guarantee. Theoretically, we show that the proposed the dynamic matching algorithm delivers an agent-optimal stable matching result with high probability. In particular, we prove a logarithmic regret upper bound $\mathcal{O}(\log(T))$ and construct a corresponding instance-dependent matching regret lower bound.

3.1 Introduction

Two-sided online matching platforms are utilized in various marketplaces, including college admissions (GS62; Rot08), ride-sharing (LC18; SWS23), medical doctor placement (Rot84),

dating markets (GI89; Knu97; ZBB18), and job-seeking (MKO13; ATK14; GM20; VPD22). In modern job matching platforms, the two sides are represented by recruiters and job-seekers. The platform's objective is to recommend job-seekers to recruiters to determine if these recommendations meet the companies' talent demands. Recruiters provide a matching score for each recommended job-seeker, which the platforms use as feedback to enhance their recommendation mechanisms. However, optimizing this recommendation process is significantly complicated by two intrinsic factors: (1) *competing characteristic*—the supply of job seekers and demand from companies create competition within the market; (2) *dynamic and two-sided preferences*—preferences are not static and are two-sided, with recruiters and job-seekers each having their own criteria and preferences. Recruiters' preferences vary based on the dynamic fitness of candidate profiles for current positions. Similarly, job-seekers have fixed preferences regarding potential employers, roles, locations, salaries, and other job-related aspects. These challenges significantly complicate the formulation of an effective dynamic matching problem. The platform must continuously adapt its algorithms and strategies to cater to the changing preferences and the competitive nature of the job market. This adaptation requires a sophisticated understanding of market dynamics and the ability to dynamically adjust recommendations based on online feedback and evolving preferences on both sides of the job market.

The two-sided preference structure has been extensively studied in the literature as *static* but not *dynamic*. In (LMJ20), the authors assume a static preference and one-sided preference structure (from job-seekers to companies) is known, which is impractical in environments with a large number of job-seekers where it is prohibitively expensive and time-consuming for recruiters to rank them to have dynamic preferences over job-seekers. Similarly, (LCD23) assumes knowledge of a single-sided preference structure and provides an extensive study on *static* complementary preferences, overlooking the dynamic nature of job-matching, such as the constantly changing talent pool. While these prior efforts advance the understanding of matching in static talent market environments and deliver efficient algorithm designs, chal-

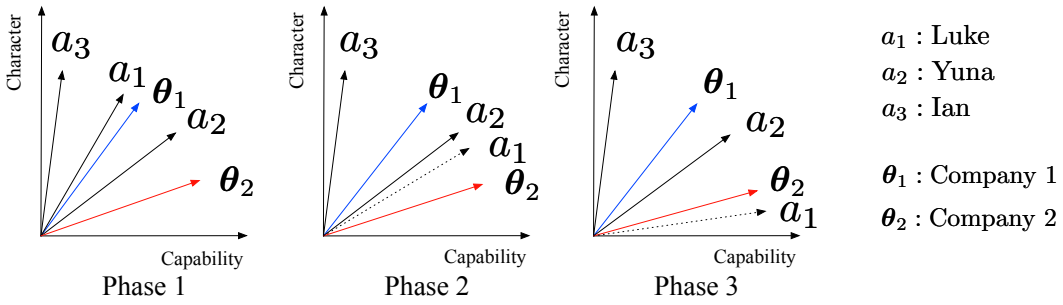

Figure 3.1: Arm a_1's profile changes with an angular velocity, which results in different optimal matching results. Phase 1's optimal matching: (company 1, a_1), (company 2, a_2), Phase 2's optimal matching: (company 1, a_2), (company 1, a_1), and Phase 3's optimal matching: (company 1, a_2), (company 2, a_1).

lenges arise when engineers implement these algorithms in environments with *dynamic preferences*. For instance, as job-seekers regularly update their skills, experiences, and wage expectations, companies dynamically change their preferences over these job-seekers (GAH16).

This concept of dynamic preference is illustrated in Figure 3.1. The scenario includes two companies (Company 1 and Company 2) and three job applicants (a_1, a_2, a_3). The profiles of these job applicants are depicted along two dimensions: capability level (represented on the x-axis) and character level (on the y-axis). The true preference parameters of Company 1 and Company 2 are denoted as $\{\theta_1, \theta_2\} \in \mathbb{R}^2$. The elements within θ_1 and θ_2 represent the respective companies' preference magnitudes for the capability and character traits of the job applicants. It is assumed that all job applicants uniformly prefer Company 1 over Company 2.

In this scenario, job applicant a_1's profile transitions from Phase 1 to Phase 3, while the profiles of a_2 and a_3 remain unchanged. The preference of a company for a job applicant is determined by the fitness (inner product) $\langle \theta_i, x_a \rangle$, where x_a represents the profile of job applicant a for $a \in \{1, 2, 3\}$. The higher this fitness, the more preferable the job applicant is to the company. An interesting observation from this example is that as a_1's profile updates, the company's preference for job applicants shifts, and correspondingly, the optimal matching changes. Such a dynamic nature of preferences and its impact on optimal matchings highlight

the primary challenge in the dynamic online matching market.

The primary goal of the matching platform is to continuously pair companies with the most suitable job applicants, thereby optimizing the overall matching outcome. However, achieving this objective presents a significant challenge: platforms often struggle to accurately estimate companies' true preferences in an ever-changing pool of job applicants. Furthermore, the matching process is complicated by the concept of *bandit feedback*. Specifically, a company only receives feedback—namely, the level of satisfaction—from the job applicant with whom it is currently matched, while the counterfactual (other applicants not matched) outcomes remain unobserved (LS20). This interdependency implies that the feedback received at any given step not only reflects the outcome of the current match but also influences and shapes subsequent matching decisions. This interdependent nature of feedback and decision-making introduces an additional layer of complexity to the dynamic matching process, underscoring the need for adaptive algorithms capable of navigating these complexities effectively.

3.1.1 Major Contributions

In this study, we leverage a critical observation: the optimality of matching decisions in a dynamic environment depends on the sufficient exploration of two-sided preferences. This insight emerges from an elegant integration of online ridge regression with bandit learning strategies, which aims to achieve optimal matching decisions. This integration leads us to propose a novel two-sided matching algorithm in a dynamic environment. We quantify the uncertainties over learned preference parameters to identify a sufficient exploration horizon that enables us to make optimal matching decisions. Consequently, a successful two-sided matching algorithm will yield optimal decisions once the sample size surpasses this sufficient exploration horizon.

We refer to our novel two-sided online matching algorithm as the Dynamic Matching Algorithm (see Section 3.4). The dynamic matching algorithm offers three major advantages:

it centralizes all matching decisions within the platform, addresses the continuously changing dynamics in preference learning, and produces optimal dynamic matching decisions. These attributes ensure the validity and robustness of our algorithm in practical two-sided matching scenarios. Theoretically, we establish an upper bound on agent regret and a corresponding theoretical lower bound in a two-agent and three-arms scenario to demonstrate the optimality of our algorithm. Experimentally, we evaluate the performance of dynamic matching algorithm using both synthetic and real datasets.

In summary, our work advances the algorithmic matching literature with the following three major contributions:

1. Conceptually, we formulate the two-sided online matching problem as a Dynamic Matching Problem (DMP) (see Section 3.2). The DMP encapsulates the ever-changing nature of the talent pool in the job-matching market (see Figure 3.1) and highlights the intrinsic challenges associated with preference learning in dynamic recommendation environments.

2. Methodologically, we introduce a novel dynamic matching algorithm (see Section 3.4, Algorithm 1) that addresses the DMP through a bandit algorithm design. The dynamic matching algorithm initially estimates the dynamic preferences for agents (companies) using a penalized statistical estimation method to construct complete ranking lists over arms (job-seekers). After collecting these rankings, the platform employs the classic deferred-acceptance (DA) algorithm (GS62) to provide the matching object for all participants (agents and arms).

   The design of our multi-agent dynamic matching algorithm extends the single-agent bandit algorithm framework (LS20). Furthermore, we demonstrate that existing online matching algorithms based on the Upper Confidence Bound (UCB) approach fail in the DMP context and suffer from a linear regret (see Figure 3.2), due to the non-shrinking upper confidence bounds for specific arms inherent in the dynamic matching problem's

characteristics. Our algorithm circumvents this issue by employing a sufficient and theoretically-guided optimal exploration sample size. Additionally, through a simple simulation example, we demonstrate this phenomenon (Section 3.3.1).

3. Empirically, we demonstrate that our algorithm exhibit robustness across diverse arm-to-agent preference uncertainties, in scenarios with rapid temporal changes, preference structures, contextual dimensions, and participant sizes in Section 3.7.1. Furthermore, dynamic matching algorithm also showcases its versatility and practical applicability in a dynamic and complex real-world job market, utilizing LinkedIn data, as discussed in Section 3.7.2.

In addition to the methodological contributions listed above, we also discuss our theoretical contributions in the following:

1. **Connection Between Statistical Learing and DMP.** In Section 3.5 Claims 3.5.1 and 3.5.2, we find that a fully correct ranking or an unbiased estimation of the preference parameter are the sufficient conditions to achieve an agent-optimal matching. Our work is the first to elucidate the roles that build the bridge between the statistical learning method and the DMP. Additionally, we introduce a novel conceptualization of the DMP as essentially a dual-layered mixture of ranking and estimation challenges in Section 3.5.3.

2. **Stable Matching.** We initially demonstrate the matching stability of the dynamic matching algorithm at each time step with high probability, as highlighted in Theorem 3.2. A key characteristic is that at any given moment, and with a complete ranking list available, no participant shows a willingness to deviate from the current recommended matching assigned by dynamic matching algorithm in favor of another participant. This aspect of matching stability is crucial in the dynamic matching problem, as it underscores the efficacy and robustness of the algorithm in maintaining satisfactory recommended matchings throughout the matching process.

3. **Regret Upper Bound.** We establish that the dynamic matching algorithm achieves a logarithmic expected cumulative regret over time T (Corollary 3.1). A significant finding of is that the complexity of the dynamic matching problem is directly proportional to the job-seeker feature dimension, number of participants, and matching feedback noise level, and inversely proportional to the gap between different job-seekers. Achieving this regret upper bound presents considerable challenges due to the time-variant dynamic preferences, which makes our proof more complex compared to scenarios with fixed preferences between agents and arms over time, as considered in (LMJ20). To navigate this regret upper bound, we employ novel non-asymptotic concentration results based on the online ridge regression (LWC21) to quantify the union-bound of probability of "invalid ranking" (Lemma 3.2).

4. **Instance-Dependent Regret Lower Bound.** We utilize a two-agent, three-arm example to explore the instance-dependent regret lower bound. Specifically, we decompose the instantaneous regret based on the correctness of other agents' rankings and evaluate the probabilities of correct and incorrect ranking events (Section 3.6.4). By analyzing these events, we can assess the regret on a case-by-case basis and aggregate the regret lower bound across all six identified cases. This analysis indicates that our dynamic matching algorithm will encounter at least a logarithmic regret bound (Theorem 3.3).

3.1.2 Related Work

Our work advances the study of preference-based two-sided market matching, and bandit exploration policy design.

Matching in Two-Sided Markets. We first discuss the matching in discrete and continuous two-sided markets when the preference from both sides are known to the platform. (GS62) studied the two-sided matching markets as a pioneer and proposed the deferred-

acceptance algorithm (also known as the DA Algorithm), which achieved the stable matching. This algorithm (Rot08) has been widely used to match hospitals with residents (Rot86) and students with public schools (APR05b; APR05a) in New York City and Boston. They focused on discrete two-sided matching models without money transfer. (KTY18; NV19; ABY21) focused on the two-sided market with side constraint, e.g., different races should have the same admitting proportions in the college admission. However, these results assume that preferences from both sides are known to the platform, which is fundamentally different from our setting, where agents on the one side of the market's preferences are *unknown* and need to be learned through historical interactions.

In practice, there usually exists a centralized platform helping agents to match with each other, which exhibits the same setting as our DMP. (LMJ20) is one of the first work which considers the case that agents need to learn their preferences through bandit techniques in the centralized platform. (JWW21) considers that both sides' preferences are represented by utility functions over contexts and allow money transfer. They optimize the total utility in the viewpoint of the platform, which is different from ours. We focus on minimizing the individual agent's regret and considering the case where there is no money transfer among agents.

For example, monetary transfer is prohibited in the job application market. In a similar setting, (CS22) considers the case when both users and providers do not know their true preferences a priori and incorporate costs and money transfers among agents to faithfully model the competition among agents and discuss the fairness in the matching. (MWX22) considers the uncertain utility of matching two agents in the episodic reinforcement learning setting. (LCD23) studied the two-sided matching market with complementary preference with quota constraints. However, most of the previous work considers the case where preference is fixed.

Bandit Exploration Strategy. Bandit algorithms (LS20) and reinforcement learning (SB18) are modern strategies to solve sequential decision making problems. They have received attentions in statistics community for business and scientific applications including

dynamic pricing (CSW22; WWS23), online decision making (SZL22; CLS21b), dynamic treatment regimes (LLK19; QLF20), and online causal effect in two-sided market (SWL23). The two-sided competing matching problem can be transformed into a sequential decision-making problem (DK05; LMJ20; Sar21).

To tackle the two-sided matching problem in the bandit framework, researchers transform the matching objects into bandit notation and assume that one side of market participants can be represented as agents (preferences are unknown) and the other side participants of the market can be viewed as arms (preferences are known), and transform this problem into a multi-agent bandit competing problem. (LMJ20) considered that an agent could only match with one arm at one time, such as in the dating market, where (Sar21) considered the case that an agent could match with multiple arms, such as in the lending market. However, these works do not consider the arms' contextual information and hence are not capable of tackling our dynamic matching problem.

Notations. We denote $[N] = [1, 2, ..., N]$. Define the capital $X \in \mathbb{R}^d$ be the d-dimensional random vector. Let $x \in \mathbb{R}^d$ represents a d-dimensional vector, $x^{(r)}$ represents the r-th element of vector x, and the bold $\mathbf{X} \in \mathbb{R}^{d \times d}$ represents a real valued matrix. Let $\mathbf{I}_d = \text{diag}(1, 1, ..., 1) \in \mathbb{R}^{d \times d}$ represent a $d \times d$ diagonal identity matrix. Denote $\lceil x \rceil$ as the minimum integer greater than x. We denote T as the time horizon.

3.2 Dynamic Matching Problem

This section formulates the Dynamic Matching Problem (DMP).

3.2.1 Environment

We use matching of job applicants and companies as the running example throughout the paper. There are three primary roles in this environment: the organizer (recommendation platform), job applicants, and companies. The goal of the organizer is to recommend the

optimal job applicant to companies within this dynamic, online, competitive environment. We begin by introducing three essential elements in the DMP.

(I) Participants. In this centralized platform, there are N companies (agents) denoted by $\mathcal{N} = \{p_1, p_2, ..., p_N\}$, and K job applicants (arms) denoted by $\mathcal{K} = \{a_1, a_2, ...a_K\}$. We assume that the number of companies ($N = |\mathcal{N}|$) is fewer than the number of job applicants ($K = |\mathcal{K}|$).[1]

(II) Two-sided Preferences. For DMP, there are two types of preferences: arms to agents' preferences, and agents to arms' preferences.

Arms to agents' fixed and known preference $\pi : \mathcal{K} \mapsto \mathcal{N}$: We assume that there exist fixed preferences from job applicants to companies, and these preferences are known to the centralized platform. For instance, job applicants are typically required to submit their preferences for different companies via the platform. Let $\pi_{j,i} \in [N]$ represent the ranking for company p_i from the perspective of job applicant a_j, and $\pi_j = \{\pi_{j,1}, ..., \pi_{j,N}\}$ denote the complete set of company rankings for arm a_j. Here, π_j is a permutation of $[N]$, and it is assumed that there are no ties in rankings. Using shorthand notation, $p_i >_j p_{i'}$ indicates that job applicant a_j prefers company p_i over company $p_{i'}$. This known arm-to-agent preference is a mild and common assumption in current online matching literature (LMJ20; LRM21; LCD23).

Agents to arms' dynamic and unknown preference $r(t) : \mathcal{N} \mapsto \mathcal{K}, t \in [T]$. Preferences from companies to job applicants are dynamic and are unknown to the platform due to the large scale of K. Denote $r_{i,j}(t)$ as the ranking for the job applicant a_j from the perspective of company p_i and $r_i(t) = \{r_{i,1}(t), ..., r_{i,K}(t)\}$ represents the ranking for all job applicants at time t which is a permutation of $[K]$. We assume that there are no ties in rankings. The notation $r_{i,j}(t) < r_{i,j'}(t)$ indicates that company p_i prefers job applicant a_j over job applicant

[1]Here we also allow job applicants joining and leaving. It is important to note that these job applicants are not static entities within this platform; their composition may vary over time. However, wihtout loss of generality, we assume that at each given time, the number of job applicants remains constant.

$a_{j'}$ at time t. Similarly, $a_j >_i^t a_{j'}$ means that at time t, company p_i prefers job applicant a_j over job applicant $a_{j'}$. The key distinction between the DMP and classic two-sided matching (GS62) is that $\{r_i(t)\}_{i\in[N]}$ are both unknown and dynamic.

(III) Stable Matching and Optimal Matching. We introduce several key concepts in the two-sided matching field (Rot08).

Definition 3.1 (Blocking)**.** A matching m is *blocked by agent* p_i if p_i prefers being single to being matched with $m(p_i)$, i.e. $p_i >_i m(p_i)$. A matching m is *blocked by a pair of agent and arm* (p_i, a_j) if they each prefer each other to the partner they receive at m, i.e. $a_j >_i m(p_i)$ and $p_i >_j m^{-1}(a_j)$.

Definition 3.2 (Stable Matching)**.** A matching m is stable if it isn't blocked by any individual or pair of agent and arm applicant.

Stable matching in a two-sided market ensures that no pair of agent and arm prefers another partner over their current match. This stability is crucial because it fosters efficiency and reduces costs, leading to more satisfied participants and a robust marketplace. (1) *Efficiency* is achieved as all participants are optimally matched, with no blocking pairs present, ensuring that no participant can improve their situation without disadvantaging others. (2) *Reduced transaction costs* arise because stable matchings prevent the need for repeated renegotiations, saving time, effort, and resources. Consequently, stability contributes to the smooth and efficient operation of matching markets, providing predictable and cost-effective outcomes for all involved.

To account for the potential non-uniqueness of stable matching, we introduce further definitions to delineate agent-optimal matching:

Definition 3.3 (Valid Match)**.** With true preferences from both sides, arm a_j is called a *valid match* of agent p_i if there exist a stable matching according to those rankings such that a_i and p_j are matched.

Definition 3.4 (Agent-Optimal Match)**.** Arm a_j is an *optimal match* of agent p_i if it is the most preferred valid match.

Given true preferences, the DA algorithm shown in Appendix 3.10 (GS62) provides a stable matching and is always optimal for members of the proposing side. We use $\overline{m}_t(i)$ to represent the *agent-optimal matching arm* for agent p_i and $\overline{m}_t = \{\overline{m}_t(1), ..., \overline{m}_t(N)\}$ represent the agent-optimal matching from $\mathcal{N}$ to $\mathcal{K}$ at time t,.

3.2.2 Matching Protocol

At time t, the platform recommends a job applicant a_j from $\mathcal{K}$ for company p_i according to the current matching policy $m_t(\cdot)$. This recommendation is based on the contextual information of the job applicant a_j, $x_j(t) \in \mathbb{R}^d$, which may include demographics, geography, or capabilities, etc,. In response, company p_i evaluates the recommended arm a_j by providing a *noisy matching score* $y_{i,j}(t)$ written as:

$$y_{i,j}(t) = \mu_{i,j}(t) + \epsilon_{i,j}(t), \forall i \in [N], j \in [K], t \in [T], \tag{3.1}$$

where $\mu_{i,j}(t) = \theta_{i,*}^T x_j(t)$ represents the *true matching score*, $\epsilon_{i,j}(t)$ is subgaussian noise (Assumption 3.1), and $\theta_{i,*} \in \mathbb{R}^d$ denotes the *true preference parameter* for company p_i, indicating preference priority across different contexts. Additionally, for company p_i, we define $\overline{\Delta}_{i,j}(t)$ as the *score gap* between the optimal matching arm $\overline{m}_t(i)$ and the currently recommended arm a_j at time t:

$$\overline{\Delta}_{i,j}(t) = \mu_{i,\overline{m}_t(i)}(t) - \mu_{i,j}(t). \tag{3.2}$$

Unlike the score gap always positive in single agent bandit problems, this score gap in DMP can be positive, negative, or zero. Detailed discussion of this gap can be found in Section 3.5.

REGRET. Based on model (3.1), we define the *agent-optimal regret* for p_i as

$$R_i(T) = \sum_{t=1}^{T} \mu_{i,\overline{m}_t(i)}(t) - \mu_{i,m_t(i)}(t). \tag{3.3}$$

This agent-optimal regret represents the difference between the capability of a policy $\mathbf{m}(i) \triangleq \{m_1(i), m_2(i), ..., m_T(i)\}$ in hindsight and the agent-optimal stable matching oracle policy $\overline{\mathbf{m}}(i) \triangleq \{\overline{m}_1(i), \overline{m}_2(i), ..., \overline{m}_T(i)\}$.

SOCIAL WELFARE GAP. We define social welfare gap as the sum of the absolute value of agent-optimal regret $R_i(T)$ across all agents,

$$\text{SOCIAL WELFARE GAP} = \sum_{i=1}^{N} |R_i(T)|.$$

It indicates the difference between the total optimal matching score that could have been achieved under ideal conditions and the actual outcome achieved under the current strategy. Since in DMP, social welfare gap is always non-negative and is easier to compare among different policies, which can be used to provide crucial insights into the efficiency of the matching process.

3.3 Challenges and Resolutions

The challenges of the DMP stem from the ever-changing contextual information of job-seekers, which lead to dynamic preferences. To accurately evaluate these dynamic preferences, the platform must learn from historical data, influenced by the policy it employs. An ideal matching algorithm should effectively balance the trade-off between exploring these contextual information and exploiting them to minimize the agent-optimal regret.

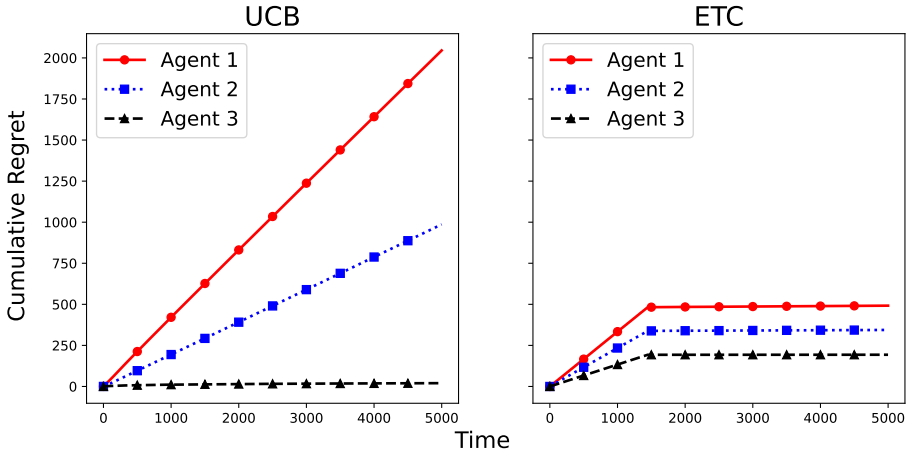

Figure 3.2: Left: upper confidence bound (UCB) algorithm, Right: our algorithm. Incapable exploration of UCB method.

3.3.1 Pitfall: Incapable Exploration of UCB in DMP

In this part, we demonstrate why directly applying the Upper Confidence Bound (UCB) method (refer to Chapter 7 in (LS20)) to balance exploration—by adaptively shrinking the upper confidence bound to quickly find the optimal arm—and exploitation—by frequently pulling the optimal arm to minimize the agent-optimal regret—is infeasible. We show that centralized UCB suffers a *linear* agent-optimal regret in the following DMP example.

Let $\mathcal{N} = \{p_1, p_2, p_3\}$ and $\mathcal{K} = \{a_1, a_2, a_3\}$, with true preferences at time t given below:

$$p_1 : a_1 > a_2 > a_3 \qquad a_1 : p_2 > p_3 > p_1$$

$$p_2 : a_2 > a_1 > a_3 \qquad a_2 : p_1 > p_2 > p_3$$

$$p_3 : a_3 > a_1 > a_2 \qquad a_3 : p_3 > p_1 > p_2$$

Based on the above preference design, the agent-optimal stable matching is (p_1, a_1), (p_2, a_2), (p_3, a_3). However, if the platform wrongly estimates p_3's preference as $a_1 > a_3 > a_2$ based on the UCB estimator, the output stable matching is $(p_1, a_2), (p_2, a_1), (p_3, a_3)$. As a result, p_1 and p_2 suffer positive regrets since their optimal matching arms are a_1 and a_2. In this case,

p_3 will never have the opportunity to correct its mistake $a_1 > a_3$, as it will never be matched with a_1 where arm a_1 has a higher upper confidence bound. *Therefore, the upper confidence bound for a_1 will never shrink, maintaining the preference $a_1 > a_3$.* Consequently, this leads to p_1 and p_2 experiencing linear regrets. We empirically demonstrate this phenomenon in Figure 3.2 and the detailed setting is available Section 4.9.3.1 at the appendix.

However, as shown in Figure 3.2, our algorithm to be introduced in Section 3.4 can avoid this situation through a dedicated design to balance the exploration and exploitation. The advantage of our algorithm is that it can utilize the historical matching data to acquire a good estimate of θ_i^* and $r_i(t)$ with a high probability (Lemma 3.2).

Remark 1. *The above example illustrates that the mechanism to achieve the optimal matching within the DMP is fundamentally different from the single agent bandit problem since the best fitness (optimal) matching arm is not always the top-1 arm (with the highest matching score) for agent due to the competitive characteristics.*

Based on the previous finding, our goal is to design a matching policy $\{m_t(i)\}_{i=1,t=1}^{N,T}$ recommending arms for agents. It seems that we need our algorithm to possess the ability to (i) *learn* the true agent-specific preference parameter $\theta_{i,*}$ to uncover the underlying true preference model, (ii) *design* an exploration strategy based on bandit matching feedback. This strategy efficiently explores potential matching pairs by extracting dynamic ranking information, thereby assisting the algorithm in minimizing agent-optimal matching. To summarize, we have to following to challenges.

3.3.2 Challenge 1: Dynamic Preference Learning

Learning companies' preferences given dynamic job applicants' profiles is challenging since there are numbers of possible matchings between companies and job applicants. Recovering true preference parameters from noisy matching scores requires modeling the relationship between companies and job applicants. We resolve this challenge by considering the para-

metric model (3.1) to capture the relationship between the matching score and job applicants' profiles. Therefore, the main task becomes estimating the underlying preference parameter by adaptively and sequentially conducting matching experiments to have a good statistical property of these estimators. Such an estimate is important for inferring a true preference scheme and informing future matching decisions.

3.3.3 Challenge 2: Bandit Feedback

The platform also needs to balance the exploration (collecting enough job applicants' profiles and companies' matching information) to estimate companies' true preference parameters and the exploitation (providing the optimal matching for companies) tradeoff at each matching time point. Compared to the single-agent bandit problem, the multi-agent competing matching problem is more challenging since the platform needs to handle the multi-agent exploration and exploitation simultaneously. We resolve this challenge by using a new dynamic matching algorithm to balance the multi-agent exploration-exploitation trade-off.

3.4 Dynamic Matching Algorithm

In this section, we propose the dynamic matching algorithm to learn all agents' preference parameters $\{\theta_{i,*}\}_{i=1}^{N}$ and to minimize agent-optimal regret $R_i(T)$. The dynamic matching algorithm functions as a online statistical estimation method, which achieves optimal matching at most of time. This characteristic underscores the algorithm's efficacy in balancing the trade-offs between estimation accuracy and sample efficiency within dynamic matching problem.

Dynamic matching algorithm includes two major steps, the *learning step*, and the *exploitation step*. In the learning step, the platform recommends a_j to p_i randomly. After the learning step ends, platform estimates agents' preference parameters $\{\theta_{i,*}\}_{i=1}^{N}$, constructs estimated preference ranking $\{\widehat{r}_i(t)\}_{i=1}^{N}$, and collects arms preference $\{\pi_j\}_{j=1}^{K}$ in Stage 2 of

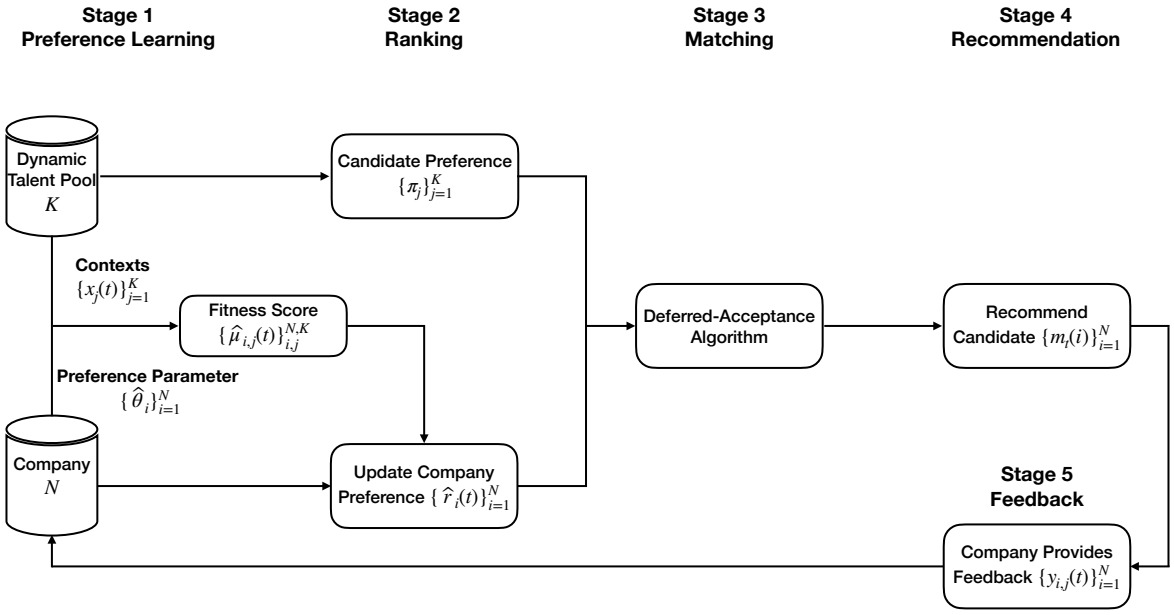

Figure 3.3: A generic design of dynamic matching platform.

Figure 3.3. Then the platform operates the DA algorithm 4 in the appendix with previous estimated preference ranking in Stage 3 of Figure 3.3 to recommend arms to agents in Stage 4 of Figure 3.3. Finally, agents provide matching score $\{y_{i,j}(t)\}_{i=1,j=1}^{N,K}$ to the platform in Stage 5 of Figure 3.3. The detailed dynamic matching algorithm is summarized in Algorithm 1. Below we discuss these two major steps in details.

3.4.1 Learning Step

Let h denote the learning length of dynamic matching algorithm. The key challenge is to find a sufficient learning length, which is a lower bound on h such that the resulting algorithm secures a sub-linear regret. Determine the lower bound of h is a challenging task due to many factors in DMP. We overcome this challenge by utilizing concentration results of the online ridge regression (LWC21) to control probability of invalid ranking such that the agent will enjoy valid ranking with high probability. The theoretical choice of h is provided in

Algorithm 1: `Dynamic Matching (DM) Algorithm`

1 Input: Time horizon T; exploration loop h; ridge parameters $\lambda_i, \forall i \in [N]$; preference $\pi_j, \forall j \in [K]$.
2 *Learn*: Get all companies' estimated true parameters: $(\widehat{\theta}_1(h), ..., \widehat{\theta}_N(h)) =$ `Learning`($\mathcal{N}$, $\mathcal{K}$, $\pi_{j\in[K]}$, $\lambda_{i\in[N]}$, h) from Algorithm 2.
3 *Plan*: Get the matching result: `Planning`(T, $\mathcal{N}, \mathcal{K}$, $\pi_{j\in[K]}$, $\widehat{\theta}_{i\in[N]}(h)$) from Algorithm 3.

Corollary 3.1 in Section 3.6.

After h rounds, the platform collects the historical matching data $\mathbb{D}_i(h) = \{\mathbf{X}_i(h), \mathbf{y}_i(h)\}_{i=1}^N$, where $\mathbf{X}_i(t) = [x_i(1), x_i(2), ..., x_i(t)]^T \in \mathbb{R}^{t\times d}$ denotes p_i's historical matched arms' profiles and $\mathbf{y}_i(t) = [y_i(1), y_i(2), ..., y_i(t)]^T \in \mathbb{R}^t$ represents p_i's historical noisy matching scores. With data $\mathbb{D}_i(h)$, the platform estimates $\{\theta_{i,*}\}_{i=1}^N$ through minimizing the mean square error with an l_2 penalty. Specifically, the objective function is

$$\min_{\theta_i\in\mathbb{R}^d} \|\mathbf{y}_i(h) - \mathbf{X}_i(h)\theta_i\|_2^2 + \lambda_i \|\theta_i\|_2^2, \quad \forall i \in [N], \tag{3.4}$$

where $\lambda_i > 0$ is the penalty parameter. The corresponding *online ridge estimator* for company p_i is

$$\widehat{\theta}_i(h) = \left(\mathbf{X}_i(h)^T\mathbf{X}_i(h) + \lambda_i\mathbf{I}_d\right)^{-1}\mathbf{X}_i(h)^T\mathbf{y}_i(h), \quad \forall i \in [N]. \tag{3.5}$$

The learning step is available in Algorithm 2. From lines 3-7, the platform sequentially updates the collected contextual information $\mathbf{\Sigma}_i(t)$ and matching scores' information $\mathbf{S}_i(t)$. In the end, dynamic matching algorithm obtains the estimated preference parameter $\{\widehat{\theta}_i(h)\}_{i=1}^N$.

3.4.2 Exploitation Step

In the Exploitation Step (Algorithm 3), given the estimated preference parameter $\widehat{\theta}_i(h)$ from the learning step, platform constructs the estimated preference rankings $\{\widehat{r}_t(i)\}_{i=1}^N$ as follows.

Algorithm 2: `Learning Step`

Input: Number of companies N; number of job applicants K; preference $\pi_j, \forall j \in [K]$; ridge parameters $\lambda_i, \forall i \in [N]$; learning length h.
Initialization: $\mathbf{\Sigma}_i(0) = \lambda_i \mathbf{I}_d$, $\mathbf{S}_i(0) = \mathbf{0}_d$, $\widehat{\theta}_i(0) = \mathbf{0}_d$, for $\forall i \in [N]$.
for $t \in \{1, ..., h\}$ **do**
  for $i \in \{1, ..., N\}$ **do**
    MATCH ARM: Recommend job applicant $m_t(i)$ to company p_i.
    COLLECT RESPONSE: Receive matching score $y_i(t)$ from company p_i.
    UPDATE INFORMATION: Update the collected information for company p_i. $\mathbf{\Sigma}_i(t) = \mathbf{\Sigma}_i(t-1) + x_{m_t(i)}(t) x_{m_t(i)}(t)^T, \mathbf{S}_i(t) = \mathbf{S}_i(t-1) + x_{m_t(i)}(t) y_i(t)$.
for $i \in \{1, ..., N\}$ **do**
  ESTIMATE PARAMETERS: Estimate preference parameter $\widehat{\theta}_i(h) = \mathbf{\Sigma}_i^{-1}(t) \mathbf{S}_i(t)$.

At $t = h + 1$, the platform estimates all arms' matching score for agent p_i as

$$\widehat{\mu}_{i,j}(t) = \langle \widehat{\theta}_i(h), x_j(t) \rangle, \quad \forall i \in [N], j \in [K]. \tag{3.6}$$

According these estimated matching scores $\widehat{\mu}_{i,j}(t)$, the platform ranks all arms in descending order. Denote the ranking list as $\widehat{r}_{i,[K]}(t) = \{\widehat{r}_{i,1}(t), ..., \widehat{r}_{i,K}(t)\}$ for agent p_i. The platform then collects the estimated preferences of agents towards arms, $\{\widehat{r}_{i,[K]}(t)\}_{i=1}^N$, along with the arms' true preferences towards agents, $\{\pi_j\}_{j=1}^K$, which are assumed to be known in the DMP (see Section 3.2.1). Following this, the platform executes the DA algorithm. Subsequently, the platform recommends job applicants $\{m_t(i)\}_{i=1}^N$ to each agent. In response, the companies provide their matching scores $\{y_i(t)\}_{i=1}^N$ to the platform, as illustrated in Stage 5 of Figure 3.3.

Remark 2. *(Doubling Trick for Unknown T for Dynamic Matching Algorithm). If T is unknown, the platform can employ the doubling trick (ACF95; BK18). This approach involves initially setting a small T, and if more decisions are required beyond this horizon, the platform restarts the algorithm with a doubled horizon $T := 2T$ and restart the learning step followed by the exploitation step, which suffers the same order regret upper bound as the dynamic matching algorithm with known T.*

Algorithm 3: `Exploitation Step`

Input: Time horizon T; number of companies N; number of job applicants K; estimated true parameters $\widehat{\theta}_i(h), \forall i \in [N]$; preference $\pi_j, \forall j \in [K]$.
for $t \geq h+1$ **do**
  for $i \in \{1, ..., N\}$ **do**
    RANK CANDIDATES: Estimate scores $\widehat{\mu}_{i,j}(t) = \widehat{\theta}_i(h)^T x_j(t), \forall j \in [K]$. Rank all job applicants in descending order by $\{\widehat{\mu}_{i,j}(t)\}_{j=1}^K$ and get the preference ranking list $\widehat{r}_{i,[K]}(t)$.
  MATCH: With two-sided preferences $\{\widehat{r}_{i,[K]}(t)\}_{i=1}^N$ and $\{\pi_j\}_{j=1}^K$, platform computes stable matching m_t via DA Algorithm 4.
  RECEIVE RESPONSE: Company $\mathcal{N}$ provide their matching score $\{y_i(t)\}_{i=1}^N$.

Remark 3. *(Computational Complexity). The computational costs for dynamic matching algorithm consists of the learning step and exploitation step. In the learning step, it has the one time estimation with cost $\mathcal{O}(d^3)$ and matching cost $\mathcal{O}(NK)$. At each exploitation step, it has the ranking cost $\mathcal{O}(K \log K)$ and matching cost $\mathcal{O}(NK)$. So the total computational cost for T steps' DMP is $\mathcal{O}((T-h)(K \log K + NK) + d^3 + NK)$. If T is large, d is small and $N \geq \log K$, the computational cost for DMP is $\mathcal{O}(TNK)$.*

Remark 4. *(Comparison with ϵ-greedy Algorithm). The strategy of a learning step followed by a exploitation step is a typical approach in bandit learning (LS20), particularly when historical data is available. This method is widely used in applications such as website optimization (GLK16) and clinical trials (LRS83). It shares a similar exploration-exploitation tradeoff with the ϵ-greedy algorithm (ACF02), but differs in the timing of exploration. Specifically, our approach conducts explorations initially, while ϵ-greedy employs a randomized strategy with gradually reduced exploration over time. In our real data study, we compare the ϵ-greedy method and our dynamic matching algorithm in Section 3.7.2.*

3.5 Connection Between Statistical Learning and DMP

In this section, we mainly focus on the underlying relationship between the statistical learning and the dynamic matching problem. First in Section 3.5.1, we explore two types of measure

to characterize the correctness of ranking — *correct ranking* and *valid ranking* — that lead to optimal matching. In Section 3.5.2, we find that both unbiased and biased estimations can achieve the optimal matching, and later we provide the motivation of our algorithm's design based on this findings. Finally, in Section 3.5.3, we discuss the foundational terms determining the complexity of the DMP. It is obvious to achieve the optimal matching for

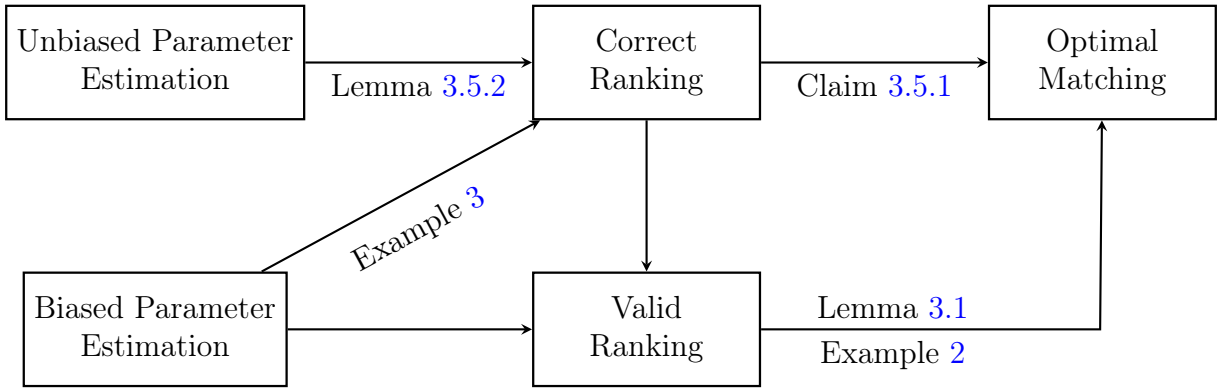

Figure 3.4: Flow of sufficient conditions for optimal matching.

all agents hinges on the construction of correct ranking lists through the DA algorithm. However, given that the platform operates within an online matching framework, there is a non-neglectable possibility that it might generate partially accurate ranking lists due to insufficient matching data. Such inaccuracies can significantly impact the matching results, leading to suboptimal outcomes for agents.

In the following part, we find that the key quantity for assessing the accuracy of ranking lists in the context of DMP is not merely the number of correctly ranked positions but rather the concept of a *valid ranking*, which is a more precise and comprehensive measure that directly influences the ability to achieve optimal matching outcomes.

3.5.1 Correct Ranking and Valid Ranking

In the toy example provided below, we illustrate an intriguing scenario where having zero correct ranking positions can still yield the optimal matching result.

Example 1. Suppose the platform provides correct rankings for all agents except p_i, and assume the optimal matching arm for p_i is at rank j. All ranks from 1 to $j-2$ are permuted (i.e., $\widehat{r}_{i,k} \neq r_{i,k}$ and $\widehat{r}_{i,k} \in \{1, 2, ..., j-2\}$ for all $k \in [j-2]$). Similarly, all ranks from $j+1$ to K are permuted ($\widehat{r}_{i,k} \neq r_{i,k}$ and $\widehat{r}_{i,k} \in \{j+1, j+2, ..., K\}$ for all $k \in [j+1, K]$). Additionally, the platform swaps the arm at rank $j-1$ with the arm at rank j (the optimal matching arm). Despite this arrangement, agent p_i can still achieve an optimal match. This is because all arms ranked before $j-1$ will be rejected based on the preferences from the other side (arm side), as per the DA algorithm, even when the positions of the arm at rank $j-1$ and the optimal matching arm at rank j are switched.

The above example illustrates that the number of correct rankings is not the prime key determinant in achieving optimal matching. We provide the following claim to summarize. Correct ranking is a sufficient condition for the optimal matching. Building on the above insight, we propose that the relative position of a wrongly ranked arm to the optimal arm is crucial in determining the achievement of optimal matching. Consequently, we introduce the term valid ranking to quantify this concept. To better present the concept of valid ranking, we first classify arms based on its relative position over the optimal arm.

Definition 3.5 (Types of Arms)**.** Arms can be classified into two types.

- *Sub-optimal matching arms set*: $\mathcal{K}_{i,\text{sub}}(t) = \{a_j | \overline{\Delta}_{i,j}(t) > 0, j \in [K]\}$, which is similar to the single bandit problem's definition.
- *Super-optimal matching arms set*: $\mathcal{K}_{i,\text{sup}}(t) = \{a_j | \overline{\Delta}_{i,j}(t) < 0, j \in [K]\}$, which is unique for DMP.

Recall the score gap $\overline{\Delta}_{i,j}(t) = \mu_{i,\overline{m}_t(i)}(t) - \mu_{i,j}(t)$ which is defined in Eq. (3.2).

Definition 3.6. (Valid and Invalid Ranking). Ranking $\widehat{r}_{i,[K]}(t)$ is *valid* if whenever arm a_j from the super-optimal matching arms set ranked lower than the optimal matching arm $\overline{m}_t(i)$, i.e, $\widehat{r}_{i,j}(t) > \widehat{r}_{i,\overline{m}_t(i)}(t)$, it follows that score $\mu_{i,j}(t) > \mu_{i,\overline{m}_t(i)}(t)$. On the other hand, if

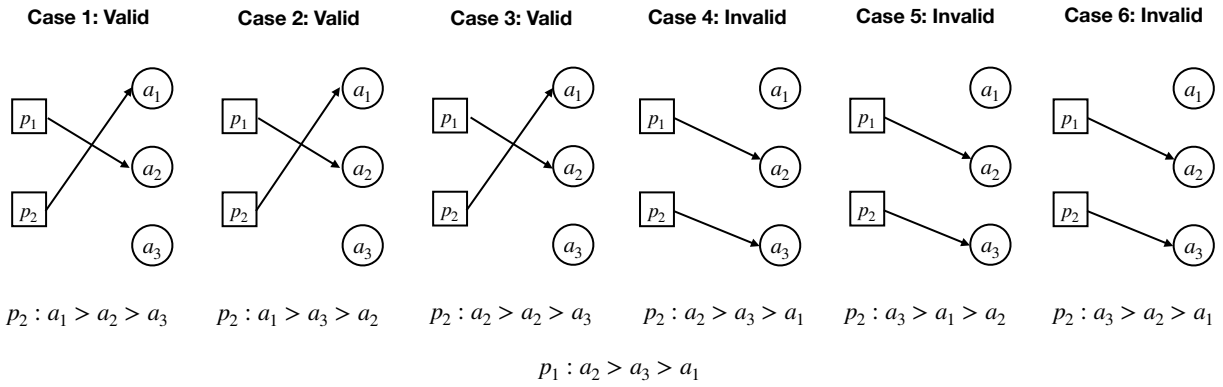

Figure 3.5: The corresponding matching results for p_1 and p_2 if p_1 has valid ranking $a_2 > a_3 > a_1$ and p_2 has six possible rankings. Valid ranking for both and optimal matching: Case 1, 2, and 3. Single invalid ranking and non-optimal matching: Case 4, 5, and 6.

an agent ranks arms from sub-optimal matching arms set is ranked higher than the agent-optimal arm, then it is *invalid*.

Valid ranking necessitates that the arms from the sub-optimal group $\mathcal{K}_{i,\text{sub}}(t)$ are not ranked higher than the optimal arms $a_{\overline{m}_t(i)}$ for agent p_i at time t, rather than requiring fully correct ranking. This perspective contrasts with focusing solely on the number of correct rankings. We conclude that if agent p_i maintains a valid ranking and all other agents also possess valid rankings, then all agents can achieve optimal matching. This indicates that keep all rankings valid is inherently easier, which obviously simplifies the learning objectives and the matching process.

Lemma 3.1. *If all agents maintain valid rankings, they all obtain the agent-optimal matching.*

The detailed proof of Lemma 3.1 is available in Section 3.11 of Appendix. To illustrate Lemma 3.1, we consider a simplified scenario with two agents and three arms.

Example 2. We assume agent p_1 has the valid ranking $a_2 > a_3 > a_1$ and agent p_2 has any one of the six possible rankings (3! permutations), and preferences from agents to arms and

arms to agents are

$$p_1 : a_2 > a_1 > a_3, \qquad p_2 : a_2 > a_1 > a_3$$

$$\pi_1 : p_1 > p_2 \qquad \pi_2 : p_1 > p_2, \qquad \pi_3 : p_1 > p_2.$$

Given this preference setup, the agent-optimal matching for agents is $\{(p_1, a_2), (p_2, a_1)\}$. In addition, the classification of the final matching result (Figure 3.5) is shown in as follows:

Case 1,2,3 (Optimal matching). *If agent p_2 has valid ranking as in Case 1,2,3, the matching result is still (p_1, a_2) and (p_2, a_1). As long as $p_2 : a_3 > a_1$ (omit a_2) and p_1 has valid ranking, no agents suffers regret.*

Case 4,5,6 (Non-optimal matching). *If agent p_2 has invalid ranking as in Case 4,5,6, the matching result is no longer the (p_1, a_2) and (p_2, a_1). Since $p_2 : a_1 > a_3$ (omit a_2) and even p_1 has valid ranking, p_2 suffers regret.*

We observe that even if agent p_2 does not have a correct ranking, it can still achieve an optimal match in Cases 1, 2, and 3. The analysis across these six cases offers insights into the conditions that allow an agent to attain an optimal matching result, despite incorrect in the rankings.

3.5.2 Unbiased Estimation and Biased Estimation

Lemma 3.1 highlights that the valid ranking property is crucial for achieving optimal matching. Intuitively, obtaining a valid ranking initially requires a good estimate of the preference parameters to secure the correct ranking, which in turn ensures convergence to the correct ranking as the sample size increases. Naturally, an unbiased estimator is a good option. Unbiased estimation is a sufficient condition for the optimal matching. However, we present an example demonstrating that correct/valid ranking can be achieved even with biased estimations of the preference parameters.

Example 3. (1).*Correct ranking.* If $\widehat{\theta}_i = \theta_i + b$, where $b \in \mathbb{R}^d$, this results in a biased matching score $\widehat{\mu}_{i,j} = \widehat{\theta}_i^\top x_j = \theta_i^\top x_j + b^\top x_j = \mu_{i,j} + b'$ for all $j \in [K]$. Despite this bias, the

correct ranking $\widehat{r}_i = r_i$ can still be maintained because the shift b' is consistently applied across all arms. (2). *Valid ranking.* This approach involves applying a personalized bias to these estimators, ensuring that as long as we maintain a valid ranking, it can produce an optimal matching.

It is obvious that unbiased estimator can lead to the optimal matching. However, pursuing an unbiased estimator often comes at a high computational cost and can be infeasible in practice. A more practical approach is to adopt a biased version of online estimation but close enough to $\theta_{i,*}$. In this case, biased estimation is still capable of recovering the correct/valid ranking, thereby attaining the optimal matching. That's the reason why we design our algorithm with the online ridge regression method as discussed in Section 3.4.

3.5.3 Foundations of DMP

In addressing the DMP, it's crucial to differentiate between online ranking and online estimation challenges, as they fundamentally guide the strategy of an algorithm's design and implementation. The key quantity that distinguishes between these two types of problems is the "rate of error decay" in relation to sample size (CGZ22). (CGZ22) proposed that online ranking problems tend to be less challenging than online estimation problems in terms of the sample size needed. This is primarily because the number of incorrect rankings in online ranking problems decays exponentially with the increase in sample size, while in online estimation problems, the decay is polynomial. This distinction suggests that the ranking method can swiftly approach the optimal matching in simpler scenarios. However, in more complex problems, the ranking method may struggle due to the influence of feedback noise.

Indeed, the primary challenge that impedes the straightforward application of either the online ranking or online estimation approaches lies in the information about the DMP's complexity available to determine which algorithm is optimal. This complexity leads to the characterization of DMP as typically presenting an online *dual-layered mixture of ranking*

and estimation challenges. Firstly, DMP results in a divergence of difficulty experiences due to its multi-agent nature; at any given decision point, some agents are primarily dealing with a ranking problem while others grapple with an estimation problem. This variation is influenced by the interplay between the preference parameters θ_i for each agent $i \in [N]$ and the contextual attributes $x_j(t)$ for each arm $j \in [K]$. Secondly, the dynamic and online nature of DMP means that the difficulty dynamically shifts for each agent between ranking and estimation challenges, corresponding to continuously evolving contexts.

This insight underscores the need for flexible estimation strategies and decision-making within the DMP framework. Therefore, we have designed our algorithm from the perspective that the worst-case scenario is one where all agents are confronted with dynamic online estimation problems. It ensures that our algorithm is robust and capable of adapting to the most challenging conditions, providing reliable performance even under significant variability and uncertainty in agent preferences and market dynamics.

3.6 Regret Optimality of Dynamic Matching Algorithm

In this section, we outline the properties that our algorithm possesses. We first state several necessary assumption in Section 3.6.1. Next, we provide the agent-optimal logarithmic shape of regret upper bound of dynamic matching algorithm in Section 3.6.2, followed by the critical step in decomposing regret and its key lemma. Furthermore, we demonstrate that our algorithm produce stable matching with a high probability in Section 3.6.3. In addition, we also provide the instance-dependent regret lower bound in Section 3.6.4 and its proof outline.

3.6.1 Regularity Conditions

We first assume the noise follows the subguassian distribution.

Assumption 3.1 (Subgaussian Noise)**.** *The noise $\epsilon_{i,j}(t)$'s are drawn independently from a σ-subgaussian distribution for $t \in [T], i \in [N], j \in [K]$. That is, for every $\alpha \in \mathbb{R}$, it is satisfied that $\mathbb{E}[\exp(\alpha\epsilon_{i,j}(t))] \leq \exp(\alpha^2\sigma^2/2)$.*

Next, we assume that the context $x_j(t)$ distribution of the arm a_j is from distribution $\mathcal{D}_{\mathcal{X}_j}$ and the joint distribution of all arms $\mathcal{D}_{\mathcal{X}} = \mathcal{D}_{\mathcal{X}_1} \times ... \times \mathcal{D}_{\mathcal{X}_K}$ is independent product of individual context distribution $\{\mathcal{D}_{\mathcal{X}_j}\}_{j=1}^K$.

Assumption 3.2. *(Unit Sphere). $\|x_j(t)\|_\infty < 1, \forall j \in [K], t \in [T]$.*

This assumption is common in literature (BB20; LCD23; WWS23) and easy to hold when normalization is applied.

Assumption 3.3. *(Positive-Definiteness). Define $V = \mathbb{E}[XX^T | X \in \mathcal{D}_{\mathcal{X}}]$. Then there exists a deterministic constant $\phi_0 \in \mathbb{R}^+$ such that for all $X \in \mathcal{D}_{\mathcal{X}}$ we have the minimum eigenvalue of the covariance matrix $\lambda_{\min}(V) \geq \phi_0^2$.*

Assumption 3.3 is referred to as the *compatibility condition* in online statistical learning literature (LWC21) and is to ensure that the online ridge estimate trained on samples $X \in \mathcal{D}_{\mathcal{X}}$ converges to the true preference parameter $\{\theta_{i,*}\}_{i=1}^N$ with high probability as the number of samples grows to infinity.

Assumption 3.4. *(Uniform Sub-optimal Minimal Gap Condition). The difference in terms of the scaled matching score between the agent-optimal matching arm and arms from sub-optimal arm $\mathcal{K}_{i,sub}(t)$ for all agents over T is uniformly greater than $\rho > 0$. That is,*

$$\widetilde{\Delta}_{i,\min} = \min_{t\in[T]} \min_{j\in\mathcal{K}_{i,sub}(t)} \overline{\Delta}_{i,j}(t)/||x_{\overline{m}_t(i)}(t) - x_j(t)||_2 > \rho, \forall i \in [N].$$

Assumption 3.4 assures the uniqueness of the agent-optimal match. This assumption extends the fixed uniform sub-optimal minimal gap condition in static matching contexts

(LMJ20) to the dynamic matching framework of DMP where the true matching score $\mu_{i,j}(t)$ varies.

Without loss of generality, we have the following assumption over the preference parameter.

Assumption 3.5. *(Positive Preference).* $\theta_{i,*}^{(r)} > 0, \forall r \in [d], i \in [N]$.

Assumption 3.5 captures the fact that agents evaluate arms' attributes positively but with varying priorities based on the fitness of the arm to the agent.[2]

3.6.2 Regret Upper Bound

In this section, we provide the result of the agent-optimal regret upper bound of our algorithm as follows.

Theorem 3.1. *With Assumptions 3.1 - 3.5 and given the learning length h, if the platform follows the dynamic matching algorithm, agent p_i's regret up to T is upper bounded by*

$$R_i(T) \leq \underbrace{\sum_{t=1}^{h} \overline{\Delta}_{i,m_t(i)}(t)}_{\textit{Part I Regret}} + \underbrace{2\, C_0(\lambda_i) N d \left[\sum_{t=h+1}^{T} \overline{\Delta}_{i,\max}(t)(K - \min_{i\in[N]} \tau_i(t)) \right] \exp\left[- h \frac{2\phi_0^4 \rho^2}{d^2 \sigma^2} \right]}_{\textit{Part II Regret}}, \tag{3.7}$$

where $C_0(\lambda_i) = \exp\left[-4\lambda_i \phi_0^2 \rho^2 / d^2 \sigma^2\right]$, $\tau_i(t)$ is the agent p_i's optimal matching object's ranking position $r_{i,\overline{m}_t(i)}(t)$, and $x_{i,\max} = \|\mathbf{X}_i(h)\|_\infty$ is the maximum absolute value of the context entry.

Proof. We split the regret into two parts, the learning step's regret ("Part I Regret") and the exploitation step's regret ("Part II Regret"). The detailed decomposition procedure can be found in Section 3.6.2.1. To summarize, (i) the learning step's regret is obtained through directly adding the expected score gap between optimal matching arm $a_{\overline{m}_i(t)}$ and policy

[2] In practice, if this assumption does not hold, the platform can initially estimate it, find that parameter entries are negative, and subsequently adjust the sign of the context.

recommended arm $a_{m_i(t)}$ at each time step; (ii) in the exploitation step, the regret is caused by the "bad event", which is the occurring of the *invalid ranking*.

Theorem 3.1 provides the decomposed regret upper bound of the dynamic matching. With optimized h, we show dynamic matching algorithm has a logarithmic regret in the following corollary.

Corollary 3.1. *With* $h = \left\lceil \max_{i\in[N]} \frac{d^2\sigma^2}{2\phi_0^4\rho^2} \log \frac{4C_0(\lambda_i)NK\phi_0^4\rho^2}{d\sigma^2\overline{\Delta}_{i,\max}} T \right\rceil$, *we have*

$$R_i(T) \leq C_1(1 + \log C_2 T) = \tilde{\mathcal{O}}\left(\frac{d^2\sigma^2}{\rho^2} \log(NKT)\right) \tag{3.8}$$

where $C_1 = d^2\sigma^2\overline{\Delta}_{i,\max}/(2\phi_0^4\rho^2)$ *and* $C_2 = 4C_0(\lambda_i)NK\phi_0^4\rho^2/(d\sigma^2\overline{\Delta}_{i,\max})$.

For part II regret in Eq.(3.7), it depends on the gap between the minimum optimal arm rank $\min_{i\in[N]} \tau_i(t)$ across all arms and the worst arm, which measures the difficulty of dynamic matching problem's characteristic. If there exists an agent's optimal arm rank $\tau_i(t) = 1$, the gap is $K - \min_{i\in[N]} \tau_i(t)) = K - 1$. Given the optimal learning length h from Eq. (3.8), we find the regret upper bound depends logarithmically over T, which means that it is a no regret learning method.

We next discuss the dependence of the upper bound on several critical parameters. The quantity $\rho^2/d^2\sigma^2 \log NK$ represents the signal-to-noise ratio for dynamic online matching problem. When the signal-to-noise ratio is high, the complexity level of the DMP is low; conversely, in the low signal-to-noise regime, the complexity of the DMP is high. From another perspective, if the uniform sub-optimal minimal gap $\overline{\Delta}_{i,\min}$ is small, dynamic matching algorithm faces a challenging task as it becomes difficult to distinguish between the optimal arm and the suboptimal arm. Consequently, the ranking provided by the platform is prone to errors, potentially leading to non-optimal stable matching results. Moreover, we observe that both N and K increase at a logarithmic rate in terms of regret when the number of participants increases. We further provide an instance-dependent lower bound, which matches

the order of the regret upper bound (see Section 3.6.4).

3.6.2.1 Proof Outline

We provide key steps to prove Theorem 3.1. We decompose the agent regret $R_i(T)$ into the learning step regret and the exploitation step regret as follows:

$$R_i(T) = \sum_{t=1}^{T} \mu_{i,\overline{m}_t(i)}(t) - \mu_{i,m_t(i)}(t) \leq \underbrace{\sum_{t=1}^{h} \overline{\Delta}_{i,m_t(i)}(t)}_{\text{Part I Regret}} + \underbrace{\left[\sum_{t=h+1}^{T} \overline{\Delta}_{i,\max}(t)(N\mathbb{P}(\widehat{r}_{i,t} \text{ is invalid})) \right]}_{\text{Part II Regret}}.$$

The "Part I regret" is the sum of gaps between the optimal arm and the arm recommended by dynamic matching algorithm during the learning rounds. The Part II regret accumulates during the exploitation step. Based on Lemma 3.1, it is necessary to quantify the probability of an invalid ranking to calculate the instantaneous regret. At time t, the instantaneous regret $\Delta_{i,m_t(i)}(t)N\mathbb{P}(\widehat{r}_{i,t} \text{ is invalid}) \leq \overline{\Delta}_{i,\max}(t)N\mathbb{P}(\widehat{r}_{i,t} \text{ is invalid})$. We quantify the probability of invalid ranking $\mathbb{P}(\widehat{r}_{i,t} \text{ is invalid})$ in the following Lemma 3.2. We then sum all instantaneous regrets from time $h+1$ to T to determine the "Part II Regret," as shown in Equation (3.7). Following this, we provide the upper bound of the invalid ranking probability.

Lemma 3.2. *Assume all agents receive recommended arms from dynamic matching algorithm, the invalid ranking probability's upper bound,*

$$\mathbb{P}(\widehat{r}_{i,t} \textit{ is invalid}) \leq 2d(K - \tau_i(t)) \exp\left(- h\frac{2\lambda_i^2\rho^2\phi_1^4}{d^2\sigma^2} \right). \tag{3.9}$$

3.6.3 Matching Stability of Dynamic Matching Algorithm

In this section, we prove that our algorithm provide stable matching result with high probability.

As we know, the DA algorithm can deliver stable matching based on the two-sided true

preferences. However, this scenario is usually not available in the initial decision rounds of the online setting. Our theory identifies the optimal minimum learning length in Corollary 3.1, which guarantees that dynamic matching algorithm delivers stable matching with high probability (Theorem 3.2). This result connects the online learning techniques and offline matching algorithm design, providing key insights for designing more general online dynamic matching algorithms. In the following theorem, we demonstrate that our algorithm provides a stable match with high probability Ψ. That is, no agents will deviate from the recommended matching arm with a probability of at least Ψ after time t.

Theorem 3.2 (Stability of dynamic matching algorithm)**.** *Given*

$$t \geq \lceil \frac{d^2\sigma^2}{2\lambda_{\min}^2 \rho^2 \phi_1^2} [\log(2d(K-1)) - \log(1-\Psi^{1/N})] \rceil,$$

the dynamic matching algorithm provides an agent-optimal stable matching solution with probability $\Psi > 0$.

Proof. The sketch proof of the stability property of dynamic matching algorithm consists of two steps, naturally following the design of dynamic matching algorithm. In the exploitation step, DA still produces a stable matching result based on estimated preferences and there are no blocked pairs during the matching procedure with high probability and the main proof follows Lemma 3.2 with a union bound of the valid ranking.

3.6.4 Instance-Dependent Regret Lower Bound

We next provide the instance-dependent regret lower bound over a two-agent, three-arm instance and demonstrate the matching lower bound of our algorithm.

In the following lower bound analysis, we consider that there are two agents and three arms in the platform. Contexts are generated from the uniform distribution, $x_j(t) \sim U(0,1)^d, \forall t \in [T], \forall j \in [K]$. We also assume the true preference parameter are designed

as follows $\theta_{1,*} = (\sqrt{1-1/h}, 1/\sqrt{h}, 0, ..., 0)^T \in \mathbb{R}^d, \theta_{2,*} = (\sqrt{1-1/h}, 0, 1/\sqrt{h}, 0, ..., 0)^T \in \mathbb{R}^d$. Noise follows Gaussian distribution with variance σ^2. Then the estimator from Eq. (3.5) satisfies,

$$\widehat{\theta}_i(h)|\mathcal{F}_i(h) \sim N(\bar{\theta}_i, \sigma^2 \mathbf{M}_i), \quad i \in [N],$$

where $\bar{\theta}_i = \big(\mathbf{X}_i(h)^T\mathbf{X}_i(h)+\lambda_i\mathbf{I}\big)^{-1}\mathbf{X}_i(h)^T\mathbf{X}_i(h)\theta_{i,*} \in \mathbb{R}^d$, and $\mathrm{Cov}[\widehat{\theta}_i(h)|\mathcal{F}_i(h)] = \sigma^2\big(\mathbf{X}_i(h)^T\mathbf{X}_i(h)+\lambda_i\mathbf{I}\big)^{-1}\mathbf{X}_i(h)^T\mathbf{X}_i(h)\big(\mathbf{X}_i(h)^T\mathbf{X}_i(h)+\lambda_i\mathbf{I}\big)^{-1} \in \mathbb{R}^{d\times d}$.

Theorem 3.3. *Consider the designed two-agent three-arms instance above. The regret lower bound for agent p_i is,*

$$R_i(T) \geq \sum_{t=1}^{h} \Delta_{i,m_t(i)}(t) + \sum_{t=h+1}^{T} \overline{\Delta}_{i,\min}\big[\mathcal{L}_i^b(t)\mathcal{L}_j^b(t) + \mathcal{L}_i^b(t)\mathcal{L}_j^g(t)\big], \tag{3.10}$$

where $\mathcal{L}_i^g(t) = 1-(3/c_5(t)\sqrt{2})\exp(-c_5^2(t)h/2)$, $\mathcal{L}_i^b(t) = (1/c_7(t)\sqrt{h}-1/c_7^3(t)h^{3/2})\exp(-c_7^2(t)h/2)$, and $c_5(t), c_7(t)$ are contextual time-dependent constants but independent of designing exploration rounds h. With the optimized h provided by dynamic matching algorithm, the order of the regret lower bound is $R_i(T) = \Omega(\log(T))$.

From Theorem 3.3, we find that our algorithm achieve a matching regret lower bound. This lower bound not only depends on both agents' incorrect ranking's probability lower bound $\mathcal{L}_i^b(t), i = 1, 2$, but also the other agent's (p_j) correct ranking estimate's probability lower bound $\mathcal{L}_j^g(t)$.

Remark 5. *Similarly, (LMJ20) provided a regret lower bound by considering other agents submitting truthful rather than strategic rankings to the platform and bounding the maximum number of pulls of non-optimal arms in order to obtain the regret lower bound without context consideration. (JWW21) presented a lower bound for the MAB problem instance with money transfer instead of strict preference constraints compared with ours. (LCW22) considered the minimax lower bound for the multi-agent Markov game where it shares the same action space for all agents. However, the DMP setting is different from the two-sided competing matching*

setting due to the exclusive action selection characteristic. In DMP, for agents, there is exclusivity in action selection and this exclusivity is ubiquitous since one arm cannot be matched with two agents.

3.6.4.1 Proof Outline

The agent-optimal regret $R_{i,t}$ for agent p_i at time t can be decomposed as

$$\begin{aligned} R_{i,t} &= \mathbb{E}[\mathbb{E}[R_{i,t}|\text{other agents' ranking status}]] \\ &= \mathbb{E}[R_{i,t}|\bigcap_{j\neq i}\{\widehat{r}_j(t) = r_j(t)\}]\mathbb{P}(\underbrace{\bigcap_{j\neq i}\{\widehat{r}_j(t) = r_j(t)\}}_{\text{Event I}}) \\ &+ \mathbb{E}[R_{i,t}|\underbrace{\bigcup_{j\neq i}\{\widehat{r}_j(t) \neq r_j(t)\}}_{\text{Event II}}]\mathbb{P}(\bigcup_{j\neq i}\{\widehat{r}_j(t) \neq r_j(t)\}). \end{aligned} \tag{3.11}$$

Here if we assume $i = 1$, Event I becomes $\{\widehat{r}_2(t) = r_2(t)\}$, p_2 has a correct ranking. Event II becomes $\{r_2(t) \neq r_2(t)\}$, p_2 has incorrect ranking. $R_{1,t}(\widehat{r}_2(t) = r_2(t))$ is the first component of Eq.(3.11), which is the expected instantaneous regret for p_1 conditioning on p_2 having correct ranking at time t. Similarly, $R_{1,t}(\widehat{r}_2(t) \neq r_2(t))$ is the second component of Eq.(3.11), the expected instantaneous regret for p_1 conditioning on p_2 having incorrect ranking at time t. $R_{1,t}(\widehat{r}_2(t) = r_2(t))$ and $R_{1,t}(\widehat{r}_2(t) \neq r_2(t))$ are product of the Event I and II's probabilities and corresponding expected regret. Thus, the next step is to quantify the lower bound of two probabilities and expected regret, which we provided in Lemmas 3.6 and 3.7 of appendix.

3.7 Experiments

This section demonstrates the effectiveness and robustness of dynamic matching algorithm in simulation and real data, where the simulation studies include five different settings, its robustness under different context distributions (S1 & S2). The additional experiments such

as different minimal margins (S3), different feature vector dimensions (S4), and different sizes of agents and arms (S5) are available at Section 3.16 of appendix. In real data, we apply the dynamic matching algorithm in a online job-seeking market.

3.7.1 Simulation

From Scenario 1 to Scenario 4, we consider that there are two agents $N = 2$ and three arms $K = 3$. In Scenario 5, we consider that $N = K = 5$. The penalty parameters for all agents are set to be $\lambda = 0.1$ in all scenarios and $T = 1000$.

Scenario 1 (S1): Contexts are generated from a d-dimensional normal distribution with four different fluctuation variance $\zeta = [0.01, 0.05, 0.1, 0.2]$ and $d = 2$, and normalized to have unit norm. $\{\theta_{i,*}\}_{i=1}^{2}$ are randomly generated from uniform distribution and scaled to have unit norm. The uniform minimal sub-optimal condition for this scenario is set to be $\rho = 0.2$. In addition, the noise is generated from normal distribution with $\sigma = 0.05$. We assume that arms to agents' preference π are $a_1 : p_1 > p_2, a_2 : p_2 > p_1, a_3 : p_1 > p_2$. According to Corollary 3.1, the optimal learning step length h is 312.

Scenario 2 (S2): Contextual features move with an *angular velocity* $w_t = 0.005t$ which is different from S1, and $d = 2$. Contexts for arms are still generated from normal distribution and normalized. But for $x_1(t)$, its mean is constantly increasing with a velocity w_t. The true parameters $\{\theta_{i,*}\}_{i=1}^{2}$ are the same as these in S1. The uniform minimal sub-optimal for this scenario is set to be $\rho = 0.2$. The example of moving context with an angular velocity is illustrated in Figure 3.1. We consider three levels of noise $\sigma = [0.01, 0.02, 0.05]$ to test the robustness of our algorithm. In S2, the agent-optimal matching is no longer fixed even when fluctuation level $\zeta = 0$ since context $x_1(t)$ is dynamic. h for three noise levels are $h = [24, 66, 312]$, correspondingly.

Additional experiments settings and results are available in appendix.

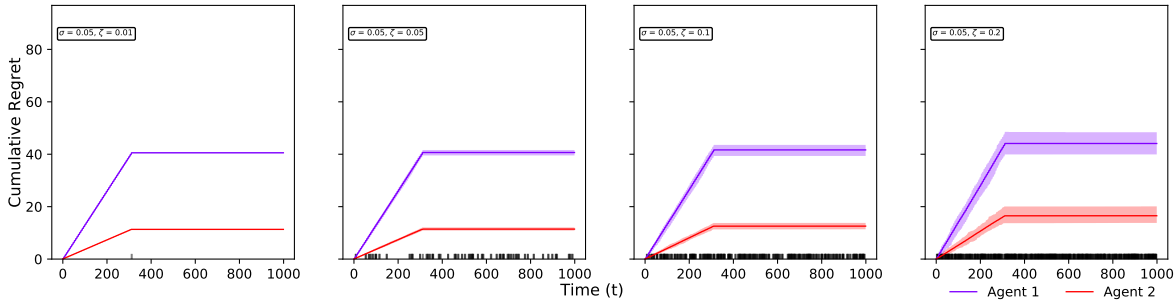

Figure 3.6: S1: Cumulative regret for different context variation levels ζ. Each black stick means a change of optimal matching.

3.7.1.1 Results and Analysis

In Figures 3.6 and 3.7, the horizontal axis represents the time point and the vertical axis represents the cumulative regret. In all figures, we plot the maximum (worst) regret represented as the upper bound shaded line, mean regret represented as the solid line, and minimum (best) regret represented as the lower bound shaded line, over 100 replications.

S1: dynamic matching algorithm is robust to different contexts' variance levels ζ. In Figure 3.6, our dynamic matching algorithm shows the logarithmic regret shape which demonstrates that it is robust to contexts' noise levels. When contexts' variance level ζ increases, the shaded area becomes wider, indicating the uncertainty of the regret increasing, and indicates that the complexity of the DMP is also larger. In this figure and following figures, we use the short black sticks to represent the change of the optimal matching between two adjacent-time points due to the contextual information change. We mark the short black stick at time $t+1$ on the horizontal axis if $\overline{\mathbf{m}}(t) \neq \overline{\mathbf{m}}(t+1)$, which means that the optimal matching result is different on two continuous-time points. The denser the black stick is, the more frequently the agent-optimal matching changes over time. In other words, when the contexts' fluctuation magnitudes increase, the optimal stable matching changes more frequently, and it exhibits the dynamic property of DMP.

S2: dynamic matching algorithm is robust to mean shifting context distributions w_t *and*

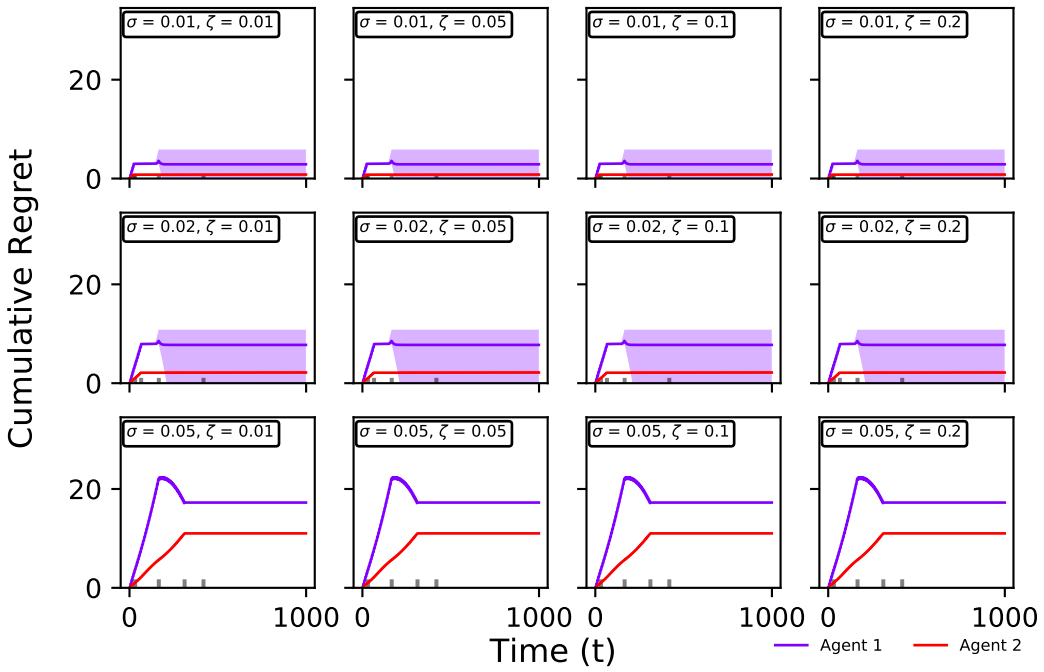

Figure 3.7: Cumulative regret for different noise levels and context variation levels of mean shifting context in Scenario S2.

different levels of observed score noise σ. In Figure 3.7, we present the S2's results when the mean of arm a_1 changes with an angular velocity w_t. In row 1, we find there is a small "bump" in the mean regret of the cumulative regret in each plot, and the slight bump occurs at the exploitation step where the occurring time of the bump is greater than h.

Two reasons cause the occurrence of the small bump. One is the coarse estimation of parameters. Another is that the context's angular velocity changes too slowly, violating the uniform sub-optimal minimal condition assumption. Both of these will cause the incorrect ranking estimated by the platform, resulting in regret. A similar pattern can also be found in the second row of the figure. In order to demonstrate the conjecture of violating uniform sub-optimal minimal condition assumption, we decrease it $\overline{\Delta}_{i,\min}$ from 0.2 to 0.1 in S3 (Figure 3.11 in appendix), which indirectly extends the learning step h and therefore increases the estimation accuracy because platform would gather more data to acquire more accurate estimates. In addition, in the third row, the shaded area disappears because the learning

step is long enough to accumulate sufficient data to get a reasonable estimate compared with the first row and the second row, which demonstrate our conjecture.

Another interesting finding is the decreasing regret phenomena after the bump. In the first and second rows of Figure 3.7, the phenomenon of decreasing regret occurs because the agent, p_1, needs to *recover* from the violation of the "uniform sub-optimal minimal condition" assumption. This implies that agent p_1 is unable to distinguish the differences between arms when the uniform sub-optimal minimal condition is violated. In the third row's, the regret decreases over a long period because agent p_1 is in the learning step and the *super-optimal* arms have a much larger gain over *sub-optimal* arms. These significant gains will result in a negative regret. So the cumulative regret will decrease. This interesting phenomenon is only occurring in DMP when considering the contextual information. In all, we find that dynamic matching algorithm is robust to changing the context format.

3.7.2 Real Data

We next apply the dynamic matching algorithm in the job application market with job applicants' profile information and companies' job description information from LinkedIn.

3.7.2.1 Background

We have three job applicants and two companies in the market.

Job Applicants' and Companies' Preferences: Based on the profiles of the job applicants, three candidates with diverse backgrounds are seeking job opportunities in the market:

- a_1: a data scientist (ds),
- a_2: a software development engineer (sde),
- a_3: a quantitative researcher (qr).

In addition, two companies provides two job descriptions indicating that they are interested in hiring candidates with specific skill sets as follows:

- Company p_1 is looking for a candidate with quantitative research skills,
- Company p_2 is looking for a candidate with software development skills.

Given this setup, the preferences of the three job applicants for companies can be described as follows:

- For the data scientist, a_1, the preferences are $\pi_{ds}(a_1) : p_1(\text{qr}) > p_2(\text{sde})$,
- For the software development engineer, a_2, the preferences are $\pi_{sde}(a_2) : p_2(\text{sde}) > p_1(\text{qr})$,
- For the quantitative researcher, a_3, the preferences are $\pi_{qr}(a_3) : p_1(\text{qr}) > p_2(\text{sde})$.

This indicates that each applicant prefers the company whose job description best matches their professional background and skills. Detailed description is provided at Section 3.16.5 of appendix.

Dynamic Contextual Information: To simulate the dynamic contextual information, we take the following steps to construct the dynamic matching environment ($T = 10800$):

- Job applicants' dynamic contextual information:
  - At $t = 0$: the job applicant a_j has textual information $\mathbf{w}_j(0)$, a sequence of words represented as $\mathbf{w}_j(0) = \{w_j^1(0), w_j^2(0), ..., w_j^{q_0}(0)\}$ and q_0 is the length of the sequence of the words at time $t = 0$, profile like `research projects on modeling of high-dimensional and multi-modal (partially observed), inputs for classification, regression and clustering tasks, leveraging a wide range of techniques`.

– At $t = 600z, z \in \mathbb{N}$: we assume that job applicants learn new skills, update profile like `(1)t=600, Strong interested in data science, (2)t=1200, machine learning, (3)t=1800, data visualization...`, and updates his profile, so the textual information becomes $\mathbf{w}_j(t)$, the sequence of words is represented as $\mathbf{w}_j(t) = \{w_j^1(t), w_j^2(t), ..., w_j^{q_t}(t)\}$ for all $t = 600z, z = 1, 2, ..., 18$.[3][4]

- Companies' fixed job descriptions:

  – The job descriptions from companies are fixed texts over time denoted by $\mathbf{w}_i = \{w_i^1, w_i^2, ..., w_i^{p_i}\}$ where p_i is the length of words for company p_i, job descriptions are like `Strong passion in quant finance, strong mathematical and statistical knowledge. Proficiency in programming languages like Python or R, etc.`

The detailed text data is available in Tables 3.1 and 3.2 at Section 3.16.5 of appendix.

Text-to-Embedding: We use the encoder of the Transformer model (DCL18) f to generate the word embedding of these textual information from job applicants' profiles $\{\mathbf{w}_j(t)\}_{j=1,2,3;t\in[T]}$, and companies' job descriptions $\{\mathbf{w}_i\}_{i=1,2}$.

$$h_j(t) = f(\mathbf{w}_j(t)), h_i = f(\mathbf{w}_i), \tag{3.12}$$

where $h_j(t), h_i \in \mathbb{R}^{d_{raw}}$ and $d_{raw} = 768$ is the commonly output dimension of the transformer model (DCL18). For simplicity, here we use PCA method (Pea01; JC16) to extract the most significant dimension from these word embedding vectors for job applicants and add Gaussian noise to $h_j(t)$ at every time step to transform it into streaming data. So the observed contextual information for each job applicant is $x_j(t) = \text{PCA}(h_j(t) + N(0, \zeta^2)) \in \mathbb{R}^d$ where $d = 3, \zeta = 1e - 6$.

[3]One agent update profile every 1800 steps for different updating frequency.

[4]Here we create the streaming data is through adding additional textual information over time.

Figure 3.8: Total regret for agent p_1 and p_2 under noise $\sigma = 0.1$ (Left) and $\sigma = 0.2$ (Right) of methods dynamic matching algorithm, greedy, 0.05-greedy, $1/t$-greedy.

True Response: The true response is determined by the similarity of the job applicant's profile and job description with an added Guassian noise $y_{i,j}(t) = h_j^T h_i(t) + \epsilon_{i,j}(t), \epsilon_{i,j}(t) \sim N(0, \sigma^2), t \in [T]$, $\sigma = 0.1, 0.2$.

Comparison Methods: Here we compare our algorithm with three methods:

- **Greedy method**: This approach constructs the ranking based purely on previously collected data to form estimate $\widehat{\mu}_{i,j}(t)$, without regard for exploration.
- **ϵ-greedy method (where $\epsilon = 0.05$)**: This method usually exploits the ranking list based on previously collected data, but with a probability of ϵ exploration (5% random matching in this case), it will randomly explore other options (randomly permute the ranking list).
- **$1/t$-greedy method with a decaying rate $1/t$**: This technique adjusts the balance between exploring and exploiting by decreasing the exploration rate over time, specifically using a rate that inversely decays with the number of matching t.

3.7.2.2 Results

In Figure 3.8, we demonstrate the social welfare gap—a measure of the absolute difference between the optimal and actual total matching score across all agents—of different methods at various noise levels. The dynamic matching algorithm consistently achieves the minimum social welfare gap under these conditions. The sub-optimality of other comparison methods can be attributed to their failure to utilize dynamic contextual information within the DMP to adaptively design the exploration rate ϵ.

Additionally, we use a vertical dashed line to indicate the transition point of the optimal matching pattern. It is noteworthy that our findings underscore the robustness of our method

in the face of changes in the optimal matching pattern, a characteristic that is absent in the greedy group method. For instance, examining the social welfare gap around the $t = 5,000$ time step reveals a marked increase in the regret pattern associated with the comparison methods.

3.8 Appendix

This appendix is organized as follows. In Section 3.9, we provide the Bernstein concentration lemma and tail probability's upper bound and lower bound for the normal distribution. In Section 3.10, the detail of the DA Algorithm 4 under the job application scenario is provided. In Section 3.11, we prove that if agents can submit valid rankings to the platform, agents will acquire the matching which is as least as good as the stable matching. In Section 3.12, we provide detailed proof of the regret upper bound of dynamic matching algorithm. In Section 3.13, we prove the stable matching holding with high probability. In Section 3.14, the detailed instantaneous regret decomposition at time t is available when we consider there are two agents and three arms in this online matching market. Finally, we provide detailed proof of the instance-dependent regret lower bound in Section 3.15. In Section 3.16, we provide more experimental results of UCB method and dynamic matching algorithm. In addition, various simulation settings' result is presented in Section 3.16.2 and real data related materials are available in Section 3.16.4.

3.9 Miscellaneous Lemmas

Lemma 3.3 (Bernstein Concentration)**.** *Let* $\{D_k, \mathcal{F}_k\}_{k=1}^{\infty}$ *be a martingale difference, and suppose that* D_k *is a* σ*-subgaussian in an adapted sense, i.e., for all* $\alpha \in \mathbb{R}$*.* $\mathbb{E}[e^{\alpha D_k}|\mathcal{F}_{k-1}] \leq$

$e^{\frac{\alpha^2\sigma^2}{2}}$ *almost surely. Then, for all* $t \geq 0$,

$$\mathbb{P}[|\sum_{k=1}^{n} D_k| \geq t] \leq 2e^{-\frac{t^2}{2n\sigma^2}}. \tag{3.13}$$

Lemma 3.3 is from Theorem 2.3 of Wainwright (2019) (Wai19) when $\alpha_* = \alpha_k = 0$ and $\nu_k = \sigma$ for all k.

Lemma 3.4 (Tails of Normal distribution)**.** *Let* $g \sim N(0,1)$*. Then for all* $t > 0$*, we have*

$$(\frac{1}{t} - \frac{1}{t^3})\frac{1}{\sqrt{2\pi}}e^{(-t^2/2)} \leq \mathbb{P}(g \geq t) \leq \frac{1}{t}\frac{1}{\sqrt{2\pi}}e^{(-t^2/2)}. \tag{3.14}$$

Lemma 3.5. *With probability at most* δ*, we have the sample covariance matrix minimum eigenvalue over* $n \geq n_0 = \log(d/\delta)/\tilde{C}_2(\phi_0)$ *i.i.d samples is bounded below by* $\lambda_i/2h + \phi_0^2/2$ *with probability* $1 - \delta$*.*

$$Pr\left[\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}(n)) > \frac{\lambda}{2n} + \frac{\phi_0^2}{2}\right] \geq 1 - \exp\left[-\tilde{C}_2(\phi_0)n + \log(d)\right] \tag{3.15}$$

where $\tilde{C}_2(\phi_0) = \min(1/2, \phi_0^2/8(x_{\max}^2 + \lambda))$*.*

Proof. First, note that

$$\begin{aligned}
\lambda_{\max}\big(\widehat{\Sigma}(\mathbf{X}(n))\big) &= \max_{\|u\|=1} u^T\widehat{\Sigma}(\mathbf{X}(n))u \\
&= \max_{\|u\|=1} \frac{1}{n}\sum_{t\in[n]}(X_t^T u)^2 + \lambda \\
&\leq x_{\max}^2 + \lambda
\end{aligned} \tag{3.16}$$

Algorithm 4: `DA Algorithm`

Input: Companies set $\mathcal{N}$, job applicants set $\mathcal{K}$, companies to job applicants' preferences, job applicants to companies' preferences.
Initialize: An empty set S.
while $\exists$ *A company p who is not matched and has not contacted to every job applicant* **do**
    Let a be the highest ranking job applicant in company p's preference, to whom company p has not yet contacted.
    Now company p contacts the job applicant a.
    if *Job applicant a is free* **then**
        (p, a) become matched (add (p, a) to S).
    else
        Job applicant a is matched to company p' (add (p', a) to S).
        if *Job applicant a prefers company p' to company p* **then**
            Company p remains free (remove (p, a) from S).
        else
            Job applicant a prefers company p to company p'.
            Company p' becomes free (remove (p', a) from S).
            (p, a) are paired (add (p, a) to S).
Output: Matching result S.

Then, it follows from the matrix Chernoff inequality, Corollary 5.2 in (Tro15), that

$$\begin{aligned} \Pr\left[\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}(n)) > \frac{\lambda}{2n} + \frac{\phi_0^2}{2}\right] &\geq 1 - d\exp\left[-\frac{n\phi_0^2}{8(x_{\max}^2 + \lambda)}\right] \\ &\geq 1 - d\exp\left[-\tilde{C}_2(\phi_0)n\right], \end{aligned} \tag{3.17}$$

if we take $\tilde{\delta} = 1/2$ and $R = x_{\max}^2 + \lambda$. □

3.10 Deferred Acceptance (DA) Algorithm

In algorithm 4, we present the DA algorithm in the example of job seeking scenario.

3.11 Proof of Lemma 3.1

Lemma 3.1 states that if all agents have valid rankings to the platform, the DA-Algorithm will provide a matching m_t as least as good as $\overline{m}_t$.

Proof. First, we show that the agent-optimal matching $\overline{m}(t)$ is *stable* according to the rankings submitted by agents when all those rankings are valid.

Let a_j be an arm such that $\widehat{r}_{i,j}(t) < \widehat{r}_{i,\overline{m}_t(i)}(t)$ for agent p_i. Since $\widehat{r}_{i,[K]}(t)$ is a valid ranking, which means that p_i prefers a_j over $\overline{m}_t(i)$ according to the true preference. However, since $\overline{m}(t)$ is *stable* according to the true preference, arm a_j must prefer agent $\overline{m}_j(t)^{-1}$ over p_i because arm a_j has no incentive to deviate the current matching $\overline{m}(t)$, where $\overline{m}_j(t)^{-1}$ is a_j's matching object according to the agent-optimal $\overline{m}(t)$ or the empty set if a_j does not have a match. Therefore, according to the ranking $\widehat{r}_{i,[K]}(t)$, p_i has no incentive to deviate to arm a_j because that arm a_j would reject him.

Since $\overline{m}(t)$ is a stable matching according to the valid ranking $\widehat{r}_{i,[K]}(t)$, we know that the DA-algorithm will output a matching which is at least as good as $\overline{m}(t)$ for all agents according to rankings $\widehat{r}_{i,[K]}(t)$ since this $\widehat{r}_{i,[K]}(t)$ ranking is an agent-optimal ranking if it were the true ranking. Since all rankings are valid rankings, it follows that the DA algorithm will output a matching $m(t)$ which is as least as good as $\overline{m}(t)$.

3.12 Proof of Theorem 3.1 - Regret Upper Bound

3.12.1 Proof of Lemma 3.2

Proof. We consider one time step t at the exploitation step throughout this proof. We first show how to quantify the invalid ranking probability.

If the ranking $\widehat{r}_{i,[K]}(t)$ is *invalid*, there must exist an arm a_j where $j \neq \overline{m}_t(i)$ such that $\mu_{i,\overline{m}_t(i)}(t) > \mu_{i,j}(t)$, but $\widehat{r}_{i,j}(t) < \widehat{r}_{i,\overline{m}_t(i)}(t)$, due to the inaccurate estimation of the true

parameter, which is equivalent to $\widehat{\mu}_{i,j}(t) > \widehat{\mu}_{i,\overline{m}_t(i)}(t)$. So we have

$$
\begin{aligned}
&\mathbb{P}(\widehat{\mu}_{i,j}(t) > \widehat{\mu}_{i,\overline{m}_t(i)}(t)) \\
&= \mathbb{P}\Big[\widehat{\mu}_{i,j}(t) - \mu_{i,j}(t) - \widehat{\mu}_{i,\overline{m}_t(i)}(t) + \mu_{i,\overline{m}_t(i)}(t) \geq \mu_{i,\overline{m}_t(i)}(t) - \mu_{i,j}(t)\Big] \\
&= \mathbb{P}\Big[\widehat{\theta}_i(h)^T x_j(t) - \theta_{i,*}^T x_j(t) - \widehat{\theta}_i(h)^T x_{\overline{m}_t(i)}(t) + \theta_{i,*}^T x_{\overline{m}_t(i)}(t) \geq \mu_{i,\overline{m}_t(i)}(t) - \mu_{i,j}(t)\Big] \\
&= \mathbb{P}\Big[(\widehat{\theta}_i(h) - \theta_{i,*})^T (x_j(t) - x_{\overline{m}_t(i)}(t)) \geq \theta_{i,*}^T (x_{\overline{m}_t(i)}(t) - x_j(t))\Big] \\
&\leq \mathbb{P}\Big[\Big\|\widehat{\theta}_i(h) - \theta_{i,*}\Big\|_2 \Big\|x_j(t) - x_{\overline{m}_t(i)}(t)\Big\|_2 \geq \theta_{i,*}^T (x_{\overline{m}_t(i)}(t) - x_j(t))\Big] \\
&= \mathbb{P}\Big[\Big\|\widehat{\theta}_i(h) - \theta_{i,*}\Big\|_2 \geq \langle \theta_{i,*}, \frac{x_{\overline{m}_t(i)}(t) - x_j(t)}{\Big\|x_{\overline{m}_t(i)}(t) - x_j(t)\Big\|_2} \rangle\Big],
\end{aligned}
\tag{3.18}
$$

where in the inequality, we use the Cauchy inequality to upper bound the left inner product. Here we find an interesting term called, *similarity difference* (SD), which is

$$
\begin{aligned}
\text{SD} &\triangleq \langle \theta_{i,*}, \frac{x_{\overline{m}_t(i)}(t) - x_j(t)}{\Big\|x_{\overline{m}_t(i)}(t) - x_j(t)\Big\|_2} \rangle \\
&= \|\theta_{i,*}\|_2 \, \langle \frac{\theta_{i,*}}{\|\theta_{i,*}\|_2}, \frac{x_{\overline{m}_t(i)}(t) - x_j(t)}{\Big\|x_{\overline{m}_t(i)}(t) - x_j(t)\Big\|_2} \rangle \\
&= \|\theta_{i,*}\|_2 \cos(\phi_{i,j}(t)),
\end{aligned}
\tag{3.19}
$$

where $\phi_{i,j}(t)$ represents the *angle* between the normalized true parameter $\theta_{i,*}$ and the normalized arms difference at time step t, which is the similarity difference between arm a_j and arm $a_{\overline{m}_t(i)}$ from the viewpoint of agent p_i.

Here we discuss the boundary scenario of the similarity difference. If SD $= 0$, there are three possible reasons.

1. The first possible reason is that if the true parameter $\theta_{i,*} = 0$. Since we assume all agents' true parameters are meaningful and positive, with Assumption 3.5, we can rule out this case.

2. The second possible reason is that if arm a_j and arm $a_{\overline{m}_t(i)}$ are identical such that

$x_j(t) = x_{\overline{m}_t(i)}$. Since we assume all arms are different, we can also rule out this case.

3. The third possible reason is that if $\cos(\phi_{i,j}(t)) = 0$. That means from the view point of agent p_i at time t, arm a_j and arm $a_{\overline{m}_t(i)}$ are symmetric. Since we assume there are no ties in ranking over time, we can also rule out this scenario.

The last case we also discussed in Assumption 3.4 where we assume that the uniform sub-optimal minimal condition over time is greater than zero. That means there is no symmetric case for the agent to distinguish two arms between the agent-optimal and the sub-optimal arm—*That is the key difference between the DMP and the MAB competing bandit problem. The MAB competing bandit only has one constant gap over time and no existence of the interesting symmetric arms.* We now restate the *uniform sub-optimal minimal condition* $\overline{\Delta}_{i,\min}$ for agent p_i over time t, that is $\overline{\Delta}_{i,\min} = \min_{j\in[K],t\in[T]} \|\theta_{i,*}\|_2 \cos(\phi_{i,j}(t)) > 0$.

With Assumption 3.4, we consider the estimation error of the true parameter is lower bounded by the *uniform sub-optimal minimal condition* $\overline{\Delta}_{i,\min}$. So the probability of the invalid ranking is upper bounded by

$$\mathbb{P}\left[\left\|\widehat{\theta}_i(h) - \theta_{i,*}\right\|_2 \geq \|\theta_{i,*}\|_2 \cos(\phi_{i,j}(t))\right] \leq \mathbb{P}\left[\left\|\widehat{\theta}_i(h) - \theta_{i,*}\right\|_2 \geq \overline{\Delta}_{i,\min}\right]. \tag{3.20}$$

To get the upper bound of this tail event's probability, we use the technique of quantifying the confidence ellipsoid from (LWC21). Notation $\widehat{\Sigma}(\mathbf{X}_i(t))$ represents the normalized covariance matrix, so $\widehat{\Sigma}(\mathbf{X}_i(t)) = \Phi_i(t)/t = (\mathbf{X}_i(t)^T\mathbf{X}_i(t) + \lambda_i\mathbf{I}_d)/t$ for $t \geq 1$, where we define $\Phi_i(0) = \lambda_i\mathbf{I}_d$ and λ_i is the prespecified penalty hyperparameter for agent p_i. Note that the event $\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}_i(t))) \geq \phi_0^2/2 + \lambda_i/2t$ holds for $t \geq 1$ with having that $\lambda_{\min}\big(\mathbf{X}_i(t)^T\mathbf{X}_i(t)\big)/t \geq \phi_0^2/2$, based on the high probability in exponential decay wrt t (see Corollary 5.2 in (Tro15) and Lemma 3.5. Thus after the learning step, agents have already gathered length h historical data, which include actions, rewards and contexts. For notation simplicity, we use $\widehat{\theta}_i$ to

replace $\widehat{\theta}_i(h)$ and $\mathbf{X}_i$ to replace $\mathbf{X}_i(h)$. So we have

$$\begin{aligned}
&\left\|\widehat{\theta}_i - \theta_{i,*}\right\|_2 \\
&= \left\|(\mathbf{X}_i^T\mathbf{X}_i + \lambda_i\mathbf{I})^{-1}\mathbf{X}_i^T(\mathbf{X}_i\theta_{i,*} + \epsilon) - \theta_{i,*}\right\|_2 \\
&= \left\|(\mathbf{X}_i^T\mathbf{X}_i + \lambda_i\mathbf{I})^{-1}\mathbf{X}_i^T\epsilon + \theta_{i,*} - \lambda_i(\mathbf{X}_i^T\mathbf{X}_i + \lambda_i\mathbf{I})^{-1}\theta_{i,*} - \theta_{i,*}\right\|_2 \\
&= \left\|(\mathbf{X}_i^T\mathbf{X}_i + \lambda_i\mathbf{I})^{-1}(\mathbf{X}_i^T\epsilon - \lambda_i\theta_{i,*})\right\|_2 \\
&\leq \frac{1}{\lambda_i + h\phi_0^2}\left\|\mathbf{X}_i^T\epsilon - \lambda_i\theta_{i,*}\right\|_2.
\end{aligned} \tag{3.21}$$

Here we use a constant $\chi > 0$ to get the estimation error. So we have

$$\begin{aligned}
&\Pr\left[\left\|\widehat{\theta}_i - \theta_{i,*}\right\|_2 \leq \chi\right] \\
&\geq \Pr\left[\left(\left\|\mathbf{X}_i^T\epsilon - \lambda_i\theta_{i,*}\right\|_2 \leq 2\chi(\lambda_i + h\phi_0^2)\right) \cap \left(\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}_i) > \frac{\lambda_i}{2h} + \frac{\phi_0^2}{2}\right)\right] \\
&\geq 1 - \sum_{r=1}^{d} \underbrace{\Pr\left[\epsilon^T\mathbf{X}_i^{(r)} > \lambda_i\theta_{i,*}^{(r)} + \frac{2\chi(\lambda_i + h\phi_0^2)}{\sqrt{d}}\right]}_{\text{Part I}} - \underbrace{\Pr\left[\epsilon^T\mathbf{X}_i^{(r)} < \lambda_i\theta_{i,*}^{(r)} - \frac{2\chi(\lambda_i + h\phi_0^2)}{\sqrt{d}}\right]}_{\text{Part II}} \\
&\qquad - \Pr\left[\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}_i) \leq \frac{\lambda_i}{2h} + \frac{\phi_0^2}{2}\right]
\end{aligned} \tag{3.22}$$

where we let $\mathbf{X}_i^{(r)}(t)$ denote the r^{th} column of $\mathbf{X}_i(t)$. To make $\Pr[\epsilon^T\mathbf{X}_i^{(r)} < \lambda_i\theta_{i,*}^{(r)} - \frac{2\chi(\lambda_i+h\phi_0^2)}{\sqrt{d}}]$ have a relative small probability, based on the Assumption 3.5 that $\theta_{i,*}^{(r)}$ is positive and let $\lambda_i < \frac{2\chi(\lambda_i+h\phi_0^2)}{\sqrt{d}\theta_{i,*}^{(r)}}$, with the analysis from Case B.2.3 from (LWC21), the part II's probability will be small. So when $\lambda_i < \frac{2\chi(\lambda_i+h\phi_0^2)}{\sqrt{d}\theta_{i,*}^{(r)}}$ is small, part I and part II's probability will be similar[5]. So the previous probability lower bound will be

$$\begin{aligned}
&\Pr\left[\left\|\widehat{\theta}_i - \theta_{i,*}\right\|_2 \leq \chi\right] \\
&\geq 1 - \sum_{r=1}^{d} 2\Pr\left[\epsilon^T\mathbf{X}_i^{(r)} > \lambda_i\theta_{i,*}^{(r)} + \frac{2\chi(\lambda_i + h\phi_0^2)}{\sqrt{d}}\right] - \Pr\left[\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}_i) \leq \frac{\lambda_i}{2h} + \frac{\phi_0^2}{2}\right]
\end{aligned} \tag{3.23}$$

[5]Or we can follow (LWC21)'s analysis for part I and part II separately. However, based on the Assumption 3.5, the probability difference is minor.

We can expand $\epsilon^T \mathbf{X}_i^{(r)}(t) = \sum_{j\in[t]} \epsilon(j) x_{i,j}^{(r)}$, where we note that $D_{i,j,r} \equiv \epsilon(k) x_{i,j}^{(r)}$ is a $x_{i,\max}\sigma$-subgaussian random variable, where $x_{i,\max} = \|\mathbf{X}_i(t)\|_\infty$, conditioned on the sigma algebra $\mathcal{F}_{j-1}$ that is generated by random variable $X_1, ..., X_{j-1}, Y_1, ..., Y_{j-1}$. Defining $D_{i,0,r} = 0$, the sequence $D_{i,0,r}, D_{i,1,r}, ..., D_{(i,j,r)}$ is a martingale difference sequence adapted to the filtration $\mathcal{F}_1 \subset \mathcal{F}_2 \subset ...\mathcal{F}_j$, since $E[\epsilon(j)x_j^{(r)}|\mathcal{F}_{j-1}] = 0$. Using the Bernstein concentration inequality from Lemma 3.3,

$$
\begin{aligned}
&\Pr\left[\left\|\widehat{\theta}_i - \theta_{i,*}\right\|_2 \le \chi\right] \\
&\ge 1 - \sum_{r=1}^{d} 2\Pr\left[\epsilon^T \mathbf{X}_i^{(r)} > \lambda_i \theta_{i,*}^{(r)} + \frac{2\chi(\lambda_i + h\phi_0^2)}{\sqrt{d}}\right] - \Pr\left[\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}_i) \le \frac{\lambda_i}{2h} + \frac{\phi_0^2}{2}\right] \\
&\ge 1 - \sum_{r=1}^{d} 2\Pr\left[\epsilon^T \mathbf{X}_i^{(r)} > \frac{2h\chi\lambda_i\phi_1^2(h,\lambda_i)}{\sqrt{d}}\right] - \Pr\left[\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}_i) \le \frac{\lambda_i}{2h} + \frac{\phi_0^2}{2}\right] \\
&\ge 1 - 2d\exp\left[-\frac{2h\chi^2\lambda_i^2\phi_1^4}{d\left\|\mathbf{X}_i\right\|_\infty^2 \sigma^2}\right] - \Pr\left[\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}_i) \le \frac{\lambda_i}{2h} + \frac{\phi_0^2}{2}\right],
\end{aligned}
\tag{3.24}
$$

where we denote $\phi_1^2 := \phi_1^2(h,\lambda_i) = (\lambda_i + h\phi_0^2)/(h\lambda_i) = 1/h + \phi_0^2/\lambda_i$. So we have the probability upper bound for the estimation error for any constant $\chi > 0$,

$$
\begin{aligned}
&\Pr\left[\left\|\widehat{\theta}_i - \theta_{i,*}\right\|_2 \ge \chi\right] \\
&\le 2d\exp\left[-\frac{2h\chi^2\lambda_i^2\phi_1^4}{d\left\|\mathbf{X}_i\right\|_\infty^2 \sigma^2}\right] - \Pr\left[\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}_i) \le \frac{\lambda_i}{2h} + \frac{\phi_0^2}{2}\right].
\end{aligned}
\tag{3.25}
$$

Now we replace χ with $\overline{\Delta}_{i,\min}$, and we have $x_{i,\max} \le 1$ by Assumption 3.2. So we get the following upper bound of the invalid ranking probability,

$$
\begin{aligned}
&\mathbb{P}\Big[\left\|\widehat{\theta}_i(h) - \theta_{i,*}\right\|_2 \ge \overline{\Delta}_{i,\min}\Big] \\
&\le \exp\left[-h\frac{2\lambda_i^2\rho^2\phi_1^4}{d^2 x_{i,\max}^2\sigma^2} + \log(2d)\right] - \mathbb{P}\Big[\lambda_{\min}(\widehat{\Sigma}(\mathbf{X}_i(h)) \le \frac{\lambda_i}{2h} + \frac{\phi_0^2}{2}\Big] \\
&\lesssim \exp\left[-h\frac{2\lambda_i^2\rho^2\phi_1^4}{d^2\sigma^2} + \log(2d)\right].
\end{aligned}
\tag{3.26}
$$

So the invalid ranking's probability created by agent p_i at time t is upper bounded by

$$\mathbb{P}(\widehat{\mu}_{i,j}(t) > \widehat{\mu}_{i,\overline{m}_t(i)}(t)) \lesssim 2d \exp\left[- h\frac{2\lambda_i^2\rho^2\phi_1^4}{d^2\sigma^2}\right], \tag{3.27}$$

and because we consider all such sub-optimal arms a_j, we have the following upper bound of the invalid ranking probability,

$$\mathbb{P}(\widehat{r}_{i,[K]}(t) \text{ is invalid}) \leq 2d(K - \tau_i(t)) \exp\left[- h\frac{2\lambda_i^2\rho^2\phi_1^4}{d^2\sigma^2}\right], \tag{3.28}$$

where we use $\tau_i(t)$ to represent the agent p_i's optimal ranking position when matched with $\overline{m}_t(i)$.

With Lemma 3.2, we can quantify the regret at $t > h$. So the instantaneous regret for agent p_i at time t will be upper bounded by

$$\begin{aligned} R_{i,t} &\triangleq \overline{\Delta}_{i,j}(t)\mathbb{P}(\text{at least one } \widehat{r}_{i,[K]}(t) \text{ is invalid}, \forall i \in [N]) \\ &\leq N(K - \min_{i\in[N]} \tau_i(t))\overline{\Delta}_{i,\max}(t)\mathbb{P}(\widehat{r}_{i,[K]}(t) \text{ is invalid}). \end{aligned} \tag{3.29}$$

Then we add the part I regret and part II regret together and get the regret upper bound of dynamic matching algorithm.

$$R_i(n) \leq \sum_{t=1}^{h} \overline{\Delta}_{i,j}(t) + 2Nd\left[\sum_{t=h+1}^{T} \overline{\Delta}_{i,\max}(t)(K - \min_{i\in[N]} \tau_i(t))\right] \exp\left[- \frac{2\lambda_i^2\rho^2\phi_1^4}{d^2\sigma^2}h\right]. \tag{3.30}$$

By $\phi_1^4 = (1/h + \phi_0^2/\lambda_i)^2 = \frac{1}{h^2} + \frac{2\phi_0^2}{\lambda_i h} + \frac{\phi_0^4}{\lambda_i^2}$, we have $\phi_1^4 h = \frac{1}{h} + \frac{2\phi_0^2}{\lambda_i} + h\frac{\phi_0^4}{\lambda_i^2} \geq \frac{2\phi_0^2}{\lambda_i} + h\frac{\phi_0^4}{\lambda_i^2}$,

the regret upper bound is

$$
\begin{aligned}
R_i(n) &\leq \sum_{t=1}^{h} \overline{\Delta}_{i,j}(t) + 2Nd\left[\sum_{t=h+1}^{T} \overline{\Delta}_{i,\max}(t)(K - \min_{i\in[N]} \tau_i(t))\right] \exp\left[-\frac{2\lambda_i^2\rho^2\phi_1^4}{d^2\sigma^2}h\right] \\
&< \sum_{t=1}^{h} \overline{\Delta}_{i,j}(t) + 2Nd\left[\sum_{t=h+1}^{T} \overline{\Delta}_{i,\max}(t)(K - \min_{i\in[N]} \tau_i(t))\right] \exp\left[-\frac{2\lambda_i^2\rho^2}{d^2\sigma^2}(\frac{2\phi_0^2}{\lambda_i} + h\frac{\phi_0^4}{\lambda_i^2})\right] \\
&= \sum_{t=1}^{h} \overline{\Delta}_{i,j}(t) + 2C_0(\lambda_i)Nd\left[\sum_{t=h+1}^{T} \overline{\Delta}_{i,\max}(t)(K - \min_{i\in[N]} \tau_i(t))\right] \exp\left[-\frac{2\phi_0^4\rho^2}{d^2\sigma^2}h\right]
\end{aligned} \tag{3.31}
$$

where $C_0(\lambda_i) = \exp\left[-\frac{4\lambda_i\phi_0^2\rho^2}{d^2\sigma^2}\right]$.

3.12.2 Proof of Corollary 3.1

Proof. Moreover, in order to analyze the order of the regret upper bound, we optimize the the exploration horizon,

$$
\begin{aligned}
R_i(n) &\leq \sum_{t=1}^{h} \overline{\Delta}_{i,j}(t) + 2C_0(\lambda_i)Nd\left[\sum_{t=1}^{T} \overline{\Delta}_{i,\max}(t)(K - \min_{i\in[N]} \tau_i(t))\right] \exp\left[-\frac{2\phi_0^4\rho^2}{d^2\sigma^2}h\right] \\
&\leq h\overline{\Delta}_{i,\max} + 2C_0(\lambda_i)Nd\left[\sum_{t=1}^{T} \overline{\Delta}_{i,\max}(t)(K - \min_{i\in[N]} \tau_i(t))\right] \exp\left[-\frac{2\phi_0^4\rho^2}{d^2\sigma^2}h\right] \\
&\leq h\overline{\Delta}_{i,\max} + 2C_0(\lambda_i)NKdT\overline{\Delta}_{i,\max} \exp\left[-\frac{2\phi_0^4\rho^2}{d^2\sigma^2}h\right],
\end{aligned} \tag{3.32}
$$

where we know that $\overline{\Delta}_{i,j}(t) \leq \overline{\Delta}_{i,\max}(t) \leq \overline{\Delta}_{i,\max}, \forall t \in [T]$ and $K - \tau_i(t) < K$. Taking the derivative on the RHS of Eq. (3.32) with respect to h to obtain the optimal h,

$$
\overline{\Delta}_{i,\max} + 2C_0(\lambda_i)NKdT\overline{\Delta}_{i,\max} \exp\left[-\frac{2\phi_0^4\rho^2}{d^2\sigma^2}h\right] \times (-\frac{2\phi_0^4\rho^2}{d^2\sigma^2}) = 0, \tag{3.33}
$$

and get

$$
h = \frac{d^2\sigma^2}{2\phi_0^4\rho^2} \log \frac{4C_0(\lambda_i)TNK\phi_0^4\rho^2}{d\sigma^2\overline{\Delta}_{i,\max}}, \tag{3.34}
$$

when we set the optimal learning step to $h \leftarrow \lceil h \rceil$, we can achieve the minimum regret,

$$\begin{aligned} R_i(T) &\leq \max\left\{h\overline{\Delta}_{i,\max}, \frac{d^2\sigma^2\overline{\Delta}_{i,\max}}{2\phi_0^4\rho^2}\log\frac{4C_0(\lambda_i)NK\phi_0^4\rho^2}{d\sigma^2\overline{\Delta}_{i,\max}}T\right\} + \frac{d^2\sigma^2\overline{\Delta}_{i,\max}}{2\phi_0^4\rho^2} \\ &= C_1(d,\sigma,\overline{\Delta}_{i,\min},\overline{\Delta}_{i,\max},\lambda_i,x_{i,\max})\log\left[C_2(N,K,d,\sigma,\overline{\Delta}_{i,\min},\overline{\Delta}_{i,\max},\lambda_i,x_{i,\max})\times T\right] \\ &\quad + C_1(d,\sigma,\overline{\Delta}_{i,\min},\overline{\Delta}_{i,\max},\lambda_i,x_{i,\max}) \\ &= \tilde{\mathcal{O}}\left(\frac{d^2\sigma^2}{\rho^2}\log(NKT)\right) \end{aligned} \tag{3.35}$$

where the constants C_1, C_2 are given by

$$C_1 = \frac{d^2\sigma^2\overline{\Delta}_{i,\max}}{2\phi_0^4\rho^2}, \qquad C_2 = \frac{4C_0(\lambda_i)NK\phi_0^4\rho^2}{d\sigma^2\overline{\Delta}_{i,\max}}. \tag{3.36}$$

3.13 Proof of Theorem 3.2 - Stable Matching

Proof. Based on Lemmas 3.1 and 3.2, as long as all agents have valid rankings, then the matching solution is stable. In order to have the $\mathbb{P}(\text{matching solution is stable}) \geq \Psi$, we have

$$\begin{aligned} \mathbb{P}(\text{Matching solution is stable}) &= \mathbb{P}(\text{all agents have valid rankings}) \\ &\geq \prod_{i=1}^{N}\left[1 - 2d(K-\tau_i(t))\exp\left(-t\frac{2\lambda_i^2\rho^2\phi_1^4}{d^2\sigma^2}\right)\right] \\ &\geq \left[1 - 2d(K-1)\exp\left(-t\frac{2\lambda_{\min}^2\rho^2\phi_1^4}{d^2\sigma^2}\right)\right]^N. \end{aligned} \tag{3.37}$$

Thus, given $t \geq \lceil \frac{d^2\sigma^2}{2\lambda_{\min}^2\rho^2\phi_1^2}[\log(2d(K-1)) - \log(1-\Psi^{1/N})]\rceil$ based on Corollary 3.1, we have the matching solution provided by dynamic matching algorithm is stable with probability at least Ψ.

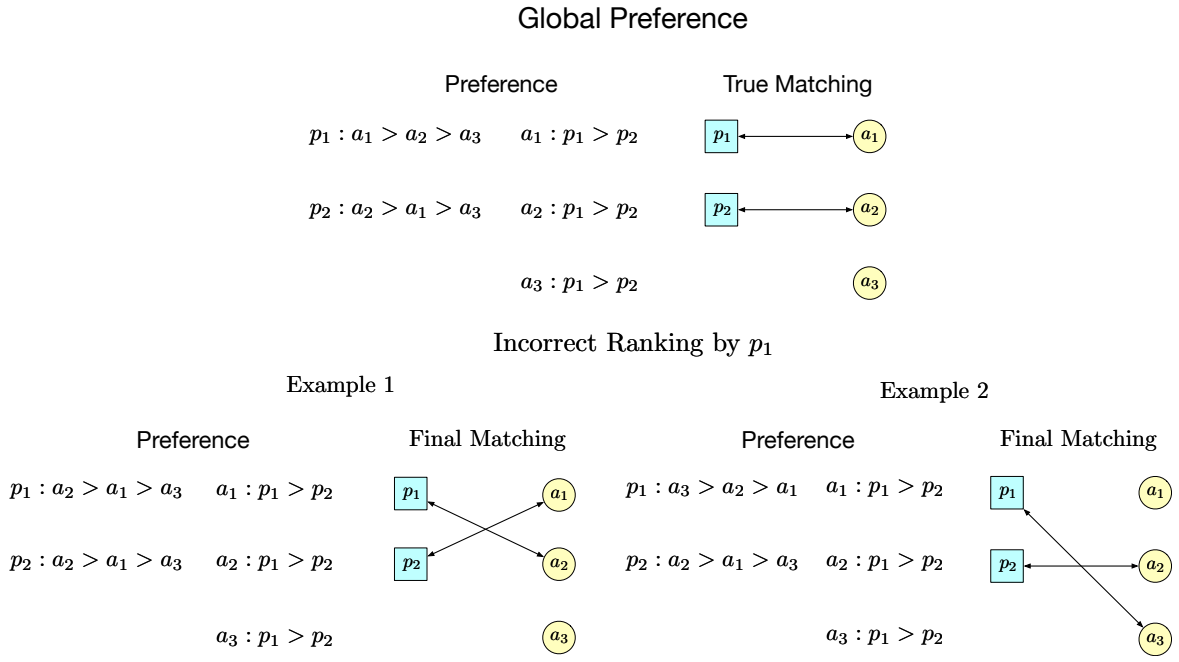

Figure 3.9: Examples of the matching result caused by the incorrect ranking provided by agent p_1 when agent p_2 submits the correct ranking list under the global preference. In Example 1, Agent p_1 provides an incorrect ranking $p_1 : a_2 > a_1 > a_3$. The final matching result is $\{(p_1, a_2), (p_2, a_2)\}$. It creates a positive regret for both agents. In Example 2: Agent p_1 provides an incorrect ranking $p_1 : a_3 > a_2 > a_1$. The final matching result is $\{(p_1, a_3), (p_2, a_2)\}$. It creates a positive regret for p_1 and no regret for p_2.

3.14 Detailed Regret Analysis for Two Agents and Three Arms

The expected instantaneous regret $R_{1,t}(\widehat{r}_2(t) = r_2(t)) = \mathbb{P}(\mathcal{G}_2(t)) \sum_{z=1}^{6} \mathbb{P}_t^{C_z} R_{1,t}^{C_z}(\widehat{r}_2(t) = r_2(t))$, where $\mathcal{G}_2(t)$ is the correct ranking. $\mathbb{P}_t^{C_z}$ is the probability of occurring matching case z at time t. $R_{1,t}^{C_z}(\widehat{r}_2(t) = r_2(t))$ is the conditional instantaneous regret of occurring matching case z at time t if p_2 submits correct ranking list. Meanwhile $\sum_{z=1}^{6} \mathbb{P}_t^{C_z} R_{1,t}^{C_z}(\widehat{r}_2(t) = r_2(t))$ represents the expected regret for p_1 when p_2 submits the correct ranking list. We find that there are six cases in total if p_2 submits the correct ranking list shown in Figures 3.9 and 3.10.

After collecting all probabilities' lower bounds, we can compute the instantaneous regret for agent p_1 at time t. Then we can sum all instantaneous regret to get the regret lower

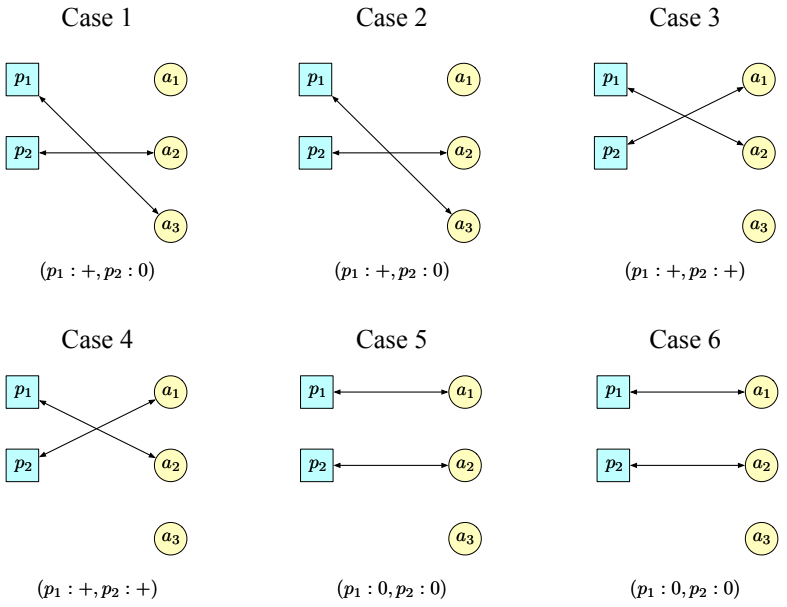

Figure 3.10: The corresponding matching results and regret status in six cases when agent p_1 submits an incorrect ranking. *Single agent suffers regret*: Case 1 and Case 2. *Both agents suffer regret*: Case 3 and Case 4. *No regret*: Case 5 and Case 6.

bound.

Due to the *incorrect raking* from agent p_1, it creates six cases in total. In the following passage, we will analyze them case by case.

Case 1. If agent p_1 wrongly estimates the ranking over arms as $p_1 : a_3 > a_1 > a_2$, the matching result by DA Algorithm is shown in Figure 3.5 Case 1. Agent p_1 is matched with a_3 and agent p_2 is matched with a_2. In this case p_1 suffers a positive regret. The instantaneous regret can be decomposed as

$$R_{1,t}^{C_1} \triangleq \theta_{1,*}^T x_1(t) - \theta_{1,*}^T x_3(t) = \theta_{1,*}^T (x_1(t) - x_3(t)), \tag{3.38}$$

where we define $R_{1,t}^{C_1}$ is the case 1 instantaneous regret for agent p_1 at time t. Here C_1 represents the case 1, "1" in the subscript represents agent p_1, and t in the subscript represents the time step. Similar definitions are used in the following analysis. In addition, agent p_2 does not suffer regret in case 1.

This incorrect ranking's joint probability for agent p_1 is the product of two ranking probabilities $\mathbb{P}_t^{C_1} \triangleq \mathbb{P}_1\big(\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t)\big)$ from agent p_1 and $\mathbb{P}(\mathcal{G}_2(t)) \triangleq \mathbb{P}_2\big(\widehat{\mu}_{2,2}(t) > \widehat{\mu}_{2,1}(t) > \widehat{\mu}_{2,3}(t)\big)$ from agent p_2. Here we define $\mathbb{P}_t^{C_1}$ as the probability of occurring case 1 of agent p_1 and $\mathcal{G}_2(t)$ represents that agent p_2 submits correct ranking list to the centralized platform and we call this as the correct ranking in the following analysis, which is equivalent to agent submitting the correct ranking list to the platform. And the bad event is equivalent to agent submitting the incorrect rankings. So this $\{\widehat{\mu}_{2,2}(t) > \widehat{\mu}_{2,1}(t) > \widehat{\mu}_{2,3}(t)\}$ is a *good* event because agent p_2 correctly estimate its preference scheme over arms. $\{\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t)\}$ is a *bad* event because agent p_1 wrongly estimate its preference scheme over arms. The decomposed instantaneous regret for agent p_1 is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_1} R_{1,t}^{C_1} \\ &= \mathbb{P}\big(\widehat{\mu}_{1,3} > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t) > \widehat{\mu}_{2,1}(t) > \widehat{\mu}_{2,3}(t)\big)\theta_{1,*}^T(x_1(t) - x_3(t)),\end{aligned} \tag{3.39}$$

where the above instantaneous regret is greater than zero. For agent p_2, the decomposed instantaneous regret is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_1} R_{2,t}^{C_1} \\ &= \mathbb{P}\big(\widehat{\mu}_{1,3} > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t) > \widehat{\mu}_{2,1}(t) \geq \widehat{\mu}_{2,3}\big)\theta_{2,*}^T(x_2(t) - x_2(t)) = 0.\end{aligned} \tag{3.40}$$

Case 2. If agent p_1 wrongly estimates the ranking over arms as $p_1 : a_3 > a_2 > a_1$. The matching result by DA Algorithm is in Figure 3.5 Case 2. Agent p_1 is matched with arm a_3 and agent p_2 is matched with arm a_2, where agent p_1 suffers a positive regret. The instantaneous regret is

$$R_{1,t}^{C_2} = \theta_{1,*}^T x_1(t) - \theta_{1,*}^T x_3(t) = \theta_{1,*}^T(x_1(t) - x_3(t)). \tag{3.41}$$

In addition, agent p_2 does not suffer regret in case 2.

This bad event's joint probability is the product of two ranking probabilities $\mathbb{P}_t^{C_2} =$

$\mathbb{P}\big(\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,1}(t)\big)$ by agent p_1 and $\mathbb{P}(\mathcal{G}_2(t))$ by agent p_2. $\{\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,1}(t)$ is the *bad* event that agent p_1 wrongly estimate its preference scheme over arms. The decomposed instantaneous regret for agent p_1 is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_2} R_{1,t}^{C_2} \\ &= \mathbb{P}\big(\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,1}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t) > \widehat{\mu}_{2,1}(t) > \widehat{\mu}_{2,3}(t)\big)\theta_{1,*}^T(x_1(t) - x_3(t)) > 0.\end{aligned} \tag{3.42}$$

For agent p_2, the decomposed instantaneous regret is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_2} R_{2,t}^{C_2} \\ &= \mathbb{P}\big(\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,1}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t) > \widehat{\mu}_{2,1}(t) > \widehat{\mu}_{2,3}(t)\big)\theta_{2,*}^T(x_2(t) - x_2(t)) = 0.\end{aligned} \tag{3.43}$$

This case is the same as the case 1.

Case 3. If agent p_1 wrongly estimates the ranking over arms as $p_1 : a_2 > a_3 > a_1$. The matching result by DA Algorithm is in Figure 3.5 Case 3. Agent p_1 is matched with arm a_2 and agent p_2 is matched with arm a_1. The decomposed instantaneous regret for agent p_1 is

$$R_{1,t}^{C_3} = \theta_{1,*}^T x_1(t) - \theta_{1,*}^T x_2(t) = \theta_{1,*}^T(x_1(t) - x_2(t)) > 0. \tag{3.44}$$

In addition, agent p_2 suffers a regret. The decomposed instantaneous regret for agent p_2 is

$$R_{2,t}^{C_3} = \theta_{2,*}^T x_2(t) - \theta_{2,*}^T x_1(t) = \theta_{2,*}^T(x_2(t) - x_1(t)) > 0. \tag{3.45}$$

This bad event's joint probability is the product of two ranking probabilities $\mathbb{P}_t^{C_3} = \mathbb{P}\big(\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t)\big)$ by agent p_1 and $\mathbb{P}(\mathcal{G}_2(t))$ by agent p_2. $\{\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t)\}$ is the *bad* event that agent p_1 wrongly estimate its preference scheme over arms.

The decomposed instantaneous regret for agent p_1 is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_3}R_{1,t}^{C_3}\\&=\mathbb{P}\big(\widehat{\mu}_{1,2}(t)>\widehat{\mu}_{1,3}(t)>\widehat{\mu}_{1,1}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t)>\widehat{\mu}_{2,1}(t)>\widehat{\mu}_{2,3}(t)\big)\theta_{1,*}^T(x_1(t)-x_2(t))>0.\end{aligned}\tag{3.46}$$

For agent p_2, the decomposed instantaneous regret is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_3}R_{2,t}^{C_3}\\&=\mathbb{P}\big(\widehat{\mu}_{1,3}(t)>\widehat{\mu}_{1,1}>\widehat{\mu}_{1,2}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t)>\widehat{\mu}_{2,1}(t)>\widehat{\mu}_{2,3}(t)\big)\theta_{2,*}^T(x_2(t)-x_1(t))>0.\end{aligned}\tag{3.47}$$

Case 4. If agent p_1 wrongly estimates the ranking over arms as $p_1 : a_2 > a_1 > a_3$. The matching result by DA Algorithm is in Figure 3.5 Case 4. Agent p_1 is matched with arm a_2 and agent p_2 is matched with arm a_1. The decomposed instantaneous regret for agent p_1 is

$$R_{1,t}^{C_4}=\theta_{1,*}^Tx_1(t)-\theta_{1,*}^Tx_2(t)=\theta_{1,*}^T(x_1(t)-x_2(t))>0.\tag{3.48}$$

In addition, agent p_2 suffers a positive regret. The decomposed instantaneous regret for agent p_2 is

$$R_{2,t}^{C_4}=\theta_{2,*}^Tx_2(t)-\theta_{2,*}^Tx_1(t)=\theta_{2,*}^T(x_2(t)-x_1(t))>0.\tag{3.49}$$

This bad event's joint probability is the product of two ranking probabilities $\mathbb{P}_t^{C_4}=\mathbb{P}\big(\widehat{\mu}_{1,2}(t)>\widehat{\mu}_{1,1}(t)>\widehat{\mu}_{1,3}(t)\big)$ by agent p_1 and $\mathbb{P}(\mathcal{G}_2(t))$ by agent p_2. $\{\widehat{\mu}_{1,2}(t)>\widehat{\mu}_{1,1}(t)>\widehat{\mu}_{1,3}(t)\}$ is the *bad* event that agent p_1 wrongly estimate its preference scheme over arms. The decomposed instantaneous regret for agent p_1 is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_4}R_{1,t}^{C_4}\\&=\mathbb{P}\big(\widehat{\mu}_{1,2}(t)>\widehat{\mu}_{1,1}(t)>\widehat{\mu}_{1,3}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t)>\widehat{\mu}_{2,1}(t)>\widehat{\mu}_{2,3}(t)\big)\theta_{1,*}^T(x_1(t)-x_2(t))>0.\end{aligned}\tag{3.50}$$

For agent p_2, the decomposed instantaneous regret is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_4}R_{2,t}^{C_4}\\&=\mathbb{P}\big(\widehat{\mu}_{1,2}(t)>\widehat{\mu}_{1,1}(t)>\widehat{\mu}_{1,3}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t)>\widehat{\mu}_{2,1}(t)>\widehat{\mu}_{2,3}(t)\big)\theta_{2,*}^T(x_2(t)-x_1(t))>0.\end{aligned}\tag{3.51}$$

Case 5. If agent p_1 wrongly estimates the ranking over arms as $p_1 : a_1 > a_3 > a_2$. The matching result by DA Algorithm is in Figure 3.5 Case 5. Agent p_1 is matched with arm a_1 and agent p_2 is matched with arm a_2. This pair will not suffer regret. The decomposed instantaneous regret for agent p_1 is

$$R_{1,t}^{C_5}=\theta_{1,*}^Tx_1(t)-\theta_{1,*}^Tx_1(t)=\theta_{1,*}^T(x_1(t)-x_1(t))=0.\tag{3.52}$$

In addition, agent p_2 will not suffer a regret. The decomposed instantaneous regret for agent p_2 is

$$R_{2,t}^{C_5}=\theta_{2,*}^Tx_2(t)-\theta_{2,*}^Tx_2(t)=\theta_{2,*}^T(x_2(t)-x_2(t))=0.\tag{3.53}$$

This bad event's joint probability is the product of two ranking probabilities $\mathbb{P}_t^{C_5}=\mathbb{P}\big(\widehat{\mu}_{1,1}(t)>\widehat{\mu}_{1,3}(t)>\widehat{\mu}_{1,2}(t)\big)$ by agent p_1 and $\mathbb{P}(\mathcal{G}_2(t))$ by agent p_2. $\{\widehat{\mu}_{1,1}(t)>\widehat{\mu}_{1,3}(t)>\widehat{\mu}_{1,2}(t)\}$ is the *bad* event that agent p_1 wrongly estimate its preference scheme over arms. The decomposed instantaneous regret for agent p_1 is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_5}R_{1,t}^{C_5}\\&=\mathbb{P}\big(\widehat{\mu}_{1,1}(t)>\widehat{\mu}_{1,3}(t)>\widehat{\mu}_{1,2}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t)>\widehat{\mu}_{2,1}(t)>\widehat{\mu}_{2,3}(t)\big)\theta_{1,*}^T(x_1(t)-x_1(t))=0.\end{aligned}\tag{3.54}$$

For agent p_2, the decomposed instantaneous regret is

$$\begin{aligned}&\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_5}R_{2,t}^{C_5}\\&=\mathbb{P}\big(\widehat{\mu}_{1,1}(t)>\widehat{\mu}_{1,3}(t)>\widehat{\mu}_{1,2}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t)>\widehat{\mu}_{2,1}(t)>\widehat{\mu}_{2,3}(t)\big)\theta_{2,*}^T(x_2(t)-x_2(t))=0.\end{aligned}\tag{3.55}$$

This setting will not create any regret.

Case 6. If agent p_1 correctly estimates the ranking over arms as $p_1 : a_1 > a_2 > a_3$. The matching result by DA Algorithm is in Figure 3.5 Case 6. Agent p_1 is matched with arm a_1 and agent p_2 is matched with arm a_2. This pair will not suffer regret. The decomposed instantaneous regret for agent p_1 is

$$R_{1,t}^{C_6} = \theta_{1,*}^T x_1(t) - \theta_{1,*}^T x_1(t) = \theta_{1,*}^T (x_1(t) - x_1(t)) = 0. \tag{3.56}$$

In addition, agent p_2 will not suffer a regret. The decomposed instantaneous regret for agent p_1 is

$$R_{2,t}^{C_6} = \theta_{2,*}^T x_2(t) - \theta_{2,*}^T x_2(t) = \theta_{2,*}^T (x_2(t) - x_2(t)) = 0. \tag{3.57}$$

This bad event's joint probability is the product of two ranking probabilities $\mathbb{P}_t^{C_6} = \mathbb{P}\big(\widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t)\big)$ by agent p_1, which in fact is a good event and $\mathbb{P}(\mathcal{G}_2(t))$ by agent p_2. $\{\widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t)\}$ is the *good* event that agent p_1 correctly estimate its preference scheme over arms. The decomposed instantaneous regret for agent p_1 is

$$\begin{aligned} &\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_6} R_{1,t}^{C_6} \\ &= \mathbb{P}\big(\widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t) > \widehat{\mu}_{2,1}(t) > \widehat{\mu}_{2,3}(t)\big)\theta_{1,*}^T (x_1(t) - x_1(t)) = 0, \end{aligned} \tag{3.58}$$

For agent p_2, the decomposed instantaneous regret is

$$\begin{aligned} &\mathbb{P}(\mathcal{G}_2(t))\mathbb{P}_t^{C_6} R_{2,t}^{C_6} \\ &= \mathbb{P}\big(\widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,2}(t)\big)\mathbb{P}\big(\widehat{\mu}_{2,2}(t) > \widehat{\mu}_{2,1}(t) > \widehat{\mu}_{2,3}(t)\big)\theta_{2,*}^T (x_2(t) - x_2(t)) = 0. \end{aligned} \tag{3.59}$$

This setting will also not create any regret.

In summary, for agent p_1, the four regret occurred cases are represented in Case 1 to Case 4, two regret vanished cases happen at Case 5 and Case 6. For agent p_2, the two regret occurred cases are represented in Case 3 and Case 4, four regret vanishing cases happen at Case 1, Case 2, Case 5, and Case 6. These six cases represent all the possible regret occurring

cases when p_1 submits incorrect ranking and p_2 submits correct ranking.

3.15 Proof of Theorem 3.3 - Instance - Dependent Lower Bound

Based on the setting constructed in Section 3.6.4, we conduct the regret analysis to get the lower bound. After h rounds of exploration, for agent p_i, its estimator $\widehat{\theta}_i(t)$ is acquired through the penalized linear regret. Thus at time step t, the estimated mean reward for arm a_j from the viewpoint of agent p_i is $\widehat{\mu}_{i,j}(t) = \widehat{\theta}_i(t)^T x_j(t)$, which provides the basis to construct the ranking list $\widehat{r}_{i,[K]}(t)$. Besides, since all contexts are from uniform distribution, conditioning on all previous information $\mathcal{F}_i(h)$ and contextual information of $x_j(t)$, we have the distribution of the estimated mean reward $\widehat{\mu}_{i,j}(t)$ following the normal distribution

$$\widehat{\mu}_{i,j}(t) = \widehat{\theta}_i(t)^T x_j(t) | \mathcal{F}_i(h) \sim N(\bar{\theta}_i^T x_j(t), \sigma^2 x_j(t)^T \mathbf{M}_i x_j(t)), \quad \forall j \in [K], \tag{3.60}$$

where $\mathbb{E}[\widehat{\theta}_i(t)|\mathcal{F}_i(h)] = \bar{\theta}_i = \big(\mathbf{X}_i(h)^T\mathbf{X}_i(h)+\lambda_i\mathbf{I}\big)^{-1}\mathbf{X}_i(h)^T\mathbf{X}_i(h)\theta_{i,*} \in \mathbb{R}^d$, and $\text{Cov}[\widehat{\theta}_i(t)|\mathcal{F}_i(h)] = \sigma^2\mathbf{M}_i = \sigma^2\big(\mathbf{X}_i(h)^T\mathbf{X}_i(h) + \lambda_i\mathbf{I}\big)^{-1}\mathbf{X}_i(h)^T\mathbf{X}_i(h)\big(\mathbf{X}_i(h)^T\mathbf{X}_i(h) + \lambda_i\mathbf{I}\big)^{-1} \in \mathbb{R}^{d\times d}$.

Denote the true preference for p_i at t is $a_{j_1} <_i^t a_{j_2} <_i^t a_{j_3}$ and the correct ranking event and partial correct ranking rank event as $\mathcal{G}_i(t) = \{\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_2}(t) > \widehat{\mu}_{i,j_3}(t)\}$ and $\mathcal{G}_i^c(t) = \{\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_2}(t) > \widehat{\mu}_{i,j_3}(t)\}^c$. The lower bound probability of the correct ranking estimate (good event) and partial correct ranking estimate (bad event) is provided as follows.

Lemma 3.6. *(1) Define* $\mathbf{M}_i = \big(\mathbf{X}_i(h)^T\mathbf{X}_i(h)+\lambda_i\mathbf{I}\big)^{-1}\mathbf{X}_i(h)^T\mathbf{X}_i(h)\big(\mathbf{X}_i(h)^T\mathbf{X}_i(h)+\lambda_i\mathbf{I}\big)^{-1} \in \mathbb{R}^{d\times d}$, *and* $\Sigma_{i,(j,k)}(t) = \sigma^2[x_j(t)^T\mathbf{M}_i x_j(t) + x_k(t)^T\mathbf{M}_i x_k(t)]$. *If the true preference for* p_i *over arms is* $a_{j_1} < a_{j_2} < a_{j_3}$ *at time step* t, *the probability of* $\mathcal{G}_i(t)$ *is lower bounded by*

$$\mathbb{P}(\mathcal{G}_i(t)) \geq 1 - \frac{1}{\sqrt{2\pi}}\big[\Psi_{i,t}(j_1,j_2) + \Psi_{i,t}(j_2,j_3) + \Psi_{i,t}(j_1,j_3)\big], \tag{3.61}$$

where $\Psi_{i,t}(j,k) = \exp\left(-\nu^2_{i,(j,k)}(t)/2\right)/\nu_{i,t}(j,k)$ *and* $\nu_{i,t}(j,k) = \bar{\theta}_i^T[x_j(t) - x_k(t)]/\Sigma_{i,(j,k)}(t)$ *represents the scaled mean difference of* a_j *and* a_k *from the perspective of* $\bar{\theta}_i$ *at time* t.

(2) Define $\tilde{\nu}_{i,t}(j,k) = \bar{\theta}_i^T[x_j(t) - x_k(t)]/\tilde{\Sigma}_{i,t}(j,k)$ *and* $\tilde{\Sigma}_{i,t}(j,k) = \sigma^2[x_j(t)^T\mathbf{M}_i x_j(t) + x_k(t)^T\mathbf{M}_i x_k(t) - 2x_j(t)^T\mathbf{M}_i x_k(t)]$. *If the true preference for* p_i *over arms is* $a_{j_1} < a_{j_2} < a_{j_3}$ *at time step* t, *the* $\mathcal{G}_i^c(t)$ *probability lower bound is,*

$$\mathbb{P}\big(\mathcal{G}_i^c(t)\big) \geq \min\left\{\Gamma_{i,t}(j_1,j_2), \Gamma_{i,t}(j_2,j_3)\right\} \tag{3.62}$$

where $\Gamma_{i,t}(j,k) = (1/\tilde{\nu}_{i,t}(j,k) - 1/\tilde{\nu}_{i,t}^3(j,k))\exp\left(-\tilde{\nu}_{i,t}^2(j,k)/2\right)$.

Lemma 3.6 is used to getting the $\mathcal{G}_i(t)$ and $\mathcal{G}_i^c(t)$'s lower bounds via the sharp Gaussian tail probability lower bound at each time step. In addition, the conditional expectation regret is provided in Section 3.14. The following lemma provides the order of lower bounds of $\mathcal{G}_i(t)$ and $\mathcal{G}_i^c(t)$.

Lemma 3.7. *Considering the problem instance in appendix, the order of the probability's lower bound are*

$$\mathbb{P}(\mathcal{G}_i(t)) \geq \mathcal{L}_i^g(t) \text{ and } \mathbb{P}\big(\mathcal{G}_i^c(t)\big) \geq \mathcal{L}_i^b(t), \tag{3.63}$$

where $\mathcal{L}_i^g(t) = 1-(3/c_5(t)\sqrt{2})\exp\left(-c_5^2(t)h/2\right)$, $\mathcal{L}_i^b(t) = (1/c_7(t)\sqrt{h}-1/c_7^3(t)h^{3/2})\exp\left(-c_7^2(t)h/2\right)$, *and* $c_5(t), c_7(t)$ *are contextual time-dependent constants but independent of designing exploration rounds* h.

With the distribution of $\widehat{\mu}_{i,j}(t)$, to derive the regret lower bound, we provide the proof of good events $\mathcal{G}_1(t)$ and $\mathcal{G}_2(t)$'s probability lower bound in Section 3.15.1, and bad events $\mathcal{G}_1^c(t)$ and $\mathcal{G}_2^c(t)$'s lower bound in Section 3.15.2. In addition, we provide these events' probability lower bounds' order at time t, which is provided in Section 3.15.3. Finally, with the previous technical lemmas, we provide the final instance-dependent regret lower bound as a whole.

To get the regret of agent p_1, we first assume that p_2 correctly estimates its preference at time step t in the exploitation step. So the instantaneous regret $R_{1,t}(\widehat{r}_2(t) = r_2(t))$ for agent

p_1, if agent p_2 submits correct ranking, can be decomposed as follows,

$$\begin{aligned} R_{1,t}(\widehat{r}_2(t) = r_2(t)) &= \mathbb{P}\big(\mathcal{G}_2(t)\big)\mathbb{E}[R_1] \\ &= \mathbb{P}\big(\mathcal{G}_2(t)\big)\sum_{z=1}^{6}\mathbb{P}_t^{C_z}R_{1,t}^{C_z}(\widehat{r}_2(t) = r_2(t)). \end{aligned} \tag{3.64}$$

where these six cases' regret analysis can be found at Appendix 3.14. So we can decompose these six cases' regrets into

$$\begin{aligned} \sum_{z=1}^{6}\mathbb{P}_t^{C_z}R_{1,t}^{C_z}(\widehat{r}_2(t) = r_2(t)) = \theta_{1,*}^T\Big[&\mathbb{P}\big(\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t)\big)(x_1(t) - x_3(t)) \\ &+ \mathbb{P}\big(\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,1}(t)\big)(x_1(t) - x_3(t)) \\ &+ \mathbb{P}\big(\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t)\big)(x_1(t) - x_2(t)) \\ &+ \mathbb{P}\big(\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,3}(t)\big)(x_1(t) - x_2(t))\Big], \end{aligned} \tag{3.65}$$

because there are four cases suffering regret and two cases without suffering regret. Combining case 1 and case 2 as a whole, and case 3 and case 4 together, we obtain

$$\begin{aligned} = \theta_{1,*}^T\Big[&\Big(\mathbb{P}\big(\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t)\big) + \mathbb{P}\big(\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t)\big)\Big)(x_1(t) - x_3(t)) \\ &+ \Big(\mathbb{P}\big(\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t)\big) + \mathbb{P}\big(\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,3}(t)\big)\Big)(x_1(t) - x_2(t))\Big]. \end{aligned} \tag{3.66}$$

With Lemma 3.6, we have the bad event's probability lower bound, and define $\overline{\Delta}_{1,\min}(t) = \min_{j\in[3],\overline{\Delta}_{1,j}(t)>0}\overline{\Delta}_{1,j}(t) = \min_{j\in[K],\overline{\Delta}_{1,j}(t)>0}\langle\theta_{1,*}, x_{\overline{m}_t(1)}(t) - x_j(t)\rangle$, we can get this instantaneous regret

as follows

$$\begin{aligned}
&\geq \Bigg[\Big(\mathbb{P}\big(\widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,2}(t)\big) + \mathbb{P}\big(\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t)\big)\Big)\overline{\Delta}_{1,\min}(t) \\
&\quad + \Big(\mathbb{P}\big(\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,3}(t) > \widehat{\mu}_{1,1}(t)\big) + \mathbb{P}\big(\widehat{\mu}_{1,2}(t) > \widehat{\mu}_{1,1}(t) > \widehat{\mu}_{1,3}(t)\big)\Big)\overline{\Delta}_{1,\min}(t)\Bigg] \qquad (3.67) \\
&= \mathbb{P}\big(\mathcal{G}_1^c(t)\big)\overline{\Delta}_{1,\min}(t)
\end{aligned}$$

So the regret for agent p_1 is lower bounded by

$$R_{1,t}(\widehat{r}_2(t) = r_2(t)) \geq \mathbb{P}\big(\mathcal{G}_2(t)\big)\mathbb{P}\big(\mathcal{G}_1^c(t)\big)\overline{\Delta}_{1,\min}(t) \geq \mathbb{P}\big(\mathcal{G}_2(t)\big)\mathcal{L}_1^b(t)\overline{\Delta}_{1,\min}(t). \qquad (3.68)$$

Based on Lemma 3.6, we have the good event $\mathcal{G}_2(t)$'s probability lower bound and get

$$R_{1,t}(\widehat{r}_2(t) = r_2(t)) \geq \mathcal{L}_2^g(t)\mathcal{L}_1^b(t)\overline{\Delta}_{1,\min}(t). \qquad (3.69)$$

With the same rule, we obtain similar result when p_2 is incorrect,

$$R_{1,t}(\widehat{r}_2(t) \neq r_2(t)) \geq \overline{\Delta}_{1,\min}(t)\prod_{i=1}^{2}\mathcal{L}_i^b(t). \qquad (3.70)$$

By considering agent p_2's preference at time t, the regret for agent p_1 at time t is lower bounded by

$$R_1(t) \geq \overline{\Delta}_{1,\min}(t)\Big(\prod_{i=1}^{2}\mathcal{L}_i^b(t) + \mathcal{L}_2^g(t)\mathcal{L}_1^b(t)\Big). \qquad (3.71)$$

The agent p_2 gets similar regret lower bound by symmetry. So the overall lower bound regret for agent p_1 is

$$R_1(T) \geq \sum_{t=1}^{h}\Delta_{i,m_t(i)}(t) + \sum_{t=h+1}^{T}\overline{\Delta}_{1,\min}(t)\Big(\prod_{i=1}^{2}\mathcal{L}_i^b(t) + \mathcal{L}_2^g(t)\mathcal{L}_1^b(t)\Big). \qquad (3.72)$$

Besides, we analyze the order of the two probability lower bounds' product.

$$\mathcal{L}_2^g(t)\mathcal{L}_1^b(t) = (\frac{1}{\sqrt{h}} - \frac{1}{\sqrt{h^3}})e^{-\frac{h}{2}}(1 - \frac{1}{\sqrt{h}}e^{-\frac{h}{2}}) = \frac{c_8(t)}{\sqrt{h}}e^{-\frac{h}{2}}, \tag{3.73}$$

where $c_8(t)$ is a context-dependent constant, but independent of h. And the product order of these bad events' probability lower bounds between two agents is,

$$\mathcal{L}_1^b(t)\mathcal{L}_2^b(t) = (\frac{1}{\sqrt{h}} - \frac{1}{\sqrt{h^3}})e^{-\frac{h}{2}}(\frac{1}{\sqrt{h}} - \frac{1}{\sqrt{h^3}})e^{-\frac{h}{2}} = \frac{1}{h}e^{-c_9(t)h}, \tag{3.74}$$

where $c_9(t)$ is a context-dependent constant, but independent of h. So the sum of $\mathcal{L}_2^g(t)\mathcal{L}_1^b(t)$ and $\mathcal{L}_1^b(t)\mathcal{L}_2^b(t)$ will be $\frac{c_{10}(t)}{\sqrt{h}}e^{-h}$. By $1/\sqrt{h} > 1/\sqrt{T}$, the $R_1(T)$ will be lower bounded by $h\overline{\Delta}_{i,\min} + \sqrt{T}\overline{\Delta}_{1,\min}c_{10}(t)e^{-h}$. Then by similar analysis derived in the upper bound order analysis of dynamic matching in Appendix 3.12.2, we find that the order of the regret lower bound will be $\Omega(\log(T))$.

3.15.1 Proof of Lemma 3.6 - Good Event

Proof. First, without loss of generality, suppose that the true preference from agent p_i to all arms is $a_{j_1} <_i^t a_{j_2} <_i^t a_{j_3}$ at time t. In order to present the competing status of those agents, we need to quantify the probability of the good event $\mathcal{G}_i(t) = \{\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_2}(t) > \widehat{\mu}_{i,j_3}(t)\}$. Here we denote $\mathcal{A}(t) = \{\widehat{\mu}_{i,j_1}(t) > \max(\widehat{\mu}_{i,j_2}(t), \widehat{\mu}_{i,j_3}(t))\}$ as the *1st-good event* and $\mathcal{B}(t) = \{\widehat{\mu}_{i,j_2}(t) > \widehat{\mu}_{i,j_3}(t)\}$ as the *2nd-good event*, where $\mathcal{G}_i(t) = \mathcal{A}(t) \cap \mathcal{B}(t)$. Here we omit the index 'i' in *1st-good event* and *2nd-good event*. The *1st-good event* $\mathcal{A}$ can also be divided into to the event $\mathcal{A}_1(t) = \{\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_2}(t)\}$ and the event $\mathcal{A}_2(t) = \{\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_3}(t)\}$ happening simultaneously, where $\mathcal{A}(t) = \mathcal{A}_1(t) \cap \mathcal{A}_2(t)$. By the property of $\mathbb{P}(\mathcal{A}(t) \cap \mathcal{B}(t)) \geq \mathbb{P}(\mathcal{A}(t)) + \mathbb{P}(\mathcal{B}(t)) - \mathbb{P}(\mathcal{A}(t) \cup \mathcal{B}(t)) \geq \mathbb{P}(\mathcal{A}(t)) + \mathbb{P}(\mathcal{B}(t)) - 1$, we use the same technique again

and have $\mathbb{P}(\mathcal{A}(t)) \geq \mathbb{P}(\mathcal{A}_1(t)) + \mathbb{P}(\mathcal{A}_2(t)) - 1$. So the event $\mathcal{A}(t)$'s probability lower bound is,

$$\begin{aligned}
\mathbb{P}(\mathcal{A}(t)) &= \mathbb{P}(\widehat{\mu}_{i,j_1}(t) > \max(\widehat{\mu}_{i,j_2}(t), \widehat{\mu}_{i,j_3}(t))) \\
&= \mathbb{P}(\{\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_2}(t)\} \cap \{\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_3}(t)\}) \\
&\geq \mathbb{P}(\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_2}(t)) + \mathbb{P}(\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_3}(t)) - 1 \\
&= \mathbb{P}(\mathcal{A}_1(t)) + \mathbb{P}(\mathcal{A}_2(t)) - 1.
\end{aligned} \tag{3.75}$$

Now we have to quantify the event $\mathcal{A}_1(t)$ and event $\mathcal{A}_2(t)$'s probabilities' lower bound. We first define the estimated mean reward difference for agent p_i at time t between arm a_{j_1} and arm a_{j_2} as $\widehat{Z}_{i,(j_1,j_2)} = \widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t)$. Given all contextual information at time t, we get

$$\widehat{Z}_{i,(j_1,j_2)} | \mathcal{F}_i(h) \sim N(\bar{\theta}_i^T [x_{j_1}(t) - x_{j_2}(t)], \tilde{\Sigma}_{i,(j_1,j_2)}(t)), \tag{3.76}$$

where $\tilde{\Sigma}_{i,(j_1,j_2)}(t) = \sigma^2 [x_{j_1}(t)^T \mathbf{M}_i x_{j_1}(t) + x_{j_2}(t)^T \mathbf{M}_i x_{j_2}(t) - 2 x_{j_1}(t)^T \mathbf{M}_i x_{j_2}(t)] \in \mathbb{R}$ is the variance of the estimated mean reward $\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t)$. We know that $\widehat{\mu}_{i,j_1}(t)$ and $\widehat{\mu}_{i,j_2}(t)$ are positively correlated because $\mathbf{M}_i$ is positive semi-definite since x_{j_1} and x_{j_2}'s coordinates are follow uniform distribution $U(0,1)^d$. So the variance of the difference $\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t)$ of the two correlated normal random variables is less than the variance of the difference of two independent normal random variables by the property $var(\varpi_1 - \varpi_2) \leq var(\varpi_1) + var(\varpi_2)$ if ϖ_1 and ϖ_2 are positively correlated random variables. Besides, we know if two normal random variables $\widehat{\mu}_{i,j_1}(t)$ and $\widehat{\mu}_{i,j_2}(t)$ are independent,

$$\tilde{\Sigma}_{i,(j_1,j_2)}(t) \leq \sigma^2 [x_{j_1}(t)^T \mathbf{M}_i x_{j_1}(t) + x_{j_2}(t)^T \mathbf{M}_i x_{j_2}(t)], \tag{3.77}$$

where $x_{j_1}(t)^T \mathbf{M}_i x_{j_1}(t)$ and $x_{j_2}(t)^T \mathbf{M}_i x_{j_2}(t)$ are the variances of $\widehat{\mu}_{i,j_1}(t)$ and $\widehat{\mu}_{i,j_2}(t)$ correspondingly, and we define $\Sigma_{i,(j_1,j_2)}(t) = \sigma^2 [x_{j_1}(t)^T \mathbf{M}_i x_{j_1}(t) + x_{j_2}(t)^T \mathbf{M}_i x_{j_2}(t)]$. We use the *proxy* random variable $Z_{i,(j_1,j_2)}$ to define the difference of two independent Gaussian random

variables. $Z_{i,(j_1,j_2)}$'s distribution follows the normal distribution

$$Z_{i,(j_1,j_2)}|\mathcal{F}_i(h) \sim N(\bar{\theta}_i^T[x_{j_1}(t) - x_{j_2}(t)], \Sigma_{i,(j_1,j_2)}(t)), \tag{3.78}$$

where $\bar{\theta}_i^T[x_{j_1}(t)-x_{j_2}(t)]$ is the $Z_{i,(j_1,j_2)}$'s expectation and $\Sigma_{i,(j_1,j_2)}(t)$ is the variance of $Z_{i,(j_1,j_2)}$. In the following passage, we omit the filtration $|\mathcal{F}_i(h)$ in argument. Then we can obtain the probability lower bound of arm a_{j_1} is ranked higher than the arm a_{j_2} at time step t from the viewpoint of agent p_i via the proxy random variable $Z_{i,(j_1,j_2)}$, that is

$$\begin{aligned}
\mathbb{P}(\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_2}(t)) &= \mathbb{P}(\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0) \\
&= \mathbb{P}(\widehat{Z}_{i,(j_1,j_2)} > 0) \\
&\geq \mathbb{P}(Z_{i,(j_1,j_2)} > 0), \text{ by the inequality (3.77)} \\
&= \mathbb{P}\left(\frac{Z_{i,(j_1,j_2)} - \bar{\theta}_i^T[x_{j_1}(t) - x_{j_2}(t)]}{\Sigma_{i,(j_1,j_2)}(t)} \geq -\nu_{i,(j_1,j_2)}(t)\right),
\end{aligned} \tag{3.79}$$

where $\nu_{i,(j_1,j_2)}(t) = \frac{\bar{\theta}_i^T[x_{j_1}(t)-x_{j_2}(t)]}{\Sigma_{i,(j_1,j_2)}(t)}$ greater than zero, is based on the true preference's setting that $a_{j_1} >_i^t a_{j_2} >_i^t a_{j_3}$ for agent p_i at time t. The aim of the last equality is to transform the proxy random variable to the standard normal variable and quantify the event $\mathcal{A}_1(t)$'s probability lower bound. Now this event $\mathcal{A}_1(t)$'s probability lower bound is

$$\begin{aligned}
\mathbb{P}(\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_2}(t)) &\geq 1 - \mathbb{P}\left(\frac{Z_{i,(j_1,j_2)} - \bar{\theta}_i^T[x_{j_1}(t) - x_{j_2}(t)]}{\Sigma_{i,(j_1,j_2)}(t)} \geq \nu_{i,(j_1,j_2)}(t)\right) \\
&\geq 1 - \frac{1}{\nu_{i,(j_1,j_2)}(t)} \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{\nu_{i,(j_1,j_2)}^2(t)}{2}\right)},
\end{aligned} \tag{3.80}$$

where the last inequality is by Lemma 3.4, which provides the tail probability of the normal distribution since $\nu_{i,(j_1,j_2)}(t)$ is positive. With the same technique, we can acquire the event $\mathcal{A}_2(t)$'s probability's lower bound and we also define $\nu_{t,(j_1,j_3)} = \frac{\bar{\theta}_i^T[x_{j_1}(t)-x_{j_3}(t)]}{\Sigma_{i,(j_1,j_3)}(t)}$, which is greater

than zero. Then we have

$$\mathbb{P}(\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_3}(t)) \geq 1 - \frac{1}{\nu_{t,(j_1,j_3)}} \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{\nu^2_{t,(j_1,j_3)}}{2}\right)}. \tag{3.81}$$

So the *1st-good event*'s lower bound probability is,

$$\begin{aligned}
&\mathbb{P}(\widehat{\mu}_{i,j_1}(t) > \max(\widehat{\mu}_{i,j_2}(t), \widehat{\mu}_{i,j_3}(t))) \\
&\geq 1 - \frac{1}{\nu_{i,(j_1,j_2)}(t)} \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{\nu^2_{i,(j_1,j_2)}(t)}{2}\right)} + 1 - \frac{1}{\nu_{t,(j_1,j_3)}} \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{\nu^2_{t,(j_1,j_3)}}{2}\right)} - 1 \\
&= 1 - \frac{1}{\nu_{i,(j_1,j_2)}(t)} \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{\nu^2_{i,(j_1,j_2)}(t)}{2}\right)} - \frac{1}{\nu_{t,(j_1,j_3)}} \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{\nu^2_{t,(j_1,j_3)}}{2}\right)}.
\end{aligned} \tag{3.82}$$

And the *2nd-good event*'s lower bound probability is,

$$\mathbb{P}(\widehat{\mu}_{i,j_2}(t) > \widehat{\mu}_{i,j_3}(t)) \geq 1 - \frac{1}{\nu_{i,(j_2,j_3)}(t)} \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{\nu^2_{i,(j_2,j_3)}(t)}{2}\right)}, \tag{3.83}$$

where $\nu_{t,(j_2,j_3)} = \frac{\bar{\theta}_i^T[x_{j_2}(t) - x_{j_3}(t)]}{\Sigma_{i,(j_2,j_3)}(t)} > 0$ and we define $\Sigma_{i,(j_2,j_3)}(t) = \sigma^2[x_{j_2}(t)^T \mathbf{M}_i x_{j_2}(t) + x_{j_3}(t)^T \mathbf{M}_i x_{j_3}(t)]$.

Here we provide all definitions of $\Sigma_{i,(j_1,j_2)}(t)$, $\Sigma_{i,(j_2,j_3)}(t)$, and $\Sigma_{i,(j_1,j_3)}(t)$,

$$\begin{aligned}
\nu_{i,(j_1,j_2)}(t) &= \frac{\bar{\theta}_i^T[x_{j_1}(t) - x_{j_2}(t)]}{\Sigma_{i,(j_1,j_2)}(t)}, \Sigma_{i,(j_1,j_2)}(t) = \sigma^2[x_{j_1}(t)^T \mathbf{M}_i x_{j_1}(t) + x_{j_2}(t)^T \mathbf{M}_i x_{j_2}(t)] \\
\nu_{i,(j_2,j_3)}(t) &= \frac{\bar{\theta}_i^T[x_{j_2}(t) - x_{j_3}(t)]}{\Sigma_{i,(j_2,j_3)}(t)}, \Sigma_{i,(j_2,j_3)}(t) = \sigma^2[x_{j_2}(t)^T \mathbf{M}_i x_{j_2}(t) + x_{j_3}(t)^T \mathbf{M}_i x_{j_3}(t)] \\
\nu_{i,(j_1,j_3)}(t) &= \frac{\bar{\theta}_i^T[x_{j_1}(t) - x_{j_3}(t)]}{\Sigma_{i,(j_1,j_3)}(t)}, \Sigma_{i,(j_1,j_3)}(t) = \sigma^2[x_{j_1}(t)^T \mathbf{M}_i x_{j_1}(t) + x_{j_3}(t)^T \mathbf{M}_i x_{j_3}(t)]
\end{aligned} \tag{3.84}$$

So the final good event $\mathcal{G}_i(t)$'s probability lower bound is

$$\begin{aligned}\mathbb{P}(\mathcal{G}_i(t)) \geq 1 &- \frac{1}{\nu_{i,(j_1,j_2)}(t)}\frac{1}{\sqrt{2\pi}}e^{(-\frac{\nu^2_{i,(j_1,j_2)}(t)}{2})} \\ &- \frac{1}{\nu_{i,(j_1,j_3)}(t)}\frac{1}{\sqrt{2\pi}}e^{(-\frac{\nu^2_{i,(j_1,j_3)}(t)}{2})} - \frac{1}{\nu_{i,(j_2,j_3)}(t)}\frac{1}{\sqrt{2\pi}}e^{(-\frac{\nu^2_{i,(j_2,j_3)}(t)}{2})}.\end{aligned} \tag{3.85}$$

3.15.2 Proof of Lemma 3.6 - Bad Event

Proof. To get the probability lower bound of the bad event $\mathcal{G}_i^c(t)$, we can obtain the upper bound of the good event $\mathcal{G}_i(t)$ probability first. The proof path is similar to the proof of Lemma 3.6 but with the upper bound of the tail probability of the normal distribution and exists some nuances. We have

$$\begin{aligned}&\mathbb{P}\big(\mathcal{G}_i(t)\big) \\ &= \mathbb{P}\big(\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0, \widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t) > 0\big) \\ &= \mathbb{P}\big(\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0 | \widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t) > 0\big)\mathbb{P}\big(\widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t) > 0\big) \\ &\leq \mathbb{P}\big(\widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t) > 0\big),\end{aligned} \tag{3.86}$$

where the last inequality holds because $\mathbb{P}\big(\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0 | \widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t) > 0\big) \leq 1$. Similarly we have

$$\begin{aligned}&\mathbb{P}\big(\mathcal{G}_i(t)\big) \\ &= \mathbb{P}\big(\widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t) > 0 | \widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0\big)\mathbb{P}\big(\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0\big) \\ &\leq \mathbb{P}\big(\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0\big),\end{aligned} \tag{3.87}$$

where the last inequality holds because $\mathbb{P}\big(\widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t) > 0 | \widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0\big) \leq 1$.

To get the upper bound of $\mathbb{P}\big(\mathcal{G}_i(t)\big)$, we need to quantify the maximum value of $\mathbb{P}\big(\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0\big)$ and $\mathbb{P}\big(\widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t) > 0\big)$. Here we use the same definition in Lemma 3.6, $\mathcal{A}_1(t) = \{\widehat{\mu}_{i,j_1}(t) > \widehat{\mu}_{i,j_2}(t)\}$ and $\mathcal{B}(t) = \widehat{\mu}_{i,j_2}(t) > \widehat{\mu}_{i,j_3}(t)$. In the following, we provide

the proof of getting upper bound probability of $\mathcal{B}(t)$ and $\mathcal{A}_1(t)$. The proof of getting the probability upper bound of two quantities is similar, so we get the upper bound of $\mathbb{P}\big(\mathcal{B}(t)\big)$ first.

Let's use the similar notation defined in Lemma 3.6.

$$\widehat{Z}_{i,(j_2,j_3)} = \widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t)|\mathcal{F}_i(h) \sim N(\bar{\theta}_i^T[x_{j_2}(t) - x_{j_3}(t)], \tilde{\Sigma}_{i,(j_2,j_3)}), \tag{3.88}$$

where $\tilde{\Sigma}_{i,(j_2,j_3)} = \sigma^2[x_{j_2}(t)^T\mathbf{M}_i x_{j_2}(t) + x_{j_3}(t)^T\mathbf{M}_i x_{j_3}(t) - 2x_{j_2}(t)^T\mathbf{M}_i x_{j_3}(t)]$, greater than zero, is the true variance of $\widehat{Z}_{i,(j_2,j_3)}$. So

$$\begin{aligned}
\mathbb{P}\big(\mathcal{B}(t)\big) &= \mathbb{P}\bigg(\widehat{Z}_{i,(j_2,j_3)} > 0\bigg) \\
&= \mathbb{P}\bigg(\frac{\widehat{Z}_{i,(j_2,j_3)} - \bar{\theta}_i^T[x_{j_2}(t) - x_{j_3}(t)]}{\tilde{\Sigma}_{i,(j_2,j_3)}} \geq -\tilde{\nu}_{i,(j_2,j_3)}(t)\bigg) \\
&= 1 - \mathbb{P}\bigg(\frac{\widehat{Z}_{i,(j_2,j_3)} - \bar{\theta}_i^T[x_{j_2}(t) - x_{j_3}(t)]}{\tilde{\Sigma}_{i,(j_2,j_3)}} \geq \tilde{\nu}_{i,(j_2,j_3)}(t)\bigg)
\end{aligned} \tag{3.89}$$

where the last equality holds by the symmetry property of normal distribution and define $\tilde{\nu}_{i,(j_2,j_3)}(t) = \frac{\bar{\theta}_i^T[x_{j_2}(t) - x_{j_3}(t)]}{\tilde{\Sigma}_{i,(j_2,j_3)}}$, greater than zero. Besides, $\tilde{\nu}_{i,(j_1,j_2)}(t)$ can be defined similarly,

$$\begin{aligned}
\tilde{\nu}_{i,(j_1,j_2)}(t) &= \frac{\bar{\theta}_i^T[x_{j_1}(t) - x_{j_2}(t)]}{\tilde{\Sigma}_{i,(j_1,j_2)}(t)}, \\
\tilde{\Sigma}_{i,(j_1,j_2)}(t) &= \sigma^2[x_{j_1}(t)^T\mathbf{M}_i x_{j_1}(t) + x_{j_2}(t)^T\mathbf{M}_i x_{j_2}(t) - 2x_{j_1}(t)^T\mathbf{M}_i x_{j_2}(t)].
\end{aligned} \tag{3.90}$$

So by the Lemma 3.4's lower bound of normal tail probability, we have the upper bound probability of $\mathcal{B}(t)$,

$$\mathbb{P}\big(\mathcal{B}(t)\big) \leq 1 - \Big(\frac{1}{\tilde{\nu}_{i,(j_2,j_3)}(t)} - \frac{1}{\tilde{\nu}^3_{i,(j_2,j_3)}(t)}\Big)e^{\left(-\frac{\tilde{\nu}^2_{i,(j_2,j_3)}(t)}{2}\right)}. \tag{3.91}$$

So the similar result can be obtained for $\mathcal{A}_1(t)$,

$$\mathbb{P}\big(\mathcal{A}_1(t)\big) \leq 1 - \Big(\frac{1}{\tilde{\nu}_{i,(j_1,j_2)}(t)} - \frac{1}{\tilde{\nu}^3_{i,(j_1,j_2)}(t)}\Big)e^{\left(-\frac{\tilde{\nu}^2_{i,(j_1,j_2)}(t)}{2}\right)}. \tag{3.92}$$

Since

$$\begin{aligned}\mathbb{P}\big(\mathcal{G}_i(t)\big) &\leq \max\Big\{\mathbb{P}\big(\widehat{\mu}_{i,j_2}(t) - \widehat{\mu}_{i,j_3}(t) > 0\big), \mathbb{P}\big(\widehat{\mu}_{i,j_1}(t) - \widehat{\mu}_{i,j_2}(t) > 0\big)\Big\}, \\ &\leq 1 - \min\Big\{\Big(\frac{1}{\tilde{\nu}_{i,(j_2,j_3)}(t)} - \frac{1}{\tilde{\nu}^3_{i,(j_2,j_3)}(t)}\Big)e^{\left(-\frac{\tilde{\nu}^2_{i,(j_2,j_3)}(t)}{2}\right)}, \\ &\qquad \Big(\frac{1}{\tilde{\nu}_{i,(j_1,j_2)}(t)} - \frac{1}{\tilde{\nu}^3_{i,(j_1,j_2)}(t)}\Big)e^{\left(-\frac{\tilde{\nu}^2_{i,(j_1,j_2)}(t)}{2}\right)}\Big\}.\end{aligned} \tag{3.93}$$

Meanwhile we get the lower bound of $\mathcal{G}_i^c(t)$ as follows,

$$\begin{aligned}\mathbb{P}\big(\mathcal{G}_i^c(t)\big) \geq \min\Big\{&\Big(\frac{1}{\tilde{\nu}_{i,(j_2,j_3)}(t)} - \frac{1}{\tilde{\nu}^3_{i,(j_2,j_3)}(t)}\Big)e^{\left(-\frac{\tilde{\nu}^2_{i,(j_2,j_3)}(t)}{2}\right)}, \\ &\Big(\frac{1}{\tilde{\nu}_{i,(j_1,j_2)}(t)} - \frac{1}{\tilde{\nu}^3_{i,(j_1,j_2)}(t)}\Big)e^{\left(-\frac{\tilde{\nu}^2_{i,(j_1,j_2)}(t)}{2}\right)}\Big\}\end{aligned} \tag{3.94}$$

3.15.3 Proof of Lemma 3.7

Proof. In order to get the good event $\mathcal{G}_i(t)$ and bad event $\mathcal{G}_i^c(t)$'s probability order. We first need to analyze the order of $\nu_{i,(j_1,j_2)}(t)$, $\tilde{\nu}_{i,(j_1,j_2)}(t)$, $\Sigma_{i,(j_2,j_3)}(t)$, $\tilde{\Sigma}_{i,(j_2,j_3)}(t)$ and other similar terms.

Based on the definition of $\nu_{i,(j_1,j_2)}(t) = \frac{\bar{\theta}_i^T[x_{j_1}(t) - x_{j_2}(t)]}{\Sigma_{i,(j_1,j_2)}}$, we know that the context difference at time t, which is $x_{j_1}(t) - x_{j_2}(t)$, independent of h. The expected ridge parameter is $\bar{\theta}_i = \big(\mathbf{X}_i(h)^T\mathbf{X}_i(h) + \lambda_i\mathbf{I}\big)^{-1}\mathbf{X}_i(h)^T\mathbf{X}_i(h)\theta_{i,*}$. Based on the problem design in Eq. (**??**), $\bar{\theta}_i$ can be rewritten as $\{\sqrt{1-1/h}\mathbf{c}_1 + 1/\sqrt{h}\mathbf{c}_2\} = 1/\sqrt{h}\mathbf{c}_3$, where $\mathbf{c}_1, \mathbf{c}_2$ are time-

dependent constants based on $\left(\mathbf{X}_i(h)^T\mathbf{X}_i(h)+\lambda_i\mathbf{I}\right)^{-1}\mathbf{X}_i(h)^T\mathbf{X}_i$ but independent of h, and $\mathbf{c}_3$ is also a context-constant, but independent of h. Besides, we know that $x_{j_1}(t)^T\mathbf{M}_i x_{j_1}(t) \leq \lambda_{\max}(\mathbf{M}_i)||x_{j_1}(t)||_2^2 \leq \lambda_{\max}(\mathbf{M}_i)L$ by the property $||x_{j_1}(t)||_2^2 \leq L$, where L is a constant and we assume $L=1$. So $\lambda_{\max}(\mathbf{M}_i) = c_4/h$ by Chapter 4 from (Ver18), where c_4 can be viewed as a context-constant, independent of h. Thus $\nu_{i,(j_1,j_2)}(t) = \frac{\mathbf{c}_3^T[x_{j_1}(t)-x_{j_2}(t)]/\sqrt{h}}{c_4/h} = c_5(t)\sqrt{h}$, where $c_5(t)$ is a context-dependent constant, but independent of h.

From Lemma 3.6, we get the lower bound of of probability $\mathbb{P}(\mathcal{G}_i(t))$ such as

$$
\begin{aligned}
&\mathbb{P}(\mathcal{G}_i(t)) \\
&\geq 1 - \frac{1}{\nu_{i,(j_1,j_2)}(t)}\frac{1}{\sqrt{2\pi}}e^{\left(-\frac{\nu^2_{i,(j_1,j_2)}(t)}{2}\right)} \\
&\quad - \frac{1}{\nu_{i,(j_1,j_3)}(t)}\frac{1}{\sqrt{2\pi}}e^{\left(-\frac{\nu^2_{i,(j_1,j_3)}(t)}{2}\right)} - \frac{1}{\nu_{i,(j_2,j_3)}(t)}\frac{1}{\sqrt{2\pi}}e^{\left(-\frac{\nu^2_{i,(j_2,j_3)}(t)}{2}\right)} \\
&= 1 - 3\max\left\{\frac{1}{\nu_{i,(j_1,j_2)}(t)}\frac{1}{\sqrt{2\pi}}e^{\left(-\frac{\nu^2_{i,(j_1,j_2)}(t)}{2}\right)},\right. \\
&\quad \left.\frac{1}{\nu_{i,(j_1,j_3)}(t)}\frac{1}{\sqrt{2\pi}}e^{\left(-\frac{\nu^2_{i,(j_1,j_3)}(t)}{2}\right)}, \frac{1}{\nu_{i,(j_2,j_3)}(t)}\frac{1}{\sqrt{2\pi}}e^{\left(-\frac{\nu^2_{i,(j_2,j_3)}(t)}{2}\right)}\right\},
\end{aligned}
\tag{3.95}
$$

and its corresponding order,

$$
\mathbb{P}\big(\mathcal{G}_i(t)\big) \geq \mathcal{L}_i^g(t), \tag{3.96}
$$

where we define $\mathcal{L}_i^g(t) \triangleq 1 - \frac{3}{\sqrt{2\pi}}\frac{1}{c_5(t)\sqrt{h}}e^{\left(-\frac{c_5^2(t)}{2}h\right)}$ as the good event $\mathcal{G}_i(t)$'s probability lower bound.

Based on Lemma 3.6, we can get the bad event $\mathcal{G}_i^c(t)$' probability lower bound, which is

$$\mathbb{P}\big(\mathcal{G}_i^c(t)\big) \geq \min\Big\{(\frac{1}{\tilde{\nu}_{i,(j_2,j_3)}(t)} - \frac{1}{\tilde{\nu}^3_{i,(j_2,j_3)}(t)})e^{\left(-\frac{\tilde{\nu}^2_{i,(j_2,j_3)}(t)}{2}\right)}, \\ (\frac{1}{\tilde{\nu}_{i,(j_1,j_2)}(t)} - \frac{1}{\tilde{\nu}^3_{i,(j_1,j_2)}(t)})e^{\left(-\frac{\tilde{\nu}^2_{i,(j_1,j_2)}(t)}{2}\right)}\Big\} \tag{3.97}$$

where $\tilde{\nu}_{i,(j_1,j_2)}(t), \tilde{\nu}_{i,(j_2,j_3)}(t)$ and $\tilde{\Sigma}_{i,(j_1,j_2)}(t), \tilde{\Sigma}_{i,(j_2,j_3)}(t)$ are defined in Lemma 3.6.

In addition, we know $\bar{\theta}_i = \sqrt{h}\mathbf{c}_3$ by the instance design. Since $x_{j_2}(t)^T\mathbf{M}_i x_{j_2}(t) \geq \lambda_{\min}(\mathbf{M}_i)||x_{j_2}(t)||_2^2 \geq \lambda_{\min}(\mathbf{M}_i)c_{\min,j_2}(t)$ where $c_{\min,j_2}(t) = \min_{t\in[h,T]}||x_{j_2}(t)||_2^2$ and we assume contexts are meaningful, so $||x_{j_2}(t)|| \neq 0$. Because we know that $\langle x_{j_2}(t), x_{j_3}(t)\rangle \geq 0$, $2x_{j_2}(t)^T\mathbf{M}_i x_{j_3}(t) \geq 2\lambda_{\min}(\mathbf{M}_i)\langle x_{j_2}(t), x_{j_3}(t)\rangle \geq 2\lambda_{\min}(\mathbf{M}_i)c_{\min,(j_2,j_3)}(t)$, where $c_{\min,(j_2,j_3)}(t) = \min_{t\in[h+1,T]}\langle x_{j_2}(t), x_{j_3}(t)\rangle$. Then $\tilde{\Sigma}_{i,(j_2,j_3)} \geq 2\sigma^2 L\lambda_{\min}(\mathbf{M}_i) - 2\sigma^2\lambda_{\max}(\mathbf{M}_i)$ $c_{\min,(j_2,j_3)}(t) = c_{6,(j_2,j_3)}(t)/h$. Thus $\tilde{\nu}_{i,(j_2,j_3)}(t)$ is less $\frac{\mathbf{c}_3^T[x_{j_2}(t)-x_{j_3}(t)]/\sqrt{h}}{c_{6,(j_2,j_3)}(t)/h} \triangleq c_7(t)\sqrt{h}$, where $c_7(t)$ is a context-dependent constant, but independent of h.

So we get the lower bound order of $\mathbb{P}\big(\mathcal{G}_i^c(t)\big)$,

$$\mathbb{P}\big(\mathcal{G}_i^c(t)\big) \geq \mathcal{L}_i^b(t), \tag{3.98}$$

where we define $\mathcal{L}_i^b(t) \triangleq (\frac{1}{c_7(t)\sqrt{h}} - \frac{1}{c_7^3(t)h^{3/2}})e^{\left(-\frac{c_7^2(t)}{2}h\right)}$ as the bad event $\mathcal{G}_i^c(t)$'s probability lower bound.

3.16 More Simulations

3.16.1 Section 3.3.1 Example - Incapable Exploration

We set the true matching reward for three firms to $(0.8, 0.4, 0.2)$, $(0.5, 0.7, 0.2)$, $(0.6, 0.3, 0.65)$. All preferences from companies over workers can be derived from the true matching reward.

As we can view, company p_3 has a similar preference over a_1 (0.6) and a_3 (0.65). Thus, the small difference can lead the incapable exploration as described in Section 3.3.1 by the UCB algorithm.

Next we present the experiment settings of S3, S4, and S5.

3.16.2 More Simulation Settings

Scenario 3 (S3): The uniform sub-optimal minimal condition for this scenario is set to be $\overline{\Delta}_{i,\min} = 0.05, \forall i \in [N]$. The time horizon is set to be $T = 5000$ to have a long enough learning length since we decrease the uniform sub-optimal minimal condition. The learning length h for the three noise levels are $h = [264, 876, 4014]$, correspondingly. Thus the difference between S3 and S2 is the time horizon T and different hyperparameters. The data generation process for S3 and S2 are the same.

Scenario 4 (S4): The difference between S4 and S1 is that the context dimension changes from $d = 2$ to $d = 10$. The time horizon is set to be $T = 10000$ to accommodate the large dimension. Besides, the contextual features $\mu_j \in \mathbb{R}^{10}, \forall j \in [3]$, follow similar data generation process as it in S1. Here we consider the global preference, i.e, we assume that arms to agents' preference is the global preference, $a_1 : p_1 > p_2, a_2 : p_1 > p_2, a_3 : p_1 > p_2$. When contexts are noiseless ($\rho = 0$), the true optimal matching is $\{(p_1, a_1), (p_2, a_2)\}$. The uniform sub-optimal minimal condition for this scenario is set as $\overline{\Delta}_{i,\min} = 0.2, \forall i \in [N]$. The learning step length h is 5856.

Scenario 5 (S5): The setting in S5 is the same as the setting in S4 except $N = 5, K = 5$, and $d = 5$. The time horizon is set to be $T = 15000$ to accommodate the increasing number of participants. The uniform sub-optimal minimal condition for this scenario is set to be $\overline{\Delta}_{i,\min} = 0.1, \forall i \in [N]$. The learning step length h is 1975.

3.16.3 Additional Simulation Results

Here we present the experimental analysis of S3 - S5.

Scenario 3 (S3): dynamic matching algorithm is robust to different uniform minimal margin scenarios. In Figure 3.11, we present the result of S3. As we change $\overline{\Delta}_{i,\min} = 0.2$ in S2 to $\overline{\Delta}_{i,\min} = 0.05, \forall i \in [1,2]$, the learning length becomes larger and the estimation becomes better. Compared with S2's first row and second row, the shaded area in S3's first row and second row becomes narrower, which substantiates our conjecture. In the second row and third row, we find that agent p_1 achieves the negative cumulative regret mainly because in the learning step, agent p_1 is periodically matched with the super-optimal arm with a huge (in absolute value) negative regret.

Scenario 4 (S4): dynamic matching algorithm is robust to different dimensions. In Figure 3.12, we present the result of S4. Compared with all previous results, we find that when dimension d increases, the regret increases, and the logarithm regret pattern indicates that dynamic matching algorithm is still robust to the dimension.

Scenario 5 (S5): dynamic matching algorithm is robust to multiple participants. In Figure 3.13, we present the result of S5, which includes five agents and five arms. Based on the analysis from previous figures and results, dynamic matching algorithm is robust to the choice of preference, context dimension, and context changing format (fixed mean and dynamic mean). Furthermore, we find that dynamic matching is also robust to multiple participants. The cumulative regret still shows the logarithmic shape.

3.16.4 Additional Real Data Result

In Figure 3.14, we exhibit the regret of two companies and find that dynamic matching algorithm's individual regret is superior over all comparison methods under different noise levels for both agents. The shaded area represents the upper and lower bound regret over 100 replications. Lines are used to represent the regret mean over these replications.

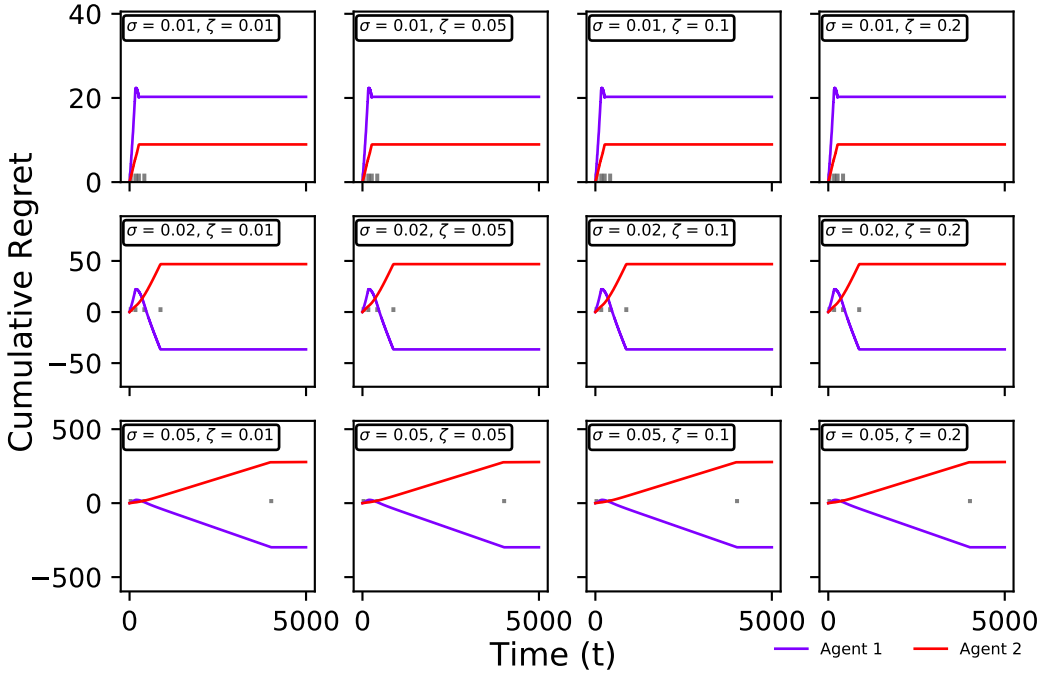

Figure 3.11: Cumulative regret for different noise levels and context variation levels in Scenario S3.

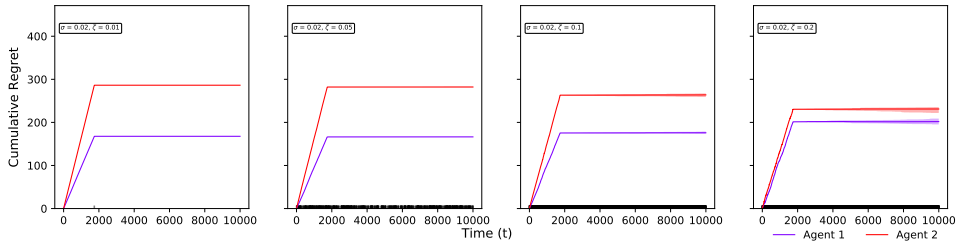

Figure 3.12: Cumulative regret for different context dimensions in Scenario S4.

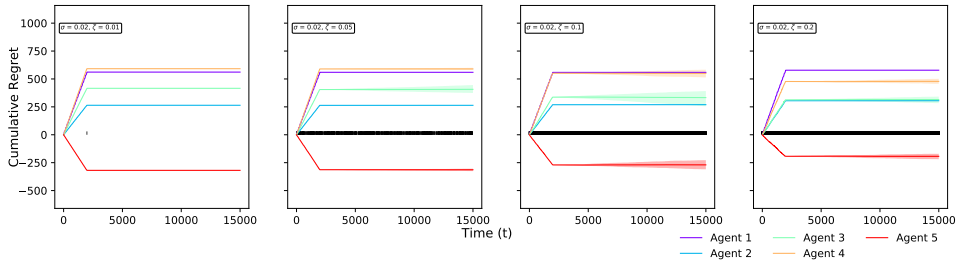

Figure 3.13: Cumulative regret for different number of agents and arms in Scenario S5.

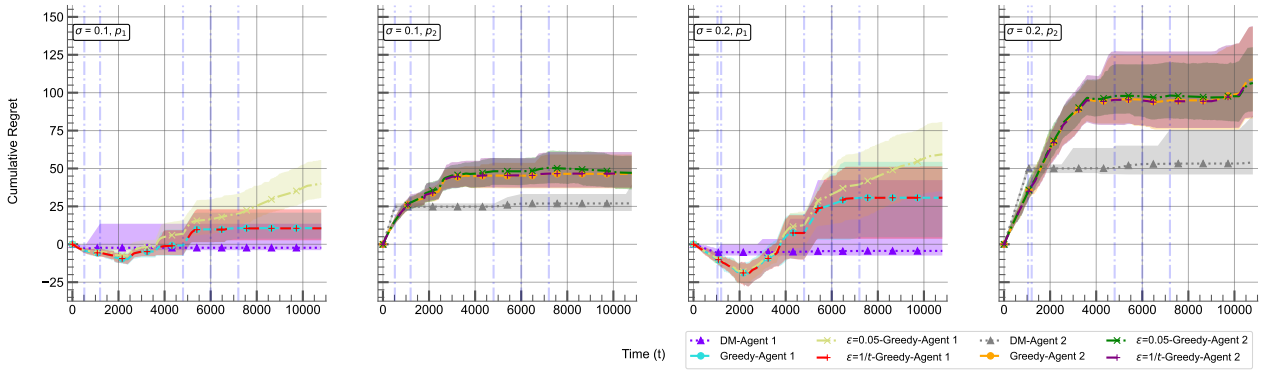

Figure 3.14: Individual regret for agent p_1 and p_2 under noise $\sigma = 0.1$ (Left two) and $\sigma = 0.2$ (Right two) of methods dynamic matching algorithm, greedy, 0.05-greedy, $1/t$-greedy.

3.16.5 Textual Information of job applicants and job description

o the space $\Theta_{i,K}(\gamma, C_0)$, we will develop a clear but nontrivial understanding of the full ranking problem in this paper. $\mathcal{D}_{\mathcal{X}_j}$ and $\mathcal{D}_{\mathcal{X}_k}$ are independent, the variance of the difference between $y_{i,j}$ and $y_{i,k}$ is $2\sigma^2$. With the deign of dynamic matching algorithm, the initial exploration is independent sampling from the $\mathcal{D}_{\mathcal{X}}$. For each agent p_i, it has $h = \sum_{j=1}^{K} A_{i,j}$ total number of observations for $y_{i,j}$ with observation number $A_{i,j}$, so it has $A_{i,(j,k)} = A_{i,j} * A_{i,k}$ observations of $y_{i,(j,k)}$. The joint distribution $\{A_{i,(j,k)}\}$ and $\{y_{i,(j,k)}\}$ under the above generating process, is denoted by $\mathbb{P}((\mu_i, \sigma^2, \mathbf{r}))$.

Candidate Profile	Text
DS profile	research projects on modeling of high-dimensional and multi-modal (partially observed) inputs for classification, regression and clustering tasks, leveraging a wide range of techniques.
DS update info (18)	(1) Strong interested in data science, (2) machine learning, (3) data visualization, (4) data analysis, (5) statistical model, (6) deep learning, (7) natural language processing, (8) coding, (9) options, (10) derivatives, (11) futures, (12) analyze investments, (13) assess risk, (14) assess return profiles, (15) knowledge in math, (16) statistical model, (17) programming python, (18) R.
SDE profile	Experienced Software Engineer working at Cisco, skilled in Go, Java, and C++, (1) Working on APIC (Application Policy Infrastructure Controller) and a virtualization project of CMTS (Cable Modem Terminal System). (2) Working on a Cloud-native system utilizing containerized microservices using Kubernetes, Docker, etc.
SDE update info (18)	(1) Algorithms, (2) data structures, (3) Architecture, (4) Artificial Intelligence, (5) Machine Learning, (6) Compilers, (7) Database, (8) Distributed Systems, (9) Networking, (10) Systems, (11) C, (12) C++, (13) C, (14) Java, (15) JavaScript, (16) go, (17) Python, (18) objective C.
Quant profile	Strong passion in quant finance. Well-coordinated skill sets consisting of math, finance, statistics and programming. Industrial experiences in equity space including both linear and non-linear products. Pricing desk quant covering equity exotic derivatives, hybrid derivatives.
Quant update info (6)	(1) strong math, (2) statistics modeling, (3) Programming, (4) Python, (5) R, (6) economics.

Table 3.1: Job applicants' profile

Job description	Text
Quantitative job description	Strong passion in quant finance, strong mathematical and statistical knowledge. Proficiency in programming languages like Python or R. Data analysis and visualization skills. Understanding of quantitative modeling and statistical methods. Domain-specific knowledge (e.g., finance, economics). know equitable product and derivatives.
SDE job description	Research experience in Algorithms, Architecture, Artificial Intelligence, Compilers, Database, Data Mining, Distributed Systems, Machine Learning, Networking, or Systems. Programming experience in one or more of the following: C/C++, C, Java, JavaScript, Python Objective C, Go, or similar. Experience in computer science, with competencies in data structures, algorithms and software design.

Table 3.2: Job description

CHAPTER 4

Two-sided Competing Matching Recommendation Markets With Quota and Complementary Preferences Constraints

In this project, we propose a new recommendation algorithm for addressing the problem of two-sided matching markets with complementary preferences and quota constraints, where agents' preferences are unknown a priori and must be learned from data. The presence of mixed quota and complementary preferences constraints can lead to instability in the matching process, making this problem challenging to solve. To overcome this challenge, we formulate the problem as a bandit learning framework and propose the Multi-agent Multi-type Thompson Sampling (MMTS) algorithm. The algorithm combines the strengths of Thompson Sampling for exploration with a double matching technique to achieve a stable matching outcome. Our theoretical analysis demonstrates the effectiveness of MMTS as it can achieve stability at every matching step and has a total $\widetilde{\mathcal{O}}(Q\sqrt{K_{\max}T})$-Bayesian regret, which exhibits linearity with respect to the total firm's quota Q and the square root of the maximum size of available type workers $\sqrt{K_{\max}}$.

4.1 Introduction

Two-sided matching markets with recommendation have been a mainstay of theoretical research and real-world applications for several decades since the seminal work by (GS62). Matching markets are used to allocate indivisible "goods" to multiple decision-making agents

based on mutual compatibility as assessed via sets of preferences. Preferences are usually unknown in the recommendation process due to large volume of participants and hard to be explicit. Besides, matching markets embody a notion of scarcity in which the resources on both sides of the market are limited. One of the key concepts that contribute to the success of matching markets is *stability*, which criterion ensures that all participants have no incentive to block a prescribed matching (Rot82). Matching markets often consist of participants with *complementary* preferences that can lead to instability (CKK19). Examples of complementary preferences in matching markets include: firms seeking workers with skills that complement their existing workforce, sports teams forming teams with players that have complementary roles, and colleges admitting students with diverse backgrounds and demographics that complement each other. Studying the stability issue in the context of complementary preferences is crucial in ensuring the successful functioning of matching markets with complementarities.

In this paper, we propose a novel algorithm and present an in-depth analysis of the problem of complementary preferences in matching markets. Specifically, we focus on a many-to-one matching scenario and use the job market as the example. In our proposed model, there are a set of agents (e.g., firms), each with limited quota, and a set of arms (e.g., workers), each of which can be matched to at most one agent. Each arm belongs to a unique type, and each agent wants to match with a minimum quota of arms from each type. This leads to complementarities in agents' preferences. Additionally, the agents' preference of arms from each type is unknown a priori and must be learned from data, which we refer to as the *competing matching under complementary preference recommendation problem* (CMCPR).

Our first result is the formulation of CMCPR into a bandit learning framework as described in (LS20). Using this framework, we propose a new algorithm, the Multi-agent Multi-type Thompson Sampling (MMTS), to solve CMCPR. Our algorithm builds on the strengths of Thompson Sampling (TS) (Tho33; AG12; RVK18) in terms of exploration and further enhances it by incorporating a *double matching* technique to find a stable solution for

CMCPR, illustrated in Section 4.5.2. Unlike the upper confidence bound (UCB) algorithm, TS method can achieve sufficient exploration by incorporating a deterministic, non-negative bias inversely proportional to the number of matches into the observed empirical means. Furthermore, the double matching technique proposed in this paper uses two stages of matching to satisfy both the type quota and total quota requirements. These two stages mainly consist of using the deferred-acceptance (DA) algorithm from (GS62).

Second, we provide the theoretical analysis of the proposed MMTS algorithm. Our analysis shows that MMTS can achieve stability at each matching step and show the incentive compatibility (IC) of the MMTS. The proof of stability is obtained through a two-stage design of the *double matching* technique, and the proof of IC is obtained through the regret lower bound. To the best of our knowledge, MMTS is the first algorithm to achieve stability and IC in the CMCPR.

Finally, our theoretical results indicate that MMTS achieves a Bayesian regret that scales $\widetilde{\mathcal{O}}(\sqrt{T})$ and is near linear in terms of total quota of all firms (Q). Besides, we find that the Bayesian regret only depends on the square root of the *maximum* number of workers ($K_{\max}$) in one type rather than the square root of the total number of workers ($\sum_m K_m$) in all types, which is important for the large market. This is a more challenging setting than that considered in previous works such as (LMJ20; JWW21), which only consider a single type of worker in the market and a quota of one for each firm. To address these challenges, we use the eluder dimension (RV13) to measure the uncertainty set widths and bound the instantaneous regret for each firm, and use the concentration results to measure the probability of *bad events* occurring to get the final regret. Bounding the uncertainty set width is the key step for deriving the regret upper bound of MMTS.

The rest of this paper is organized as follows. Section 4.2 discussed related works. Section 4.3 introduces seven elements of CMCPR. Section 4.4 states the challenges of this problem. Section 4.5 provides MMTS algorithm, its comparison with UCB-family algorithms, and show the incapable exploration of UCB algorithm in CMCPR. Then we present the stability,

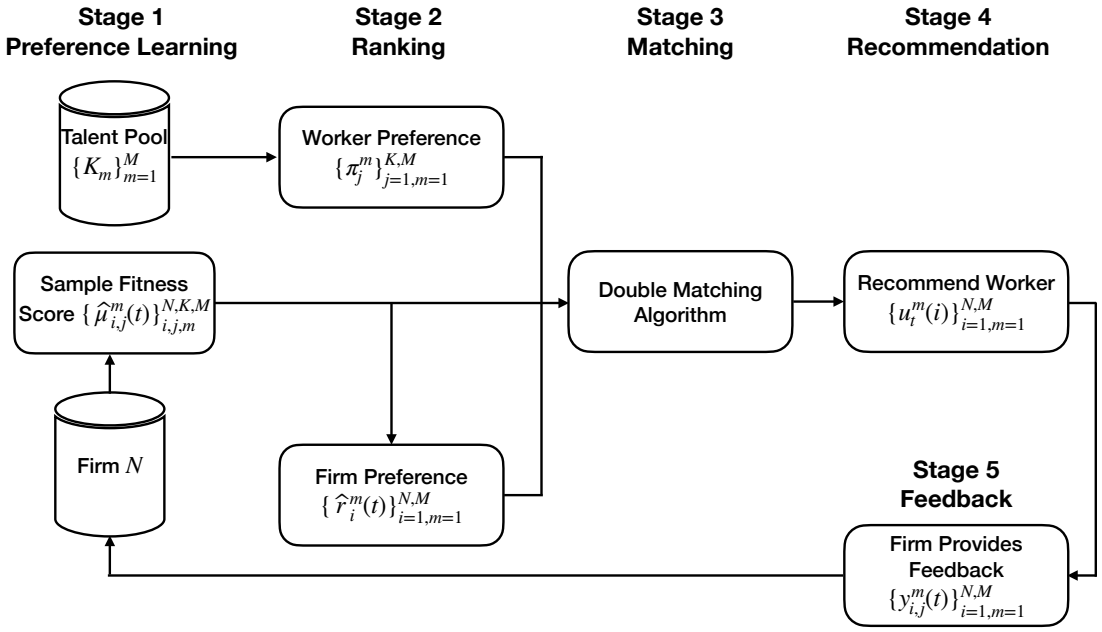

Figure 4.1: MMTS Algorithm for CMCPR with its application in the job market, including five stages: *preference learning*, *ranking construction*, *matching*, *recommendation*, *feedback collection*.

regret upper bound, and the incentive-compatibility of MMTS in Section 4.6. Finally, Section 4.7 shows the application of MMTS in simulations including the distribution of learning parameters, and demonstrating the robustness of MMTS in large markets.

4.2 Related Works

This section reviews two-sided matching market with unknown preferences, multi-agent systems, assortment optimization, and matching markets.

Multi-Agent Systems and Game theory. There are some papers considering the multi-agent in the sequential decision-making systems including the cooperative setting (Lit01; GH13; ZYL18; PPP18; SWS22) and competing setting (Lit94; AO06; ZJB07; WHL17; FCR19; JNJ20). (ZYW21) study the multi-player general-sum Markov games with one of the players designated as the leader and the other players regarded as followers and establish

the efficient RL algorithms to achieve the Stackelberg-Nash equilibrium.

Assortment Optimization. To maximize the number of matches between the two sides (customers and suppliers), the platform must balance the inherent tension between recommending customers more potential suppliers to match with and avoiding potential collisions. (AKM22) introduce a stylized model to study the above trade-off. Motivated by online labor markets (AS22) consider the online assortment optimization problem faced by a two-sided matching platform that hosts a set of suppliers waiting to match with a customer. (ILM21) consider a two-sided matching assortment optimization under the continuum model and achieve the optimized meeting rates and maximize the equilibrium social welfare. (RSZ22) discuss the application of assortment optimization in dating markets. (Shi22) studies the minimum communication needed for a two-sided marketplace to reach an approximately stable outcome with the transaction price.

Two-sided Matching Markets with Known Preferences. One strand of related literature is two-sided matching, which is a stream of papers that started in (GS62). They propose the DA algorithm with its application in the marriage problem and college admission problem. A series work (Knu76; Rot82; RS92; Rot08) discuss the history of the DA algorithm and summarize theories about stability, optimality, and incentive compatibility, and finally provide its practical use and further open questions. In particular, (Rot85; Son97) propose that the college admissions problem is not equivalent to the marriage problem, especially when a college can manipulate its capacity and preference. Notably, in the hospital doctor matching example, since hospitals want diversity of specializations, or demographic diversity, or whatever, they care about the combination (group of doctors) they get. (Rot86) state that when all preferences are strict, and hospitals (firms) have responsive preferences, the set of doctors (workers) employed and positions filled is the same at every stable match. However, when there exist *couples* in the preference list (not *responsive preference* (KK05)), which might make the set of stable matchings empty. Even when stable matchings exist, there need not be an optimal stable matching for either side. Later, (ABH11) revisit this

couple matching problem and provide the *sorted deferred acceptance algorithm* that can find a stable matching with high probability in large random markets. (BMM14) provide an integer programming model for hospital/resident problems with couples (HRC) and ties (HRCT). (MMT17) release the HRC with minimal blocking pairs and show that if the preference list of every single resident and hospital is of length at most 2, their method can find a polynomial-time algorithm. (NV18; NV22) find the stable matching in the nearby NRC problem, which is that the quota constraints are soft. (AH18; CKK19; GK21) discuss the existence and uniqueness of stable matching with complementaries and its relationship with substitutable preferences in large economies. Besides, there are also papers considering stability and optimality of the refugee allocation matching (ACG18; HT22). (Tom18; BH22) consider a case that firms have hard constraints both on the minimum and maximum type-specific quotas and other type-specific quota consideration works.

Two-sided Matching Market with Unknown Preferences. (LMJ20) considers the multi-agent multi-armed competing problem in the centralized platform with explore-then-commit (ETC) and upper confidence bound (UCB) style algorithms where preferences from agents to arms are unknown and need to be learned through streaming interactive data. (JWW21) considers the two-sided matching problem where preferences from both sides are defined through dynamic utilities rather than fixed preferences and provide regret upper bounds over different contexts settings, and (MWX22) apply it to the Markov matching market. (CS22) show that if there is transfer between agents, then the three desiderata (stability, low regret, and fairness) can be simultaneously achieved. (LWC22) discuss the two-sided matching problem when the arm side has dynamic contextual information and preference is fixed from the arm side and propose a centralized contextual ETC algorithm to obtain the near-optimal regret bound. Besides, there are a plethora of works discussing the two-sided matching problem in the decentralized markets (LRM21; BSS21; SBS21; DJ21a; DJ21b; DQJ22). In particular, (DJ21b) study the college admission problem and provides an optimal strategy for agents, and shows its incentive-compatible property. Moreover, (JJH22)

explores the phenomenon of the two-sided matching problem with two competing markets.

4.3 Problem

We now describe the problem formulation of the **C**ompeting **M**atching under **C**omplementary **P**references **R**ecommendation problem (CMCPR).

Notations. We define T as the time horizon and assume it is known[1]. We denote $[N] = [1, 2, ..., N]$ where $N \in \mathbb{N}^+$. Define the bold $\mathbf{x} \in \mathbb{R}^d$ be a d-dimensional random vector.

4.3.1 Environment

The matching of workers and firms will be our running example throughout the paper. The organizer is the centralized platform and the overall goal of the platform is to recommend the best fit worker and match two-sided participants with their ideal objects over time. We first introduce seven elements in CMCPR.

(I) Participants. In the centralized, there are N firms (agents), denoted by $\mathcal{N} = \{p_1, p_2, ..., p_N\}$, and various types of workers (arms), represented $\mathcal{K}_m = \{a_1^m, a_2^m, ...a_{K_m}^m\}, \forall m \in [M]$, where K_m is the number of m-th type workers and M types in total.

(II) Quota. p_i has a minimum quota q_i^m for m-type workers, and a maximum total quota Q_i (e.g., seasonal headcount in company) and we assume $\sum_{i=1}^{M} q_i^m \leq Q_i$. Define the total market quota as $Q = \sum_{i=1}^{N} Q_i$ and the total number of workers as $K = \sum_{m=1}^{M} K_m$. And we assume that $Q \ll K$ and T is large.

(III) Two-sided Complementary Preferences. There are two kinds of preferences: workers to firms' preferences, firms to workers' preferences.

a. Preferences of m-type workers towards firms $\boldsymbol{\pi}^m : \mathcal{K}_m \mapsto \mathcal{N}$. We assume that there exists fixed preferences from workers to firms, and these preferences are known for the plat-

[1]The unknown T can be handled with the doubling trick (ACF95).

form. For instance, workers submit their preferences for different firms on the platform. $\pi^m_{j,i}$ represents the rank for p_i from the view of a^m_j, and we assume that there are no ties in the rank orders, $\boldsymbol{\pi}^m_j \subseteq \{\pi^m_{j,1}, ..., \pi^m_{j,N}\}$. In other words, $\boldsymbol{\pi}^m_j$ is a subset of the permutation of $[N]$. And $\pi^m_{j,i} < \pi^m_{j,i'}$ implies that $m-$type worker a^m_j prefers firm p_i over firm $p_{i'}$ and as a shorthand, denoted as $p_i <^m_j p_{i'}$. This known worker-to-firm preference is a mild and common assumption in matching market literature (LMJ20; LRM21; LWC22).

b. Preferences of firms towards $m-$type workers $\mathbf{r}^m : \mathcal{N} \mapsto \mathcal{K}_m$. The true *unknown* preferences of firms towards workers are fixed, but unknown. The goal of the platform is to infer these unknown preferences through historical matching data. We denote $r^m_{i,j}$ as the true rank of worker a^m_j in the preference list of firm p_i, and assume there are no ties. p_i's preferences towards workers is represented by $\mathbf{r}^m_i$, which is a subset of the permutation of $[\mathcal{K}_m]$. $r^m_{i,j} < r^m_{i,j'}$ implies that firm p_i prefers worker a^m_j over worker $a^m_{j'}$.

4.3.2 Policy

(IV) Matching Policy. $u^m_t(p_i) : \mathcal{N} \mapsto \mathcal{K}_m$ is a recommendation mapping function from p_i to $m-$type workers at time t.

(V) Stable Matching and Optimal Matching. We introduce key concepts in matching fields (Rot08).

Definition 4.1. (Blocking pair). A matching u is blocked by firm p_i if p_i prefers being single to being matched with $u(p_i)$, i.e. $p_i >_i u(p_i)$. A matching u is blocked by a pair of firm and worker (p_i, a_j) if they each prefer each other to the partner they receive at u, i.e. $a_j >_i u(p_i)$ and $p_i >_j u^{-1}(a_j)$.

Definition 4.2. (Stable Matching). A matching u is stable if it isn't blocked by any individual or pair of worker and firm.

Definition 4.3 (Valid Match)**.** With true preferences from both sides, arm a_j is called a *valid match* of agent p_i if there exist a stable matching according to those rankings such that

a_i and p_j are matched.

Definition 4.4 (Agent Optimal Match)**.** Arm a_j is an *optimal match* of agent p_i if it is the most preferred valid match.

Given two-sided true preferences, the DA algorithm (GS62) will provide a stable matching. The matching result by the DA algorithm is always optimal for members of the proposing side and denote the agent-optimal policy as $\{\overline{u}_i^m\}_{m=1}^M$ for CMCPR.

In CMCPR, however, each firm has a minimum quota constraint $\mathbf{q}_i = [q_i^1, ..., q_i^M]$ for all type workers to fill. Therefore, we define the concept of stability as the absence of "blocking pairs" across all types of workers and firms. Based on the definition of stable matching, we discussed the feasibility of the stable matching in CMCPR in Appendix 4.9.1.

(VI) Matching Score. If p_i is matched with a_j^m at time t, p_i provides a noisy reward $y_{i,j}^m(t)$ which is sampled from the Bernoulli distribution with the *true matching score* $\mu_{i,j}^m(t)$,

$$y_{i,j}^m(t) \sim \text{Ber}(\mu_{i,j}^m(t)), \tag{4.1}$$

$\forall i, j, m, t \in [N], [K_m], [M], [T]$, where we know the noise follows the sub-Gaussian random variable with parameter $\sigma = 1/2$. That is, for every $\alpha \in \mathbb{R}$, it is satisfied that $\mathbb{E}[\exp(\alpha\epsilon_{i,j}^m(t))] \leq \exp(\alpha^2\sigma^2/2)$.

(VII) Regret. Based on model (4.1), we denote the matching score for p_i as $\mathbf{y}_i^m(t) := \mathbf{y}_{i,u_t^m(p_i)}(t)$. Define the *firm-optimal regret with m-type worker* for p_i as

$$R_i^m(T, \theta) := \sum_{t=1}^{T} [\mu_{i,\overline{u}_i^m} - \mu_{i,u_t^m(p_i)}(t)], \tag{4.2}$$

where denote θ as the sampled problem instance from the distribution Θ. $R_i^m(T, \theta)$ represents the total expected score difference between the policy $u_i^m := \{u_t^m(p_i)\}_{t=1}^T$ and the optimal policy $\overline{u}_i^m$ in hindsight.

As each firm have to recruit M types workers with total quota Q_i, the *total firm-optimal stable regret* for p_i is defined as

$$R_i(T, \theta) := \sum_{m=1}^{M} R_i^m(T, \theta). \tag{4.3}$$

Finally, define the *Bayesian Social Welfare Gap* (BSWG) $\mathfrak{R}(T)$ as the expected regret over all firms and problem instance,

$$\mathfrak{R}(T) := \mathbb{E}_{\theta \in \Theta} \left[\sum_{i=1}^{N} R_i(T, \theta) \right]. \tag{4.4}$$

The goal of the centralized platform is to design a learning algorithm that achieves stable matchings through learning the firms' complementary preferences for multiple types of workers preciously from the previous matchings for better recommendation. This is equivalent to design an algorithm that minimizes BSWG $\mathfrak{R}(T)$.

4.4 Challenges and Solutions

When preferences are unknown a priori in matching markets, the stability issue while satisfying complementary preferences and quota requirements is a challenging problem due to the interplay of multiple factors.

4.4.1 Challenge 1: How to design a stable matching algorithm to solve complementary preferences?

This is a prevalent issue in real-world applications such as hiring workers with complementary skills in hospitals and high-tech firms or admitting students with diverse backgrounds in college admissions. Despite its importance, no implementable algorithm is currently available to solve this challenge. In this paper, we propose a novel approach to resolving this issue

by utilizing a *double matching* (Algorithm 7) to marginalize complementary preferences and achieve stability. Our algorithm can efficiently learn a stable matching using historical matching data, providing a practical solution to CMCPR.

4.4.2 Challenge 2: How to balance the exploration and exploitation to achieve the sublinear regret?

The platform must find a way to recommend the most fit workers to firms to establish the credibility among workers and firms to stay at the platform, towards achieving optimal matching. Compared to traditional matching algorithms, the CMCPR is not an one-time recommendation algorithm but a *recycled* recommendation matching algorithm with supply and demand consideration (workers and firms), which is more challenging as it requires more time to balance this trade-off. In addition, the classic UCB bandit methods could function well in exploration and suffer sublinear regret demonstrated in Section 4.5.2. To overcome this challenge, we propose the use of sampling algorithm which allows for better exploration and achieves sublinear regret.

4.4.3 Challenge 3: How to solving CMCPR with quota constraints in large markets?

Unlike the classic DA algorithm (GS62), our problem involves type-specific and quota requirements for each firm. Can we find a stable matching algorithm that satisfies these constraints while also adapting to unknown preferences? Furthermore, can this algorithm be applied in large markets with efficiency? We address these challenges by proposing a novel algorithm, MMTS, that effectively balances exploration and exploitation while can also be partially parallel implemented.

Algorithm 5: Multi-agent Multi-type Thompson Sampling Algorithm(MMTS)

Input : Time horizon T; firms' priors $(\boldsymbol{\alpha}_i^{m,0}, \boldsymbol{\beta}_i^{m,0}), \forall i, m \in [N], [M]$; workers' preference $\boldsymbol{\pi}^m, \forall m \in [M]$.

for $t \in \{1, ..., T\}$ **do**

- **Step 1: preference learning Stage**
  - Sample estimated mean reward $\widehat{\boldsymbol{\mu}}_i^m(t)$ over all types of workers (Algo. 6)
- **Step 2: Ranking Construction Stage**
  - Construct all firms' estimated rankings $\{\widehat{\mathbf{r}}_i^m(t)\}_{i=1,m=1}^{N,M}$ according $\widehat{\boldsymbol{\mu}}_i^m(t)$.
- **Step 3: Double Matching Stage**
  - Get the matching result $\mathbf{u}_t^m(p_i), \forall i \in [N], m \in [M]$ from the *double matching* in Algo 7.
- **Step 4: Collecting Feedback Stage**
  - Each firm receives its corresponding rewards from all types of workers $\mathbf{y}_i^m(t)$.
- **Step 5: Updating Belief Stage**
  - Based on received rewards, the platform updates firms' posterior belief.

4.5 MMTS Algorithm

In this section, we propose the Multi-agent Multi-type Thompson Sampling algorithm (MMTS), which aims to learn the true preferences of all firms over all types of workers, achieve stable matchings, and minimize firms' Bayesian regret. We provide a description of MMTS and demonstrate its benefits of using sampling method. The overall MMTS algorithm procedure is in Figure 4.1. The computational complexity of MMTS is in Appendix 4.9.2.

4.5.1 Algorithm Description - 3 Stages

The MMTS (Algorithm 5) is composed of five stages, *preference learning stage*, *ranking construction stage*, *double matching stage*, *collecting feedback stage*, and *updating belief stage*. At each matching step t, MMTS iterates these four steps.

Step 1: Preference Learning Stage. (Algorithm 6). For p_i, platform samples the mean feedback (reward) $\widehat{\mu}_{i,j}^m(t)$ of a_j^m from distribution $\mathcal{P}_j^m$ with estimated parameters $(\alpha_{i,j}^{m,t-1}, \beta_{i,j}^{m,t-1})$ learned from the historical matching data.

Step 2: Ranking Construction Stage. Then the platform sorts these workers within each type according $\{\widehat{\mu}_{i,j}^m(t)\}_{j=1}^{K_m}$ in descending order and gets the estimated rank $\widehat{\mathbf{r}}^m(t) = \{\widehat{\mathbf{r}}_i^m(t)\}_{i=1,m=1}^{N,M}$ where $\widehat{\mathbf{r}}_i^m(t) = \{\widehat{\mathbf{r}}_{i,j}^m(t)\}_{j=1}^{K_m}$.

Step 3: Double Matching Stage. (Algorithm 7). With sampled mean reward $\widehat{\boldsymbol{\mu}}(t) := \{\widehat{\mu}_{i,j}^m(t)\}_{i=1,j=1,m=1}^{N,K_m,M}$, estimated ranks $\widehat{\mathbf{r}}(t)$, quota constraints $\{Q_i\}_{i=1}^N$, the double matching algorithm provides the final matching result with two-stage matchings.

The goal of the first match is to allow all firms to satisfy their minimum type-specific quota q_i^m first followed by sanitizing the status quo as a priori. The second match is to fill the left-over positions $\widetilde{Q}_i$ (defined below) for each firm and match firms and workers without type consideration.

3.1. First Match: The platform implements the type-specific DA (Algo. 8) given quota constraints $\{q_i^m\}_{i=1,m=1}^{N,M}$. The matching road map starts from matching all firms with type from 1 to M and returns the matching result $\{\widetilde{u}_t^m(p_i)\}_{m\in[M]}$. This step can be implemented in parallel.

3.2. Sanitize Quota: After the first match, the centralized platform sanitizes each firm's left-over quota $\widetilde{Q}_i = Q_i - \sum_{m=1}^M q_i^m$. If there exists a firm $p_i, s.t., \widetilde{Q}_i > 0$, then the platform will step into the second match. For those firms like p_i whose leftover quota is zero $\widetilde{Q}_i = 0$, they and their matched workers will skip the second match.

3.3. Second Match: When rest firms and workers continue to join in the second match, the centralized platform implements the standard DA in Algorithm 9 without type consideration. That is, the platform re-ranks the rest M types of workers who do not have a match in the first match for firms, and fill available vacant positions. It is worth noting that in Algorithm 9, each firm will not propose to the previous workers who rejected him/her already or matched in Step 1. Then firm p_i gets the corresponding matched workers $\breve{u}_t(p_i)$

Algorithm 6: Preference Learning Stage

Input : Time horizon T; firms' priors $(\boldsymbol{\alpha}_i^{m,0}, \boldsymbol{\beta}_i^{m,0}), \forall i \in [N], \forall m \in [M]$.

Sample: Sample mean reward $\widehat{\mu}_{i,j}^m(t) \sim \mathcal{P}(\alpha_{i,j}^{m,t-1}, \beta_{i,j}^{m,t-1})$, $\forall i, m, j \in [N], [M], [\mathcal{K}_m]$.

Sort: Sort estimated mean feedback $\widehat{\mu}_{i,j}^m(t)$ in descending order and get the estimated rank $\widehat{\mathbf{r}}_i^m(t)$.

Output: The estimated rank $\widehat{\mathbf{r}}_i^m(t)$ and the estimated mean feedback $\widehat{\boldsymbol{\mu}}_i^m(t)$, $\forall i, m \in [N], [M]$.

Algorithm 7: Double Matching

Input : Estimated rank $\widehat{\mathbf{r}}(t)$, estimated mean $\widehat{\boldsymbol{\mu}}_i^m(t)$, type quota $q_i^m, \forall m \in [M], i \in [N]$ and total quota $Q_i, \forall i \in [N]$; workers' preference $\{\boldsymbol{\pi}^m\}_{m\in[M]}$.

Step 1: First Match

Given estimated ranks $\widehat{\mathbf{r}}(t)$ and all workers' preferences $\boldsymbol{\pi}^m$, the platform operate the firm-propose DA Algo and return the matching $\{\widetilde{u}_t^m(p_i)\}_{i=1,m}^{N,M}$.

Step 2: Sanitize Quota

Sanitize whether all firms' positions have been filled. For each company p_i, if $Q_i - \sum_{m=1}^M q_i^m > 0$, set the left quota as $\widetilde{Q}_i \leftarrow Q_i - \sum_{m=1}^M q_i^m$ for firm p_i.

Step 3: Second Match

if $\widetilde{\mathbf{Q}} \neq 0$ **then**

Given left quota $\{\widetilde{Q}_i\}_{i\in[N]}$, estimated means $\widehat{\boldsymbol{\mu}}(t)$, and workers' preferences $\{\boldsymbol{\pi}^m\}_{m\in[M]}$, the platform runs the firm-propose DA and return the matching $\breve{u}_t(p_i)$.

else

Set the matching $\breve{u}_t(p_i) = \emptyset$.

Output: The matching $u_t^m(p_i) \leftarrow \text{Merge}(\widetilde{u}_t^m(p_i), \breve{u}_t(p_i))$ for all firms.

in the second match. Finally, the platform merges the first and second results to obtain a final matching $\mathbf{u}_t^m(p_i) = \text{Merge}(\widetilde{u}_t^m(p_i), \breve{u}_t(p_i)), \forall i, m \in [N], [M]$.

Step 4: Collecting Feedback Stage. When the platform broadcasts the matching result $\mathbf{u}_t^m(p_i)$ to all firms, each firm then receives its corresponding stochastic reward $\mathbf{y}_i^m(t), \forall i \in [N], m \in [M]$.

Step 5: Updating Belief Stage. After receiving these noisy rewards, the platform updates firms' belief (posterior) parameters as follows, $(\boldsymbol{\alpha}_i^{m,t}, \boldsymbol{\beta}_i^{m,t}) = \text{Update}(\boldsymbol{\alpha}_i^{m,t-1}, \boldsymbol{\beta}_i^{m,t-1}, \mathbf{y}_i^m(t)), \forall i \in [N], \forall m \in [M]$.

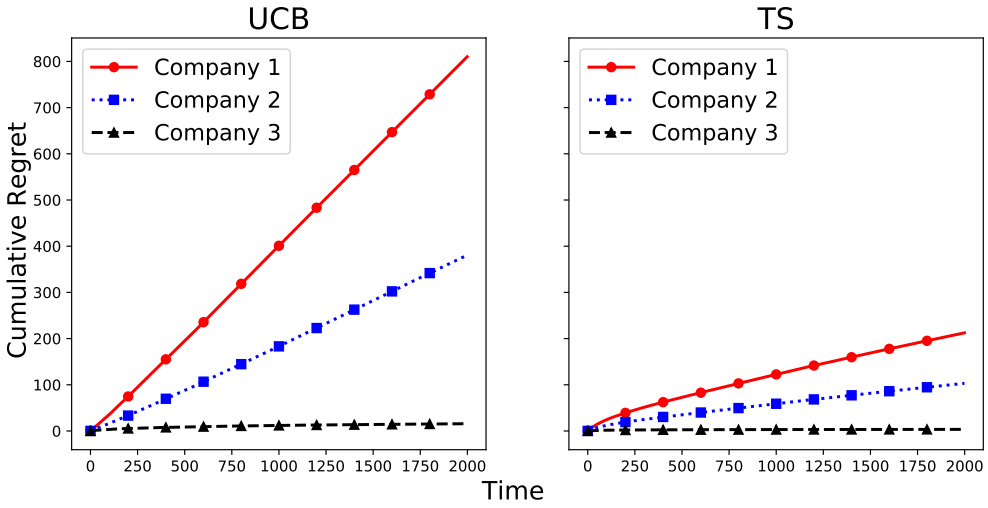

Figure 4.2: A comparison of centralized UCB and TS. A demonstrate of the incapable exploration of UCB.

4.5.2 Incapable Exploration

We show why the sampling method has an advantage over the UCB style method in estimating the ranks of workers. We find that centralized UCB suffers linear firm-optimal stable regret in some cases and show it in Appendix 4.9.3 with detailed experimental setting and analysis.

Why sampling method is capable of avoiding the curse of linear regret? By the property of sampling shown in Algorithm 6. Firm p_i's initial prior over worker a_i is a uniform random variable, and thus $r_j(t) > r_i(t)$ with probability $\widehat{\mu}_j \approx \mu_j$, rather than *zero*! This differs from the UCB style method, which cannot update a_i's upper bound due to lacking exploration over a_i. The benefit of TS is that it can occasionally explore different ranking patterns, especially when there exists such a previous example. In Figure 4.2, we show a quick comparison of centralized UCB (LMJ20) in the settings shown above and MMTS when $M = 1, Q = 1, N = 3, K = 3$. The UCB method occurs a linear regret for firm 1 and firm 2. However, TS method achieves a sublinear regret in firm 1 and firm 2.

4.6 Properties of MMTS: Matching Stability, Bayesian Regret Upper Bound, and Incentive Compatible

Section 4.6.1 demonstrates the double matching algorithm can provide the stability property for CMCPR. Section 4.6.2 establishes the Bayesian regret upper bound for all firms when they follow the MMTS. Section 4.6.3 discusses the incentive-compatibility property of the MMTS.

4.6.1 Matching Stability

In the following theorem, we show the double matching algorithm (Algo.7) provides stable matching solution in the following theorem.

Theorem 4.1. *Given two sides' preferences from firms and M types of workers. The double-matching procedure can provide a firm-optimal stable matching solution $\forall t \in [T]$.*

Proof. The sketch proof of the stability property of MMTS is two steps, naturally following the design of MMTS. The first match is conducted in parallel, and the output is stable and guaranteed by (GS62). As the need of MMTS, before the second match, firms without leftover quotas ($\tilde{Q} = 0$) will quit the second round of matching, which will not affect the stability. After the quota sanitizing stage, firms and leftover workers will continue to join in the second matching stage, where firms do no need to consider the type of workers designed by double matching. And the DA algorithm still provides a stable result based on each firm's *sub-preference* list. The reason is that for firm p_i, all previous possible favorite workers have been proposed in the first match. If they are matched in the first match, they quit together, which won't affect the stability property; otherwise, the worker has a better candidate (firm) and has already rejected the firm p_i. So for each firm p_i, it only needs to consider a sub-preference list excluding the already matched workers in the first match and the proposed workers in the first match. It will provide a stable match in the second match and won't

be affected by the first match. So the overall double matching is a stable algorithm. The detailed proof can be found in Appendix Section 4.9.5. □

4.6.2 Bayesian Regret Upper Bound

Next we provide MMTS's Bayesian total firm-optimal regret upper bound.

Theorem 4.2. *Assume* $K_{\max} = \max\{K_1, ..., K_M\}, K = \sum_{m=1}^{M} K_m$, *with probability* $1 - 1/QT$, *when all firms follow the MMTS algorithm, firms together will suffer the Bayesian expected regret*

$$\mathfrak{R}(T) \leq 8Q \log(QT)\sqrt{K_{\max} T} + NK/Q.$$

Proof. The detailed proof can be found in Appendix 4.9.6. □

The derived Bayesian regret bound, which is dependent on the square root of the time horizon T and a logarithmic term, is nearly rate-optimal. Additionally, we examine the dependence of this regret bound on other key parameters. The first of which is a near-linear dependency on the total quota Q. Secondly, the regret bound is dependent only on the *square root* of the maximum worker $K_{\max}$ of one type, as opposed to the total number of workers, $\sum_{m=1}^{M} K_m$ in previous literature (LMJ20; JWW21). This highlights the ability of our proposed algorithm, MMTS, to effectively capture the interactions of multiple types of matching in CMCPR. The second term in the regret is a constant which is only dependent on constants N, K and the total quota Q. Notably, if we assume that each $q_i = 1$ and $Q_i = M$, then NK/Q will be reduced to $NK/(NM) = K/M$, which is an unavoidable regret term due to the exploration in bandits (LS20). This also demonstrates that the Bayesian total cumulative firm-optimal exploration regret is only dependent on the *average* number of workers of each type available in the market, as opposed to the *total* number of workers or the maximum number of workers available of all types. Additionally, if one Q_i is dominant over other firms' Q_i, then the regret will mainly be determined by that dominant quota Q_i and $K_{\max}$, highlighting the inter-dependence of this complementary matching problem.

4.6.3 Incentive-Compatibility

In this section, we discuss the incentive-compatibility property of MMTS. That is, if one firm does not match the worker that MMTS (platform) recommended when all other firms follow MMTS recommended matching objects which is equivalent to that firm submits a ranking preferences different from the sampled ranking list from MMTS, and we know that firm cannot benefit (matched with a better worker than his optimal stable matching worker) over a sublinear order. As we know, (DF81) discussed the *Machiavelli* firm could not benefit from incorrectly stating their true preference when there exists a unique stable matching. However, when one side's preferences are unknown and need to be learned through data, this result no longer holds. Thus, the maximum benefits that can be gained by the Machiavelli firm are under-explored in the setting of learning in matching. (LMJ20) discussed the benefits that can be obtained by Machiavelli firm when other firms follow the centralized-UCB algorithm with the problem setting of one type of worker and quota equal one in the market.

We now show in CMCPR, when all firms except one p_i accept their MMTS recommended workers from the matching platform, the firm p_i has an incentive also to follow the sampling rankings in a *long horizon*, so long as the matching result do not have multiple stable solutions. Now we establish the following lemma, which is an upper bound of the expected number of pulls that a firm p_i can match with a m-type worker that is better than their optimal m-type workers, regardless of what workers they want to match.

Let's use $\mathcal{H}_{i,l}^m$ to define the achievable *sub-matching* set of u^m when all firms follow the MMTS, which represents firm p_i and $m-$type worker a_l^m is matched such that $a_l^m \in u_i^m$. Let $\Upsilon_{u^m}(T)$ be the number of times sub-matching u^m is played by time t. We also provide the blocking triplet in a matching definition as follows.

Definition 4.5. (Blocking triplet) A blocking triplet $(p_i, a_k, a_{k'})$ for a matching u is that there must exist a firm p_i and worker a_j that they both prefer to match with each other than their current match. That is, if $a_{k'} \in u_i$, $\mu_{i,k'} < \mu_{i,k}$ and worker a_k is either unmatched or

$\pi_{k,i} < \pi_{k,u^{-1}(k)}$.

The following lemma presents the upper bound of the number of matching times of p_i and a_l^m by time T, where a_l^m is a *super optimal* $m-$type worker (preferred than all stable optimal $m-$type workers under true preferences), when all firms follow the MMTS.

Lemma 4.1. *Let $\Upsilon_{i,l}^m(T)$ be the number of times a firm p_i matched with a m-type worker such that the mean reward of a_l^m for firm p_i is greater than p_i's optimal match $\overline{u}_i^m$, which is $\mu_{i,a_l^m}^m > \max_{a_j^m \in \overline{u}_i^m} \mu_{i,j}^m$. Then the expected number of matches between p_i and a_l^m is upper bounded by*

$$\mathbb{E}[\Upsilon_{i,l}^m(T)] \leq \min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \sum_{(p_j, a_k^m, a_{k'}^m) \in S^m} \Big(C_{i,j,k'}^m(T) + \frac{\log(T)}{d(\mu_{j,\overline{u}_{i,\min}^m}, \mu_{j,k'})} \Big), \tag{4.5}$$

where $\overline{u}_{i,\min}^m = \underset{a_k^m \in \overline{u}_j^m}{\operatorname{argmin}} \mu_{i,k}^m$, and $C_{i,j,k'}^m = \mathcal{O}((\log(T))^{-1/3})$.

Then we show the benefit (lower bound of the regret) of Machiavelli firm p_i can gain by not following the MMTS recommended workers. Let's define the *super reward gap* as $\overline{\Delta}_{i,l}^m = \max_{a_j^m \in \overline{u}_i^m} \mu_{i,j}^m - \mu_{i,l}^m$, where $a_l^m \notin \overline{u}_i^m$.

Theorem 4.3. *Suppose all firms other than firm p_i follow the preferences according to the MMTS to the centralized platform. Then the following upper bound on firm p_i's optimal regret for m-type workers holds:*

$$R_i^m(T, \theta) \geq \sum_{l: \overline{\Delta}_{i,l}^m < 0} \overline{\Delta}_{i,l}^m \left[\min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \sum_{(p_j, a_k^m, a_{k'}^m) \in S^m} \left(C_{i,j,k'}^m + \frac{\log(T)}{d(\mu_{j,\overline{u}_{i,\min}^m}, \mu_{j,k'})} \right) \right] \tag{4.6}$$

where $\overline{u}_{i,\min}^m = \underset{a_k^m \in \overline{u}_j^m}{\operatorname{argmin}} \mu_{i,k}^m$, and $C_{i,j,k'}^m = \mathcal{O}((\log(T))^{-1/3})$.

This result can be directly derived from Lemma 4.1. Theorem 4.3 demonstrates that there is no sequence of preferences that a firm can manipulate and does not follow MMTS recommended workers that would achieve negative optimal regret and its absolute value greater than $\mathcal{O}(\log T)$. Considering M types together for firm p_i, this magnitude remains $\mathcal{O}(M \log T)$.

Theorem 4.3 confirms that, when there is a unique stable matching, firms cannot gain significant advantage in terms of firm-optimal stable regret due to incorrect estimated preferences if others follow MMTS.

An example is provided in Section 4.7.1 to illustrate this incentive compatibility property. Figure 4.3(a) illustrates the total regret, with solid lines representing the aggregate regret over all types for each firm, and dashed lines representing the each type's regret. It is observed that the type I regret of p_1 is negative, owing to the inaccuracies in the rankings estimated for both p_1 and p_2. A detailed analysis of this negative regret pattern is given in Appendix Section 4.9.11.1.

4.7 Experiments

In this section, we present simulation results to demonstrate the effectiveness of MMTS in learning firms' unknown preferences. The overall experiment setup can be found in Appendix Section 4.9.10. Section 4.7.1 presents two examples to analyze the underlying causes of the novel phenomenon of negative regret (*gain benefit by matching with over-optimal workers*) and large market effect. Appendix Section 4.9.11.1 showcases the distribution of learning parameters and provides insight of reasons for non-optimal stable matchings. Additionally, we demonstrate the robustness of MMTS in large markets in Appendix 4.9.11.2. All simulation results are run in 100 trials.

4.7.1 Two Examples: Small Market and Large Market

Example 1. There are $N = 2$ firms, $M = 2$ types of workers, and there are $K_m = 5, \forall m \in [M]$. The quota q_i^m for each type and each firm p_i is 2, and the total quota/capacity for each firm is $Q_i = 5$. The time horizon is $T = 2000$.

Preferences. True preferences from workers to firms and from firms to workers are all randomly generated. Preferences from workers to firms' $\{\boldsymbol{\pi}^m\}_{m=1}^M$ are fixed and known. We

use the data scientist (D or DS) and software developer engineer (S or SDE) as our example. The following are true preferences,

$$\begin{aligned}
&D_1 : p_1 \succ p_2, D_2 : p_1 \succ p_2, D_3 : p_2 \succ p_1, D_4 : p_1 \succ p_2, D_5 : p_2 \succ p_1, \\
&S_1 : p_1 \succ p_2, S_2 : p_1 \succ p_2, S_3 : p_2 \succ p_1, S_4 : p_2 \succ p_1, S_5 : p_1 \succ p_2, \\
&\pi_1^1 : D_4 \succ D_2 \succ D_3 \succ D_5 \succ D_1, \pi_1^2 : S_1 \succ S_4 \succ S_5 \succ S_2 \succ S_3, \\
&\pi_2^1 : D_2 \succ D_3 \succ D_1 \succ D_5 \succ D_4, \pi_2^2 : S_4 \succ S_2 \succ S_5 \succ S_1 \succ S_3.
\end{aligned} \tag{4.7}$$

The true matching score of each worker for firms are sampled from $U([0,1])$, and are available in Appendix Table 4.1. In addition, feedback $y_{i,j}^m(t)$ (0 or 1) provided by firms is generated by Bernoulli($\mu_{i,j}^m(t)$). If two sides' preferences are known, the firm optimal stable matching is $\bar{u}_1 = \{[D_2, D_4], [S_5, S_1, S_3]\}$, $\bar{u}_2 = \{[D_3, D_1, D_5], [S_4, S_2]\}$ by the double matching algorithm. However, if firms' preferences are unknown, MMTS can learn these unknown preferences and attain the optimal stable matching while achieving a sublinear regret for each firm.

MMTS Parameters. We set priors $\alpha_{i,j}^{m,0} = \beta_{i,j}^{m,0} = 0.1, \forall i \in [N], \forall j \in [K_m], \forall m \in [M]$ to limit the strong impact of the prior belief. The update formula for each firm p_i at time t of the m-type worker a_j^m: $\alpha_{i,j}^{m,t+1} = \alpha_{i,j}^{m,t} + 1$ if the worker a_j^m is matched with the firm p_i, that is $a_j^m \in \mathbf{u}_t^m(p_i)$, and the provided score is $y_{i,j}^m(t) = 1$; otherwise $\alpha_{i,j}^{m,t+1} = \alpha_{i,j}^{m,t}$; $\beta_{i,j}^{m,t+1} = \beta_{i,j}^{m,t} + 1$ if the provided score is $y_{i,j}^m(t) = 0$, otherwise $\beta_{i,j}^{m,t+1} = \beta_{i,j}^{m,t}$. For other unmatched pairs (firm, $m-$ type worker), parameters retain.

Results. In Figure 4.3(a), we find that firm 1, 2 achieve a total *negative* sublinear regret and a total *positive* sublinear regret separately (solid lines). However, we find that due to the incorrect rankings estimated for firms, firm 1 benefits from this non-optimal matching result to achieve *negative* sublinear regret specifically for matching with type 1 workers often (blue dashed line). More discussion about the negative regret phenomenon is available in Appendix 4.9.11.

Example 2. We enlarge the market by expanding the DS market, particularly wanting to

explore interactions between two types of workers. $N = 2$ firms, $M = 2$ types, $K_1 = 20$ (DS) and $K_2 = 6$ (SDE). The DS quota for two firms is $q_1^1 = q_2^1 = 1$ and the SDE quota for two firms is $q_1^2 = q_2^2 = 3$, and the total quota is $Q_i = 6$ for both firms. Preferences from firms to workers and workers to firms are still randomly generated. Therefore, the optimal matching result for each firm should consist of three workers for each type, and type II workers will be fully allocated in the first match, and the rest workers are all type II workers. All MMTS initial parameters are set the same procedure as it in Example 1.

Results. In Figure 4.3(b), we show when excessive type II workers exist, and type I workers are just right. Both firms can achieve positive sublinear regret. We find that since type II worker $K_2 = q_1^2 + q_2^2 = 6$, which means in the first match stage, those type II workers are fully allocated into two firms. Thus, in the second match stage, the left quota would be all allocated to the type I workers for two firms. Two dotted lines represent type II regret suffered by two firms. Both firms can quickly find the type II optimal matching since finding the optimal type II match just needs the first stage of the match. However, the type I workers' matching takes a longer time to find the optimal matching (take two stages), represented by dashed lines, and both are positive sublinear regret. Therefore, these two types of matching are fully independent, which is different from Example 1.

4.8 Discussion

In this project, we proposed a new algorithm, MMTS to solve the CMCPR. MMTS builds on the strengths of TS for exploration and employs a *double matching* method to find a stable solution for complementary preferences and quota constraints. Through theoretical analysis, we show the effectiveness of the algorithm in achieving stability at every matching step under these constraints, achieving a $\widetilde{\mathcal{O}}(Q\sqrt{K_{\max}T})$-Bayesian regret over time, and exhibiting the incentive compatibility property.

There are several directions for future research. One is to investigate more efficient ex-

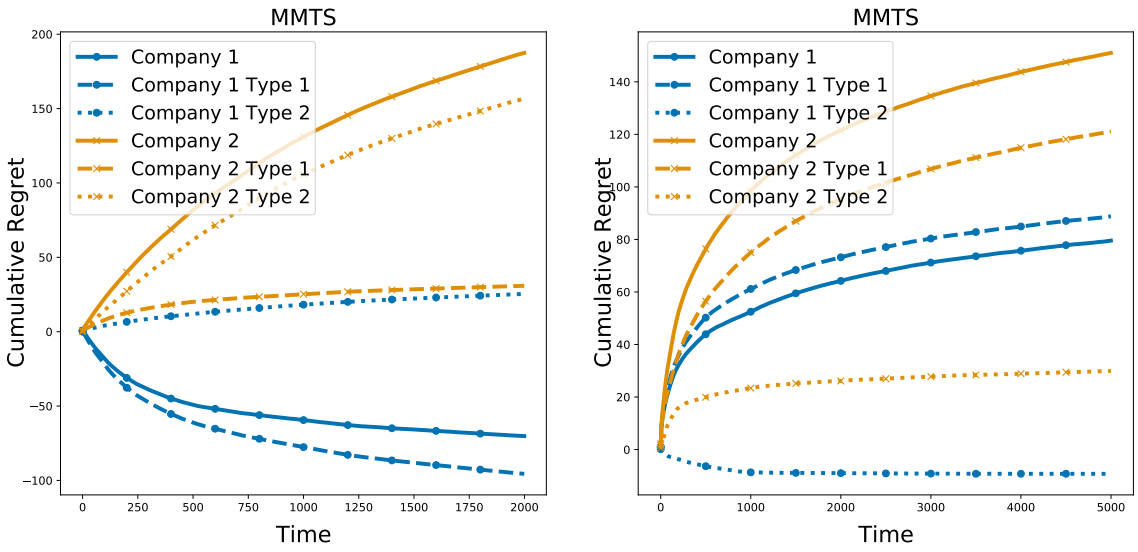

Figure 4.3: Firms and their sub-types regret for Example 1 and, firms and their sub-types regret for Example 2.

ploration strategies to reduce the time required to learn the agents' unknown preferences. Another is to examine scenarios where agents have indifferent preferences, and explore the optimal strategy for breaking ties. Additionally, it is of interest to incorporate real-world constraints such as budget or physical locations into the matching process, which could be studied using techniques from constrained optimization. Moreover, it is interesting to incorporate side information, such as agents' background information, into the matching process. This can be approached using techniques from recommendation systems or other machine learning algorithms that incorporate side information. Finally, it would be interesting to extend the algorithm to handle time-varying matching markets where preferences and the number of agents may change over time.

4.9 Appendix

This supplement is organized as follows. In Section 4.9.1, we discuss the feasibility and its corresponding assumption of the stable matching. In Section 4.9.2, we show the computa-

tional complexity of MMTS. In Section 4.9.3, we exhibit why the centralized UCB suffers insufficient exploration. In Section 4.9.4, we provide the Hoeffding concentration lemma. In Section 4.9.5, we provide the stability property of MMTS. In Section 4.9.6, we give the detailed proof of the regret upper bound of MMTS and decompose its proof into three parts, regret decomposition (4.9.6.1), bound for confidence width (4.4), and bad events' probabilities' upper bound (4.9.6.3). In Section 4.9.8.1, we prove MMTS's strategy-proof property. Besides, as a reference, we append the DA with type and without type algorithms in Section 4.9.9. In Section 4.9.10, we provide details of experiments and the explanation of the negative regret, and also demonstrate the robustness of MMTS in large markets.

4.9.1 Feasibility of the Stable Matching

The feasibility solution is an interesting and well-discussed problem in the stable matching problem.

Assumption of the feasibility: In the finite market, it is the marginal preference assumption for the feasibility. But for the large market, it requires more assumptions such as the substitutability and indifferences, etc,. The difference between the infinite and finite (AH18; GK21) lies in matching problem and the techniques they use. In the infinite market, we assume that there is an uncountable number of agents on both sides of the market. This essentially means that the number of agents is so large that it can be treated as continuous, and you can't assign a specific numerical value to it. An example of an infinite market could be the matching of agents is extremely large and cannot be practically counted. In the finite market, the number of agents on both sides is limited and countable. You can assign a specific numerical value to the number of agents. An example could be the matching of agents where there is a definite small number of agents. However, such an exploration in the infinite market is beyond the scope of our current study.

In our case, if the complementary preference can be marginalized (or referred as the responsive preference (Rot85), $(a_1, b_1) > (a_1, b_2)$ as long as $b_1 > b_2$, verse visa for $(a_1, b_1) >$

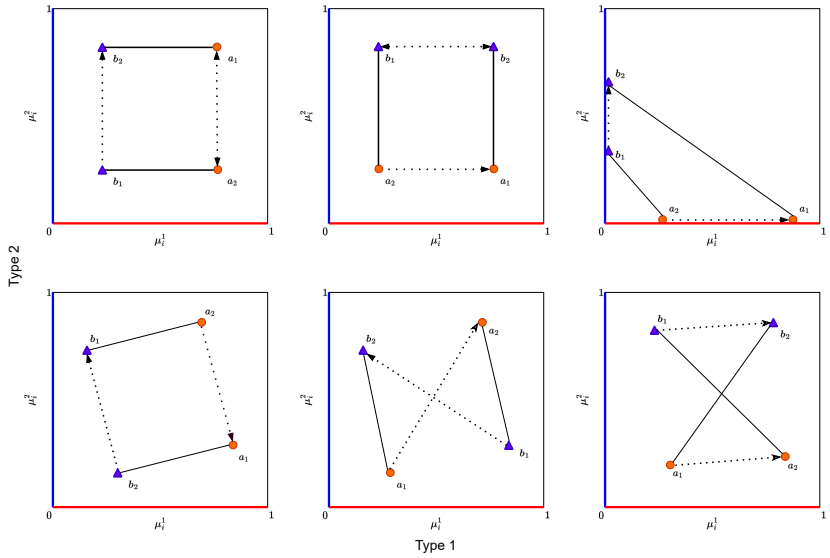

Figure 4.4: Complementary Preference.

(a_2, b_1) as long as $a_1 > a_2$, which is at the top of Figure 4.4, then based on our proposed double matching algorithm and Theory 1, it exists such a stable matching solution. However, as discussed in the related works, if there exists couples in the preference list, which could potentially lead to an empty set of stable matchings.

(CKK19) discussed that if there exists couples in the preference list in a infinite market (large) with a continuum of workers, provided that each firm's choice is convex and changes continuously as the set of available workers changes. They proved the existence and structure of stable matchings under preferences exhibiting substitutability and indifferences in a large market. The difference between our result and (CKK19)'s result is in two ways: (1) we consider the finite market and they consider the infinite market. (2) we consider one side's preferences are unknown and (CKK19)'s both sides preferences are known. (3) (CKK19) proved the existence of stable matching in the infinite market and no algorithm provided. However, in our paper, we provide the double matching algorithm to find it effectively.

4.9.2 Complexity

Based on (GS62; Knu97), the stable marriage problem's DA algorithm's worst total proposal number is $N^2 - 2N + 2 = \mathcal{O}(N^2)$ when the number of participants on both sides is equal ($N = K$). The computational complexity of the college admission matching problem with quota consideration is also $\mathcal{O}(NK)$. MMTS algorithm consists of two steps of matching. The computational complexity of the first step matching is $\mathcal{O}(\sum_{m=1}^{M} NK_m)$ if we virtually consider each type's matching process is organized in parallel. The second step's computation cost is also $\mathcal{O}(\sum_{m=1}^{M} NK_m)$. That is, in the first match, if all firms are matched with their best workers, this step meets the lower bound quota constraints. Then the second match will be reduced to the standard college admission problem without type consideration and the computational complexity is $\mathcal{O}(N \sum_{m=1}^{M} K_m)$. So the total computational complexity is still $\mathcal{O}(\sum_{m=1}^{M} NK_m)$, which is polynomial in the of firm (N) and the number of workers $\sum_{m=1}^{M} K_m$ in the market.

4.9.3 Incapable Exploration

In this section, we show why the TS strategy has an advantage over the UCB style method in estimating the ranks of workers. We even find that centralized UCB does achieve linear firm-optimal stable regret in some cases. In the following example (Example 6 from (LMJ20)), we show the firm achieves linear optimal stable regret if follow the UCB algorithm.[2]

Let $\mathcal{N} = \{p_1, p_2, p_3\}$, $\mathcal{K}_m = \{a_1, a_2, a_3\}$, and $M = 1$, with true preferences given below:

$$p_1 : a_1 \succ a_2 \succ a_3 \qquad a_1 : p_2 \succ p_3 \succ p_1$$

$$p_2 : a_2 \succ a_1 \succ a_3 \qquad a_2 : p_1 \succ p_2 \succ p_3$$

$$p_3 : a_3 \succ a_1 \succ a_2 \qquad a_3 : p_3 \succ p_1 \succ p_2$$

[2]Here we only consider one type of worker, and the firm's quota is one.

The firm optimal stable matching is $(p_1, a_1), (p_2, a_2), (p_3, a_3)$. However, due to incorrect ranking from firm p_3, $a_1 \succ a_3 \succ a_2$, and the output stable matching is $(p_1, a_2), (p_2, a_1), (p_3, a_3)$ based on the DA algorithm. In this case, p_3 will never have a chance to correct its mistake because p_3 will never be matched with a_1 again and cause the upper confidence bound for a_1 will never shrink and result in this rank $a_1 \succ a_3$. Thus, it causes that p_1 and p_2 suffer linear regret.

However, the TS is capable of avoiding this situation. By the property of sampling showed in Algorithm 6, firm p_1's initial prior over worker a_1 is a uniform random variable, and thus $r_3(t) > r_1(t)$ (if we omit a_2) with probability $\widehat{\mu}_3 \approx \mu_3$, rather than *zero*! This differs from the UCB style method, which cannot update a_1's upper bound due to lacking exploration over a_1. The benefit of TS is that it can occasionally explore different ranking patterns, especially when there exists such a previous example.

In Figure 4.2, we show a quick comparison of centralized UCB (LMJ20) in the settings shown above and MMTS when $M = 1, Q = 1, N = 3, K = 3$. The UCB method occurs a linear regret in firm 1 and firm 2 and achieves a low matching rate (0.031)[3]. However, the TS method suffers a sublinear regret in firm 1 and firm 2 and achieves a high matching rate (0.741). All results are averaged over 100 trials. See Section 4.9.3.1 for the experimental details.

4.9.3.1 Section 4.5.2 Example - Insufficient Exploration

We set the true matching score for three firms to $(0.8, 0.4, 0.2)$, $(0.5, 0.7, 0.2)$, $(0.6, 0.3, 0.65)$. All preferences from companies over workers can be derived from the true matching score. As we can view, company p_3 has a similar preference over a_1 (0.6) and a_3 (0.65). Thus, the small difference can lead the incapable exploration as described in Section 4.5.2 by the UCB algorithm.

[3]We count 1 if the matching at time t is fully equal to the optimal match when two sides' preferences are known. Then we take an average over the time horizon T.

4.9.4 Hoeffding Lemma

Lemma 4.2. *For any $\delta > 0$, with probability $1-\delta$, the confidence width for a $m-$type worker $a_j^m \in \mathcal{A}_{i,t}^m$ at time t is upper bounded by*

$$w_{i,\mathcal{F}_{i,t}^m}^m(a_j^m) \leq \min\left(2\sqrt{\frac{\log(\frac{2}{\delta})}{n_{i,j}^m(t)}}, 1\right) \tag{4.8}$$

where $n_{i,j}^m(t)$ is the number of times that the pair (p_i, a_j^m) has been matched at the start of round t.

Proof. Let $\widehat{\mu}_{i,j,t}^{m,LS} = \frac{\sum_{s=1}^{t} \mathbf{1}(a_j^m \in \mathcal{A}_{i,s}^m) y_{i,j}^m(s)}{n_{i,j}^m(t)}$ denote the empirical mean reward from matching firm p_i and $m-$type worker a_j^m up to time t. Define upper and lower confidence bounds as follows:

$$U_{i,t}^m(a_j^m) = \min\left\{\widehat{\mu}_{i,j,t}^{m,LS} + \sqrt{\frac{\log(\frac{2}{\delta})}{n_{i,j}^m(t)}}, 1\right\}, L_{i,t}^m(a_j^m) = \max\left\{\widehat{\mu}_{i,j,t}^{m,LS} - \sqrt{\frac{\log(\frac{2}{\delta})}{n_{i,j}^m(t)}}, 0\right\}. \tag{4.9}$$

The the confidence width is upper bounded by $\min\left(2\sqrt{\frac{\log(\frac{2}{\delta})}{n_{i,j}^m(t)}}, 1\right)$. □

4.9.5 Proof of the Stability of MMTS

Proof. We shall prove existence by giving an iterative procedure to find a stable matching.

Part I To start, in the *first match* loop, based on the double matching procedure, we can discuss M types of matching in parallel. So we will only discuss the path for seeking the type-m company-worker stable matching.

Suppose firm p_i has q_i^m quota for m-type workers. We replace each firm p_i by q_i^m copies of p_i denoted by $\{p_{i,1}, p_{i,2}, ..., p_{i,q_i^m}\}$. Each of these $p_{i,h}$ has preferences identical with those of p_i but with a quota of 1. Further, each m-type worker who has p_i on his/her preference list now replace p_i by the set $\{p_{i,1}, p_{i,2}, ..., p_{i,q_i^m}\}$ in that order of preference. It is now easy to verify

that the stable matchings for the firm m-type worker matching problem are in natural one-to-one correspondence with the stable matchings of this modified version problem. Then in the following, we only need to prove that stable matching exists in this transformed problem where each firm has quota 1, which is the standard stable marriage problem (GS62). The existence of stable matching has been given in (GS62). Here we reiterate it to help us to find the stable matching in the *second match*.

Let each firm propose to his favorite m-type worker. Each worker who receives more than one offer rejects all but her favorite from among those who have proposed to her. However, the worker does not fully accept the firm, but keeps the firm on a string to allow for the possibility that some better firm come along later.

Now we are in the second stage. Those firms who were rejected in the first stage propose to their second choices. Each m-type worker receiving offers chooses her favorite from the group of new firms and the firm on her string, if any. The worker rejects all the rest and again keeps the favorite in suspense. We proceed in the same manner. Those firms who are rejected at the second stage propose to their next choices, and the m-type workers again reject all but the best offer they have had so far.

Eventually, every m-type worker will have rejected a proposal, for as long as any worker has not been proposed to there will be rejections and new offers[4], but since no firm can propose the same m-type worker more than once, every worker is sure to get a proposal in due time. As soon as the last worker gets her offer, the "recruiting" is declared over, and each m-type worker is now required to accept the firm on her string.

We asset that this set of matching is stable. Suppose firm p_i and m-type worker a_j are not matched to each other but firm p_i prefers a_j to his current matching m-type worker $a_{j'}$. Then p_i must have proposed to a_j at some stage (since the proposal is ordered by the preference list) and subsequently been rejected in favor of some firm $p_{i'}$ that a_j liked better. It is clear

[4]Here we assume the number of firms is less than or equal to the number workers, and those workers unmatched finally will be matched to themselves and assume their matching object is on the firm side.

that a_j must prefer her current matching firm $p_{i'}$ and there is no instability/blocking pair.

Thus, each m-type firm-worker matching established on the first match is stable. Then each firm p_i's matching object in the first match with quota q_i^m can be recovered as grouping all matching objects of firm $\{p_{i,h}\}_{h=1}^{q_i^m}$.

Part II To start the second match, we first check the left quota $\widetilde{Q}_i$ for each firm. If the left quota is zero for firm p_i, then firm p_i and its matching workers will quit the matching market and get its stable matching object. Otherwise, the left firm will continue to participate in the second match.

In the second match, preferences from firms to workers are un-categorized. Based on line 19 in Algorithm 7, all types of workers will be ranked to fill the left quota. Thus, it reduces to the problem in part I, and the result matching in the second match is also stable. What is left to prove is that the overall double matching algorithm can provide stable matching. In the second match, each firm proposes to workers in his left concatenate ordered preference list, and all previous workers not in the second match preference list have already been matched or rejected. So it cannot form a blocking pair between the firm p_i with leftover workers. □

4.9.6 MMTS Regret Upper Bound

4.9.6.1 Regret Decomposition

In this part, we provide the road map of the regret decomposition and key steps to prove Theorem 4.6.2. First, we define the history for firm p_i up to time t of type m as $H_{i,t}^m := \{\mathcal{A}_{i,1}^m, \mathbf{y}_{i,\mathcal{A}_{i,1}^m}^m(1), \mathcal{A}_{i,2}^m, \mathbf{y}_{i,\mathcal{A}_{i,2}^m}^m(2), ..., \mathcal{A}_{i,t-1}^m, \mathbf{y}_{i,\mathcal{A}_{i,t-1}^m}^m(t-1)\}$, composed by actions (matched workers) and rewards, where $\mathcal{A}_{i,t}^m := \mathbf{u}_t^m(p_i)$ is a set of workers (based on quota requirement q_i^m and Q_i) belong to m-type which is matched with firm p_i at time t, $\mathbf{y}_{i,\mathcal{A}_{i,t-1}^m}^m(t-1)$ are realized rewards when firm p_i matched with $m-$type workers $\mathcal{A}_{i,t}^m$. Define $\widetilde{H}_{i,t} := \{H_{i,t}^1, H_{i,t}^2, ..., H_{i,t}^M\}$ as the aggregated interaction history between firm p_i and all types of workers up to time t.

Next, we define the *good event* for firm p_i when matching with $m-$type worker at time t and the true mean Matching Score falls in the uncertainty set as $E_{i,t}^m = \{\boldsymbol{\mu}_{i,\mathcal{A}_{i,t}^m}^m \in \mathcal{F}_{i,t}^m\}$, where $\boldsymbol{\mu}_{i,\mathcal{A}_{i,t}^m}^m$ is the true mean reward vector of actually pulled arms (matched with $m-$type workers) at time t for firm p_i, and $\mathcal{F}_{i,t}^m$ is the uncertainty set for $m-$type worker at time t for firm p_i. Similarly, the good event for firm p_i when matching with all types of workers at time t is $E_{i,t} = \bigcap_{m=1}^M E_{i,t}^m$, over all firms $E_t = \bigcap_{i=1}^N E_{i,t}$. And the corresponding *bad event* is defined as $\overline{E}_{i,t}^m, \overline{E}_{i,t}, \overline{E}_t$ respectively. That represents the true mean vector/tensor reward of the pulled arms is not in the uncertainty set.

Lemma 4.3. *Fix any sequence $\{\widetilde{\mathcal{F}}_{i,t} : i \in [N], t \in \mathbb{N}\}$, where $\widetilde{\mathcal{F}}_{i,t} \subset \mathcal{F}$ is measurable with respect to $\sigma(\widetilde{H}_{i,t})$. Then for any $T \in \mathbb{N}$, with probability 1,*

$$\mathfrak{R}(T) \leq \mathbb{E} \sum_{t=1}^{T} \left[\sum_{i=1}^{N} \sum_{m=1}^{M} \widetilde{W}_{i,\mathcal{F}_{i,t}^m}^m(\mathcal{A}_{i,t}^m) + C\mathbf{1}(\overline{E}_t) \right] \tag{4.10}$$

where $\widetilde{W}_{i,\widetilde{\mathcal{F}}_{i,t}^m}^m(\cdot) = \sum_{a_j^m \in \mathcal{A}_{i,t}^m} w_{i,\widetilde{\mathcal{F}}_{i,t}^m}^m(a_j^m)$ represents the sum of the element-wise value of uncertainty width at $m-$type worker a_j^m. The uncertainty width $w_{i,\widetilde{\mathcal{F}}_{i,t}^m}^m(a_j^m) = \sup_{\bar{\mu}_i^m, \underline{\mu}_i^m \in \widetilde{\mathcal{F}}_{i,t}^m} (\bar{\mu}_i^m(a_j^m) - \underline{\mu}_i^m(a_j^m))$ is a worst-case measure of the uncertain about the mean reward of $m-$type worker a_j^m. Here C is a constant less than 1.

Proof. The key step of regret decomposition is to split the instantaneous regret by firms, types, and quotas. Then we categorize regret by the happening of good events and bad events. The good events' regret is measured by the uncertainty width, and the bad events' regret is measured by the probability of happening it.

To reduce notation, define element-wise upper and lower bounds $U_{i,t}^m(a) = \sup\{\mu_i^m(a) : \mu_i^m \in \mathcal{F}_{i,t}^m, a \in \mathcal{K}_m\}$ and $L_{i,t}^m(a) = \inf\{\mu_i^m(a) : \mu_i^m \in \mathcal{F}_{i,t}^m, a \in \mathcal{K}_m\}$, where μ_i^m is the mean reward function $\mu_i^m \in \mathcal{F}_{i,t}^m : \mathbb{R} \mapsto \mathbb{R}, \forall i \in [N], \forall m \in [M]$. Whenever $\mu_{i,\widetilde{\mathcal{A}}_i^m}^m \in \mathcal{F}_{i,t}^m$, the bounds $L_{i,t}^m(a) \leq \mu_{i,\widetilde{\mathcal{A}}_i^m}^m(a) \leq U_{i,t}^m(a)$ hold for all types of workers. Here we define $\mathcal{A}_{i,t}^m = \mathbf{u}_i^m(t)$ as the matched $m-$type workers for firm p_i at time t and $\mathcal{A}_{i,t}^{m,*} = \overline{\mathbf{u}}_i^m(t)$ as the firm p_i's optimal

stable matching result of $m-$type workers at time t. Since the firm-optimal stable matching result is fixed, given both sides' preferences, we can omit time t here. The firm-optimal stable matching result set is also denoted as $\mathcal{A}_i^{m,*} = \mathcal{A}_{i,t}^{m,*}$.

As for type-m workers' matching for the firm p_i at time t, the instantaneous regret with a given instance θ can be implied as follows, here for simplicity, we omit the instance conditional notation

$$\begin{aligned}\mathcal{I}_{i,t}^m = \mu_i^m(\mathcal{A}_i^{m,*}) - \mu_i^m(\mathcal{A}_{i,t}^m) &\leq \sum_{a\in\mathcal{A}_i^{m,*}} U_{i,t}^m(a) - \sum_{a\in\mathcal{A}_{i,t}^m} L_{i,t}^m(a) + C\mathbf{1}(\boldsymbol{\mu}_{i,\widetilde{\mathcal{A}}_i}^m \notin \mathcal{F}_{i,t}^m) \\ &= \widetilde{U}_{i,t}^m(\mathcal{A}_i^{m,*}) - \widetilde{L}_{i,t}^m(\mathcal{A}_{i,t}^m) + C\mathbf{1}(\boldsymbol{\mu}_{i,\widetilde{\mathcal{A}}_i}^m \notin \mathcal{F}_{i,t}^m) \\ &= \widetilde{W}_{i,\mathcal{F}_{i,t}^m}(\mathcal{A}_{i,t}^m) + [\widetilde{U}_{i,t}^m(\mathcal{A}_i^{m,*}) - \widetilde{U}_{i,t}^m(\mathcal{A}_{i,t}^m)] + C\mathbf{1}(\boldsymbol{\mu}_{i,\widetilde{\mathcal{A}}_i}^m \notin \mathcal{F}_{i,t}^m),\end{aligned} \tag{4.11}$$

where $C \leq 1$ is a constant, and we let $\widetilde{U}_{i,t}^m(\cdot) = \sum_a U_{i,t}^m(a)$ and $\widetilde{W}_{i,\mathcal{F}_{i,t}^m}(\cdot) = \sum_a w_{i,\mathcal{F}_t}^m(a)$ represent the sum of the element-wise value of $U_{i,t}^m(\cdot), w_{i,\mathcal{F}_{i,t}}^m(\cdot)$, respectively. Define the good event for firm p_i, matching with $m-$type worker at time t is $E_{i,t}^m = \{\boldsymbol{\mu}_{i,\widetilde{\mathcal{A}}_i}^m \in \mathcal{F}_{i,t}^m\}$, over all types $E_{i,t} = \bigcap_{m=1}^M E_{i,t}^m$, over all firms $E_t = \bigcap_{i=1}^N E_{i,t}$. And the corresponding bad event is defined as $\overline{E}_{i,t}^m, \overline{E}_{i,t}, \overline{E}_t$ respectively.

Now consider Eq. (4.10), summing over the previous equation over time t, firms p_i, and workers' type m, we get

$$\begin{aligned}\mathfrak{R}(T) &\leq \mathbb{E}\sum_{i=1}^N\sum_{t=1}^T\sum_{m=1}^M[\widetilde{W}_{i,\mathcal{F}_{i,t}^m}(\mathcal{A}_{i,t}^m) + C\mathbf{1}(\overline{E}_t)] + \sum_{i=1}^N \mathbb{E}M_{i,T} \\ &= \mathbb{E}\sum_{t=1}^T[C\mathbf{1}(\overline{E}_t) + \sum_{i=1}^N\sum_{m=1}^M \widetilde{W}_{i,\mathcal{F}_{i,t}^m}(\mathcal{A}_{i,t}^m)] + \sum_{i=1}^N \mathbb{E}M_{i,T}\end{aligned} \tag{4.12}$$

where $M_{i,T} = \sum_{t=1}^T\sum_{m=1}^M[\widetilde{U}_{i,t}^m(\mathcal{A}_i^{m,*}) - \widetilde{U}_{i,t}^m(\mathcal{A}_{i,t}^m)]$. Now by the definition of TS, $\mathbb{P}_m(\mathcal{A}_{i,t}^m \in \cdot|H_{i,t}^m) = \mathbb{P}_m(\mathcal{A}_i^{m,*} \in \cdot|H_{i,t}^m)$ for all types, where $\mathbb{P}_m(\cdot|H_{i,t}^m)$ represents this probability is conditional on history $H_{i,t}^m$ and the selected action (worker) belongs in m-type workers for firm p_i. That is $\mathcal{A}_{i,t}^m$ and $\mathcal{A}_i^{m,*}$ within type-m is identically distributed under the posterior.

Besides, since the confidence set $\mathcal{F}_{i,t}^m$ is $\sigma(H_{i,t}^m)$-measurable, so is the induced upper confidence bound $U_{i,t}^m(\cdot)$. This implies $\mathbb{E}_m[U_{i,t}^m(\mathcal{A}_{i,t}^m)|H_t^m] = \mathbb{E}_m[U_{i,t}^m(\mathcal{A}_i^{m,*})|H_t^m]$, and there for $\mathbb{E}[M_{i,T}] = 0$ and $\sum_{i=1}^N \mathbb{E}M_{i,T} = 0$. Then we can obtain the desired result. □

4.9.6.2 Uncertainty Widths

In this part, we provide the upper bound of the accumulated uncertainty widths over all types of workers and all firms, which is the first part in Eq. (4.10).

Lemma 4.4. *If* $(\beta_{i,j,t}^m \geq 0 | t \in \mathbb{N})$ *is a non-decreasing sequence and* $\mathcal{F}_{i,j,t}^m := \{\mu_{i,j}^m \in \mathcal{F}_{i,j}^m : \left\|\mu_{i,j}^m - \widehat{\mu}_{i,j,t}^{m,LS}\right\|_1 \leq \sqrt{\beta_{i,j,t}^m}\}$*, then with probability 1,*

$$\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M} \widetilde{W}_{i,\mathcal{F}_{i,t}^m}^m(\mathcal{A}_{i,t}^m) \leq 8Q\log(QT)\sqrt{K_{\max}T}.$$

The proof of this lemma builds upon Lemma 4.5, which establishes the number of instances where the widths of uncertainty sets for a chosen set of $m-$type workers $\mathcal{A}_{i,t}^m$ greater than ϵ. We show that this number is determined by the *Eluder dimension* (RV14).

Proof. By Lemma 4.3, the instantaneous regret $\mathcal{I}_t$ over all firms and all types, can be decomposed by types and by firms and shown as

$$\begin{aligned}
\mathcal{I}_t &= \sum_{m=1}^{M} \mathcal{I}_t^m = \sum_{i=1}^{N}\sum_{m=1}^{M} \mathcal{I}_{i,t}^m \\
&\leq \sum_{i=1}^{N}\sum_{m=1}^{M} \widetilde{W}_{i,\mathcal{F}_{i,t}^m}(\mathcal{A}_{i,t}^m), \quad \text{if } E_t \text{ holds.} \\
&\leq 2 \sum_{i\in[N],m\in[M],a_j^m\in\mathcal{K}_m} \sqrt{\frac{\log(\sum_{i=1}^N Q_i T)}{n_{i,j}^m(t)}}, \quad \text{with prob } 1-\delta
\end{aligned} \tag{4.13}$$

where the first inequality is based on Lemma 4.3 and if E_t holds for $t \in \mathbb{N}, m \in M, i \in [N]$, $n_{i,j}^m(t)$ is the number of times that the pair (p_i, a_j^m) has been matched at the start of round t.

The second inequality is constructed from a union concentration inequality based on Lemma 4.2, and we set $\delta = 2/\sum_{i=1} Q_i T$. We denote $z_{i,j}^m(t) = \frac{1}{\sqrt{n_{i,j}^m(t)}}$ as the size of the scaled confidence set (without the log factor) for the pair (p_i, a_j^m) at the time t.

At each time step t, let's consider the list consisting of $z_{i,j}^m(t)$ and reorder the overall list consisting of concatenating all those scaled confidence sets over all rounds and all types in decreasing order. Then we obtain a list $\tilde{z}_1 \geq \tilde{z}_2 \geq ..., \geq \tilde{z}_L$, where $L = \sum_{t=1}^T \sum_{i=1}^N Q_i = T\sum_{i=1}^N Q_i$. We reorganize the Eq. (4.13) to get

$$\sum_{t=1}^T \mathcal{I}_t \leq \sum_{t=1}^T \sum_{m=1}^M \sum_{i=1}^N \widetilde{W}_{i,\mathcal{F}_{i,t}^m}(\mathcal{A}_{i,t}^m) \leq 2\log(\sum_{i=1}^N Q_i T) \sum_{l=1}^L \tilde{z}_l. \tag{4.14}$$

By Lemma 4.5, the number of rounds that a pair of a firm and any $m-$type worker can have it confidence set have size at least $\tilde{z}_l$ is upper bounded by $(1+\frac{4}{\tilde{z}_l^2})K_m$ when we set $\epsilon = \tilde{z}_l$ and know $\beta_{i,j,t}^m \leq 1$. Thus, the total number of times that any confidence set can have size at least $\tilde{z}_l$ is upper bounded by $\left(1+\frac{4}{\tilde{z}_l^2}\right)\sum_{i=1}^N \sum_{m=1}^M |\mathcal{A}_{i,t}^m| K_m$. To determine the minimum condition for $\tilde{z}_l$, which is equivalent to determine the maximum of l, we have $l \leq \left(1+\frac{4}{\tilde{z}_l^2}\right)\sum_{i=1}^N \sum_{m=1}^M |\mathcal{A}_{i,t}^m| K_m$. So we claim that

$$\tilde{z}_l \leq \min\left(1, \frac{2}{\sqrt{\frac{l}{\sum_{i=1}^N \sum_{m=1}^M |\mathcal{A}_{i,t}^m| K_m} - 1}}\right) \leq \min\left(1, \frac{2}{\sqrt{\frac{l}{\sum_{i=1}^N Q_i K_{\max}} - 1}}\right), \tag{4.15}$$

where the second inequality above is by $\sum_{i=1}^N \sum_{m=1}^M |\mathcal{A}_{i,t}^m| K_m \leq K_{\max} \sum_{i=1}^N \sum_{m=1}^M |\mathcal{A}_{i,t}^m| \leq K_{\max} \sum_{i=1}^N Q_i = Q K_{\max}$ and $K_{\max} = \max\{K_1, ..., K_M\}, Q = \sum_{i=1}^N Q_i$. Putting all these

together, we have

$$
\begin{aligned}
2\log(\sum_{i=1}^{N} Q_i T)\sum_{l=1}^{L}\tilde{z}_l &\le 2\log(QT)\sum_{l=1}^{L}\min(1,\frac{2}{\sqrt{\frac{l}{QK_{\max}}-1}}) \\
&= 4\log(QT)\sum_{l=1}^{QT}\frac{1}{\sqrt{\frac{l}{QK_{\max}}-1}} \\
&\le 8\log(QT)\sqrt{QK_{\max}}\sqrt{QT}
\end{aligned}
\tag{4.16}
$$

where the last inequality is by intergral inequality

$$
\sum_{l=1}^{QT}\frac{1}{\sqrt{\frac{l}{QK_{\max}}-1}} \le \sqrt{QK_{\max}}\sum_{l=1}^{QT}\frac{1}{\sqrt{l}} \le \sqrt{QK_{\max}}\int_{x=0}^{QT}\frac{1}{\sqrt{x}}dx = 2\sqrt{QK_{\max}}\sqrt{QT}.
$$

Based on Eq. (4.14) and the above result, we can get the regret

$$
\sum_{t=1}^{T}\mathcal{I}_t \le 8Q\log(QT)\sqrt{K_{\max}T}, \tag{4.17}
$$

if E_t holds. □

Lemma 4.5. *If* $(\beta_{i,j,t}^m \ge 0 | t \in \mathbb{N})$ *is a nondecreasing sequence for* $i \in [N], a_j^m \in \mathcal{K}_m, m \in [M]$ *and* $\mathcal{F}_{i,j,t}^m := \{\mu_{i,j}^m \in \mathcal{F}_{i,j}^m : \left\|\mu_{i,j}^m - \widehat{\mu}_{i,j,t}^{m,LS}\right\|_1 \le \sqrt{\beta_{i,j,t}^m}\}$, *for all* $T \in \mathbb{N}$ *and* $\epsilon > 0$, *then*

$$
\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{a_j^m \in \mathcal{A}_{i,t}^m}\mathbf{1}\big(w_{i,\mathcal{F}_{i,t}^m}^m(a_j^m) > \epsilon\big) \le \big(\frac{4\widetilde{\beta}_{i,T}}{\epsilon^2}+1\big)\sum_{m=1}^{M}|\mathcal{A}_{i,t}^m|K_m.
$$

Here $\widehat{\mu}_{i,j,t}^{m,LS} = \frac{\sum_{s=1}^{t}\mathbf{1}(a_j^m \in \mathcal{A}_{i,s}^m)y_{i,j}^m(s)}{n_{i,j}^m(t)}$ *is the estimated average reward for* $m-$*type worker* a_j^m *from the view point of firm* p_i *at time* t, *and* $n_{i,j}^m(t)$ *is the number of matched times up to time* t *of firm* p_i *with* $m-$*type worker* a_j^m. *Besides, we define* $\widetilde{\beta}_{i,T} = \max_{a_j^m \in \mathcal{K}_m, m\in[M]}\beta_{i,j,T}^m$ *as the maximum uncertainty bound over all types of workers at time* T *for firm* p_i.

The proof of this result is based on techniques from (RV13; RV14). This result demonstrates

that the upper bound of the number of times the widths of uncertainty sets exceeds ϵ is dependent on the error $\mathcal{O}(\epsilon^{-2})$ and linearly proportional to the product of the number of $m-$type worker and the type quota size q_i^m.

Proof. Based on the Proposition 3 from (RV13), we can use the *eluder dimension* $dim_E(\mathcal{F}_i^m, \epsilon)$ to bound the number of times the widths of confidence intervals for a selection of set of $m-$type workers $\mathcal{A}_{i,t}^m$ greater than ϵ.

$$\begin{aligned}\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{a_j^m\in\mathcal{A}_{i,t}^m}\mathbf{1}\left(w_{i,\mathcal{F}_{i,t}^m}^m(a_j^m)>\epsilon\right) &\leq \sum_{m=1}^{M}\sum_{a_j^m\in\mathcal{A}_{i,t}^m}\left(\frac{4\beta_{i,j,T}^m}{\epsilon^2}+1\right)\dim_E(\mathcal{F}_i^m,\epsilon)\\ &\leq \left(\frac{4\max\limits_{a_j^m\in\mathcal{K}_m,m\in[M]}\beta_{i,j,T}^m}{\epsilon^2}+1\right)\sum_{m=1}^{M}|\mathcal{A}_{i,t}^m|\dim_E(\mathcal{F}_i^m,\epsilon),\end{aligned} \tag{4.18}$$

where the eluder dimension of a multi-arm bandit problem is the number of arms, we get

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{a_j^m\in\mathcal{A}_{i,t}^m}\mathbf{1}\left(w_{i,\mathcal{F}_t}^m(a_j^m)>\epsilon\right)\leq\left(\frac{4\widetilde{\beta}_{i,T}}{\epsilon^2}+1\right)\sum_{m=1}^{M}|\mathcal{A}_{i,t}^m|K_m\leq\left(\frac{4\widetilde{\beta}_{i,T}}{\epsilon^2}+1\right)Q_iK_{\max} \tag{4.19}$$

where $\widetilde{\beta}_{i,T}=\max\limits_{a_j^m\in\mathcal{K}_m,m\in[M]}\beta_{i,j,T}^m$. Besides, we know that $Q_i=\sum_{m=1}^{M}|\mathcal{A}_{i,t}^m|$ and define $K_{\max}=\max\limits_{m\in[M]}K_m$, so we can get the second inequality. □

4.9.6.3 Bad Event Upper Bound

In this part, we provide an upper bound of the second part of Eq. (4.10). The regret caused by the happening of the bad event at each time step is quantified by the following lemma.

Lemma 4.6. *If* $\mathcal{F}_{i,j,t}^m:=\{\mu_{i,j}^m\in\mathcal{F}_{i,j}^m:\left\|\mu_{i,j}^m-\widehat{\mu}_{i,j,t}^{m,LS}\right\|_1\leq\sqrt{\beta_{i,j,t}^m}\}$ *holds with probability* $1-\delta$*, then the bad event* $\overline{E}_t$ *happening's probability is upper bounded by* $\mathbb{E}\mathbf{1}(\overline{E}_t)\leq NK\delta$*. In particular, if* $\delta=1/QT$*, the accumulated bad events' probability is upper bounded by* $\sum_{t=1}^{T}\mathbb{E}\mathbf{1}(\overline{E}_t)\leq NK/Q$*.*

To bound the probability of bad events, we use a union bound to obtain the desired result. Specifically, if $Q_i = 1$, which means each firm has a total quota of 1 and only considers one type of worker, then $\sum_{t=1}^{T} \mathbb{E}\mathbf{1}(\overline{E}_t) \leq NK/(N \times 1) = K$. This shows that each firm needs to explore a single type of worker, and the worst total regret is less than K. If $Q_i = 1, M = 1$, which means all firms have the same recruiting requirements, the result reduces to the general competitive matching scenario, and the worst regret is the number of workers of type K_M in the market.

Proof. If E_t does not hold, the probability of the true Matching Score is not in the confidence interval we constructed is upper bounded by

$$\begin{aligned}
\mathbb{E}\mathbf{1}(\overline{E}_t) = \mathbb{P}(\overline{E}_t) &= \mathbb{P}\left(\left(\bigcap_{i\in[N]} \bigcap_{m\in[M]} \bigcap_{a_j^m \in \mathcal{K}_m} \{\mu_{i,j}^m \in \mathcal{F}_{i,j,t}^m\} \right)^c \right) \\
&= \mathbb{P}\left(\bigcup_{i\in[N]} \bigcup_{a_j^m \in \mathcal{K}_m} \bigcup_{m\in[M]} \{\mu_{i,j}^m \notin \mathcal{F}_{i,j,t}^m\} \right) \\
&= \mathbb{P}\left(\bigcup_{i\in[N]} \bigcup_{a_j^m \in \mathcal{K}_m} \bigcup_{m\in[M]} \left\{ \left\| \mu_{i,j}^m - \widehat{\mu}_{i,j,t}^{m,LS} \right\|_{2,E_t} \geq \sqrt{\beta_{i,j,t}^m} \right\} \right) \\
&= \mathbb{P}\left(\bigcup_{i\in[N]} \bigcup_{a_j^m \in \mathcal{K}_m} \bigcup_{m\in[M]} \left\{ \left\| \mu_{i,j}^m - \widehat{\mu}_{i,j,t}^{m,LS} \right\|_{1} \geq \sqrt{\frac{\log(\frac{2}{\delta})}{n_{i,j}^m(t)}} \right\} \right) \\
&\leq \sum_{i\in[N]} \sum_{a_j^m \in \mathcal{K}_m} \sum_{m\in[M]} \mathbb{P}\left(\left\| \mu_{i,j}^m - \widehat{\mu}_{i,j,t}^{m,LS} \right\|_{1} \geq \sqrt{\frac{\log(\frac{2}{\delta})}{n_{i,j}^m(t)}} \right)
\end{aligned} \tag{4.20}$$

where the third equality is by De-Morgan's Law of sets. In the last inequality, we use the union bound to control the probability. Since each $\widehat{\mu}_{i,j}^{m,LS} - \mu_{i,j}^m$ is a mean zero and $\frac{1}{2n_{i,j}^m}$-sub-Gaussian random variable, based on Lemma 4.2, have $\mathbb{P}(\left\| \mu_{i,j}^m - \widehat{\mu}_{i,j,t}^{m,LS} \right\|_1 \geq \sqrt{\frac{\log(\frac{2}{\delta})}{n_{i,j}^m(t)}}) \leq \delta$. The overall bad event's probability's upper bound is

$$\mathbb{P}(\overline{E}_t) \leq NK\delta \tag{4.21}$$

Based on our confidence width is less than 1, so $C = 1, \forall i \in [N]$. The expected regret from this bad event is not in the confidence interval at most

$$NK\delta \cdot CT \leq NK \frac{1}{\sum_{i=1}^{N} Q_i T} T = \frac{NK}{Q} \tag{4.22}$$

This part's regret is negligible compared with the regret from Lemma 4.4. In particular, if there is only one type and each firm has only one position to be filled. Thus, $Q = N$, the bad event's upper bounded probability will shrink to K, the number of workers to be explored. □

In this part, we provide the proof of MMTS's Bayesian regret upper bound.

4.9.7 Proof of Theorem 4.2

Theorem 4.4. *When all firms follow the MMTS algorithm, the platform will incur the Bayesian total expected regret*

$$\mathfrak{R}(T) \leq 8 \log(QT) \sqrt{QK_{\max}} \sqrt{QT} + NK/Q \tag{4.23}$$

where $K_{\max} = \max\{K_1, ..., K_M\}, K = \sum_{m=1}^{M} K_m$.

Proof. We decompose the Bayesian Social Welfare Gap for all firms by

$$\begin{aligned}
\mathfrak{R}(T) &= \mathbb{E}_{\theta \in \Theta}\left[\sum_{i=1}^{N} R_i(T, \theta)\right] = \mathbb{E}_{\theta \in \Theta}\left[\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} \mu_{i,\overline{u}_i^m(t)}(t) - \sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} \mu_{i,u_i^m}(t) | \theta\right] \\
&= \sum_{i=1}^{N}\sum_{t=1}^{T} \mathbb{E}_{\theta \in \Theta}\left[\sum_{m=1}^{M} (\mu_{i,\overline{u}_i^m(t)}(t) - \mu_{i,u_i^m}(t)) | \theta\right] \\
&= \mathbb{E}_{\theta \in \Theta}\left[\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M} \mathcal{I}_{i,t}^m | \theta\right] \\
&= \mathbb{E}_{\theta \in \Theta}\left[\sum_{t=1}^{T} \mathcal{I}_t | \theta\right]
\end{aligned} \tag{4.24}$$

where we define $\mathcal{I}_{i,t}^m = \mu_{i,\boldsymbol{\theta}}^m(\mathcal{A}_i^{m,*}) - \mu_{i,\boldsymbol{\theta}}^m(\mathcal{A}_{i,t}^m)$ and $\mathcal{I}_t = \sum_{i=1}^N \sum_{m=1}^M \mathcal{I}_{i,t}^m$. Here $\mathcal{A}_i^{m,*}$ is the optimal matched workers for firm p_i of type m and $\mathcal{A}_{i,t}^m$ is the actual matched workers for firm p_i of type m at time t under the instance θ.

Based Lemma 4.3, $\mathfrak{R}(T)$ is upper bounded by $\mathbb{E}\sum_{t=1}^T \left[C\mathbf{1}(\overline{E}_t) + \sum_{i=1}^N \sum_{m=1}^M \widetilde{W}_{i,\mathcal{F}_{i,t}^m}(\mathcal{A}_{i,t}^m)\right]$. The first term, the sum of the bad event probability $\mathbb{E}\sum_{t=1}^T C\mathbf{1}(\overline{E}_t) = C\sum_{t=1}^T \mathbb{P}(\overline{E}_t)$, which is upper bounded by NK/Q based on Lemma 4.6 and $C \leq 1$. The second term, the sum of confidence widths is upper bounded by $8Q\log(QT)\sqrt{TK_{\max}}$ based on Lemma 4.4. Thus the Bayesian regret is upper bounded by $8Q\log(QT)\sqrt{TK_{\max}} + NK/Q$. □

4.9.8 Incentive-Compatibility

In this section, we discuss the incentive-compatibility property of MMTS. That is, if one firm does not follow the MMTS when all other firms submit their MMTS preferences, that firm cannot benefit (matched with a better worker than his optimal stable matching worker) over a sublinear order. As we know, (DF81) discussed the *Machiavelli* firm could not benefit from incorrectly stating their true preference when there exists a unique stable matching. However, when one side's preferences are unknown and need to be learned through data, this result no longer holds. Thus, the maximum benefits that can be gained by the Machiavelli firm are under-explored in the setting of learning in matching. (LMJ20) discussed the benefits that can be obtained by Machiavelli firm when other firms follow the centralized-UCB algorithm with the problem setting of one type of worker and quota equal one in the market.

We now show in CMCPR, when all firms except one p_i submit their MMTS-based preferences to the matching platform, the firm p_i has an incentive also to submit preferences based on their sampling rankings in a *long horizon*, so long as the matching result do not have multiple stable solutions. Now we establish the following lemma, which is an upper bound of the expected number of pulls that a firm p_i can match with a m-type worker that is better than their optimal m-type workers, regardless of what preferences they submit to

the platform.

Let's use $\mathcal{H}_{i,l}^m$ to define the achievable *sub-matching* set of $\mathbf{u}^m$ when all firms follow the MMTS, which represents firm p_i and $m-$type worker a_l^m is matched such that $a_l^m \in \mathbf{u}_i^m$. Let $\Upsilon_{\mathbf{u}^m}(T)$ be the number of times sub-matching $\mathbf{u}^m$ is played by time t. We also provide the blocking triplet in a matching definition as follows.

Definition 4.6. (Blocking triplet) A blocking triplet $(p_i, a_k, a_{k'})$ for a matching u is that there must exist a firm p_i and worker a_j that they both prefer to match with each other than their current match. That is, if $a_{k'} \in \mathbf{u}_i$, $\mu_{i,k'} < \mu_{i,k}$ and worker a_k is either unmatched or $\pi_{k,i} < \pi_{k,\mathbf{u}^{-1}(k)}$.

The following lemma presents the upper bound of the number of matching times of p_i and a_l^m by time T, where a_l^m is a *super optimal* $m-$type worker (preferred than all stable optimal $m-$type workers under true preferences), when all firms follow the MMTS.

Lemma 4.7. *Let $\Upsilon_{i,l}^m(T)$ be the number of times a firm p_i matched with a m-type worker such that the mean reward of a_l^m for firm p_i is greater than p_i's optimal match $\overline{\mathbf{u}}_i^m$, which is $\mu_{i,a_l^m}^m > \max_{a_j^m \in \overline{\mathbf{u}}_i^m} \mu_{i,j}^m$. Then the expected number of matches between p_i and a_l^m is upper bounded by*

$$\mathbb{E}[\Upsilon_{i,l}^m(T)] \leq \min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \sum_{(p_j, a_k^m, a_{k'}^m) \in S^m} \big(C_{i,j,k'}^m(T) + \frac{\log(T)}{d(\mu_{j,\overline{\mathbf{u}}_{i,\min}^m}, \mu_{j,k'})}\big),$$

where $\overline{\mathbf{u}}_{i,\min}^m = \operatorname{argmin}_{a_k^m \in \overline{\mathbf{u}}_j^m} \mu_{i,k}^m$, and $C_{i,j,k'}^m = \mathcal{O}((\log(T))^{-1/3})$.

Then we provide the benefit (lower bound of the regret) of Machiavelli firm p_i can gain by not following the MMTS from matching with m-type workers. Let's define the *super worker reward gap* as $\overline{\Delta}_{i,l}^m = \max_{a_j^m \in \overline{\mathbf{u}}_i^m} \mu_{i,j}^m - \mu_{i,l}^m$, where $a_l^m \notin \overline{\mathbf{u}}_i^m$.

Theorem 4.5. *Suppose all firms other than firm p_i submit preferences according to the MMTS to the centralized platform. Then the following upper bound on firm p_i's optimal*

regret for m-type workers holds:

$$R_i^m(T,\theta) \geq \sum_{l:\overline{\Delta}_{i,l}^m<0} \overline{\Delta}_{i,l}^m \left[\min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \sum_{(p_j,a_k^m,a_{k'}^m)\in S^m} \left(C_{i,j,k'}^m + \frac{\log(T)}{d(\mu_{j,\overline{\mathbf{u}}_{i,\min}^m}, \mu_{j,k'})} \right) \right] \tag{4.25}$$

where $\overline{\mathbf{u}}_{i,\min}^m = \underset{a_k^m \in \overline{\mathbf{u}}_j^m}{\operatorname{argmin}} \mu_{i,k}^m$, *and* $C_{i,j,k'}^m = \mathcal{O}((\log(T))^{-1/3})$.

This result can be directly derived from Lemma 4.1. Theorem 4.3 demonstrates that there is no sequence of preferences that a firm can submit to the centralized platform that would result in negative optimal regret greater than $\mathcal{O}(\log T)$ in magnitude within type m. When considering multiple types together for firm p_i, this magnitude remains $\mathcal{O}(\log T)$ in total. Theorem 4.3 confirms that, when there is a unique stable matching in type m, firms cannot gain significant advantage in terms of firm-optimal stable regret by submitting preferences other than those generated by the MMTS algorithm. An example is provided in Section 4.7.1 to illustrate this incentive compatibility property. Figure 4.3(a) illustrates the total regret, with solid lines representing the aggregate regret over all types for each firm, and dashed lines representing the regret for each type. It is observed that the type 1 regret of firm 1 is negative, owing to the inaccuracies in the rankings submitted by both firm 1 and firm 2. A detailed analysis of this negative regret pattern is given in Section 4.9.11.1.

4.9.8.1 Proof of Incentive Compatibility

Lemma 4.8. *Let* $\Upsilon_{i,l}^m(T)$ *be the number of times a firm* p_i *matched with a m-type worker such that the mean reward of* a_l^m *for firm* p_i *is greater than* p_i*'s optimal match* $\overline{u}_i^m$*, which is* $\mu_{i,a_l^m}^m > \max_{a_j^m \in \overline{u}_i^m} \mu_{i,j}^m$. *Then*

$$\mathbb{E}[\Upsilon_{i,l}^m(T)] \leq \min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \sum_{(p_j,a_k^m,a_{k'}^m)\in S^m} \left(C_{i,j,k'}^m(T) + \frac{\log(T)}{d(\mu_{j,\overline{u}_{i,\min}^m}, \mu_{j,k'})} \right) \tag{4.26}$$

where $\overline{u}^m_{i,\min} = \underset{a^m_k \in \overline{u}^m_j}{\operatorname{argmin}} \mu^m_{i,k}$, $C^m_{i,j,k'} = \mathcal{O}((\log(T))^{-1/3})$.

Proof. We claim that if firm p_i is matched with a *super optimal* $m-$type worker a^m_l in any round, the matching u^m must be unstable according to true preferences from both sides. We then state that there must exist a m-type blocking triplet $(p_j, a^m_k, a^m_{k'})$ where $p_j \neq p_i$.

We prove it by contradiction. Suppose all blocking triplets in matching u *only* involve firm p_i within $m-$type worker. By Theorem 4.2 in (AR95), we can start from any matching u to a stable matching by iteratively satisfying blocking pairs in a *gender consistent* order, which means that we can provide a well-defined order to determine which blocking triplet should be satisfied (matched) first within preferences from firm p_i[5]. Doing so, firm p_i can never get a worse match than a^m_l since a blocking pair will let firm p_i match with a better $m-$type worker than a^m_l, or become unmatched as the algorithm proceeds, so the matching will remain unstable. The matching will continue, which is a contradiction.

Hence there must exist a firm $p_j \neq p_i$ such that p_j is part of a blocking triplet in u when firm p_i is matched with $m-$type worker a^m_l under the matching u. In particular, based on the Theorem 9 (Dubins-Freedman Theorem), firm p_j must submit its TS preference.

Let $L^m_{j,k,k'}(T)$ be the number of times firm p_j matched with $m-$type worker $a^m_{k'}$ when the triplet $(p_j, a^m_k, a^m_{k'})$ is blocking the matching provided by the centralized platform. Then by the definition

$$\sum_{u^m \in B^m_{j,k,k'}} \Upsilon_{u^m}(T) = L^m_{j,k,k'}(T) \tag{4.27}$$

By the definition of a blocking triplet, we know that if p_j is matched with $m-$type worker $a^m_{k'}$ when the blocking triplet $(p_j, a^m_k, a^m_{k'})$ is blocking, the TS sample must have a higher mean reward for $a^m_{k'}$ than a^m_k. In other words, we need to bound the expected number of times that the TS mean reward for $m-$type worker $a^m_{k'}$ is greater than a^m_k. From (KHN15), we know that the number of times that $(p_j, a^m_k, a^m_{k'})$ forms a blocking pair in Thompson sampling, is

[5]This gender consistent requirement is to satisfy a blocking pair (p_j, a^m_k) and those blocking pairs can be ordered before we break their current matches if any, and then match p_j and a^m_k to get a new matching.

upper bounded by

$$\mathbb{E}L^m_{j,k,k'} \leq C^m_{i,j,k'}(T) + \frac{\log(T)}{d(\mu_{j,\overline{u}^m_{i,\min}}, \mu_{j,k'})} \tag{4.28}$$

where $\overline{u}^m_{i,\min} = \underset{a^m_k \in \overline{u}^m_j}{\operatorname{argmin}} \mu^m_{i,k}$ and $C^m_{i,j,k'} = \mathcal{O}((\log(T))^{-1/3})$. The $d(x,y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$ is the KL divergence between two Bernoulli distributions with expectation x and y.

The expected number of times $\Upsilon^m_{i,l}(T)$ a firm p_i matched with a $m-$type worker such that the mean reward of a^m_l for firm p_i is greater than p_i's optimal match $\overline{u}^m_i$, which is equivalent to the expected number of times viat the achievable sub-matching set $\Upsilon_{u^m}(T)$ where $u^m \in \mathcal{H}^m_{i,l}$. So the result then follows from the identity

$$\mathbb{E}[\Upsilon^m_{i,l}(T)] = \sum_{u^m \in \mathcal{H}^m_{i,l}} \mathbb{E}\Upsilon_{u^m}(T) \tag{4.29}$$

Given a set $\mathcal{H}^m_{i,l}$ of matchings, we say a set S^m of triplets $(p_j, a^m_k, a^m_{k'})$ is a *cover* of $\mathcal{H}^m_{i,l}$ if

$$\bigcup_{(p_j, a^m_k, a^m_{k'}) \in S^m} B^m_{j,k,k'} \supseteq H^m_{i,l} \tag{4.30}$$

Let $\mathcal{C}(H_{i,l}^m)$ denote the set of covers of $H_{i,l}^m$. Then

$$\begin{aligned}
\mathbb{E}[\Upsilon_{i,l}^m(T)] &= \mathbb{E} \sum_{u^m \in \mathcal{H}_{i,l}^m} \Upsilon_{u^m}(T) \\
&\leq \mathbb{E} \min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \sum_{(p_j, a_k^m, a_{k'}^m) \in S^m} \Upsilon_{u^m}(T) \\
&= \min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \mathbb{E} \sum_{(p_j, a_k^m, a_{k'}^m) \in S^m} \Upsilon_{u^m}(T) \\
&= \min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \sum_{(p_j, a_k^m, a_{k'}^m) \in S^m} \mathbb{E} L_{j,k,k'}^m(T) \\
&\leq \min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \sum_{(p_j, a_k^m, a_{k'}^m) \in S^m} \left(C_{i,j,k'}^m(T) + \frac{\log(T)}{d(\mu_{j,k}, \mu_{j,k'})} \right) \\
&\leq \min_{S^m \in \mathcal{C}(\mathcal{H}_{i,l}^m)} \sum_{(p_j, a_k^m, a_{k'}^m) \in S^m} \left(C_{i,j,k'}^m(T) + \frac{\log(T)}{d(\mu_{j,\overline{u}_{i,\min}^m}, \mu_{j,k'})} \right)
\end{aligned} \tag{4.31}$$

where the first inequality is from the property of cover and we select the minimum cover S^m from $\mathcal{C}(\mathcal{H}_{i,l}^m)$. And summation in the third line is equivalent to $\sum_{u^m \in B_{j,k,k'}^m}$. Based on Eq. (4.27), the third equality is obvious. From (KHN15), we know the expected number of times of matching with the sub-optimal $m-$type worker is upper bounded by Eq. (4.28). □

4.9.9 Firm DA Algorithm with type and without type consideration

In this section, we present the DA algorithm with type consideration and without type consideration.

Algorithm 8: Firm-Proposing DA Algorithm with Type Consideration.

Input : Type. firms set $\mathcal{N}$, workers set $\mathcal{K}_m, \forall m \in [M]$; firms to workers' preferences $\mathbf{r}_i^m, \forall i \in [N], \forall m \in [M]$, workers to firms' preferences $\boldsymbol{\pi}^m, \forall m \in [M]$; firms' type-specific quota $q_i^m, \forall i \in [N], \forall m \in [M]$, firms' total quota $Q_i, \forall i \in [N]$.

Initialize: Empty set $\mathcal{S} = \{\}$, empty sets $S^m = \emptyset, \forall m \in [M]$.

for $m = 1, ..., M$ **do**
  while $\exists$ *A firm p who is not fully filled with the quota q^m and has not contacted every $m-$type worker* **do**
    Let a be the highest-ranking worker in firm p's preference, to whom firm p has not yet contacted.
    Now firm p contacts the worker a.
    if *Worker a is free* **then**
      (p, a) become matched (add (p, a) to S^m).
    else
      Worker a is matched to firm p' (add (p', a) to S^m).
      if *Worker a prefers firm p' to firm p* **then**
        firm p filled number minus 1 (remove (p, a) from S^m).
      else
        Worker a prefers firm p to firm p'.
        firm p' filled number minus 1 (remove (p', a) from S^m).
        (p, a) are paired (add (p, a) to S^m).
  Update: Add S^m to $\mathcal{S}$.

Output : Matching result $\mathcal{S}$.

Algorithm 9: Firm-Proposing DA Algorithm without Type Consideration (GS62).

Input : Worker Types, firms set $\mathcal{N}$, workers set $\mathcal{K}_m, \forall m \in [M]$; firms to workers' preferences $\mathbf{r}_i^m, \forall i \in [N], \forall m \in [M]$, workers to firms' preferences $\boldsymbol{\pi}^m, \forall m \in [M]$; firms' type-specific quota $q_i^m, \forall i \in [N], \forall m \in [M]$, firms' total quota $Q_i, \forall i \in [N]$.

Initialize: Empty set S.

while $\exists$ *A firm* p *who is not fully filled with the quota* $\tilde{Q}$ *and has not contacted every worker* **do**

  Let a be the highest-ranking worker in firm p's preference over all types of workers, to whom firm p has not yet contacted.

  Now firm p contacts the worker a.

  if *Worker* a *is free* **then**

    (p, a) become matched (add (p, a) to S).

  else

    Worker a is matched to firm p' (add (p', a) to S).

    if *Worker* a *prefers firm* p' *to firm* p **then**

      firm p filled number minus 1 (remove (p, a) from S).

    else

      Worker a prefers firm p to firm p'.

      firm p' filled number minus 1 (remove (p', a) from S).

      (p, a) are paired (add (p, a) to S).

Output : Matching result S.

4.9.10 Experimental Details

In this section, we provide more details about the analysis of the negative regret, parameters, and large market.

Table 4.1: True Matching Scores of two types of workers from two firms.

Mean ID	**Type**	**1**	**2**	**3**	**4**	**5**
$\boldsymbol{\mu}_1$	1	0.406	0.956	0.738	0.970	0.695
	2	0.932	0.241	0.040	0.657	0.289
$\boldsymbol{\mu}_2$	1	0.682	0.909	0.823	0.204	0.218
	2	0.303	0.849	0.131	0.886	0.428

4.9.11 Negative Regret Phenomenon

The occurrence of negative regret in multi-agent matching schemes presents an interesting phenomenon, contrasting the single-agent bandit problem wherein negative regret is non-existent.

In the context of the single-agent bandit problem, it is known that the best arm can be pulled, resulting in instantaneous regret that can attain zero but not take negative values. Conversely, in the multi-agent competing bandit problem, the oracle firm-optimal arm is determined by the true expected reward/utility, assuming knowledge of the true parameter μ^*. However, due to the imprecise estimation of rankings/parameters at each time step, an exact match with the oracle policy cannot be guaranteed. This discrepancy leads to varied outcomes for firms in terms of benefits (negative instantaneous regret) or losses (positive instantaneous regret) from the matching process. Instances arise where firms may strategically submit inaccurate rankings to exploit these matches, a phenomenon termed machiavelli/strategic behaviors. Nevertheless, over the long term, such strategic actions do not yield utility gains in accordance with our policy.

Furthermore, it is crucial to note that our matching solution remains a stable matching at each time step. This means that the stable matching remains independent of the negative regret generated by our policy, as stable matching is a short-term discrete metric, while regret serves as a long-term evaluation continuous metric.

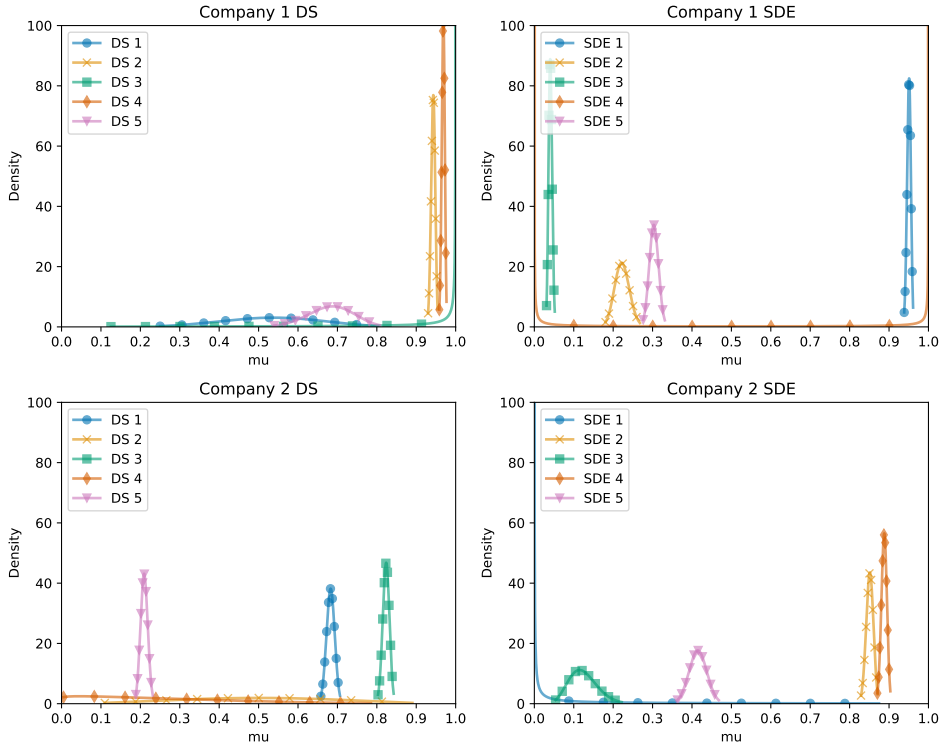

Figure 4.5: Posterior distribution of learning parameters for two firms in Example 1.

4.9.11.1 Learning

In this section, we present the learning parameters of $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of Example 1. Besides, we analyze which kind of pattern causes the non-optimal stable matching of Examples 1 and 2.

Findings from Example 1.

We show the posterior distribution of $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ in Figure 4.5. The first and second row represents the posterior distributions of firm 1 and firm 2 over two types of workers after T rounds interaction. The first and second columns in Figure 4.5 represent two firms' posterior distributions over type I and type II workers.

We find that the posterior distributions of the workers that firms most frequently match with exhibit a relatively sharp shape, indicating that firms can easily construct uncertainty sets over these workers. However, in some instances, the distributions are relatively flat, indicating a lack of exploration. This can be attributed to two possible reasons: (1) the

Table 4.2: Estimated mean reward and variance of each type of worker in view of two firms. The bold font is to represent the firm's optimal stable matching. † represents the difference between the estimated mean and the true mean less than 1%. ‡ represents the difference is less than 1.5%.

Mean & Var	**Type**	**1**	**2**	**3**	**4**	**5**
$\hat{\boldsymbol{\mu}}_1$	1 (DS)	$0.533_{0.015}$	$\mathbf{0.943}^{\ddagger}_{0.000}$	$0.917_{0.035}$	$\mathbf{0.968}^{\dagger}_{0.000}$	$0.682^{\ddagger}_{0.003}$
	2 (SDE)	$\mathbf{0.950}_{0.000}$	$0.223_{0.000}$	$\mathbf{0.041}^{\dagger}_{0.000}$	$0.500_{0.208}$	$\mathbf{0.303}^{\ddagger}_{0.000}$
$\hat{\boldsymbol{\mu}}_2$	1 (DS)	$\mathbf{0.683}^{\dagger}_{0.000}$	$0.500_{0.035}$	$\mathbf{0.823}^{\dagger}_{0.000}$	$0.262_{0.037}$	$\mathbf{0.210}^{\dagger}_{0.000}$
	2 (SDE)	$0.083_{0.035}$	$\mathbf{0.851}^{\dagger}_{0.000}$	$0.124^{\dagger}_{0.001}$	$\mathbf{0.887}^{\dagger}_{0.000}$	$0.415^{\ddagger}_{0.001}$

workers in question are not optimal stable matches for the firms, and are thus abandoned early on in the matching process, such as firm 1's DS 1 and DS 5, or (2) the workers are optimal, but are erroneously ranked by the firms and subsequently blocked, such as firm 2's SDE 3. To further illustrate this, we present the posterior mean and variance in Table 4.2. The optimal stable matches for each firm are represented in bold, and the variance of the distributions is denoted by small font. Additionally, we use the dagger symbol to indicate when the difference between the posterior mean reward and true Matching Score is less than 1% and 1.5%.

Pattern Analysis. We find that firm 1's type I matching in Figure 4.3(a), achieves a negative regret due to the high-frequency matching pattern of $\mathbf{u}_1 = \{[D_4, D_2, D_5], [S_1, S_5]\}$, and $\mathbf{u}_2 = \{[D_3, D_1], [S_4, S_2, S_3]\}$. That means firm 1 and firm 2 have a correct (stable) matching in the first match $\tilde{\mathbf{u}}_1 = \{[D_4, D_2], [S_1, S_5]\}, \tilde{\mathbf{u}}_2 = \{[D_3, D_1], [S_4, S_2]\}$. In the second match, they both need to compare worker D_5 and worker S_3, because all other workers are matched with firms or have been proposed in the first match. In Table 4.1, we find that two workers' true mean rewards for firm 1 are $\mu^1_{1,5} = 0.695, \mu^2_{1,3} = 0.040$ and two workers' estimated rewards for firm 1 are $\widehat{\mu}^1_{1,5} = 0.682, \widehat{\mu}^2_{1,3} = 0.041$. These two workers are pretty different and can be easily detected. So firm 1 has a high chance of ranking them correctly. However, two workers' true rewards for firm 2 are $\mu^1_{2,5} = 0.218, \mu^2_{2,3} = 0.131$, and two workers' estimated rewards for firm 2 are $\widehat{\mu}^1_{1,5} = 0.210, \widehat{\mu}^2_{1,3} = 0.124$. These workers are close to each

other, where these two posteriors' distributions overlap a lot and can be checked in Figure 4.5. So firm 2 has a non-negligible probability to incorrectly rank S_3 ahead of D_5. Therefore, based on the true preference, firm 2 could match with S_3 and firm 1 matches with D_5 with a non-negligible probability rather than the optimal stable matching (p_1, S_3) and (p_2, D_5) by D_5 preferring firm 2.

The above pattern links to Section 4.5.2, incapable exploration, and Section 4.6.3, incentive compatibility. Due to the insufficient exploration of S_3 and D_5, firm 2 may rank them incorrectly to get a match with S_3 rather than optimal D_3 and the regret gap is $\mu^1_{2,3} - \mu^2_{2,3} = 0.823 - 0.131 = 0.692$, which is a positive instantaneous regret. Due to the incorrect ranking from firm 2, firm 1 gets a final match with D_5 rather than optimal S_3, and suffers a regret gap $\mu^2_{1,3} - \mu^1_{1,5} = 0.040 - 0.695 = -0.655$, which is a negative instantaneous regret. Thus firm 1 benefits from firm 2's incorrect ranking and can achieve a total negative regret, as shown in Figure 4.3(a).

Findings from Example 2. In our analysis of the non-optimal stable matching in Example 2, we observed that both firms incurred positive total regret, shown in Figure 4.3(b). We find that the quota setting resulted in all workers of type II being assigned to firms in the first match. As a result, in the second match, the ranking submitted by firm 1 to the centralized platform did not affect firm 2's matching result for type II workers. This can be thought of as an analogy where firms are schools and workers are students. In the second stage of the admission process, school 2 would not participate in the competition for type II students, and its matching outcome would not be affected by the strategic behavior of other schools in the second stage, but rather by the strategic behavior of other schools in the first stage.

4.9.11.2 Large markets

In this part, we provide two large market examples to demonstrate the robustness of our algorithm. All preferences are randomly generated and all results are over 50 trials to take

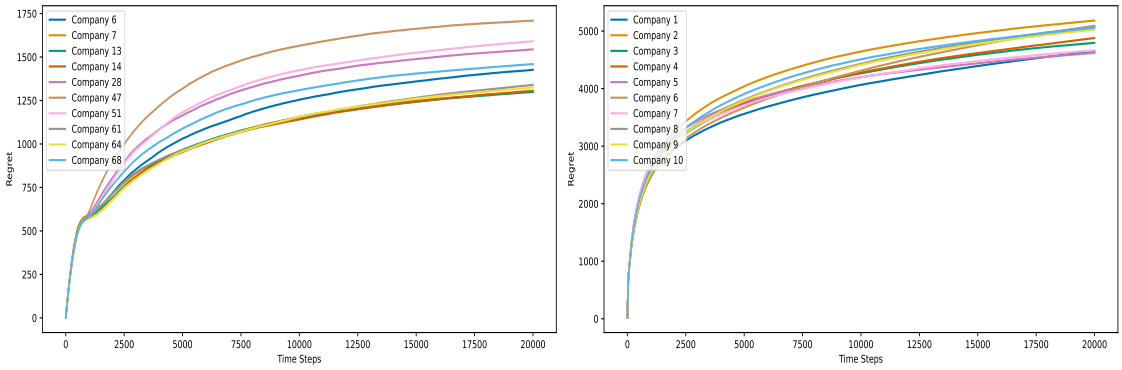

Figure 4.6: Left: 10 out of 100 randomly selected firms' total regret in Examples 3. Right: all firms' total regret in Example 4.

the average.

Example 3. We consider a large market composed of many firms ($N = 100$) and many workers ($K_1 = K_2 = 300$). Besides, we have $Q_1 = Q_2 = 3, q_1^1 = q_2^1 = q_2^1 = q_2^2 = 1$.

Example 4. We also consider a large market consisting of many workers, and each firm has a large, specified quota and an unspecified type quota. In this setting, $N = 10, M = 2, K_1 = K_2 = 500, Q_1 = Q_2 = 30, q_1^1 = q_2^1 = q_2^1 = q_2^2 = 10$.

Results. In Figure 4.6(a), we randomly select 10 out of 100 to present firms' total regret, and all those firms suffer sublinear regret. In Figure 4.6(b), we also show all 10 firms' total regret. Comparing Examples 3 and 4, we find that firms' regret in Example 3 is less than firms' regret from Example 4 because in Example 4, each firm has more quotas (30 versus 3), which demonstrates our findings from Theorem 4.2. In addition, we find there is a sudden exchange in Figure 4.6(a) nearby time $t = 1500$. We speculate this phenomenon is due to the small gap between different workers and the shifting of the explored workers.

CHAPTER 5

Conclusion

The field of AI has traditionally focused on the idea that intelligence lies solely within individual agents, such as ChatGPT, and that these agents should be able to act independently to show their intelligence without relying on human input. As a result, social complex aspects have often been overlooked when designing AI systems for use in social contexts. This limited paradigm should not be the only approach used in the development of AI. Instead, a more comprehensive approach is needed, in which AI agents are active, cooperative, and competitive, and have a vested interest in contributing to the system. To achieve this, it is important to incorporate economic and social principles into the design of AI systems and to create a more interdisciplinary approach that involves economics science, statistics science, and computer science.

In today's two-sided matching platforms, preferences are often implicit and unknown to the platform and two-sided agents involved, making it difficult for agents to match their limited best resources with those on the other side of the market. Matching problems like this arise due to the scarce resources in these markets, and agents must compete to match their best scarce resources, making it essential to design an optimal policy to maximize long-term interests. In the area of two-sided markets, preference estimation using statistical and machine learning methods have gained increasing attention in recent years because of the emergence of the large volume of data. Most of industrial-employed matching models are static and one-time recommendation, without considering the competing property, contexts shifting, existence of constraints, and incentive compatibility requirement. For instance,

state-of-the-art (LMJ20) method solved when preferences are static and one side having unknown preferences through statistical decision methods. However, it remains an open question how to handle cases where contextual information is dynamically available and how it affects the matching result over time. Another open question is how to address situations where agents' preferences are mutually or co-expressed, which can lead to instability (CKK19).

In my first project, we propose a new problem, dynamic matching problem, to make an online matching decision with dynamic preferences due to contexts shifting of arms. We find that the direct application of upper confidence bound - style estimators often fail in some matching cases if no communication mechanism exists. The reason is that the exploration collision of simultaneous pulling the same arm by multiple agents. We discover and explore this special competing characteristic in the dynamic matching market, agents' decisions interfering each other, named "incapable exploration" in short given current state-of-the-art methods. Given this competing characteristic, we design a dynamic matching algorithm. We theoretically prove that it achieves an individual $\mathcal{O}(\log(T))$ regret. We provide the regret bound analysis and show that the regret exhibits a quadratic relationship with the context dimension, noise level, and the inverse of the minimum gap, while the number of agents and decision horizon demonstrate a logarithmic correlation with the regret. In data analysis, we further show the benefit of theoretical analysis in determining the exploration length if one has no prior information about how many data points need to collect to design an optimal policy and the robustness of our algorithm with variants of noise, number of agents, and context shifting patterns and its application in the real online job market with LinkedIn text data.

In my second project, we propose a new problem, called CMCPR. This problem focuses on two-sided competing matching markets where agents have complementary preferences, meaning that their preferences are assessed through sets and one side agents (e.g., companies) have quota or headcount constraints. These complementary preferences are unknown in

advance and need be learned from historical interactive data, and the existence of these unknown preferences can lead to instability in the matching process. In CMCPR, preferences are unknown, and decisions are data driven. To solve this problem, we propose a new algorithm, called multi-agent multi-type Thompson sampling (MMTS), which formulates the problem as a two-stage bandit learning framework. MMTS uses a combination of Thompson sampling for exploration and a proposed double matching technique to achieve an individual any-time valid stable matching outcome. In theory, we first show that MMTS is effective and efficient, as it achieves stability at every matching step and provide the regret bound analysis to show the regret exhibits a square root relationship with maximum number of arms and decision horizon, while the number of quotas demonstrates a linear correlation with the regret. In addition, we prove MMTS satisfies the incentive compatibility, which is a desirable property of the mechanism where participants have a self-interested incentive to reveal their true preferences.

The thesis explores the application and limitations of AI in social contexts, challenging the traditional view that intelligence in AI systems solely relies on individual autonomous agents. It advocates for a broader, interdisciplinary approach incorporating economics, statistics, and computer science to enhance the design of AI systems. The research particularly focuses on dynamic matching problems in two-sided markets, where agents compete over scarce resources and preferences change over time due to shifting contexts. It introduces two novel online matching algorithm that addresses issues arising from simultaneous actions by multiple agents in such markets with constraints. Theoretical proofs demonstrate the algorithm's efficacy, providing a regret bound analysis showing dependencies on various factors. Practical applications of this theory are tested using data from platforms like LinkedIn, underscoring the importance of incorporating dynamic and competitive elements into AI system design to better mimic and integrate into human economic and social structures.

Bibliography

[AAS09] Yasin Abbasi-Yadkori, András Antos, and Csaba Szepesvári. "Forced-exploration based algorithms for playing in stochastic linear bandits." In *COLT Workshop on On-line Learning with Limited Feedback*, volume 92, p. 236, 2009.

[ABH11] Itai Ashlagi, Mark Braverman, and Avinatan Hassidim. "Matching with couples revisited." In *Proceedings of the 12th ACM conference on Electronic commerce*, pp. 335–336, 2011.

[ABY21] Haris Aziz, Péter Biró, and Makoto Yokoo. "Matching Market Design with Constraints." Technical report, Tech. rep., UNSW Sydney, 2021.

[ACF95] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. "Gambling in a rigged casino: The adversarial multi-armed bandit problem." In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pp. 322–331. IEEE, 1995.

[ACF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem." *Machine learning*, **47**(2):235–256, 2002.

[ACG18] Haris Aziz, Jiayin Chen, Serge Gaspers, and Zhaohong Sun. "Stability and Pareto optimality in refugee allocation matchings." In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 964–972, 2018.

[AG12] Shipra Agrawal and Navin Goyal. "Analysis of thompson sampling for the multi-armed bandit problem." In *Conference on learning theory*, pp. 39–1. JMLR Workshop and Conference Proceedings, 2012.

[AH18] Eduardo M Azevedo and John William Hatfield. "Existence of equilibrium in

large matching markets with complementarities." *Available at SSRN 3268884*, 2018.

[AKM22] Itai Ashlagi, Anilesh K Krishnaswamy, Rahul Makhijani, Daniela Saban, and Kirankumar Shiragur. "Assortment planning for two-sided sequential matching markets." *Operations Research*, **70**(5):2784–2803, 2022.

[AO06] Peter Auer and Ronald Ortner. "Logarithmic online regret bounds for undiscounted reinforcement learning." *Advances in neural information processing systems*, **19**, 2006.

[APR05a] Atila Abdulkadiroğlu, Parag A Pathak, Alvin E Roth, and Tayfun Sönmez. "The Boston public school match." *American Economic Review*, **95**(2):368–371, 2005.

[APR05b] Abdulkadiroğlu Atila, PARAG A Pathak, and ALVIN E Roth. "The New York City High School Match." In *American Economic Review, Papers and Proceedings*, volume 95, 2005.

[AR95] Hernan Abeledo and Uriel G Rothblum. "Paths to marriage stability." *Discrete applied mathematics*, **63**(1):1–12, 1995.

[AS22] Ali Aouad and Daniela Saban. "Online assortment optimization for two-sided matching platforms." *Management Science*, 2022.

[ATK14] Nikolaos D Almalis, George A Tsihrintzis, and Nikolaos Karagiannis. "A content based approach for recommending personnel for job positions." In *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, pp. 45–49. IEEE, 2014.

[Aue02] Peter Auer. "Using confidence bounds for exploitation-exploration trade-offs." *Journal of Machine Learning Research*, **3**(Nov):397–422, 2002.

[BB20] Hamsa Bastani and Mohsen Bayati. "Online decision making with high-dimensional covariates." *Operations Research*, **68**(1):276–294, 2020.

[BC12] Sébastien Bubeck and Nicolo Cesa-Bianchi. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems." *arXiv preprint arXiv:1204.5721*, 2012.

[BH22] Niclas Boehmer and Klaus Heeger. "A fine-grained view on stable many-to-one matching problems with lower and upper quotas." *ACM Transactions on Economics and Computation*, **10**(2):1–53, 2022.

[BK18] Lilian Besson and Emilie Kaufmann. "What doubling tricks can and can't do for multi-armed bandits." *arXiv preprint arXiv:1803.06971*, 2018.

[BMM14] Péter Biró, David F Manlove, and Iain McBride. "The hospitals/residents problem with couples: Complexity and integer programming models." In *International Symposium on Experimental Algorithms*, pp. 10–21. Springer, 2014.

[BSS21] 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*, pp. 705–715. PMLR, 2021.

[CGZ22] Pinhan Chen, Chao Gao, and Anderson Y Zhang. "Optimal full ranking from pairwise comparisons." *The Annals of Statistics*, **50**(3):1775–1805, 2022.

[CKK19] Yeon-Koo Che, Jinwoo Kim, and Fuhito Kojima. "Stable matching in large economies." *Econometrica*, **87**(1):65–110, 2019.

[CLS21a] Haoyu Chen, Wenbin Lu, and Rui Song. "Statistical inference for online decision making: In a contextual bandit setting." *Journal of the American Statistical Association*, **116**(533):240–255, 2021.

[CLS21b] Haoyu Chen, Wenbin Lu, and Rui Song. "Statistical inference for online decision making: In a contextual bandit setting." *Journal of the American Statistical Association*, **116**(533):240–255, 2021.

[CLS21c] Haoyu Chen, Wenbin Lu, and Rui Song. "Statistical inference for online decision making via stochastic gradient descent." *Journal of the American Statistical Association*, **116**(534):708–719, 2021.

[CS22] Sarah H Cen and Devavrat Shah. "Regret, stability & fairness in matching markets with bandit learners." In *International Conference on Artificial Intelligence and Statistics*, pp. 8938–8968. PMLR, 2022.

[CSW22] Xi Chen, David Simchi-Levi, and Yining Wang. "Privacy-preserving dynamic personalized pricing with demand learning." *Management Science*, **68**(7):4878–4898, 2022.

[DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*, 2018.

[DF81] Lester E Dubins and David A Freedman. "Machiavelli and the Gale-Shapley algorithm." *The American Mathematical Monthly*, **88**(7):485–494, 1981.

[DJ21a] Xiaowu Dai and Michael I Jordan. "Learning in Multi-Stage Decentralized Matching Markets." *Advances in Neural Information Processing Systems*, **34**, 2021.

[DJ21b] Xiaowu Dai and Michael I Jordan. "Learning strategies in decentralized matching markets under uncertain preferences." *Journal of Machine Learning Research*, **22**(260):1–50, 2021.

[DK05] Sanmay Das and Emir Kamenica. "Two-Sided Bandits and the Dating Market." In *IJCAI*, volume 5, p. 19. Citeseer, 2005.

[DQJ22] Xiaowu Dai, Yuan Qi, and Michael I Jordan. "Incentive-aware recommender systems in two-sided markets." *arXiv preprint arXiv:2211.15381*, 2022.

[FCR19] Tanner Fiez, Benjamin Chasnov, and Lillian J Ratliff. "Convergence of learning dynamics in stackelberg games." *arXiv preprint arXiv:1906.01217*, 2019.

[GAH16] Shiqiang Guo, Folami Alamudun, and Tracy Hammond. "RésuMatcher: A personalized résumé-job matching system." *Expert Systems with Applications*, **60**:169–182, 2016.

[GH13] David González-Sánchez and Onésimo Hernández-Lerma. *Discrete–time stochastic control and dynamic potential games: the Euler–Equation approach*. Springer Science & Business Media, 2013.

[GI89] Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms*. MIT press, 1989.

[GK21] Michael Greinecker and Christopher Kah. "Pairwise stable matching in large economies." *Econometrica*, **89**(6):2929–2974, 2021.

[GLK16] Aurélien Garivier, Tor Lattimore, and Emilie Kaufmann. "On explore-then-commit strategies." *Advances in Neural Information Processing Systems*, **29**, 2016.

[GM20] Akshay Gugnani and Hemant Misra. "Implicit skills extraction using document embedding and its use in job recommendation." In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13286–13293, 2020.

[GS62] David Gale and Lloyd S Shapley. "College admissions and the stability of marriage." *The American Mathematical Monthly*, **69**(1):9–15, 1962.

[HL22] Botao Hao and Tor Lattimore. "Regret bounds for information-directed reinforcement learning." *Advances in Neural Information Processing Systems*, **35**:28575–28587, 2022.

[HLQ22] Botao Hao, Tor Lattimore, and Chao Qin. "Contextual information-directed sampling." In *International Conference on Machine Learning*, pp. 8446–8464. PMLR, 2022.

[HSZ22] Qiyu Han, Will Wei Sun, and Yichen Zhang. "Online statistical inference for matrix contextual bandit." *arXiv preprint arXiv:2212.11385*, 2022.

[HT22] Justin Hadad and Alexander Teytelboym. "Improving refugee resettlement: insights from market design." *Oxford Review of Economic Policy*, **38**(3):434–448, 2022.

[ILM21] Nicole Immorlica, Brendan Lucier, Vahideh Manshadi, and Alexander Wei. "Designing approximately optimal search on matching platforms." In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 632–633, 2021.

[JC16] Ian T Jolliffe and Jorge Cadima. "Principal component analysis: a review and recent developments." *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, **374**(2065):20150202, 2016.

[JJH22] Meena Jagadeesan, Michael I Jordan, and Nika Haghtalab. "Competition, Alignment, and Equilibria in Digital Marketplaces." *arXiv preprint arXiv:2208.14423*, 2022.

[JNJ20] Chi Jin, Praneeth Netrapalli, and Michael Jordan. "What is local optimality in nonconvex-nonconcave minimax optimization?" In *International conference on machine learning*, pp. 4880–4889. PMLR, 2020.

[JWW21] Meena Jagadeesan, Alexander Wei, Yixin Wang, Michael I Jordan, and Jacob Steinhardt. "Learning Equilibria in Matching Markets from Bandit Feedback." *Advances in Neural Information Processing Systems*, **34**, 2021.

[KHN15] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. "Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays." In *International Conference on Machine Learning*, pp. 1152–1161. PMLR, 2015.

[KK05] Bettina Klaus and Flip Klijn. "Stable matchings and preferences of couples." *Journal of Economic Theory*, **121**(1):75–106, 2005.

[Knu76] Donald Ervin Knuth. "Marriages stables." *Technical report*, 1976.

[Knu97] Donald Ervin Knuth. *Stable marriage and its relation to other combinatorial problems: An introduction to the mathematical analysis of algorithms*, volume 10. American Mathematical Soc., 1997.

[KSV19] Branislav Kveton, Csaba Szepesvari, Sharan Vaswani, Zheng Wen, Tor Lattimore, and Mohammad Ghavamzadeh. "Garbage in, reward out: Bootstrapping exploration in multi-armed bandits." In *International Conference on Machine Learning*, pp. 3601–3610. PMLR, 2019.

[KTY18] Fuhito Kojima, Akihisa Tamura, and Makoto Yokoo. "Designing matching mechanisms under constraints: An approach from discrete convex analysis." *Journal of Economic Theory*, **176**:803–833, 2018.

[LC18] Mustafa Lokhandwala and Hua Cai. "Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of NYC." *Transportation Research Part C: Emerging Technologies*, **97**:45–60, 2018.

[LCD23] Yuantong Li, Guang Cheng, and Xiaowu Dai. "Double Matching Under Complementary Preferences." *arXiv preprint arXiv:2301.10230*, 2023.

[LCW22] Gen Li, Yuejie Chi, Yuting Wei, and Yuxin Chen. "Minimax-optimal multi-agent RL in markov games with a generative model." In *Advances in Neural Information Processing Systems*, 2022.

[Lit94] Michael L Littman. "Markov games as a framework for multi-agent reinforcement learning." In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

[Lit01] Michael L Littman. "Value-function reinforcement learning in Markov games." *Cognitive systems research*, **2**(1):55–66, 2001.

[LLD24] Jiayi Li, Yuantong Li, and Xiaowu Dai. "Jiayi Li, Yuantong Li and Xiaowu Dai's contribution to the Discussion of 'Estimating means of bounded random variables by betting'by Waudby-Smith and Ramdas." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **86**(1):41–43, 2024.

[LLK19] Daniel J Luckett, Eric B Laber, Anna R Kahkoska, David M Maahs, Elizabeth Mayer-Davis, and Michael R Kosorok. "Estimating dynamic treatment regimes in mobile health using v-learning." *Journal of the American Statistical Association*, 2019.

[LMJ20] Lydia T Liu, Horia Mania, and Michael I Jordan. "Competing bandits in matching markets." In *International Conference on Artificial Intelligence and Statistics*, pp. 1618–1628. PMLR, 2020.

[LR85] Tze Leung Lai and Herbert Robbins. "Asymptotically efficient adaptive allocation rules." *Advances in applied mathematics*, **6**(1):4–22, 1985.

[LRM21] Lydia T Liu, Feng Ruan, Horia Mania, and Michael I Jordan. "Bandit learn-

ing in decentralized matching markets." *Journal of Machine Learning Research*, **22**(211):1–34, 2021.

[LRS83] Tze Leung Lai, Herbert Robbins, and David Siegmund. "Sequential design of comparative clinical trials." In *Recent Advances in Statistics*, pp. 51–68. Elsevier, 1983.

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[LWC21] Yuantong Li, Chi-Hua Wang, and Guang Cheng. "Online Forgetting Process for Linear Regression Models." In *International Conference on Artificial Intelligence and Statistics*, pp. 217–225. PMLR, 2021.

[LWC22] Yuantong Li, Chi-hua Wang, Guang Cheng, and Will Wei Sun. "Rate-optimal contextual online matching bandit." *arXiv preprint arXiv:2205.03699*, 2022.

[Mai19] Odalric-Ambrym Maillard. *Mathematics of statistical sequential decision making*. PhD thesis, Université de Lille, Sciences et Technologies, 2019.

[MKO13] Tsunenori Mine, Tomoyuki Kakuta, and Akira Ono. "Reciprocal recommendation for job matching with bidirectional feedback." In *2013 Second IIAI International Conference on Advanced Applied Informatics*, pp. 39–44. IEEE, 2013.

[MMT17] David F Manlove, Iain McBride, and James Trimble. ""Almost-stable" matchings in the Hospitals/Residents problem with Couples." *Constraints*, **22**(1):50–72, 2017.

[MWX22] 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." *Advances in Neural Information Processing Systems*, **35**:19956–19970, 2022.

[NV18] Thanh Nguyen and Rakesh Vohra. "Near-feasible stable matchings with couples." *American Economic Review*, **108**(11):3154–69, 2018.

[NV19] Thành Nguyen and Rakesh Vohra. "Stable matching with proportionality constraints." *Operations Research*, **67**(6):1503–1519, 2019.

[NV22] Thanh Nguyen and Rakesh Vohra. "Complementarities and Externalities." 2022.

[Pea01] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space." *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, **2**(11):559–572, 1901.

[PPP18] Julien Perolat, Bilal Piot, and Olivier Pietquin. "Actor-critic fictitious play in simultaneous move multistage games." In *International Conference on Artificial Intelligence and Statistics*, pp. 919–928. PMLR, 2018.

[QLF20] Zhengling Qi, Dacheng Liu, Haoda Fu, and Yufeng Liu. "Multi-armed angle-based direct learning for estimating optimal individualized treatment rules with various outcomes." *Journal of the American Statistical Association*, **115**(530):678–691, 2020.

[RLY23] Pratik Ramprasad, Yuantong Li, Zhuoran Yang, Zhaoran Wang, Will Wei Sun, and Guang Cheng. "Online bootstrap inference for policy evaluation in reinforcement learning." *Journal of the American Statistical Association*, **118**(544):2901–2914, 2023.

[Rob52] Herbert Robbins. "Some aspects of the sequential design of experiments." 1952.

[Rot82] Alvin E Roth. "The economics of matching: Stability and incentives." *Mathematics of operations research*, **7**(4):617–628, 1982.

[Rot84] 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.

[Rot85] Alvin E Roth. "The college admissions problem is not equivalent to the marriage problem." *Journal of economic Theory*, **36**(2):277–288, 1985.

[Rot86] Alvin E Roth. "On the allocation of residents to rural hospitals: a general property of two-sided matching markets." *Econometrica: Journal of the Econometric Society*, pp. 425–427, 1986.

[Rot08] Alvin E Roth. "Deferred acceptance algorithms: History, theory, practice, and open questions." *International Journal of game Theory*, **36**(3):537–569, 2008.

[RS92] Alvin E Roth and Marilda Sotomayor. "Two-sided matching." *Handbook of game theory with economic applications*, **1**:485–541, 1992.

[RSZ22] Ignacio Rios, Daniela Saban, and Fanyin Zheng. "Improving match rates in dating markets through assortment optimization." *Manufacturing & Service Operations Management*, 2022.

[RV13] Daniel Russo and Benjamin Van Roy. "Eluder dimension and the sample complexity of optimistic exploration." *Advances in Neural Information Processing Systems*, **26**, 2013.

[RV14] Daniel Russo and Benjamin Van Roy. "Learning to optimize via information-directed sampling." *Advances in Neural Information Processing Systems*, **27**, 2014.

[RVK18] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. "A tutorial on thompson sampling." *Foundations and Trends® in Machine Learning*, **11**(1):1–96, 2018.

[Sar21] Soumajyoti Sarkar. "Bandit based centralized matching in two-sided markets for peer to peer lending." *arXiv preprint arXiv:2105.02589*, 2021.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[SBS21] 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*, pp. 1252–1260. PMLR, 2021.

[Shi22] Peng Shi. "Optimal Matchmaking Strategy in Two-sided Marketplaces." *Management Science*, 2022.

[Sli19] Aleksandrs Slivkins. "Introduction to multi-armed bandits." *arXiv preprint arXiv:1904.07272*, 2019.

[Son97] Tayfun Sönmez. "Manipulation via capacities in two-sided matching markets." *Journal of Economic theory*, **77**(1):197–204, 1997.

[SWL23] Chengchun Shi, Xiaoyu Wang, Shikai Luo, Hongtu Zhu, Jieping Ye, and Rui Song. "Dynamic causal effects evaluation in a/b testing with a reinforcement learning framework." *Journal of the American Statistical Association*, **118**(543):2059–2071, 2023.

[SWS22] Chengchun Shi, Runzhe Wan, Ge Song, Shikai Luo, Rui Song, and Hongtu Zhu. "A Multi-Agent Reinforcement Learning Framework for Off-Policy Evaluation in Two-sided Markets." *arXiv preprint arXiv:2202.10574*, 2022.

[SWS23] Chengchun Shi, Runzhe Wan, Ge Song, Shikai Luo, Hongtu Zhu, and Rui Song. "A multiagent reinforcement learning framework for off-policy evaluation in two-sided markets." *The Annals of Applied Statistics*, **17**(4):2701–2722, 2023.

[SZL22] Chengchun Shi, Sheng Zhang, Wenbin Lu, and Rui Song. "Statistical inference of the value function for reinforcement learning in infinite-horizon settings." *Journal of the Royal Statistical Society Series B*, **84**(3):765–793, 2022.

[Tho33] William R Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples." *Biometrika*, **25**(3-4):285–294, 1933.

[Tom18] Kentaro Tomoeda. "Finding a stable matching under type-specific minimum quotas." *Journal of Economic Theory*, **176**:81–117, 2018.

[Tro15] Joel A Tropp et al. "An introduction to matrix concentration inequalities." *Foundations and Trends® in Machine Learning*, **8**(1-2):1–230, 2015.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[VPD22] Keyon Vafa, Emil Palikot, Tianyu Du, Ayush Kanodia, Susan Athey, and David M Blei. "CAREER: Transfer Learning for Economic Prediction of Labor Sequence Data." *arXiv preprint arXiv:2202.08370*, 2022.

[Wai19] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[WHL17] Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. "Online reinforcement learning in stochastic games." *Advances in Neural Information Processing Systems*, **30**, 2017.

[WWL22] Shuang Wu, Chi-Hua Wang, Yuantong Li, and Guang Cheng. "Residual bootstrap exploration for stochastic linear bandit." In *Uncertainty in Artificial Intelligence*, pp. 2117–2127. PMLR, 2022.

[WWR22] Ian Waudby-Smith, Lili Wu, Aaditya Ramdas, Nikos Karampatziakis, and Paul Mineiro. "Anytime-valid off-policy inference for contextual bandits." *ACM/JMS Journal of Data Science*, 2022.

[WWS23] Chi-Hua Wang, Zhanyu Wang, Will Wei Sun, and Guang Cheng. "Online Regularization toward Always-Valid High-Dimensional Dynamic Pricing." *Journal of the American Statistical Association*, pp. 1–13, 2023.

[WYH20] Chi-Hua Wang, Yang Yu, Botao Hao, and Guang Cheng. "Residual bootstrap exploration for bandit algorithms." *arXiv preprint arXiv:2002.08436*, 2020.

[ZBB18] Luisa Zap, Joris van Breugel, D Bakker, and S Bels. "Swiping Right vs. Finding Mr. Right: Facebook Attempts to Reinvent Online Dating." In *New media and digital culture*. 2018.

[ZJB07] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. "Regret minimization in games with incomplete information." *Advances in neural information processing systems*, **20**, 2007.

[ZYL18] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. "Finite-sample analyses for fully decentralized multi-agent reinforcement learning." *arXiv preprint arXiv:1812.02783*, 2018.

[ZYW21] Han Zhong, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. "Can Reinforcement Learning Find Stackelberg-Nash Equilibria in General-Sum Markov Games with Myopic Followers?" *arXiv preprint arXiv:2112.13521*, 2021.