Journal of Data Science 24 (4), 785–801
October 2026
DOI: 10.6339/24-JDS1151
Statistical Data Science

Efficient UCB-Based Assignment Algorithm Under Unknown Utility with Application in Mentor-Mentee Matching

Yuyang Shi[1,*,†] and Yajun Mei[2]

[1]*H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA*
[2]*Department of Biostatistics, School of Global Public Health, New York University, New York, NY, USA*

Abstract

The assignment problem, crucial in various real-world applications, involves optimizing the allocation of agents to tasks for maximum utility. While it has been well-studied in the optimization literature when the underlying utilities between all agent-task pairs are known, research is sparse when the utilities are unknown and need to be learned from data on the fly. This paper addresses this gap, as motivated by mentor-mentee matching programs at many U.S. universities. We develop an efficient sequential assignment algorithm, with the aim of nearly maximizing the overall utility simultaneously over different time periods. Our proposed algorithm is to use stochastic bandit feedback to adaptively estimate the unknown utilities through linear regression models, integrating the Upper Confidence Bound (UCB) algorithm in the multi-armed bandit problem with the Hungarian algorithm in the assignment problem. We provide theoretical bounds of our algorithm for both the estimation error and the total regret. Additionally, numerical studies are also conducted to demonstrate the practical effectiveness of our algorithm.

Keywords *bandits; estimation; optimal assignment; upper confidence bound*

1 Introduction

The assignment problem is classical in combinatorial optimization, with many real-world applications such as allocation of workers or resources for optimal utility gain. Under a general setup, one is given an equal number of agents and tasks along with the utility associated with every possible agent-task pair, and seeks to find a one-to-one mapping between the agents and tasks that yields maximal total utility. When the underlying utilities are known, the problem is well-studied in the combinatorial optimization literature. For instance, Kuhn (1955) first proposed the well-known Hungarian algorithm, which provides the optimal solution in polynomial time.

However, in many real-world applications, the underlying utility is often unknown and must be learned from data dynamically. The motivating example of our research is the mentor-mentee program of the Office of Alumni Relations (OAR) in many U.S. colleges and universities. Such programs are typically held regularly with the goal of facilitating the professional development and network building of students under the supervision of alumni. During each matching cycle, the Alumni office needs to decide how to suitably pair mentors and mentees by considering many

*Corresponding author. Email: syuyang123@gmail.com.
†This paper is part of the first author's PhD dissertation at Georgia Institute of Technology.

© 2026 The Author(s). Published by the School of Statistics and the Center for Applied Statistics, Renmin University of China. Open access article under the CC BY license.
Received June 18, 2024; Accepted August 22, 2024

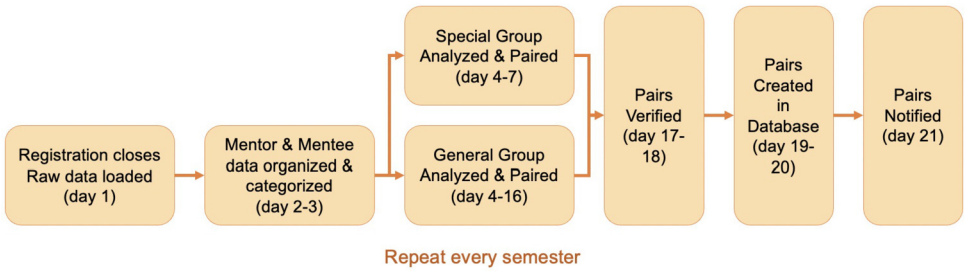


Figure 1: Example of a pipeline for information collection and manual matching between mentors and mentees for one batch of participants.

factors of background information, such as majors, status (upper-class and underclass undergraduates, M.S., Ph.D. students, or other special groups such as athletes), location preference, etc., to maximize the satisfaction of participants. At the end of each semester or year, a survey is distributed to participants to collect their feedback on satisfaction.

The matching process between mentors and mentees may vary based on the school and program. In some programs, the pairing assignments are conducted centrally by the OAR (e.g., the alumni mentoring programs at Princeton University, Yale University, and Georgia Institute of Technology). In other programs (e.g., those at UCLA, University of Georgia, and the School of Medicine at the University of Pennsylvania), student mentees may search for and send requests to potential alumni mentors independently, and mentors can accept or decline the pairing requests. We emphasize that the pairing process can be labor-intensive and time-consuming, especially when dealing with thousands of mentors or mentees. This inspires us to develop an algorithm for general sequential assignment problems with unknown utility and bandit feedback.

In this work, we focus on the case when the pairing assignments are conducted centrally. Our goal is to develop an algorithm to efficiently learn the utility function from data on the fly and find near-optimal matching for every round, in order to maximize the overall utility. Note that since the utility function is unknown, it is critical to balance the trade-off between exploration and exploitation. On the one hand, one wants to exploit the information from previous observations to infer the utility and seek the best decision for the current. On the other hand, it is also important to sample the future data wisely to improve the estimation for the unknown utility. To address this, we propose to bring the ideas from the multi-armed bandit problems to develop an efficient sequential assignment algorithm, with the objective of nearly maximizing the overall utility simultaneously for each round. At a high level, our method combines the upper confidence bound (UCB) algorithm with the Hungarian algorithm for optimal assignment in the context of the unknown utility function. Our underlying assumption is that the utility function does not change rapidly over any relatively short period, which might sound reasonable in many applications such as mentor-mentee matching.

Theoretically, we show that our method incurs a cumulative regret bounded by $\widetilde{O}(n\sqrt{dT})$, where n is the number of pairs for assignment at each round, d is the data dimension, T is the total number of rounds, and $\widetilde{O}$ hides the logarithm terms. Numerical experiments are conducted to show the usefulness of our algorithm. In addition, a further study of the mentor-mentee matching scenario is discussed to illustrate our method.

Below it is useful to provide a brief literature review. From the methodology perspective, our work is closely related to the area of combinatorial semi-bandit, where each time the player needs to pull a collection of arms (called super-arm) subjected to certain constraints and pursue

to maximize the overall reward, see for example Cesa-Bianchi and Lugosi (2012); Gai et al. (2012); Chen et al. (2013); Perrault et al. (2020). In these studies, the agents and tasks are fixed at each round, and the player wants to learn the optimal assignment through bandit feedback of utility from matched pairs, which is different from our setting, where the agents and tasks can change constantly. Based on this conception, Wen et al. (2015) investigate the contextual version of the combinatorial semi-bandit with a linear model for the payoff, and consider the general oracle algorithm for the combinatorial optimization problem as a sub-routine. Different from their study, our work utilizes the concrete Hungarian algorithm for the specific assignment problem at each time, combining with the linear regression model to learn the utility function.

We further note other fields related to our study from the application perspective. Firstly, there has been some studies on optimizing the matching between groups of subjects in single-round manner in some applications, such as in a mentoring or supporting program (Fang and Zhu, 2022; Biró and Gyetvai, 2023), which might involve more complex constraints but without the need of utility estimation. Secondly, our problem involves parameter estimation in a sequential manner, which is related to the field of sequential estimation (Anscombe, 1953; Ghosh et al., 2011), as well as online learning and optimization (Anderson, 2008; Shalev-Shwartz, 2012; Hazan, 2016), while in our problem we further consider the optimal assignment based on the estimation, and want to nearly maximize the total utility through all times. Our research is also remotely related to the so-called reciprocal recommendation systems in applications such as online friend recommendation (Pizzato et al., 2010; Xia et al., 2015), where the system recommends users potential partners based on their profiles, and learn the strategy for finding good pairs. Such a system gives a number of top recommendations for each user without conducting the assignments among users, as different from our setting. Another line of recent research, known as multitasking bandits, investigates adaptive decision making and estimation in multi-armed bandits with a multi-objective formulation (Yang et al., 2017; Deshmukh et al., 2017; Erraqabi et al., 2017; Simchi-Levi and Wang, 2023), and derives optimality results on trade-off between regret and estimation error under the classical or contextual bandits setting. In addition, a short version of this paper with a different setting has appeared in a conference paper Shi and Mei (2022), where a logistic regression model for binary outcomes was considered.

The rest of the paper is organized as below. Section 2 introduces the problem formulation and relative background. Section 3 develops our proposed UCB-based sequential assignment algorithm, and Section 4 presents the theoretical results of our algorithm. Section 5 presents the results of numerical studies, and Section 6 includes a further study under the mentor-mentee matching scenario. The concluding remarks are summarized in Section 7.

Notations For $n \in \mathbb{N}$, we denote $[n]$ as the set $\{1, 2, \ldots, n\}$. For a d-dimensional vector $\boldsymbol{v} = (v_1, \ldots, v_d)$, we define the vector ℓ_2 norm $\|\boldsymbol{v}\|_2 = \sqrt{\sum_{k=i}^{d} v_i^2}$ and the matrix norm $\|v\|_M = \sqrt{v^\top M v}$, where M is a $d \times d$ positive definite matrix. We use $P(\cdot)$ to denote the probability of events, and $\mathbb{E}[\cdot]$ to denote the expectation of random variables. We use $I(\cdot)$ to denote the indicator function.

2 Problem Formulation and Background

In this section, we present the formulation of our sequential assignment problem with unknown utility.

For each time period $t = 1, 2, \ldots, T$, we are given n_t agents and tasks, where each agent or task is associated with a vector of covariates, also referred to as side information or context. Let $\{\boldsymbol{x}_i^t : i \in [n_t]\} \subset \mathcal{X}$ and $\{\boldsymbol{z}_i^t : i \in [n_t]\} \subset \mathcal{Z}$ denote the collection of covariates for agents and tasks at time t, where $\mathcal{X}$ and $\mathcal{Z}$ are the spaces of the corresponding covariates. We need to determine an assignment, denoted by δ_t, between these agents and tasks. After the pairing is conducted, we observe the utility associated with each matched pair $(\boldsymbol{x}_i^t, \boldsymbol{z}_{\delta_t(i)}^t)$, denoted by $U(\boldsymbol{x}_i^t, \boldsymbol{z}_{\delta_t(i)}^t)$. Our objective is to maximize the total utility gained up to time T. The procedure can be summarized in an online learning framework as follows. For each round $t = 1, 2, \ldots, T$:

1. The system receives the covariates of agents and tasks, $\{\boldsymbol{x}_i^t : i \in [n_t]\}$ and $\{\boldsymbol{z}_i^t : i \in [n_t]\}$.
2. The system determines an assignment, denoted by a one-to-one mapping $\delta_t : [n_t] \to [n_t]$.
3. The system observes the utility feedback for every assigned pair, i.e., $\{U(\boldsymbol{x}_i^t, \boldsymbol{z}_{\delta_t(i)}^t) : i \in [n_t]\}$.

Our goal is to decide the assignment δ_t at each round t such that the overall expected utility $\sum_{t=1}^{T} \sum_{i=1}^{n_t} \mathbb{E}[U(\boldsymbol{x}_i^t, \boldsymbol{z}_{\delta_t(i)}^t]$ is maximized.

Although the underlying utility of any agent-task pair is unknown at the time of assignment, we assume that the utility is related to the covariates of the agent and task through some noisy function. Specifically, for any pair of covariates $(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t)$, we assume that the associated utility $U(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t)$ satisfies:

$$U\big(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t\big) = \boldsymbol{\phi}\big(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t\big)^\top \theta^* + \epsilon, \tag{1}$$

where $\boldsymbol{\phi}$ is a d-dimensional transformation, θ^* is a d-dimensional unknown parameter, and ϵ follows a σ-sub-Gaussian distribution with mean 0 and is independent of $\boldsymbol{x}$ and $\boldsymbol{z}$. We further define:

$$u\big(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t\big) = \mathbb{E}\big[U\big(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t\big)\big] = \boldsymbol{\phi}\big(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t\big)^\top \theta^*$$

as the expected utility associated with the pair. For notational simplicity, in the rest of the paper, we may use the shorthands $\boldsymbol{\phi}_{i,j}^t = \boldsymbol{\phi}(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t)$, $U_{i,j}^t = U(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t)$, and $u_{i,j}^t = u(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t)$ when there is no confusion.

In our work, we assume that the transformation function $\boldsymbol{\phi}$ is known. However, in real applications, identifying the suitable transformation $\boldsymbol{\phi}$ or function class can be challenging, as it usually depends on the specific problem and data implicitly. In such cases, one might consider non-parametric models to approximate the underlying utility function, which is of independent interest to our work.

To measure the performance of assignments conducted by a given algorithm $\mathcal{A}$, we first define the oracle assignment δ_t^* at each round t as the assignment that maximizes the total expected utility:

$$\delta_t^* \in \arg\max_{\delta} \sum_{i=1}^{n_t} u\big(\boldsymbol{x}_i^t, \boldsymbol{z}_{\delta(i)}^t\big).$$

Since u depends on the unknown parameter θ^*, δ_t^* is also unknown in practice when one conducts the assignment. With the definition of δ_t^*, we further define the cumulative regret of an algorithm $\mathcal{A}$ as:

$$R_T(\mathcal{A}) = \sum_{t=1}^{T} \left\{ \frac{1}{n_t} \sum_{i=1}^{n_t} \big[u\big(\boldsymbol{x}_i^t, \boldsymbol{z}_{\delta_t^*(i)}^t\big) - u\big(\boldsymbol{x}_i^t, \boldsymbol{z}_{\delta_t(i)}^t\big)\big] \right\}, \tag{2}$$

where δ_t is the assignment conducted by $\mathcal{A}$, and δ_t^* is the oracle assignment at time t. $R_T(\mathcal{A})$ captures the performance gap between $\mathcal{A}$ and the oracle performance in expected utility. Ideally, we aim to design an algorithm with the total regret R_T as small as possible, preferably sublinear in T.

In the regret defined in (2), we scale the utility gap by the number of pairs n_t at each time, eliminating the size differences across time and focusing on the average utility gap. Alternatively, one might consider the regret defined as the summation of the total utility gap directly, without averaging at each time. The choice of the criteria for regret can depend on the specific problem of interest. In this paper, we discuss the regret bound with regret defined as in (2), while the results can be naturally adapted to the alternative definition.

3 Our Proposed Method

In this section, we introduce our proposed UCB-based algorithm for the sequential assignment problem with unknown utility. At a high level, our method integrates two main components for every round: 1) Constructing the upper confidence bound for every agent-task pair based on past data; 2) Deciding the assignment by maximizing the total upper bound of the expected utility. Below we describe each component of our proposed UCB-based method in details, and later summarize our algorithm. For presentation convenience, we first introduce the algorithm for finding the matching to maximize the upper confidence bound on total utility, and then introduce the method for constructing the upper confidence bound.

3.1 Assignment to Maximizes Total Utility Upper Bound

In this subsection, we suppose that the upper confidence bound $b_{i,j}^t$ for every possible agent-task pair $(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t)$ at time t is readily available, and we want to find the assignment for time t to maximize the upper confidence bound for the total utility. For this goal, we solve the following optimization problem:

$$\begin{aligned} \max_{\delta_{i,j}} \quad & \sum_{i\in[n_t]}\sum_{j\in[n_t]} b_{i,j}^t \delta_{i,j} \\ \text{subject to} \quad & \sum_{i=1}^{n_t} \delta_{i,j} = 1, \ \forall j, \\ & \sum_{j=1}^{n_t} \delta_{i,j} = 1, \ \forall i, \\ & \delta_{i,j} \in \{0,1\}, \ \forall i, j. \end{aligned} \tag{3}$$

Here $\delta_{i,j}$'s are binary decision variables, and $\delta_{i,j} = 1$ indicates $\boldsymbol{x}_i^t$ and $\boldsymbol{z}_j^t$ are matched with each other, and $\delta_{i,j} = 0$ otherwise. Note that here we maximize the objective of the total upper confidence bound of the utility. Our approach is inspired by the UCB-algorithm for classical multi-armed bandit and contextual bandit with linear payoff, as discussed in Lai and Robbins (1985), Chu et al. (2011) and Abbasi-Yadkori et al. (2011) among others, where the player pull the arm with highest upper confidence bound of reward at each round. By considering the upper confidence bound instead of the utility estimate, one can balance the exploitation with exploration. In later sections, we will show that with proper choice of some tuning parameter,

our algorithm can balance the exploration and exploitation, and thus enjoys desired theoretical property in cumulative regret. To solve problem (3), we can use the classical Hungarian algorithm, among other alternative methods.

We now briefly introduce the Hungarian algorithm, which is developed by Kuhn (1955). At high level, it considers the dual problem of (3), which can be re-written as:

$$\begin{aligned} \min_{u,v} \quad & \sum_{i\in[n_t]} u_i + \sum_{j\in[n_t]} v_j, \\ \text{subject to} \quad & u_i + v_j \geqslant b^t_{i,j}, \ \forall i, j. \end{aligned}$$

The algorithm utilizes the primal-dual method to update the solution until the optimal objective is reached for both primal and dual problems. Algorithm 1 below describes the classical Hungarian algorithm in Kuhn (1955) in the matrix form. Note that the algorithm is not related with t, we will omit t from upper and lower script in the followings.

Algorithm 1 The Hungarian algorithm for optimal assignment.

Specify n the number of agents and tasks; Specify the matrix $(b_{ij}) \in \mathbb{R}^{n\times n}$. Set the matrix $C \in \mathbb{R}^{n\times n}$ such that $C_{i,j} = \max_{k,\ell} b_{k,\ell} - b_{i,j}$.

- Step i. Subtract $\min_{i,j} C_{i,j}$ from each element of C and obtain a matrix C_1.
- Step ii. Find a minimum set S_1 of lines (rows or columns) that includes all null elements in C_1. Let $n_1 = |S_1|$. If $n_1 = n$, then report the n positions of null elements as the required solution.
- Step iii. If $n_1 < n$, let h_1 be the smallest element in C_1 that is not in any line in S_1. Add h_1 to any elements in a line of S_1 and subtract h_1 from any elements in C_1. Let the resulting matrix be C_2.
- Step iv. Repeat the Steps 2 and 3 starting with C_2, until $n_k = n$ at some stage. Report the positions of these null elements.

Note that the computational complexity of Algorithm 1 is $O(n^4)$. The classical Hungarian algorithm is later improved by Tomizawa (1971) and Edmonds and Karp (1972) to achieve an $O(n^3)$ complexity. In addition, there are many approximation algorithms for optimal assignment with less computational cost and near optimality, see Kurtzberg (1962); Avis (1983); Duan and Pettie (2014) among others. One might consider adopt such approximation algorithms for problems with large scales.

3.2 Constructing Upper Confidence Bound

In this subsection, we specify the construction of the confidence bound based on past data in our method. Loosely speaking, at the beginning of each time, we use the ridge regression to fit the past observations and obtain the estimate for θ^*, and construct the confidence bound accordingly. More specifically, we maintain a $d \times d$ matrix M_t and a d-dimensional vector r_t through the process, such that with some parameter α to be determined, we have

$$M_t = \alpha I_d + \sum_{\tau=1}^{t}\sum_{i=1}^{n_\tau} \phi^\tau_{i,\delta_\tau(i)} \phi^{\tau\top}_{i,\delta_\tau(i)}, \tag{4}$$

$$r_t = \sum_{\tau=1}^{t}\sum_{i=1}^{n_\tau} U^\tau_{i,\delta_\tau(i)} \phi^\tau_{i,\delta_\tau(i)}. \tag{5}$$

At the beginning of round t, by setting $\widehat{\theta}^t = M_{t-1}^{-1} r_{t-1}$, we essentially obtain the estimate $\widehat{\theta}^t$ for θ^* through ridge regression. To maintain such M_t and r_t, it suffices to initialize $M_0 = I_d$ and $r = 0$, and update at the end of each round t that

$$M_t \leftarrow M_{t-1} + \sum_{i=1}^{n_t} \phi^t_{i,\delta_t(i)} \phi^{t\top}_{i,\delta_t(i)},$$
$$r_t \leftarrow r_{t-1} + \sum_{i=1}^{n_t} \phi^t_{i,\delta_t(i)} U^t_{i,\delta_t(i)}.$$

Since we initialize $M = \eta I_d$ in the beginning, with a proper choice of η, we can handle the potential instability issue in the least square method, especially for the early stage when we only have a few observations available. After obtaining $\widehat{\theta}^t$, we construct the confidence interval $C^t_{i,j}$ for every (i, j) pair that

$$C^t_{i,j} = \left[a^t_{i,j}, b^t_{i,j}\right] = \left[\phi^{t\top}_{i,j}\widehat{\theta}^t - \lambda s^t_{i,j}, \phi^{t\top}_{i,j}\widehat{\theta}^t + \lambda s^t_{i,j}\right],$$

where $s^t_{i,j} = \sqrt{\phi^{t\top}_{i,j} M^{-1}_{t-1} \phi^t_{i,j}}$, and λ is a tuning parameter. Intuitively, the term $\lambda s^t_{i,j}$ reflects the uncertainty that we consider for the utility estimate $\phi^{t\top}_{i,j}\widehat{\theta}^t$, and $b^t_{i,j}$ is the upper confidence bound for the utility, which will be used to decide the assignment for time t in the next step. In addition, the parameter λ here control the width of the confidence interval, and typically depends on σ^2, the variance of the Gaussian noise, as well as T, the number of rounds. Note that to in order to balance the exploration-exploitation trade-off, our algorithm requires a positive λ. For the choice $\lambda = 0$, the confidence bound $C^t_{i,j}$ shrinks to $\phi^t_{i,j}\widehat{\theta}^t$, and in this case the algorithm becomes pure greedy in the sense that it considers no uncertainty.

Here we also point out that the construction of the upper confidence bound can vary by different models. The aforementioned process is motivated by the statistical principles of linear regression and linear bandits. For more complex machine learning models where the variance of the estimate is difficult to compute or does not have a close-form, one might consider other approach to construct the upper confidence bound, such as bootstrapping (Efron and Tibshirani, 1994; DiCiccio and Efron, 1996).

In summary, our proposed algorithm conducts assignment based on the two subsections above using upper confidence bounds. For better understanding, we present our method as Algorithm 2. Note that at each time t in the algorithm, the complexity of computation is $O(n_t^3 + d^3)$, where n_t is the number of pairs to match at time t, and d is the dimension of the unknown parameter θ.

4 Theoretical Results

In this section, we discuss the theoretical property of our proposed algorithm on regret bound. Before we move on, we first introduce some mild assumptions for our results to hold, which are mild and standard in the contextual bandit literature (Chu et al., 2011), and can be achieved by proper scaling of the data.

Assumption 1. *There exists a constant R, such that the transformation ϕ in (1) satisfies that for every $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{z} \in \mathcal{Z}$, $\|\phi(\boldsymbol{x}, \boldsymbol{z})\|_2 \leqslant R$. Meanwhile, we assume $\|\theta^*\|_2 \leqslant B$ for some constant B.*

Algorithm 2 UCB-based algorithm for sequential assignment with bandit feedback.

Specify the parameters $\alpha > 0$ and $\lambda > 0$. Set $M_0 = \alpha I_d$, $r_0 = \mathbf{0}$.

for $t = 1$ to T **do**

1. Compute $\widehat{\theta}^t \leftarrow M_{t-1}^{-1} r_{t-1}$.
2. Observe new covariates $\boldsymbol{x}_1^t, \ldots, \boldsymbol{x}_{n_t}^t$ and $\boldsymbol{z}_1^t, \ldots, \boldsymbol{z}_{n_t}^t$.
3. Let $\phi_{i,j}^t = \phi(\boldsymbol{x}_i^t, \boldsymbol{z}_j^t)$ for every i and j in $[n_t]$. Then construct the confidence interval $C_{i,j}^t$ for the associated utility as

$$C_{i,j}^t = \left[a_{i,j}^t, b_{i,j}^t\right] = \left[\phi_{i,j}^{t\top}\widehat{\theta}^t - \lambda s_{i,j}^t, \phi_{i,j}^{t\top}\widehat{\theta}^t + \lambda s_{i,j}^t\right],$$

where $s_{i,j}^t = \sqrt{\phi_{i,j}^{t\top} M^{-1} \phi_{i,j}^t}$.

4. Solve the assignment δ_t from the optimization problem (3) using Algorithm 1.
5. Collect feedback $\{U_{i,\delta_t(i)}^t : i \in [n_t]\}$.
6. Update

$$M_t \leftarrow M_{t-1} + \sum_{i=1}^{n_t} \phi_{i,\delta_t(i)}^t \phi_{i,\delta_t(i)}^{t\top},$$

$$r_t \leftarrow r_{t-1} + \sum_{i=1}^{n_t} \phi_{i,\delta_t(i)}^t U_{i,\delta_t(i)}^t.$$

end for

Below we first state the result on the estimation error of our proposed UCB-based algorithm in Theorem 1. It shows that with sufficient past observations, our estimates $\widehat{\theta}^t$ are pretty close to the underlying true parameter θ^* with high probability.

Theorem 1. *Suppose Assumptions 1 holds. If we set $\alpha = R^2$ as in (4), then for any $\tau > 0$, with probability at least $1 - \delta$, we have*

$$\left\|\widehat{\theta}^t - \theta^*\right\|_{M_{t-1}}^2 \leqslant \sigma^2 d\left[\log\left(R^2 + \frac{\sum_{\tau=1}^{t-1} n_\tau}{d} R^2\right) + 2\log\left(\frac{1}{\delta}\right)\right] + 2B^2R^2, \tag{6}$$

for all $t \geqslant \tau$.

Theorem 1 implies that as long as M_τ is sufficiently large and well-posed at certain round τ (i.e., we have sufficient observations over the space of ϕ), all the estimators θ^t after τ will be close to the true underlying θ^* with high probability. This guarantees that our algorithm can conduct near-optimal matchings in the long run, and the regret for individual round decreases rapidly as t grows. We provide the proof of Theorem 1 in Section B.1 of our supplementary materials.

Following the theorem above, we next present our analysis on the cumulative regret bound of our method. Before delving into our main theoretical result, we first introduce two useful lemmas as below.

Lemma 1. *With $s_{i,j}^t = \sqrt{\phi_{i,j}^{t\top} M^{-1} \phi_{i,j}^t}$, we define the event*

$$\mathcal{E}_{i,j}^t := \left\{\left|\phi_{i,j}^{t\top}\widehat{\theta}^t - \phi_{i,j}^{t\top}\theta^*\right| \leqslant \lambda s_{i,j}^t\right\}.$$

Then for any $\delta \in (0,1)$, when $\alpha = R^2$ and λ is chosen as in (7), we have

$$P\left(\bigcup_{t\in[T],\ i,j\in[n_t]} \mathcal{E}_{i,j}^t\right) \geqslant 1-\delta.$$

Lemma 2. *Let $\{\phi_i^t : t \in [T], i \in [n_t]\}$ be an arbitrary collection of d-dimensional vectors satisfying $\|\phi_i^t\|_2 \leqslant R$ for every t and i. Let $M_0 = R^2 I_d$, $M_t = M_{t-1} + \sum_{i=1}^{n_t} \phi_i^t \phi_i^{t\top}$. Denote $\|\phi\|_M = \sqrt{\phi^\top M \phi}$ the norm induced by a positive definite matrix M. Then we have*

$$\sum_{t=1}^{T}\sum_{i=1}^{n_t} \frac{\|\phi_i^t\|^2_{M_{t-1}^{-1}}}{n_t} \leqslant 2d\log\left(R^2 + \frac{\sum_{t=1}^T n_t}{d}R^2\right).$$

Intuitively, Lemma 1 implies that i) our estimate $\widehat{\theta}^t$ converges to the ground-truth θ^* as t grows, and ii) under our choice of λ, the underlying true utility falls within our confidence bounds $C_{i,j}^t$ for every t, i, j with high probability. Lemma 2 is then used to capture the total uncertainty for the sequence of matching tasks for our algorithm, which is closely related to the regret bound. The proof of Lemma 1 makes use of the concentration inequality of the sub-Gaussian noises, and the proof for Lemma 2 adapts the argument for UCB method in linear contextual bandits. We provide the detailed proofs for Lemmas 1 and 2 to Section A of our supplementary materials.

With the previous lemmas, we are now ready to state the main result on the regret bound of our proposed UCB-based algorithm in the theorem below.

Theorem 2. *Suppose the utility function for the given agent and task has the form in (1), where ϵ is σ-sub-Gaussian, and $\phi(\cdot,\cdot)$ and θ^* satisfies Assumption 1. Then for any $\delta \in (0,1)$, with the choice that $\alpha = R^2$ and*

$$\lambda = BR + \sigma\sqrt{2\log\frac{2\sum_{t=1}^T n_t^2}{\delta}}, \tag{7}$$

with probability at least $1-\delta$, the total regret of Algorithm 2 satisfies that

$$\begin{aligned} R_T \leqslant\ & 4\sigma\sqrt{dT\log\frac{2\sum_{t=1}^T n_t^2}{\delta}\log\left(R^2 + \frac{\sum_{t=1}^T n_t}{d}R^2\right)} \\ & + 2BR\sqrt{2dT\log\left(R^2 + \frac{\sum_{t=1}^T n_t}{d}R^2\right)}. \end{aligned} \tag{8}$$

As can be seen, our high-probability regret bound in (8) is of the rate $\widetilde{O}(\sqrt{dT})$ when neglecting the logarithm factors. Also we note that this rate matches the standard and optimal rate of regret for linear stochastic contextual bandit, as discussed in Chu et al. (2011) and Abbasi-Yadkori et al. (2011). In addition, we also point out that while the choice of α and λ above guarantees the regret lower bound, one might be interested in setting these parameters differently in practice to pursue more desired performance of the algorithm. We provide the proof of Theorem 2 in Section B.2 of our supplementary materials.

5 Numerical Studies

In this section, we conduct simulations to demonstrate the usefulness of our proposed UCB-based algorithm. More specifically, we investigate the total regret and parameter estimation under several examples, with various data dimension d and choice of tuning parameter λ. For convenience, in most settings below we take $n_t = n$ as a constant, and we also consider the case where n_t is varying over time and discuss the effect in Subsection 5.3.

5.1 Settings

In this subsection, we specify the construction of our simulation examples. Specifically, we set $\boldsymbol{x}_i^t, \boldsymbol{z}_i^t \in \mathbb{R}^d$ and set $\phi(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x} \circ \boldsymbol{z}$, where $\circ$ denotes the entry-wise product. We consider the following two settings of θ^*.

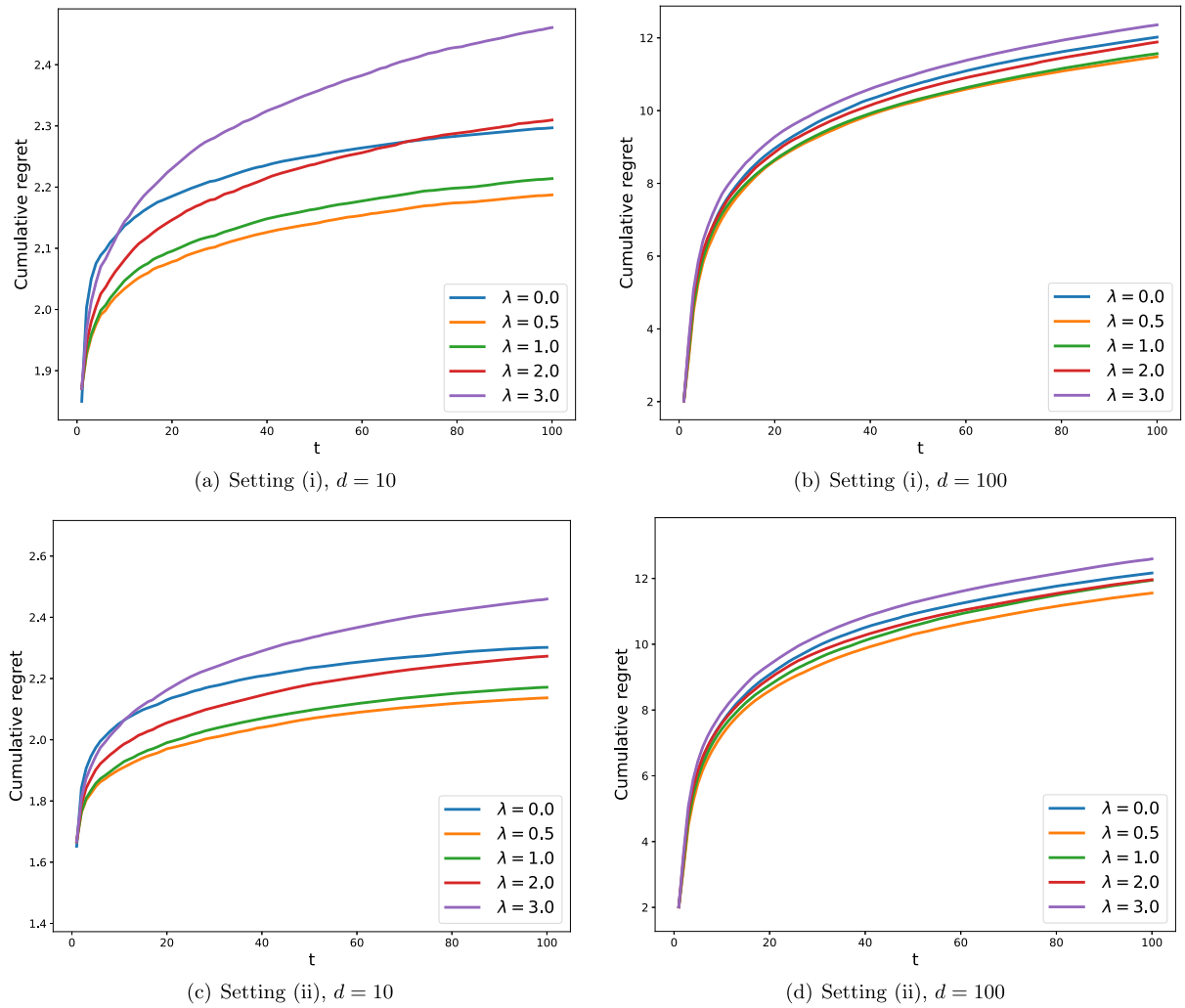
1. $\theta^* = \frac{1}{\sqrt{d}}(1, 1, \ldots, 1, -1, -1, \ldots, -1)$, where the first $d/2$ entries positive and the second $d/2$ entries negative.
2. $\theta^* = \frac{1}{\sqrt{31}}(-1, 1, 2, 3, 4, 0, \ldots, 0)$, with $\|\theta^*\|_2 = 1$.

When generating the utility outcome, we add a random noise ϵ that follows a normal distribution with mean 0 and variance 1. At each round, we randomly sample $\boldsymbol{x}_i^t$'s and $\boldsymbol{z}_i^t$'s from the multivariate normal distribution $N(0, I_d)$. While we fix $T = 100$, $n = 50$, we vary the dimension d as 10, 100, and vary the tuning parameter λ in a grid within the range $[0, 1]$. For every example, we have 20 replications of randomly sampled data, and finally we present the average performance.

5.2 Performance

In this we first present the figures that characterize the growth rate of the cumulative regret in t, followed by a table with detailed performance for different d and λ. Figure 2 presents the cumulative regret of our algorithm with respect to t under settings (i) and (ii), with different choice of λ. As can be seen, the growth of the total regret is indeed sublinear in t. Also, note that with the choice $\lambda = 0$, then the algorithm is greedy that does pure exploitation. From the figure, we can see that with a proper choice of λ, one can achieve a lower regret than the pure greedy method with $\lambda = 0$, by balancing the exploration and exploitation, while an λ too large (e.g. 3) might hurt the performance. Overall, the result validates the usefulness of our UCB-based assignment approach.

Tables 1 and 2 show the total regret up to $T = 100$ under settings (a) and (b) with different d and λ averaged over 20 replications. The standard deviation of the total regret is presented in brackets. Note that in the tables we also present the regret for the random match (i.e. assign by a random permutation with equal probability) as a baseline. Again we can see that with properly chosen λ, our algorithm can yield a total regret lower tahn the greedy approach with $\lambda = 0$. Besides, the performance is not very sensitive to λ in certain range. Furthermore, it is also worth noticing that while the choice of λ in (7) guarantees the theoretical property, in practice one might prefer to fine-tune λ for better empirical performance. As for in this example, while the choice of λ suggested by (7) can be much larger than 1, the empirical performance is more appealing with a λ smaller than 1.



(a) Setting (i), $d = 10$

(b) Setting (i), $d = 100$

(c) Setting (ii), $d = 10$

(d) Setting (ii), $d = 100$

Figure 2: Cumulative regret with different d and λ under settings (I) and (ii). The curves of cumulative regret are sub-linear in t. With proper choice of λ, the performance is improved over the pure-greedy algorithm when $\lambda = 0$, while a λ too large (e.g. 3.0) can hurt the performance.

Table 1: Total regret up to $T = 100$ under setting (a).

Regret (std)	$d = 10$	$d = 100$
Random Match	185.56 (2.11)	204.68 (1.39)
$\lambda = 0$ (Greedy)	2.30 (0.16)	12.02 (0.69)
$\lambda = 0.5$	2.19 (0.25)	11.48 (0.76)
$\lambda = 1.0$	2.21 (0.24)	11.57 (0.83)
$\lambda = 2.0$	2.31 (0.24)	11.89 (0.95)
$\lambda = 3.0$	2.46 (0.22)	12.36 (0.73)

Table 2: Total regret up to $T = 100$ under setting (b).

Regret (std)	$d = 10$	$d = 100$
Random Match	167.45 (1.84)	201.56 (1.27)
$\lambda = 0$ (Greedy)	2.30 (0.24)	12.17 (0.72)
$\lambda = 0.5$	2.14 (0.20)	11.56 (1.03)
$\lambda = 1.0$	2.17 (0.20)	11.95 (0.50)
$\lambda = 2.0$	2.27 (0.22)	11.96 (0.77)
$\lambda = 3.0$	2.46 (0.20)	12.59 (0.68)

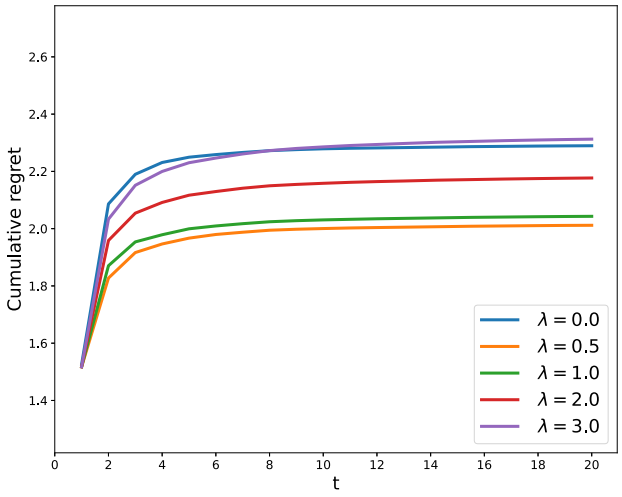


Figure 3: Cumulative regret with varying n_t.

5.3 Example with Varying n_t

To demonstrate our algorithm under the case where n_t is varying, and see how n_t affects the performance, we construct an example with the same setting of (i) with $d = 10$, except that we set $n_t = 20t$ for $t = 1, 2, \ldots, 20$, i.e. the number of pairs is increasing over the time up to $T = 20$. Again we run our algorithm with $\lambda = 0, 0.5, 1, 2, 3$ over 20 repetitions. Figure 3 shows the cumulative regret for difference λ values. Note that at early stage when n_t is small, a properly chosen λ yields better performance than the greedy method with a large gap of regret, while at late stage, when n_t becomes larger, all the curves become rather flat, and the incremental gap is small. This is not surprising, since in the early stage it is more challenging to learn the utility with limited observations, where the UCB method enjoys more advantage by active exploring.

6 Further Study

In this section, we go back to the motivating example of mentor-mentee matching for university mentoring programs, and illustrate the usefulness of our proposed algorithm.

At each semester, the university alumni office receives a number of mentees and mentors with context information on their background and preference. The office then needs to decide

how to pair between the mentees and mentors. At the end of each semester, a survey will be distributed to each participant to ask about their satisfaction on the experience. Due to the privacy and confidentiality constraints, we are unable to share the concrete dataset, but we will use similar data format to mimic the real dataset.

6.1 Data Format and Settings

In the mentor-mentee matching, the data includes the background and preference of each matched mentor-mentee pair, including their department, major, degree, location, industry, etc, together with their feedback. The feedback of each pair is summarized as a rating between 0 to 5. With preprocessing of the raw data and variable selection in linear regression, we select three variables (Department, Location and Industry) that are important to the experience of participants. Table 3 specifies these categories. Since the variables are all categorical, for any

Table 3: Table of variables for mentees.

Var Name	Categories
Department	1: Engineering; 2: Computing; 3: Sciences; 4: Design; 5: Liberal Arts; 6: Business.
Location	1: In State; 2: Northeast; 3: Southwest; 4: West; 5: Southeast; 6: Midwest; 7: US other; 8: International.
Industry	1: Research; 2: Technology; 3: Engineer; 4: Business; 5: Design; 6: Healthcare; 8: Other.

mentor-mentee pair, we introduce the indicator variables $I_{\text{department}}$, I_{location} and I_{indutry} to indicate whether the pair is matched for each variable. For example, $I_{\text{department}} = 1$ if the mentee and mentor are from the same department, and $I_{\text{department}} = 0$ otherwise. With the above variables, we fit the following linear model for the rating using the data:

$$\begin{aligned} \text{Rating} &= 2.48 + 0.069 I_{\text{department}} + 0.073 I_{\text{location}} + 0.058 I_{\text{indutry}} + \epsilon, \\ \epsilon &\sim N\big(0, 0.52^2\big). \end{aligned} \tag{9}$$

Alternatively, we can also encode the categorical variables with one-hot vectors for mentee and mentor, concatenated with an additional scalar 1 for the intercept. Let $\boldsymbol{x}$ and $\boldsymbol{z}$ denote such vector of covariates for mentee and mentor, respectively. Then (9) is equivalent to

$$\text{Rating}(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x} \circ \boldsymbol{z})^\top \theta^* + \epsilon. \tag{10}$$

Here the transformation is $\phi(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x} \circ \boldsymbol{z}$, the entry-wise product of $\boldsymbol{x}$ and $\boldsymbol{z}$, and

$$\theta^* = (2.48,\ 0.069\mathbb{1}_6,\ 0.073\mathbb{1}_8,\ 0.058\mathbb{1}_8),$$

where $\mathbb{1}_k$ denotes the vector of k dimensions with all the entries being 1. In the following study, after the assignment is decided for each round, we randomly generate the outcome of ratings based on (9).

Now we specify the generation of the covariates for mentors and mentees at each round. We randomly generate the data for $T = 20$ rounds. At each round, we generate the covariates of $n = 200$ mentors and mentees. For convenience, we draw from each category with equal probability for every variable, independent from other variables.

6.2 Performance

We run our proposed algorithm with $\lambda = 0, 0.5, 1, 2, 3$ on 20 repetitions and record the cumulative regret. Figure 4 shows the cumulative regret with different choice of λ averaged over 20 repetitions, with the standard deviation indicated by the shadow. The regret of the random match approach is also presented in the figure for comparison. As can be seen, in this study we observe a larger gap between our method with a properly chosen λ and the pure-greedy algorithm when $\lambda = 0$. Intuitively, this is related to the condition number of the design matrix. Because the variables are categorical, their are many zeros in $\phi(\boldsymbol{x}, \boldsymbol{z})$, which increases the difficulty to capture the underlying θ^*. In this case, the UCB-based method can better balance the tradeoff between exploration and exploitation compared to the pure-greedy algorithm, resulting in the significant improvement. Moreover, in this study we can also see that the performance of the UCB-based method is not sensitive to the choice of λ. Table 4 provides more details on the cumulative regrets of different choices of λ as well as their standard deviation at $T = 20$.

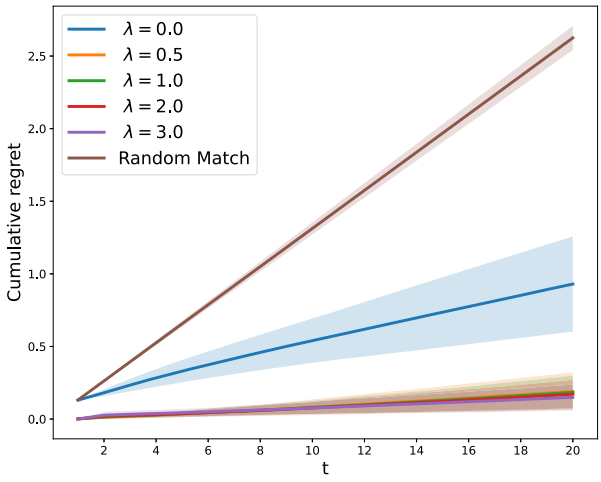


Figure 4: Cumulative regret with different λ.

Table 4: Total regret up to $T = 20$ for the simulated data for mentor-mentee matching scenario averaged over 20 repetitions. The standard deviations are presented in brackets.

Method	Regret (std)
Random Match	2.63 (0.08)
$\lambda = 0$ (Greedy)	0.96 (0.33)
$\lambda = 0.5$	0.18 (0.13)
$\lambda = 1.0$	0.17 (0.12)
$\lambda = 2.0$	0.17 (0.10)
$\lambda = 3.0$	0.14 (0.09)

7 Conclusions

In this work, we present a straightforward yet effective algorithm for the sequential assignment problem with unknown utility and stochastic feedback. We adapt the UCB-based algorithm from the multi-armed bandit problem to address the new scenario involving optimal assignment problems, providing a regret bound that aligns with findings in the stochastic contextual bandit literature. Extensive numerical studies are conducted to demonstrate the practicality and advantages of our proposed algorithm.

There are a number of interesting issues which has not been addressed here. In practice, the underlying utility function might have complicated form, and thus we might need to adopt a more sophistical model such as deep neural networks or non-parametric models. It is also interesting to investigate when the utility function is non-stationary, e.g., changing over time, by adapting the time-varying bandit algorithms in Vakili et al. (2014) and Xu et al. (2020) to our context. Moreover, it is important to develop distributed algorithm for learning the assignment strategy, especially when we face the problem of large-scale assignments in real-world applications. Therefore, this work should be interpreted as a starting point for further investigation on optimal sequential assignment problems.

Supplementary Material

The supplementary materials online includes: Proofs of Lemmas and Theorems used in the paper, and Python code needed to reproduce the results.

Funding

This research was supported in part by NSF grant DMS-2015405 and by the National Center for Advancing Translational Sciences of the National Institutes of Health under Award Number UL1TR002378. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Abbasi-Yadkori Y, Pál D, Szepesvári C (2011). Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24: 2312–2320.

Anderson T (2008). *The Theory and Practice of Online Learning*. Athabasca University Press.

Anscombe FJ (1953). Sequential estimation. *Journal of the Royal Statistical Society, Series B, Methodological*, 15(1): 1–21.

Avis D (1983). A survey of heuristics for the weighted matching problem. *Networks*, 13(4): 475–493.

Biró P, Gyetvai M (2023). Online voluntary mentoring: Optimising the assignment of students and mentors. *European Journal of Operational Research*, 307(1): 392–405.

Cesa-Bianchi N, Lugosi G (2012). Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5): 1404–1422.

Chen W, Wang Y, Yuan Y (2013). Combinatorial multi-armed bandit: General framework and applications. In: Dasgupta S and McAllester D (eds.), *Proceedings of the 30th International Conference on Machine Learning*, 151–159. PMLR.

Chu W, Li L, Reyzin L, Schapire R (2011). Contextual bandits with linear payoff functions. In: Gordon G, Dunson D, and Dudík M (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214. JMLR Workshop and Conference Proceedings.

Deshmukh AA, Dogan U, Scott C (2017). Multi-task learning for contextual bandits. *Advances in Neural Information Processing Systems*, 30: 4848–4856.

DiCiccio TJ, Efron B (1996). Bootstrap confidence intervals. *Statistical Science*, 11(3): 189–228.

Duan R, Pettie S (2014). Linear-time approximation for maximum weight matching. *Journal of the ACM*, 61(1): 1–23.

Edmonds J, Karp RM (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2): 248–264.

Efron B, Tibshirani RJ (1994). *An Introduction to the Bootstrap*. CRC Press.

Erraqabi A, Lazaric A, Valko M, Brunskill E, Liu YE (2017). Trading off rewards and errors in multi-armed bandits. In: Singh A and Zhu J (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 709–717. PMLR.

Fang A, Zhu H (2022). Matching for peer support: Exploring algorithmic matching for online mental health communities. *Proceedings of the ACM on Human–Computer Interaction*, 6(CSCW2): 1–37.

Gai Y, Krishnamachari B, Jain R (2012). Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5): 1466–1478.

Ghosh M, Mukhopadhyay N, Sen PK (2011). *Sequential Estimation*. John Wiley & Sons.

Hazan E (2016). Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3–4): 157–325.

Kuhn HW (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2): 83–97.

Kurtzberg JM (1962). On approximation methods for the assignment problem. *Journal of the ACM*, 9(4): 419–439.

Lai TL, Robbins H (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1): 4–22.

Perrault P, Boursier E, Valko M, Perchet V (2020). Statistical efficiency of Thompson sampling for combinatorial semi-bandits. *Advances in Neural Information Processing Systems*, 33: 5429–5440.

Pizzato L, Rej T, Chung T, Koprinska I Kay J (2010). RECON: A reciprocal recommender for online dating. In: *Proceedings of the Fourth ACM Conference on Recommender Systems*, 207–214.

Shalev-Shwartz S (2012). Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2): 107–194.

Shi Y, Mei Y (2022). Efficient sequential ucb-based Hungarian algorithm for assignment problems. In: *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1–8. IEEE.

Simchi-Levi D, Wang C (2023). Multi-armed bandit experimental design: Online decision-making and adaptive inference. In: Ruiz F, Dy J, and van de Meent J-W (eds.), *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, 3086–3097. PMLR.

Tomizawa N (1971). On some techniques useful for solution of transportation network problems. *Networks*, 1(2): 173–194.

Vakili S, Zhao Q, Zhou Y (2014). Time-varying stochastic multi-armed bandit problems. In: *2014 48th Asilomar Conference on Signals, Systems and Computers*, 2103–2107. IEEE.

Wen Z, Kveton B, Ashkan A (2015). Efficient learning in large-scale combinatorial semi-bandits. In: Bach F and Blei D (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, 1113–1122. PMLR.

Xia P, Liu B, Sun Y, Chen C (2015). Reciprocal recommendation system for online dating. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 234–241. IEEE.

Xu X, Dong F, Li Y, He S, Li X (2020). Contextual-bandit based personalized recommendation with time-varying user interests. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6518–6525.

Yang F, Ramdas A, Jamieson KG, Wainwright MJ (2017). A framework for multi-a(rmed)/b(andit) testing with online FDR control. *Advances in Neural Information Processing Systems*, 30: 5959–5968.