Learning Equilibria in Matching Markets from Bandit Feedback

Meena Jagadeesan[†]
mjagadeesan@berkeley.edu
UC Berkeley, EECS

Alexander Wei[†]
awei@berkeley.edu
UC Berkeley, EECS

Yixin Wang
ywang@eecs.berkeley.edu
UC Berkeley, EECS

Michael I. Jordan
jordan@cs.berkeley.edu
UC Berkeley, EECS and Statistics

Jacob Steinhardt
jsteinhardt@berkeley.edu
UC Berkeley, Statistics

Abstract

Large-scale, two-sided matching platforms must find market outcomes that align with user preferences while simultaneously learning these preferences from data. However, since preferences are inherently uncertain during learning, the classical notion of stability (Gale and Shapley, 1962; Shapley and Shubik, 1971) is unattainable in these settings. To bridge this gap, we develop a framework and algorithms for learning stable market outcomes under uncertainty. Our primary setting is matching with transferable utilities, where the platform both matches agents and sets monetary transfers between them. We design an incentive-aware learning objective that captures the distance of a market outcome from equilibrium. Using this objective, we analyze the complexity of learning as a function of preference structure, casting learning as a stochastic multi-armed bandit problem. Algorithmically, we show that "optimism in the face of uncertainty," the principle underlying many bandit algorithms, applies to a primal-dual formulation of matching with transfers and leads to near-optimal regret bounds. Our work takes a first step toward elucidating when and how stable matchings arise in large, data-driven marketplaces.

1 Introduction

Data-driven marketplaces face the simultaneous challenges of learning agent preferences and aligning market outcomes with the incentives induced by these preferences. Consider, for instance, online platforms that match two sides of a market to each other (e.g., Lyft, TaskRabbit, and Airbnb). On these platforms, customers are matched to service providers and pay for the service they receive. If agents on either side are not offered desirable matches at fair prices, they would have an incentive to leave the platform and switch to a competing platform. Agent preferences, however, are often unknown to the platform and must be learned. When faced with uncertainty about agent preferences (and thus incentives), *when can a marketplace efficiently explore and learn market outcomes that align with agent incentives?*

We center our investigation around a model called *matching with transferable utilities*, proposed by Shapley and Shubik [SS71]. In this model, there is a two-sided market of customers and service providers. Each customer has a utility that they derive from being matched to a given provider and vice versa. The platform selects a matching between the two sides and assigns a monetary transfer between each pair of matched agents. Transfers are a salient feature of most real-world matching markets: riders pay drivers on Lyft, clients pay freelancers on TaskRabbit, and guests pay hosts on Airbnb. An agent's net utility is their value for being matched to their partner plus the value of their transfer (either of which can be negative in the cases of costs and payments). In matching markets, the notion of *stability* captures alignment of a

[†]Equal contribution.

market outcome with agent incentives. Informally, a market outcome is *stable* if no pair of agents would rather match with each other than abide by the market outcome, and stable matchings can be computed when preferences are fully known.

However, in the context of large-scale matching platforms, the assumption that preferences are known breaks down. Platforms usually cannot have users report their complete preference profiles. Moreover, users may not even be aware of what their own preferences are. For example, a freelancer may not exactly know what types of projects they prefer until actually trying out specific ones. In reality, a data-driven platform is more likely to learn information about preferences from repeated feedback[1] over time. Two questions now emerge: In such marketplaces, how can stable matchings be learned? And what underlying structural assumptions are necessary for efficient learning to be possible?

Toward answering these questions, we propose and investigate a model for learning stable matchings from noisy feedback. We model the platform's learning problem using stochastic multi-armed bandits, which lets us leverage the extensive body of techniques in the bandit literature to analyze the data efficiency of learning (see Lattimore and Szepesvári [LS20] for a textbook treatment). More specifically, our three main contributions are: (i) We develop an incentive-aware learning objective—Subset Instability—that captures the distance of a market outcome from equilibrium. (ii) Using Subset Instability as a measure of regret, we show that any "UCB-based" algorithm from the classical bandit literature can be adapted to this incentive-aware setting. (iii) We instantiate this idea for several families of preference structures to design efficient algorithms for incentive-aware learning. This helps elucidate how preference structure affects the complexity of learning stable matchings.

Designing the learning objective. Since mistakes are inevitable while exploring and learning, achieving exact stability at every time step is an unattainable goal. To address this issue, we lean on approximation, focusing on learning market outcomes that are *approximately* stable. Thus, we need a metric that captures the distance of a market outcome from equilibrium.[2]

We introduce a notion for approximate stability that we call Subset Instability. Specifically, we define the Subset Instability of a market outcome to be the maximum difference, over all subsets $\mathcal{S}$ of agents, between the total utility of the maximum weight matching on $\mathcal{S}$ and the total utility of $\mathcal{S}$ under the market outcome.[3] We show Subset Instability can be interpreted as how much the platform would have to *subsidize* participants to keep them on the platform and make the resulting matching stable. We can also interpret Subset Instability as the platform's cost of learning when facing competing platforms with greater knowledge of user preferences. Finally, we show that Subset Instability is the maximum gain in utility that a coalition of agents could have derived from an alternate matching such that no agent in the coalition is worse off.

Subset Instability also satisfies the following properties, which make it suitable for learning: (i) Subset Instability is equal to 0 if and only if the market outcome is (exactly) stable; (ii) Subset Instability is robust to small perturbations to the utility functions of individual agents, which is essential for learning with noisy feedback; (iii) Subset Instability upper bounds the utility difference of a market outcome from the socially optimal market outcome.

Designing algorithms for learning a stable matching. Using Subset Instability, we investigate the problem of learning a stable market outcome from noisy user feedback using the stochastic contextual

[1] Feedback might arise from explicit sources (e.g., riders rating drivers after a Lyft ride) or implicit sources (e.g., engagement metrics on an app); in either case, feedback is likely to be sparse and noisy.

[2] Previous work [DK05; LMJ20] has investigated utility difference (i.e. the difference between the total utility achieved by the selected matching and the utility achieved by a stable matching) as a measure of regret. However, this does not capture distance from equilibrium in matching markets with monetary transfers (see Section 4) or without monetary transfers (see Section 6.3.1).

[3] This formulation is inspired by the *strong ε-core* of Shapley and Shubik [SS66].

	Regret bound
Unstructured preferences	$\widetilde{O}(N\sqrt{nT})$
Typed preferences	$\widetilde{O}(\lvert\mathcal{C}\rvert\sqrt{nT})$
Separable linear preferences	$\widetilde{O}(d\sqrt{N}\sqrt{nT})$

Table 1: Regret bounds for different preference structures when there are N agents on the platform and no more than n agents arriving in each round.

bandit model (see, e.g., [LS20]). In each round, the platform selects a market outcome (i.e., a matching along with transfers), with the goal of minimizing cumulative instability.

We develop a general approach for designing bandit algorithms within our framework. Our approach is based on a primal-dual formulation of matching with transfers [SS71], in which the primal variables correspond to the matching and the dual variables can be used to set the transfers. We find that "optimism in the face of uncertainty," the principle underlying many UCB-style bandit algorithms [ACF02; LS20], can be adapted to this primal-dual setting. The resulting algorithm is simple: maintain upper confidence bounds on the agent utilities and compute, in each round, an optimal primal-dual pair in terms of these upper confidence bounds. The crux of the analysis is the following lemma, which bounds instability by the gap between the upper confidence bound and true utilities:

Lemma 1.1 (Informal, see Lemma 5.4 for a formal statement)**.** *Given confidence sets for each utility value such that each confidence set contains the true utility, let (X, τ) be a stable matching with transfers with respect to the utility functions given by the upper confidence bounds. The instability of (X, τ) is upper bounded by the sum of the sizes of the confidence sets of pairs in X.*

We can thus analyze our algorithms by combining Lemma 1.1 with the analyses of existing UCB-style algorithms. In particular, we can essentially inherit the bounds from traditional bandits analyses on the size of the confidence bounds.

Complexity of learning a stable matching. Our main technical result is a collection of regret bounds for different structural assumptions on agent preferences. These bounds resemble the classical stochastic multi-armed bandits bounds when rewards have related structural assumptions. We summarize these regret bounds in Table 1 and elaborate on them in more detail below.

Theorem 1.2 (Unstructured Preferences, Informal)**.** *For unstructured preferences, there exists a UCB-style algorithm that incurs $\widetilde{O}(N\sqrt{nT})$ regret according to Subset Instability after T rounds, where N is the number of agents on the platform and n is the number of agents that arrive in any round. (In fact, this bound is optimal up to logarithmic factors.)*

Theorem 1.3 (Typed Preferences, Informal)**.** *Consider preferences such that each agent a has a type $c_a \in \mathcal{C}$ and the utility of a when matched to another agent a' is given by a function of the types c_a and $c_{a'}$. There exists a UCB-style algorithm that incurs $\widetilde{O}(|\mathcal{C}|\sqrt{nT})$ regret according to Subset Instability after T rounds, where n is the maximum number of agents that arrive to the platform in any round.*

Theorem 1.4 (Separable Linear Preferences, Informal)**.** *Consider preferences such that the utility of an agent a when matched to another agent a' is $\langle \phi(a), c_{a'} \rangle$ where $\phi(a) \in \mathbb{R}^d$ is unknown and $c_{a'} \in \mathbb{R}^d$ is known. There exists a UCB-style algorithm that incurs $\widetilde{O}(d\sqrt{N}\sqrt{nT})$ regret according to Subset Instability after T rounds, where N is the number of agents on the platform and n is the maximum number of agents that arrive in any round.*

These results let us elucidate the role of preferences structures on the complexity of learning a stable matching. Our regret bounds scale with $N\sqrt{nT}$ for unstructured preferences (Theorem 1.2), $|\mathcal{C}|\sqrt{nT}$ for typed preferences (Theorem 1.3), and $d\sqrt{N}\sqrt{nT}$ for linear preferences (Theorem 1.4). To illustrate these differences in a simple setting, let's consider the case where all of the agents show up every round so $n = N$. In this case, our regret bound for unstructured preferences is super-linear in N; in fact, this dependence on N is *necessary* as we demonstrate via a lower bound (see Lemma 5.5). On the other hand, the complexity of learning a stable matching changes substantially with preference structure assumptions. In particular, our regret bounds are sublinear / linear in N for typed preferences and separable linear preferences. This means that in large markets, a centralized platform can efficiently learn a stable matching with these preference structure assumptions.

Extensions. We extend our results for unstructured preferences to obtain $O(\log T)$ instance-dependent bounds (Section 6.1), we show that our regret relates to a simple notion of platform profit (see Section 6.2), and we extend our framework to matching with non-transferable utilities setting (see Section 6.3).

1.1 Related work

In the machine learning literature, starting with Das and Kamenica [DK05] and Liu, Mania, and Jordan [LMJ20], several works [DK05; LMJ20; SBS21; LRM+20; CS21; BSS21] study learning stable matchings from bandit feedback in the Gale-Shapley stable marriage model [GS62]. A major difference between this setting and ours is the absence of monetary transfers between agents. These works focus on the *utility difference* rather than the instability measure that we consider. Cen and Shah [CS21] extend this bandits model to incorporate fixed, predetermined cost/transfer rules. However, they do not allow the platform to set arbitrary transfers between agents. Moreover, they also consider a weaker notion of stability that does not consider agents negotiating arbitrary transfers: defecting agents must set their transfers according to a fixed, predetermined structure. In contrast, we follow the classical definition of stability [SS71].

Outside of the machine learning literature, several papers also consider the complexity of finding stable matchings in other feedback and cost models, e.g., communication complexity [GNO+19; ABK+20; Shi20] and query complexity [EGK20; ABK+20]. Of these works, Shi [Shi20], which studies the communication complexity of finding approximately stable matchings with transferable utilities, is perhaps most similar to ours. This work assumes agents know their preferences and focuses on the communication bottleneck, whereas we study the costs associated with learning preferences. Moreover, the approximate stability notion in Shi [Shi20] is the maximum unhappiness of any *pair* of agents, whereas Subset Instability is equivalent to the maximum unhappiness over any *subset* of agents. For learning stable matchings, Subset Instability has the advantages of being more fine-grained and having a primal view that motivates a clean UCB-based algorithm.

A complementary line of work in economics [LMP+; Bik17; Als20; Liu20] considers stable matchings under incomplete information. These works focus on defining stability when the agents have incomplete information about their own preferences, whereas we focus on the platform's problem of learning stable matchings from noisy feedback. As a result, these works relax the definition of stability to account for uncertainty in the preferences of agents, rather than the uncertainty experienced by the platform from noisy feedback.

Multi-armed bandits have also been applied to learning in other economic contexts. For example, learning a socially optimal matching (without learning transfers) is a standard application of combinatorial bandits [CL12; GKJ12; CWY13; CTP+15; KWA+15]. Other applications at the interface of bandit methodology and economics include dynamic pricing [Rot74; KL03; BKS18], incentivizing exploration [FKK+14; MSS15], learning under competition [AMS+20], and learning in matching markets without incentives [JKK21].

Finally, primal-dual methods have also been applied to other problems in the bandits literature (e.g., [ISS+19; TPR+20; LSY21]).

2 Preliminaries

The foundation of our framework is the *matching with transfers* model of Shapley and Shubik [SS71]. In this section, we introduce this model along with the concept of stable matching.

2.1 Matching with transferable utilities

Consider a two-sided market that consists of a finite set $\mathcal{I}$ of customers on one side and a finite set $\mathcal{J}$ of providers on the other. Let $\mathcal{A} \coloneqq \mathcal{I} \cup \mathcal{J}$ be the set of all agents. A *matching* $X \subseteq \mathcal{I} \times \mathcal{J}$ is a set of pairs (i, j) that are pairwise disjoint, representing the pairs of agents that are matched. Let $\mathscr{X}_{\mathcal{A}}$ denote the set of all matchings on $\mathcal{A}$. For notational convenience, we define for each matching $X \in \mathscr{X}_{\mathcal{A}}$ an equivalent functional representation $\mu_X \colon \mathcal{A} \to \mathcal{A}$, where $\mu_X(i) = j$ and $\mu_X(j) = i$ for all matched pairs $(i, j) \in X$, and $\mu_X(a) = a$ if $a \in \mathcal{A}$ is unmatched.

When a pair of agents $(i, j) \in \mathcal{I} \times \mathcal{J}$ matches, each experiences a utility gain. We denote these utilities by a global utility function $u : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$, where $u(a, a')$ denotes the utility that agent a gains from being matched to agent a'. (If a and a' are on the same side of the market, we take $u(a, a')$ to be 0 by default.) We allow these utilities to be negative, if if matching results in a net cost (e.g., if an agent is providing a service). We assume each agent $a \in \mathcal{A}$ receives zero utility if unmatched, i.e., $u(a, a) = 0$. When we wish to emphasize the role of an individual agent's utility function, we will use the equivalent notation $u_a(a') \coloneqq u(a, a')$.

A *market outcome* consists of a matching $X \in \mathscr{X}_{\mathcal{A}}$ along with a vector $\tau \in \mathbb{R}^{\mathcal{A}}$ of transfers, where τ_a is the amount of money transferred from the platform to agent a for each $a \in \mathcal{A}$. These monetary transfers are a salient feature of most real-world matching markets: riders pay drivers on Lyft, clients pay freelancers on TaskRabbit, and guests pay hosts on Airbnb. Shapley and Shubik [SS71] capture this aspect of matching markets by augmenting the classical two-sided matching model with transfers of utility between agents. Transfers are typically required to be *zero-sum*, meaning that $\tau_i + \tau_j = 0$ for all matched pairs $(i, j) \in X$ and $\tau_a = 0$ if a is unmatched. Here, X represents how agents are matched and τ_a represents the transfer that agent a receives (or pays). The net utility that an agent a derives from a matching with transfers (X, τ) is therefore $u(a, \mu_X(a)) + \tau_a$.

Stable matchings. In matching theory, stability captures when a market outcome aligns with individual agents' preferences. Roughly speaking, a market outcome (X, τ) is stable if: (i) no individual agent a would rather be unmatched, and (ii) no pair of agents (i, j) can agree on a transfer such that both would rather match with each other than abide by (X, τ). Formally:

Definition 2.1. A market outcome (X, τ) is *stable* if: (i) it is *individually rational*, i.e.,

$$u_a(\mu_X(a)) + \tau_a \geq 0 \tag{1}$$

for all agents $a \in \mathcal{A}$, and (ii) it *has no blocking pairs*, i.e.,

$$\big(u_i(\mu_X(i)) + \tau_i\big) + \big(u_j(\mu_X(j)) + \tau_j\big) \geq u_i(j) + u_j(i). \tag{2}$$

for all pairs of agents $(i, j) \in \mathcal{I} \times \mathcal{J}$.[4]

[4] We observe that (2) corresponds to no pair of agents (i, j) being able to agree on a transfer such that both would rather match with each other than abide by (X, τ). Notice that a pair (i, j) violates (2) if and only if they can find a transfer $\tau_i' = -\tau_j'$ such that $u_i(j) + \tau_i' > u_i(\mu_X(i)) + \tau_i$ and $u_j(i) + \tau_j' > u_j(\mu_X(j)) + \tau_j$.

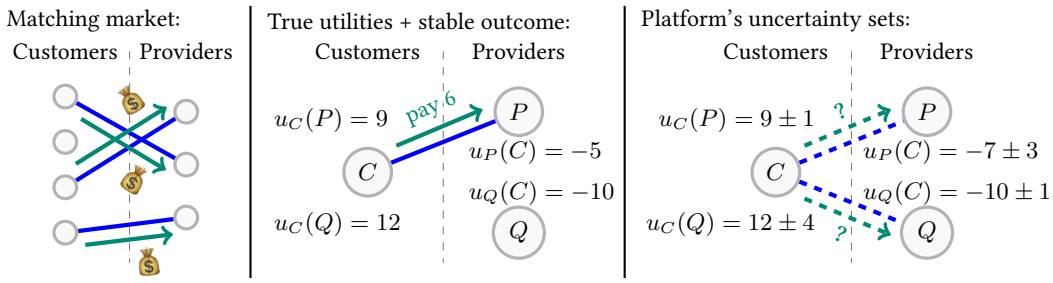

Figure 1: The left panel depicts a schematic of a matching (blue) with transfers (green). The center panel depicts a matching market with three agents and a stable matching with transfers for that market. (If the transfer 6 is replaced with any value between 5 and 7, the outcome remains stable.) The right panel depicts the same market, but with utilities replaced by uncertainty sets; note that no matching with transfers is stable for all realizations of utilities.

A fundamental property of the matching with transfers model is that if (X, τ) is stable, then X is a maximum weight matching, i.e., X maximizes $\sum_{a \in \mathcal{A}} u_a(\mu_X(a))$ over all matchings $X \in \mathscr{X}_{\mathcal{A}}$ [SS71]. The same work shows that stable market outcomes coincide with Walrasian equilibria. (For completeness, we recapitulate the basic properties of this model in Appendix A.)

To make the matching with transfers model concrete, we use the simple market depicted in the center panel of Figure 1 as a running example throughout the paper. This market consists of a customer Charlene and two providers Percy and Quinn, which we denote by $\mathcal{I} = \{C\}$ and $\mathcal{J} = \{P, Q\}$. If the agents' utilities are as given in Figure 1, then Charlene would prefer Quinn, but Quinn's cost of providing the service is much higher. Thus, matching Charlene and Percy is necessary for a stable outcome. This matching is stable for any transfer from Charlene to Percy in the interval $[5, 7]$.

3 Learning Problem and Feedback Model

We instantiate the platform's learning problem in a stochastic contextual bandits framework. Matching takes place over the course of T rounds. We denote the set of all customers by $\mathcal{I}^*$, the set of all providers by $\mathcal{J}^*$, and the set of all agents on the platform by $\mathcal{A}^* = \mathcal{I}^* \cup \mathcal{J}^*$. Each agent $a \in \mathcal{A}^*$ has an associated context $c_a \in \mathcal{C}$, where $\mathcal{C}$ is the set of all possible contexts. This context represents the side information available to the platform about the agent, e.g., demographic, location, or platform usage information. Each round, a set of agents arrives to each side of the market. The platform then selects a market outcome and incurs a regret equal to the *instability* of the market outcome (which we introduce formally in Section 4). Finally, the platform receives noisy feedback about the utilities of each matched pair (i, j).

To interpret the noisy feedback, note that platforms in practice often receive feedback both explicitly (e.g., riders rating drivers after a Lyft ride) and implicitly (e.g., engagement metrics on an app). In either instance, feedback is likely to be sparse and noisy. For simplicity, we do not account for agents strategically manipulating their feedback to the platform and focus on the problem of learning preferences from unbiased reports.

We now describe this model more formally. In the t-th round:

1. A set $\mathcal{I}^t \subseteq \mathcal{I}^*$ of customers and a set $\mathcal{J}^t \subseteq \mathcal{J}^*$ of providers arrive to the market. Write $\mathcal{I}^t \cup \mathcal{J}^t =: \mathcal{A}^t$. The platform observes the identity a and the *context* $c_a \in \mathcal{C}$ of each agent $a \in \mathcal{A}^t$.
2. The platform selects a matching with *zero-sum* transfers (X^t, τ^t) between $\mathcal{I}^t$ and $\mathcal{J}^t$.

3. The platform observes noisy utilities $u_a(\mu_{X^t}(a)) + \varepsilon_{a,t}$ for each agent $a \in \mathcal{I}^t \cup \mathcal{J}^t$, where the $\varepsilon_{a,t}$ are independent, 1-subgaussian random variables.[5]
4. The platform incurs regret equal to the *instability* of the selected market outcome (X^t, τ^t). (We define instability formally in Section 4.)

The platform's total regret R_T is thus the cumulative instability incurred up through round T.

3.1 Preference structure

In this bandits framework, we can embed varying degrees of structure on agent preferences. We capture these preference structures by the functional form of agents' utility functions and its relation to agent contexts. More formally, let $\mathcal{U}$ be the set functions $u\colon \mathcal{A}^* \times \mathcal{A}^* \to \mathbb{R}$, i.e., $\mathcal{U}$ is the set of all possible (global) utility functions. We now introduce several classes of preference structures as subsets of $\mathcal{U}$.

Unstructured preferences. The simplest setting we consider is one where the preferences are unstructured. Specifically, we consider the class of utility functions

$$\mathcal{U}_{\text{unstructured}} = \left\{u \in \mathcal{U} \mid u(a, a') \in [-1, 1]\right\}.$$

(Here, one can think of the context as being uninformative, i.e., $\mathcal{C}$ is the singleton set.) In this setup, the platform must learn each agent's utility function $u_a(\cdot) = u(a, \cdot)$.

Typed preferences. We next consider a market where each agent comes in one of finitely many *types*, with agents of the same type having identical preferences. Assuming typed preference structures is standard in theoretical models of markets (see Debreu and Scarf [DS63], Echenique et al. [ELS+13], and Azevedo and Hatfield [AH18], to name a few). We can embed types into our framework by having each agent's context represent their type, with $|\mathcal{C}| < \infty$. The global utility function is then fully specified by agents' contexts:

$$\mathcal{U}_{\text{typed}} = \left\{u \in \mathcal{U} \mid u(a, a') = f(c_a, c_{a'}) \text{ for some } f\colon \mathcal{C} \times \mathcal{C} \to [-1, 1]\right\}.$$

Separable linear preferences. We next consider markets where each agent is associated with *known* information given by their context as well as *hidden* information that must be learned by the platform. (This differs from unstructured preferences, where all information was hidden, and typed preferences, where each agent's context encapsulated their full preferences.) We explore this setting under the assumption that agents' contexts and hidden information interact linearly.

We assume that all contexts belong to $\mathcal{B}^d$ (i.e., $\mathcal{C} = \mathcal{B}^d$) where $\mathcal{B}^d$ is the ℓ_2 unit ball in $\mathbb{R}^d$. We also assume that there exists a function $\phi\colon \mathcal{A}^* \to \mathcal{B}^d$ mapping each agent to the hidden information associated to that agent. The preference class $\mathcal{U}^d_{\text{linear}}$ can then be defined as

$$\mathcal{U}^d_{\text{linear}} = \left\{u \in \mathcal{U} \;\middle|\; u(a, a') = \langle c_{a'}, \phi(a)\rangle \text{ for some } \phi\colon \mathcal{A}^* \to \mathcal{B}^d\right\}.$$

4 Measuring Approximate Stability

When learning stable matchings, we must settle for guarantees of approximate stability, since exact stability—a binary notion—is unattainable when preferences are uncertain. To see this, we return to the example from Figure 1. Suppose that the platform has uncertainty sets given by the right panel. Recall that for the

[5] Our feedback model corresponds to *semi-bandit* feedback, since the platform has (noisy) access to each agent's utility within the matching rather than the overall utility of the matching.

true utilities, all stable outcomes match Charlene with Percy. If the true utilities were instead the upper bounds of each uncertainty set, then all stable outcomes would match Charlene and Quinn. Given only the uncertainty sets, it is impossible for the platform to find an (exactly) stable matching, so it is necessary to introduce a measure of approximate stability as a relaxed benchmark for the platform; we turn to this now.

Given the insights of Shapley and Shubik [SS71]—that all stable outcomes maximize the sum of agents' utilities—it might seem natural to measure distance from stability simply in terms of the *utility difference*. To define this formally, let $\mathcal{A}$ be the set of agents participating in the market. (This corresponds to $\mathcal{A}^t$ at time step t in the bandits model.) The utility difference[6] of a market outcome (X, τ) is given by:

$$\left(\max_{X' \in \mathscr{X}_{\mathcal{A}}} \sum_{a \in \mathcal{A}} u_a(\mu_{X'}(a))\right) - \left(\sum_{a \in \mathcal{A}} u_a(\mu_X(a)) + \tau_a\right). \tag{3}$$

The first term $\max_{X' \in \mathscr{X}_{\mathcal{A}}} \sum_{a \in \mathcal{A}} u_a(\mu_{X'}(a))$ is the maximum total utility of any matching, and the second term $\sum_{a \in \mathcal{A}}(u_a(\mu_X(a)) + \tau_a)$ is the total utility of market outcome (X, τ). Since transfers are zero-sum, (3) can be equivalently written as

$$\left(\max_{X' \in \mathscr{X}_{\mathcal{A}}} \sum_{a \in \mathcal{A}} u_a(\mu_{X'}(a))\right) - \sum_{a \in \mathcal{A}} u_a(\mu_X(a)).$$

But this shows that utility difference actually ignores the transfers τ entirely! In fact, the utility difference can be zero even when the transfers lead to a market outcome that is far from stable (see Appendix B.1). Utility difference is therefore *not* incentive-aware, making it unsuitable as an objective for learning stable matchings with transfers.

In the remainder of this section, we propose a measure of instability—Subset Instability—which we will show serves as a suitable objective for learning stable matchings with transfers. Specifically, we show that Subset Instability captures the distance of a market outcome from equilibrium while reflecting both the platform's objective and the users' incentives. We additionally show that Subset Instability satisfies several structural properties that make it useful for learning.

4.1 Subset Instability

Subset Instability is based on utility difference, but rather than only looking at the market in aggregate, it takes a maximum ranging over all subsets of agents.

Definition 4.1. Given utilities u, the *Subset Instability* $I(X, \tau; u, \mathcal{A})$ of a matching with transfers (X, τ) is

$$\max_{\mathcal{S} \subseteq \mathcal{A}} \left[\left(\max_{X' \in \mathscr{X}_{\mathcal{S}}} \sum_{a \in \mathcal{S}} u_a(\mu_{X'}(a))\right) - \left(\sum_{a \in \mathcal{S}} u_a(\mu_X(a)) + \tau_a\right)\right]. \tag{$*$}$$

(The first term $\max_{X' \in \mathscr{X}_{\mathcal{S}}} \sum_{a \in \mathcal{S}} u_a(\mu_{X'}(a))$ is the maximum total utility of any matching over $\mathcal{S}$, and the second term $\sum_{a \in \mathcal{A}}(u_a(\mu_X(a)) + \tau_a)$ is the total utility of the agents in $\mathcal{S}$ under market outcome (X, τ).)

Intuitively, Subset Instability captures stability because it checks whether any subset of agents would prefer an alternate outcome. We provide a more extensive economic interpretation below; but before doing so, we first illustrate Definition 4.1 in the context of the example in Figure 1.

[6] Utility difference is standard as a measure of regret for learning a maximum weight matching in the combinatorial bandits literature (see, e.g., [GKJ12]). However, we show that for learning stable matchings, a fundamentally different measure of regret is needed.

Consider the matching $X = \{(C, Q)\}$ with transfers $\tau_C = -11$ and $\tau_Q = 11$. (This market outcome is stable for the upper bounds of the uncertainty sets of the platform in Figure 1, but not stable for the true utilities.) It is not hard to see that the subset $\mathcal{S}$ that maximizes Subset Instability is $\mathcal{S} = \{C, P\}$, in which case $\max_{X' \in \mathscr{X}_\mathcal{S}} \sum_{a \in \mathcal{S}} u_a(\mu_{X'}(a)) = 4$ and $\sum_{a \in \mathcal{S}} (u_a(\mu_X(a)) + \tau_a) = 1$. Thus, the Subset Instability of (X, τ) is $I(X, \tau; u, \mathcal{A}) = 4 - 1 = 3$. In contrast, the utility difference of (X, τ) is 2.

We now discuss several interpretations of Subset Instability, which provide further insight into why Subset Instability serves as a meaningful notion of approximate stability in online marketplaces. In particular, Subset Instability can be interpreted as *the minimum stabilizing subsidies*, as *the platform's cost of learning*, as *a measure of user unhappiness*, and as a *distance from equilibrium*.

Subset Instability as the platform's minimum stabilizing subsidy. Subset Instability can be interpreted in terms of monetary subsidies from the platform to the agents. Specifically, the Subset Instability of a market outcome equals the minimum amount the platform could subsidize agents so that the subsidized market outcome is individually rational and has no blocking pairs.

More formally, let $s \in \mathbb{R}^{\mathcal{A}}_{\geq 0}$ denote subsidies made by the platform, where the variable $s_a \geq 0$ represents the subsidy provided to agent a.[7] For a market outcome (X, τ), the *minimum stabilizing subsidy* is

$$\min_{s \in \mathbb{R}^{\mathcal{A}}_{\geq 0}} \left\{ \sum_{a \in \mathcal{A}} s_a \;\middle|\; (X, \tau + s) \text{ is stable} \right\}, \tag{4}$$

where we define stability in analogy to Definition 2.1. Specifically, we say that a market outcome (X, τ) with subsidies s is *stable* if it is individually rational, i.e., $u_a(\mu_X(a)) + \tau_a + s_a \geq 0$ for all agents $a \in \mathcal{A}$, and has no blocking pairs, i.e., $(u_i(\mu_X(i)) + \tau_i + s_i) + (u_j(\mu_X(j)) + \tau_j + s_j) \geq u_i(j) + u_j(i)$ for all pairs of agents $(i, j) \in \mathcal{I} \times \mathcal{J}$.

Given this setup, we show the following equivalence:

Proposition 4.2. *For any market outcome, the minimum stabilizing subsidy equals the Subset Instability.*

See Appendix B.2 for a full proof. The proof boils down to showing that (∗) and (5) are in some sense dual to each other. To formalize this, we rewrite the minimum stabilizing subsidy as the solution to the following linear program[8]:

$$\begin{aligned} \min_{s \in \mathbb{R}^{|\mathcal{A}|}} \; & \sum_{a \in \mathcal{A}} s_a & & \\ \text{s.t.} \; & \big(u_i(\mu_X(i)) + \tau_i + s_i\big) + \big(u_j(\mu_X(j)) + \tau_j + s_j\big) \geq u_i(j) + u_j(i) & & \forall (i, j) \in \mathcal{I} \times \mathcal{J} \\ & u_a(\mu_X(a)) + \tau_a + s_a \geq 0 & & \forall a \in \mathcal{A} \\ & s_a \geq 0 & & \forall a \in \mathcal{A}. \end{aligned} \tag{5}$$

The crux of our argument is that the dual linear program to (5) maximizes the combinatorial objective (∗). The equivalence of (∗) and (5) then follows from strong duality.

With this alternate formulation of Subset Instability in mind, we revisit the example in Figure 1. Again, consider the matching $X = \{(C, Q)\}$ with transfers $\tau_C = -11$ and $\tau_Q = 11$. (This is stable for the upper bounds of the uncertainty sets of the platform in Figure 1, but not stable for the true utilities.) We have already shown above that the Subset Instability of this market outcome is 3. To see this via the subsidy

[7] The requirement that $s_a \geq 0$ enforces that all subsidies are nonnegative; without it, (5) would reduce to the utility difference, which is not incentive-aware.

[8] In this linear program, the first set of constraints ensures there are no blocking pairs, while the second set of constraints ensures individual rationality.

formulation, note that the optimal subsidy s gives C and P a total of 3. (E.g., we give C a subsidy of $s_C = 2$ and P a subsidy of $s_P = 1$.) Indeed, if $s_C + s_P = 3$, then

$$\big(u_C(\mu_X(C)) + \tau_C + s_C\big) + \big(u_P(\mu_X(P)) + \tau_P + s_P\big) \geq u_C(P) + u_P(C)$$

holds (with equality), so the pair (C, P) could no longer gain by matching with each other.

The subsidy perspective turns out to be useful when designing learning algorithms. In particular, while the formulation in Definition 4.1 involves a maximization over the $2^{|\mathcal{A}|}$ subsets of $\mathcal{A}$, the linear programming formulation (5) only involves $O(|\mathcal{A}|)$ variables and $O(|\mathcal{A}|^2)$ constraints.

Subset Instability as the platform's cost of learning. We next connect minimum stabilizing subsidies to the platform's *cost of learning*—how much the platform would have to pay to keep users on the platform in the presence of a worst-case (but budget-balanced) competitor with perfect knowledge of agent utilities.

Observe that (4) is the minimum amount the platform could subsidize agents so that no budget-balanced competitor could convince agents to leave. The way that we formalize "convincing agents to leave" is that: (a) an agent will leave the original platform if they prefer to be unmatched over being on the platform, or (b) a pair of agents who are matched on the competitor's platform will leave the original platform if they both prefer the new market outcome over their original market outcomes. Thus, if we imagine the platform as actually paying the subsidies, then the cumulative instability (i.e., our regret) can be realized as a "cost of learning": it is how much the platform pays the agents to learn a stable outcome while ensuring that no agent has the incentive to leave during the learning process. Later on, we will see that our algorithmic approach can be extended to efficiently compute feasible subsidies for (5) that are within a constant factor of our regret bound, meaning that subsidies can be implemented using only the information that the platform has. Moreover, in Section 6.2, we show that cost of learning can also be explicitly connected to the platform's revenue.

Subset Instability as a measure of user unhappiness. While the above interpretations focus on Subset Instability from the platform's perspective, we show that Subset Instability can also be interpreted as a measure of user unhappiness. Given a subset $\mathcal{S} \subseteq \mathcal{A}$ of agents, which we call a coalition, we define the *unhappiness* of $\mathcal{S}$ with respect to a market outcome (X, τ) to be the maximum gain (relative to (X, τ)) in total utility that the members of coalition $\mathcal{S}$ could achieve by matching only among themselves, such that no member is worse off than they were in (X, τ). (See Appendix B.3 for a formal definition.) The condition that no member is worse off ensures that all agents would actually want to participate in the coalition (i.e. they prefer it to the original market outcome).

User unhappiness differs from the original definition ($*$) of Subset Instability in ($*$), because ($*$) does not require individuals to be better off in any alternative matching. However, we show that this difference is inconsequential:

Proposition 4.3. *The maximum unhappiness of any coalition $\mathcal{S} \subseteq \mathcal{A}$ with respect to (X, τ) equals the Subset Instability $I(X, \tau; u, \mathcal{A})$.*

See Appendix B.3 for a full proof. In the proof, we relate the maximum unhappiness of any coalition to the dual linear program to (5). To show this relation, we leverage the fact that optimal solutions to the dual program correspond to blocking pairs of agents as well as individual rationality violations.

The main takeaway from Proposition 4.3 is that Subset Instability not only measures costs to the platform, but also costs to users, in terms of the maximum amount they "leave on the table" by not negotiating an alternate arrangement amongst themselves.

Subset Instability as a distance from equilibrium. Finally, we connect Subset Instability to solution concepts for coalitional games, a general concept in game theory that includes matching with transfers as a special case. Coalitional games (also known as cooperative games) capture competition and cooperation amongst a group of agents. The *core* is the set of outcomes in a cooperative game such that no subset $\mathcal{S}$ of agents can achieve higher total utility among themselves than according to the given outcome. In games where the core is empty, a natural relaxation is the *strong ε-core* [SS66], which is the set of outcomes in a cooperative game such that no subset $\mathcal{S}$ of agents can achieve total utility among themselves that is at least ε greater than according to the given outcome.

Subset Instability can be seen as transporting the strong ε-core notion to a slightly different context. In particular, in the context of matching with transferable utilities, the core is exactly the set of stable matchings; since a stable matching always exists, the core is always nonempty. Even though the core is nonempty, we can nonetheless use the strong ε-core to measure *distance from the core*. More specifically, it is natural to consider the smallest ε such that (X, τ) is in the strong ε-core. This definition exactly aligns with Subset Instability, thus providing an alternate interpretation of Subset Instability within the context of coalitional game theory.

4.2 Properties of Subset Instability

We now describe additional properties of our instability measure that are important for learning. We show that Subset Instability is: (i) zero if and only if the matching with transfers is stable, (ii) Lipschitz in the true utility functions, and (iii) lower bounded by the utility difference.

Proposition 4.4. *Subset Instability satisfies the following properties:*

1. *Subset Instability is always nonnegative and is zero if and only if (X, τ) is stable.*
2. *Subset Instability is Lipschitz continuous with respect to agent utilities. That is, for any possible market outcome (X, τ), and any pair of utility functions u and $\tilde{u}$ it holds that:*

$$|I(X, \tau; u, \mathcal{A}) - I(X, \tau; \tilde{u}, \mathcal{A})| \leq 2 \sum_{a \in \mathcal{A}} \|u_a - \tilde{u}_a\|_\infty.$$

3. *Subset Instability is always at least the utility difference.*

We defer the proof to Appendix B.4.

These three properties show that Subset Instability is useful as a regret measure for learning stable matchings. The first property establishes that Subset Instability satisfies the basic desideratum of having zero instability coincide with exact stability. The second property shows that Subset Instability is robust to small perturbations to the utility functions of individual agents. The third property ensures that, when learning using Subset Instability as a loss function, the platform learns a socially optimal matching.

Note that the second property already implies the existence of an explore-then-commit algorithm that achieves $\widetilde{O}(N^{4/3}T^{2/3})$ regret in the simple setting where $\mathcal{A}^t = \mathcal{A}$ for some $\mathcal{A}$ of size N for all t.[9] In the next section, we will explore algorithms that improve the dependence on the number of rounds T to $\sqrt{T}$ and also work in more general settings.

[9] This bound can be achieved by adapting the explore-then-commit (ETC) approach where the platform explores by choosing each pair of agents $\widetilde{O}((T/N)^{2/3})$ times [LS20]. Thus, $\widetilde{O}(N^{1/3}T^{2/3})$ rounds are spent exploring, and the Subset Instability of the matching selected in the commit phase is $\widetilde{O}(N^{4/3}T^{2/3})$ with high probability. We omit further details since this analysis is a straightforward adaptation of the typical ETC analysis.

5 Regret Bounds

In this section, we develop a general approach for designing algorithms that achieve near-optimal regret within our framework. To be precise, the platform's regret is defined to be

$$R_T = \sum_{t=1}^{T} I(X^t, \tau^t; u, \mathcal{A}^t).$$

While our framework bears some resemblance to the (incentive-free) combinatorial bandit problem of learning a maximum weight matching, two crucial differences differentiate our setting: (i) in each round, the platform must choose *transfers* in addition to a matching, and (ii) loss is measured with respect to *instability* rather than the utility difference. Nonetheless, we show that a suitable interpretation of "optimism in the face of uncertainty" can still apply.

Regret bounds for different preference structures. By instantiating this optimism-based approach, we derive regret bounds for the preference structures introduced in Section 3. We start with the simplest case of unstructured preferences, where we assume no structure on the utilities.

Theorem 5.1. *For preference class $\mathcal{U}_{\text{unstructured}}$ (see Section 3),* MatchUCB *(defined in Section 5.3) incurs expected regret $\mathbb{E}(R_T) = O\big(|\mathcal{A}|\sqrt{nT\log(|\mathcal{A}|T)}\big)$, where $n = \max_t |\mathcal{A}_t|$.*

In Section 5.4, we additionally give a matching (up to logarithmic factors) lower bound showing for $n = |\mathcal{A}|$ that such scaling in $|\mathcal{A}|$ is indeed necessary. This demonstrates that the regret scales with $|\mathcal{A}|\sqrt{n}$, which is superlinear in the size of the market. Roughly speaking, this bound means that the platform is required to learn a superconstant amount of information per agent in the marketplace. These results suggest that without preference structure, it is unlikely that a platform can efficiently learn a stable matching in large markets.

The next two bounds demonstrate that, with preference structure, efficient learning of a stable matching becomes possible. First, we consider typed preferences, which are purely specified by a function f mapping finitely many pairs of contexts to utilities.

Theorem 5.2. *For preference class $\mathcal{U}_{\text{typed}}$ (see Section 3),* MatchTypedUCB *(defined in Section 5.3) incurs expected regret $\mathbb{E}(R_T) = O\big(|\mathcal{C}|\sqrt{nT\log(|\mathcal{A}|T)}\big)$, where $n = \max_t |\mathcal{A}_t|$.*

For a fixed type space $\mathcal{C}$, the regret bound in Theorem 5.2 scales sublinearly with the market size (captured by $|\mathcal{A}|$ and n). This demonstrates that the platform can efficiently learn a stable matching when preferences are determined by types. In fact, the regret bound only depends on the number of agents who arrive on the platform in any round; notably, it does not depend on the total number of agents on the platform (beyond logarithmic factors).

Finally, we consider separable linear preferences, where the platform needs to learn hidden information associated to each agent.

Theorem 5.3. *For preference class $\mathcal{U}_{\text{linear}}$ (see Section 3),* MatchLinUCB *(defined in Section 5.3) incurs expected regret $\mathbb{E}(R_T) = O\big(d\sqrt{|\mathcal{A}|}\sqrt{nT\log(|\mathcal{A}|T)}\big)$, where $n = \max_t |\mathcal{A}_t|$.*

When n is comparable to $|\mathcal{A}|$, the regret bound in Theorem 5.3 scales linearly with the market size (captured by $|\mathcal{A}|$) and linearly with the dimension d. Roughly speaking, this means that the platform learns (at most) a constant amount of information per agent in the marketplace. We interpret this as indicating that the platform can efficiently learn a stable matching in large markets for separable linear preferences, although learning in this setting is more demanding than for typed preferences.

5.1 Algorithm

Following the principle of optimism, our algorithm selects at each round a stable market outcome using upper confidence bounds as if they were the true agent utilities. To design and analyze this algorithm, we leverage the fact that, in the full-information setting, stable market outcomes are optimal solutions to a pair of primal-dual linear programs whose coefficients depend on agents' utility functions. This primal-dual perspective lets us compute a market outcome each round. A particular consequence is that any UCB-based algorithm for learning matchings in a semi-bandit setting can be transformed into an algorithm for learning *both* the matching and the prices.

Stable market outcomes via linear programming duality. Before proceeding with the details of our algorithm, we review how the primal-dual framework can be used to select a stable market outcome in the full information setting. Shapley and Shubik [SS71] show that stable market outcomes (X, τ) correspond to optimal primal-dual solutions to the following pair of primal and dual linear programs (where we omit the round index t and consider matchings over $\mathcal{A} = \mathcal{I} \cup \mathcal{J}$):

Primal (P)

$$\max_{Z \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{J}|}} \sum_{(i,j) \in \mathcal{I} \times \mathcal{J}} Z_{i,j}(u_i(j) + u_j(i))$$

$$\text{s.t.} \sum_{j \in \mathcal{J}} Z_{i,j} \leq 1 \quad \forall i \in \mathcal{I}$$

$$\sum_{i \in \mathcal{I}} Z_{i,j} \leq 1 \quad \forall j \in \mathcal{J}$$

$$Z_{i,j} \geq 0 \quad \forall (i,j) \in \mathcal{I} \times \mathcal{J}$$

Dual (D)

$$\min_{p \in \mathbb{R}^{|\mathcal{A}|}} \sum_{a \in \mathcal{A}} p_a$$

$$\text{s.t. } p_i + p_j \geq u_i(j) + u_j(i) \quad \forall (i,j) \in \mathcal{I} \times \mathcal{J}$$

$$p_a \geq 0 \quad \forall a \in \mathcal{A}$$

The primal program (P) is a linear programming formulation of the maximum weight matching problem: the Birkhoff-von Neumann theorem states that its extreme points are exactly the indicator vectors for matchings between $\mathcal{I}$ and $\mathcal{J}$. Each dual variable p_a in (D) can be interpreted as a *price* that roughly corresponds to agent a's net utility. Specifically, given any optimal primal-dual pair (Z, p), one can recover a matching μ_X from the nonzero entries of Z and set transfers $\tau_a = p_a - u_a(\mu_X(a))$ to obtain a stable outcome (X, τ). Moreover, any stable outcome induces an optimal primal-dual pair (Z, p).

Overview of the algorithm. In each round, we compute a matching with transfers by solving the primal-dual linear programs for our upper confidence bounds: Suppose we have a collection $\mathscr{C}$ of confidence sets $C_{i,j}, C_{j,i} \subseteq \mathbb{R}$ such that $u_i(j) \in C_{i,j}$ and $u_j(i) \in C_{j,i}$ for all $(i,j) \in \mathcal{I} \times \mathcal{J}$. Our algorithm uses $\mathscr{C}$ to get an upper confidence bound for each agent's utility function and then computes a stable matching with transfers as if these upper confidence bounds were the true utilities (see ComputeMatch). This can be implemented efficiently if we use, for instance, the Hungarian algorithm [Kuh55] to solve (P) and (D).

5.2 Main lemma

The key fact we need to analyze our algorithms is that Subset Instability is upper bounded by the sum of the sizes of the relevant confidence sets, assuming that the confidence sets contain the true utilities. (In the following, we again omit the round index t.)

Lemma 5.4. *Suppose a collection of confidence sets $\mathscr{C}$ is such that $u_i(j) \in C_{i,j}$ and $u_j(i) \in C_{j,i}$ for all $(i,j) \in \mathcal{I} \times \mathcal{J}$. Then the instability of the output $(X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}})$ of* ComputeMatch$(\mathscr{C})$ *satisfies*

$$I(X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u, \mathcal{A}^t) \leq \sum_{a \in \mathcal{A}} \Big(\max\big(C_{a,\mu_{X^{\mathrm{UCB}}}(a)}\big) - \min\big(C_{a,\mu_{X^{\mathrm{UCB}}}(a)}\big) \Big). \tag{6}$$

Proof. Because $(X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}})$ is stable with respect to u^{UCB}, we have that $I(X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u^{\mathrm{UCB}}, \mathcal{A}^t) = 0$. Thus, it is equivalent to bound the difference $I(X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u, \mathcal{A}^t) - I(X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u^{\mathrm{UCB}}, \mathcal{A}^t)$.

At this stage, it might be tempting to bound this difference using the Lipschitz continuity of Subset Instability (see Proposition 4.4). However, this would only allow us to obtain an upper bound of the form $\sum_{a \in \mathcal{A}} \max_{a' \in \mathcal{A}} \big(\max\big(C_{a,a'}\big) - \min\big(C_{a,a'}\big)\big)$. The problem with this bound is that it depends on the sizes of the confidence sets for all pairs of agents, including those that are *not* matched in X^{UCB}, making it too weak to prove regret bounds for UCB-style algorithms.[10] Thus, we proceed with a more fine-grained analysis.

Define the function

$$f(\mathcal{S}, X, \tau; u) = \left(\max_{X' \in \mathscr{X}_{\mathcal{S}}} \sum_{a \in \mathcal{S}} u_a(\mu_{X'}(a)) \right) - \left(\sum_{a \in \mathcal{S}} u_a(\mu_X(a)) + \tau_a \right).$$

By definition, $I(X, \tau; u, \mathcal{A}) = \max_{\mathcal{S} \subseteq \mathcal{A}} f(\mathcal{S}, X, \tau; u)$. It follows that

$$\begin{aligned} &I(X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u, \mathcal{A}^t) - I(X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u^{\mathrm{UCB}}, \mathcal{A}^t) \\ &\qquad \leq \max_{\mathcal{S} \subseteq \mathcal{A}} \left(f(\mathcal{S}, X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u) - f(\mathcal{S}, X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u^{\mathrm{UCB}}) \right). \end{aligned}$$

From here, the proof boils down to upper bounding $f(\mathcal{S}, X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u) - f(\mathcal{S}, X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u^{\mathrm{UCB}})$ for each $\mathcal{S} \subseteq \mathcal{A}$. We decompose this expression into two terms:

$$\begin{aligned} &f(\mathcal{S}, X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u) - f(\mathcal{S}, X^{\mathrm{UCB}}, \tau^{\mathrm{UCB}}; u^{\mathrm{UCB}}) \\ &\qquad = \underbrace{\left(\max_{X' \in \mathscr{X}_{\mathcal{S}}} \sum_{a \in \mathcal{S}} u_a(\mu_{X'}(a)) - \max_{X' \in \mathscr{X}_{\mathcal{S}}} \sum_{a \in \mathcal{S}} u_a^{\mathrm{UCB}}(\mu_{X'}(a)) \right)}_{\text{(A)}} \\ &\qquad\qquad + \underbrace{\left(\sum_{a \in \mathcal{S}} \big(u_a^{\mathrm{UCB}}(\mu_{X^{\mathrm{UCB}}}(a)) + \tau_a^{\mathrm{UCB}}\big) - \sum_{a \in \mathcal{S}} \big(u_a(\mu_{X^{\mathrm{UCB}}}(a)) + \tau_a^{\mathrm{UCB}}\big) \right)}_{\text{(B)}}. \end{aligned}$$

To see that (A) is nonpositive, observe that the maximum weight matching of $\mathcal{S}$ with respect to u is no larger than the maximum weight matching of $\mathcal{S}$ with respect to u^{UCB}, since u^{UCB} pointwise upper bounds u. To upper bound (B), observe that the transfers cancel out, so the expression is equivalent to

$$\sum_{a \in \mathcal{S}} \big(u_a^{\mathrm{UCB}}(\mu_{X^{\mathrm{UCB}}}(a)) - u_a(\mu_{X^{\mathrm{UCB}}}(a))\big) \leq \sum_{a \in \mathcal{A}} \Big(\max\big(C_{a,\mu_{X^{\mathrm{UCB}}}(a)}\big) - \min\big(C_{a,\mu_{X^{\mathrm{UCB}}}(a)}\big) \Big). \qquad \square$$

[10] For intuition, consider the classical stochastic multi-armed bandits setting and suppose that we could only guarantee that the loss incurred by an arm is bounded by the maximum of the sizes of the confidence sets over *all* arms. Then, we would only be able to obtain a weak bound on regret, since low-reward arms with large confidence sets may never be pulled.

Algorithm 1 ComputeMatch: Compute matching with transfers from confidence sets

procedure ComputeMatch($\mathscr{C}$)
  for $(i,j) \in \mathcal{I} \times \mathcal{J}$ **do** ▷ Instantiate UCB estimates of utilities.
    $u_i^{\mathrm{UCB}}(j) \leftarrow \max(C_{i,j})$
    $u_j^{\mathrm{UCB}}(i) \leftarrow \max(C_{j,i})$
  $(X^*, p^*) \leftarrow$ optimal primal-dual pair for (P) and (D) given utilities u^{UCB}
  for $a \in \mathcal{A}$ **do** ▷ Set transfers based on (X^*, p^*) and UCB utilities.
    $\tau_a \leftarrow p_a^* - u_a^{\mathrm{UCB}}(\mu_{X^*}(a))$
  return (X^*, τ)

5.3 Explicit algorithms

The regret bound of Lemma 5.4 suggests a simple algorithmic approach: each round, select the matching with transfers returned by ComputeMatch and update confidence sets accordingly. To instantiate this approach, it remains to construct confidence intervals that contain the true utilities with high probability. This last step naturally depends on the assumptions made about the utilities and the noise.

Unstructured preferences. For this setting, we construct confidence intervals following the classical UCB approach: for each utility value involving the pair $(i,j) \in \mathcal{I} \times \mathcal{J}$, we take a length $O\big(\sqrt{\log(|\mathcal{A}|T)/n_{ij}}\big)$ confidence interval centered around the empirical mean, where n_{ij} is the number of times the pair has been matched thus far. We describe this construction precisely in Algorithm 2 (MatchUCB).

Algorithm 2 MatchUCB: A bandit algorithm for matching with transferable utilities for unstructured preferences.

procedure MatchUCB(T)
  for $(i,j) \in \mathcal{I} \times \mathcal{J}$ **do** ▷ Initialize confidence intervals.
    $C_{i,j} \leftarrow [-1, 1]$
    $C_{j,i} \leftarrow [-1, 1]$
  for $1 \le t \le T$ **do**
    $(X^t, \tau^t) \leftarrow$ ComputeMatch($\mathscr{C}$)
    for $(i,j) \in X^t$ **do** ▷ Set confidence intervals and update means.
      Update empirical means $\hat{u}_i(j)$ and $\hat{u}_j(i)$ from feedback; increment counter n_{ij}.
      $C_{i,j} \leftarrow \big[\hat{u}_i(j) - 8\sqrt{\log(|\mathcal{A}|T)/n_{ij}}, \hat{u}_i(j) + 8\sqrt{\log(|\mathcal{A}|T)/n_{ij}}\big] \cap [-1, 1]$
      $C_{j,i} \leftarrow \big[\hat{u}_j(i) - 8\sqrt{\log(|\mathcal{A}|T)/n_{ij}}, \hat{u}_j(i) + 8\sqrt{\log(|\mathcal{A}|T)/n_{ij}}\big] \cap [-1, 1]$

To analyze MatchUCB, recall that Lemma 5.4 bounds the regret at each step by the lengths of the confidence intervals of each pair in the selected matching. Bounding the lengths of the confidence intervals parallels the analysis of UCB for classical stochastic multi-armed bandits. We give the full proof of Theorem 5.1 in Appendix C.1.

Typed Preferences. For this setting, we construct our confidence intervals as follows: for each pair of types c_1 and c_2, we take a length $O\big(\sqrt{\log(|\mathcal{A}|T)/n_{c_1c_2}}\big)$ confidence interval centered around the empirical mean, where $n_{c_1c_2}$ is the number of times that an agent with type c_1 has been matched with an agent with type c_2. We describe this construction precisely in Algorithm 3 (MatchTypedUCB). We give the full proof of Theorem 5.2 in Appendix C.2.

Algorithm 3 MATCHTYPEDUCB: A bandit algorithm for matching with transferable utilities for typed preferences.

procedure MATCHTYPEDUCB(T)
  for $(c, c') \in \mathcal{C} \times \mathcal{C}$ **do** ▷ Initialize confidence intervals and empirical means.
    $C_{c,c'} \leftarrow [-1, 1]$
  for $1 \leq t \leq T$ **do**
    $(X^t, \tau^t) \leftarrow$ COMPUTEMATCH($\mathscr{C}$)
    for $(i, j) \in X^t$ **do** ▷ Set confidence intervals and update means.
      Update empirical means $\hat{f}(c_i, c_j)$ and $\hat{f}(c_i, c_j)$ from feedback; increment counter n_{c_i,c_j}.
      $C_{c_i,c_j} \leftarrow \left[\hat{f}(c_i, c_j) - 8\sqrt{\log(|\mathcal{A}|T)/n_{c_i,c_j}}, \hat{f}(c_i, c_j) + 8\sqrt{\log(|\mathcal{A}|T)/n_{c_i,c_j}}\right] \cap [-1, 1]$
      $C_{c_j,c_i} \leftarrow \left[\hat{f}(c_j, c_i) - 8\sqrt{\log(|\mathcal{A}|T)/n_{c_i,c_j}}, \hat{f}(c_i, c_j) + 8\sqrt{\log(|\mathcal{A}|T)/n_{c_i,c_j}}\right] \cap [-1, 1]$

Algorithm 4 MATCHLINUCB: A bandit algorithm for matching with transferable utilities for separable linear preferences.

procedure MATCHLINUCB(T)
  for $(i, j) \in \mathcal{I} \times \mathcal{J}$ **do** ▷ Initialize confidence intervals.
    $C_{i,j} \leftarrow [-1, 1]$
    $C_{j,i} \leftarrow [-1, 1]$
  for $1 \leq t \leq T$ **do**
    $(X^t, \tau^t) \leftarrow$ COMPUTEMATCH($\mathscr{C}$)
    for $a \in \mathcal{A}^t$ **do** ▷ Update confidence intervals.
      Increment the counter n_a.
      $\beta \leftarrow O\left(d \log T + \frac{n_a\sqrt{\ln(n_a/(T|A|))}}{T^2}\right)$. ▷ Parameter for width of confidence set.
      if $\mu_{X^t}(a) \neq a$ **then**
        Add t to $\mathcal{T}_a$ (the set of rounds in which agent a has been matched).
        Set $\mathcal{R}_{a,t}$ equal to the observed utility for agent a in round t.
        $\phi^{\mathrm{LS}}(a) \leftarrow \operatorname{argmin}_{v \in \mathcal{B}^d}\left(\sum_{t' \in \mathcal{T}_a}\left(\langle v, c_{\mu_{X_{t'}}(a)}\rangle - \mathcal{R}_{a,t'}\right)^2\right)$ ▷ Least squares estimate.
        $C_{\phi(a)} \leftarrow \left\{v \mid \sum_{t' \in \mathcal{T}_a}\left(\langle v - \phi^{\mathrm{LS}}(a), c_{\mu_{X_{t'}}(a)}\rangle\right)^2 \leq \beta, \|v\|_2 \leq 1\right\}$ ▷ Conf. ellipsoid.
        for $a' \in \mathcal{A}$ **do**
          $C_{a,a'} \leftarrow \left\{\langle c_{a'}, v\rangle \mid v \in C_{\phi(a)}\right\} \cap [-1, 1]$. ▷ Update confidence sets involving a.

Separable Linear Preferences. To build the confidence sets, we use the high-level idea from LinUCB [RR13; LS20]. The idea is to compute a confidence set for each hidden vector $\phi(a)$ using the least squares estimate and use that to construct confidence sets for the utilities $C_{a,a'}$.

More formally, let $\mathcal{T}_a$ be the set of rounds where agent a is matched on the platform thus far, and for $t' \in \mathcal{T}_a$, let $\mathcal{R}_{a,t'}$ be the observed utility at time t' for agent a. The center of the confidence set will be given by the least squares estimate

$$\phi^{\mathrm{LS}}(a) = \arg\min_{v \in \mathcal{B}^d} \left(\sum_{t' \in \mathcal{T}_a} \left(\langle v, c_{\mu_{X_{t'}}(a)} \rangle - \mathcal{R}_{a,t'} \right) \right).$$

The confidence set for $\phi(a)$ is given by

$$C_{\phi(a)} := \left\{ v \;\middle|\; \sum_{t' \in \mathcal{T}_{a,t}} \left\langle v - \phi^{\mathrm{LS}}(a), c_{\mu_{X_{t'}}(a)} \right\rangle^2 \leq \beta \text{ and } \|v\|_2 \leq 1 \right\},$$

where $\beta = O\left(D \log T + \frac{n_a \sqrt{\ln(n_a/\delta)}}{T^2} \right)$ and n_a counts the number of times that a has appeared in selected matchings. The confidence set for $u(a, a')$ is given by

$$C_{a,a'} := \left\{ \langle c_{a'}, v \rangle \mid v \in C_{\phi(a)} \right\} \cap [-1, 1].$$

We describe this construction precisely in Algorithm 4 (MatchLinUCB). We give the full proof of Theorem 5.3 in Appendix C.3.

5.4 Matching lower bound

For the case of unstructured preferences, we now show that MatchUCB achieves optimal regret (up to logarithmic factors) by showing a lower bound that (nearly) matches the upper bound in Theorem 5.1.

Lemma 5.5. *For any algorithm that learns a stable matching with respect to unstructured preferences, there exists an instance on which it has expected regret $\widetilde{\Omega}(|A|^{3/2}\sqrt{T})$ (where regret is given by Subset Instability).*

The idea behind this lemma is to show a lower bound for the easier problem of learning a maximum weight matching using utility difference as regret. By Proposition 4.4, this immediately implies a lower bound for learning a stable matching with regret measured by Subset Instability.

This lower bound illustrates the close connection between our setting and that of learning a maximum weight matching. Indeed, by applying MatchUCB and simply disregarding the transfers every round, we recover the classical UCB-based algorithm for learning the maximum weight matching [GKJ12; CWY13; KWA+15]. From this perspective, the contribution of MatchUCB is an approach to set the dual variables while asymptotically maintaining the same regret as the primal-only problem.

6 Extensions and Discussion

In this section, we discuss several extensions of our results; these extensions illustrate the generality of our framework and also suggest several avenues for future research.

6.1 Instance-dependent regret bounds

While our analyses in Section 5.1 focused on bounds that hold uniformly for all problem instances, we now explore *instance-dependent* regret bounds. Instance-dependent bounds capture a different facet of bandit algorithms: how does the number of mistakes made by the algorithm scale on each instance with respect to T? Bounds of this nature have been explored in previous works [LMJ20; BSS21; SBS21; CS21; LRM+20] on learning stable matchings in the non-transferable utilities setting, and we show that they can be obtained within our framework as well.

Our instance-dependent regret bound depends on a gap $\Delta > 0$ determined by the true utility function u. We focus on the setting where agent utilities are unstructured (i.e., $u \in \mathcal{U}_{\text{unstructured}}$) and where the same set of agents $\mathcal{A}$ arrives in each round. As is common in analyses of combinatorial bandit problems (e.g., [KWA+15; CWY13]), the gap Δ in the bound is global to the matching. Letting X^{opt} be a maximum weight matching with respect to u, we define the gap to Δ be the difference in utility between the optimal and second-best matchings:[11]:

$$\Delta = \inf_{X \neq X^{\text{opt}}} \left\{ \sum_{a \in \mathcal{A}} u_a(\mu_{X^{\text{opt}}}(a)) - \sum_{a \in \mathcal{A}} u_a(\mu_X(a)) \right\}.$$

We prove the following regret bound:

Theorem 6.1 (Instance-Dependent Regret)**.** *Suppose that $\mathcal{A}_t = \mathcal{A}$ for all t. Let $u \in \mathcal{U}_{\text{unstructured}}$ be any utility function, and put*

$$\Delta = \inf_{X \neq X^*} \left\{ \sum_{a \in \mathcal{A}} u_a(\mu_{X^*}(a)) - \sum_{a \in \mathcal{A}} u_a(\mu_X(a)) \right\}.$$

Then MatchUCB′ *incurs expected regret $\mathbb{E}(R_T) = O(|\mathcal{A}|^5 \cdot \log(|\mathcal{A}|T)/\Delta^2)$.*

Remark. MatchUCB′ is MatchUCB with a slight adjustment to ComputeMatch needed to prove Theorem 6.1. MatchUCB′, like MatchUCB, does not depend on the gap Δ and achieves the instance-independent regret bound in Theorem 5.1.[12] That is, MatchUCB′ achieves both our instance-independent and instance-dependent regret bounds.

Our starting point for proving Theorem 6.1 is the analysis in Chen, Wang, and Yuan [CWY13] for the combinatorial bandits problem of learning a maximum weight matching. The analysis in [CWY13] proceeds by upper bounding the number of "mistakes" that a platform makes while exploring and learning, i.e., the number of rounds where the chosen matching is suboptimal. Analogously, we bound the number of rounds where the chosen market outcome is not stable with respect to the true utilities u.

To make this argument work, it turns out that we need to specify more precisely how the primal-dual solution is chosen in line 5 of ComputeMatch (which we previously did not specify). In particular, poor choices of the primal-dual solution can lead to many rounds where the chosen outcome is unstable. To see this, consider a market with a single customer C and a single provider P such that $u_C(P) = 2$ and $u_P(C) = -1$, and suppose we have nearly tight upper bounds $u_C^{\text{UCB}}(P) = 2 + \varepsilon$ and $u_P^{\text{UCB}}(C) = -1 + \varepsilon$ on the utilities. Then the market outcome with matching $\{(C, P)\}$ with $\tau_C = -2 - \varepsilon$ and $\tau_P = -\tau_C$ could be selected by ComputeMatch, since it corresponds to an optimal primal-dual pair for u^{UCB}. However, it is not stable with respect to the true utilities u (as individual rationality is violated for C), regardless

[11] Our bound is less fine-grained than the gap in [CWY13], and in particular does not allow there to be multiple maximum weight matchings. We defer improving our definition of Δ to future work.

[12] The instance-independent regret bound can be shown using the same argument as the proof for Theorem 5.1.

of how small ε is. Thus, without assuming more about how the optimal primal-dual pair is chosen in ComputeMatch, we cannot hope to bound the number of unstable market outcomes selected.

In Appendix D, we describe how we choose optimal primal-dual pairs and tweak ComputeMatch in MatchUCB$'$, and we provide a full proof of Theorem 6.1.

Theorem 6.1 opens the door to further exploring algorithmic properties of learning stable matchings. First, this result establishes fine-grained regret bounds, demonstrating the typical $O(\log T)$ regret bounds from the combinatorial bandits literature [CWY13] are achievable in our setting as well. Second, Theorem 6.1 provides insight into the number of mistakes made by the platform. In particular, we show within the proof of Theorem 6.1 that the platform fails to choose a matching that is stable with respect to u in at most $O(|\mathcal{A}|^5 \cdot \log(|\mathcal{A}|T)/\Delta^2)$ rounds.[13] This means that the platform selects a stable matching in at least $T - O(|\mathcal{A}|^5 \cdot \log(|\mathcal{A}|T)/\Delta^2) = T - O(\log T)$ of the rounds.

As we described, our bounds in Theorem 6.1 rely on choosing an appropriate primal-dual solution. An interesting direction for future work would be to provide further insight into how different methods for finding an optimal primal-dual pairs affect both regret bounds and the trajectory of the selected market outcomes over time.

6.2 Search frictions and platform revenue

Next, we further ground Subset Instability by explicitly connecting it to the platform's revenue under a stylized economic model of search frictions. A major motivation for this is that it helps explain when an online platform can earn a profit in competitive settings, even when they start out with no information about agent preferences.

More specifically, we incorporate search frictions where an agent must lose ε utility to find an alternative to the given match (e.g. from the time spent finding an alternate partner, or from a cancellation fee). These search frictions weaken the requirements for stability: the platform now only needs matchings to be *ε-stable*, i.e.,

$$u_i(j) + u_j(i) - 2\varepsilon \leq u_i(\mu_X(i)) + \tau_i + u_j(\mu_X(j)) + \tau_j$$

for all $(i, j) \in \mathcal{I} \times \mathcal{J}$ and $u_a(\mu_X(a)) + \tau_a \geq -\varepsilon$ for all $a \in \mathcal{A}$.[14]

To model revenue, we take the subsidy perspective on Subset Instability. Specifically, recall that Subset Instability was equal to the minimum subsidy needed to maintain stability (see Proposition 4.2). With search frictions, that subsidy can potentially be *negative*, thus allowing the platform to generate revenue. We are interested in analyzing the maximum revenue (minimum subsidy) the platform can generate while ensuring stability with high probability over all rounds. For realism, we also want this subsidy to be computed online using only information that the platform has access to, but it turns out we can do this with minimal modifications to our algorithm.

More formally, in this modified model, the platform must select an ε-stable matching in each round with high probability by choosing appropriate subsidies. That is, in round t, the platform selects a matching with transfers (X^t, τ^t) with the modification that the transfers need not be zero-sum. The transfers thus incorporate the amount that platform is subsidizing or charging agents for participation on the platform.

[13] The number of mistakes necessarily depends on the gap Δ because there exist utility functions u and $\tilde{u}$ where $\|u - \tilde{u}\|_\infty$ is arbitrary small, but where the stable market outcomes with respect to u and $\tilde{u}$ differ. To see this, consider a market where $\mathcal{I} = \{C\}$ and $\mathcal{J} = \{P\}$. Suppose that $u_C(P) = \tilde{u}_C(P) = 1$, while $u_P(C) = -1 + \varepsilon$ and $\tilde{u}_P(C) = -1 - \varepsilon$. Then, the maximum weight matchings under these utility functions differ: $\{(C, P)\}$ is the only maximum weight matching in the former, whereas $\emptyset$ is the only maximum weight matching in the latter.

[14] This definition corresponds to (X, τ) belonging to the weak ε-core of Shapley and Shubik [SS66]. We note that this definition also relaxes individual rationality. This formulation gives us the cleanest algorithmic results; while it can be extended to an analogue that does not relax individual rationality, it would involve bounds that (necessarily) depend on the specifics of agents' utilities.

The net profit of the platform is then $-\sum_{t=1}^{T}\sum_{a\in\mathcal{A}}\tau_a^t$. We impose the stability requirement that

$$\mathbb{P}[(X^t,\tau^t) \text{ is } \varepsilon\text{-stable for all } 1\le t\le T]\ge 0.99.$$

Given this setup, we show the following:

Theorem 6.2. *For preference class $\mathcal{U}_{\text{unstructured}}$ (see Section 3), there exists an algorithm giving the platform*

$$\varepsilon(\sum_{t=1}^{T}|\mathcal{A}_t|)T - O\big(|\mathcal{A}|\sqrt{nT}\sqrt{\log(|\mathcal{A}||T|)}\big)$$

revenue in the presence of search frictions while maintaining stability with high probability.

Remark. In particular if $\mathcal{A}_t = \mathcal{A}$ in every round, the platform will starting making a profit within $O(|\mathcal{A}|/\varepsilon^2\cdot\log(|\mathcal{A}|/\varepsilon^2))$ rounds.

We defer the proof of Theorem 6.2 to Appendix E.

Qualitatively, Theorem 6.2 captures that if the platform "pays to learn" in initial rounds, the information that it obtains will help it achieve a profit in the long run. We note that both the revenue objective and the model for search frictions that we consider in these preliminary results are stylized. An interesting direction for future work would be to integrate more realistic platform objectives and models for search frictions into the framework.

6.3 Matching with non-transferable utilities

While we have focused on matching with transferable utilities, utilities are not always transferable in practice, as in the cases of dating markets and college admissions (i.e., most people are not willing to date an undesirable partner in exchange for money, and a typical college admission slot is not sold for money). We can extend our findings to this setting following the model of matching with non-transferable utilities (NTU) [GS62], which has also been studied in previous work [DK05; LMJ20; CS21; SBS21]. The definition of Subset Instability extends naturally and has advantages over the "utility difference" metric that is commonly used in prior work. Our algorithmic meta-approach also sheds new light on the convergence properties of the centralized UCB algorithm of Liu, Mania, and Jordan [LMJ20].

The starting point of our instability measure is slightly different than in Section 4. Since stable matchings in the NTU model need not maximize total utility, we cannot define instability based on a maximum over all subsets of agents of the utility difference for that subset. On the other hand, the subsidy formulation of Subset Instability (see (4)) translates well to this setting. Our instability measure will correspond to the minimum amount the platform could subsidize agents so that individual rationality holds and no blocking pairs remain. For matching with NTU, we formalize this notion as follows:

Definition 6.3 (NTU Subset Instability)**.** For utilities u and agents $\mathcal{A}$, the *NTU Subset Instability* $I(X;u,\mathcal{A})$ of a matching X is

$$\begin{aligned}
&\min_{s\in\mathbb{R}^{|\mathcal{A}|}}\sum_{a\in\mathcal{A}}s_a && && (\dagger)\\
&\text{s.t. } \min\big(u_i(j)-u_i(\mu_X(i))-s_i,\,u_j(i)-u_j(\mu_X(j))-s_j\big)\le 0 && \forall(i,j)\in\mathcal{I}\times\mathcal{J}\\
&\quad u_a(\mu_X(a))+s_a\ge 0 && \forall a\in\mathcal{A}\\
&\quad s_a\ge 0 && \forall a\in\mathcal{A}.
\end{aligned}$$

NTU Subsidy Instability inherits some of the same appealing properties as Subsidy Instability.

Proposition 6.4 (Informal). *NTU Subset Instability satisfies the following properties:*

1. *NTU Subset Instability is always nonnegative and is zero if and only if (X, τ) is stable.*
2. *NTU Subset Instability is Lipschitz continuous with respect to agent utilities. That is, for any matching X and any pair of utility functions u and $\tilde{u}$, it holds that:*

$$|I(X; u, \mathcal{A}) - I(X; \tilde{u}, \mathcal{A})| \leq 2 \sum_{a \in \mathcal{A}} \|u_a - \tilde{u}_a\|_\infty.$$

The proofs of this and subsequent results are deferred to Appendix F. Together, the preceding properties mean that NTU Subsidy Instability is useful as a regret measure for learning stable matchings.

As in the transferable utilities setting, Property 2 implies the existence of an explore-then-commit algorithm with $\widetilde{O}(|A|^{4/3}T^{2/3})$ regret. We show that this can be improved to a $\sqrt{T}$ dependence by adapting our approach from Section 5:

Theorem 6.5. *For matchings with non-transferable utilities, there exists an algorithm that for any utility function u incurs regret $R_T = O(|\mathcal{A}|^{3/2}\sqrt{T}\sqrt{\log(|\mathcal{A}|T)})$.*

While Theorem 6.5 illustrates that our approach easily generalizes to the NTU setting, we highlight two crucial differences between these settings. First, learning a stable matching is incomparable to learning a maximum weight matching because stable matchings do not maximize the sum of agents' utilities in the NTU setting. Next, the instability measure is *not* equivalent to the cumulative unhappiness of agents, unlike in the setting with transferable utilities. Intuitively, these definitions cease to be equivalent because non-transferable utilities render the problem more "disccontinuous" and thus obstruct the duality results we applied earlier.

These results provide a preliminary application of our framework to the setting of matching with non-transferable utilities; an interesting direction for future inquiry would be to more thoroughly investigate notions of approximate stability and regret in this setting.

6.3.1 Comparison to the utility difference measure

It turns out that the algorithm underlying Theorem 6.5 is equivalent to the centralized UCB algorithm from previous work [LMJ20; CS21], albeit derived from a different angle. However, an important difference is that Theorem 6.5 guarantees low regret relative to the incentive-aware NTU Subset Instability, as opposed to the incentive-unaware "utility difference" measure in prior work. In this section, we outline several properties that make our instability measure more suitable especially in the NTU setting. In particular, we show for utility difference that:

(a) There is no canonical formalization of utility difference when multiple stable matchings exist.

(b) The utility difference of a matching can be positive even if the matching is stable and negative even if the matching is unstable.

(c) Even when restricting to markets with unique stable matchings, the utility difference of a matching can be discontinuous in the true agent utilities. As a result, it does not allow for instance-independent regret bounds that are sublinear in T.

For (a), the utility difference requires specifying a stable matching to serve as a benchmark against which to measure relative utility. However, when multiple stable matchings exist, some ambiguity arises as to which one should be chosen as the benchmark. Because of this, previous works [DK05; LMJ20; CS21;

SBS21] study two different benchmarks. In particular, they assume providers' preferences are known and benchmark with respect to the customer-optimal and customer-pessimal stable matchings. For (b), notice that the utility difference for the maximum weight matching is negative, even though it is typically not stable in the NTU setting. Moreover, because of the ambiguity in the benchmark from (a), the utility difference may not be 0 even when the matching is stable. For (c), to see that utility difference is not continuous as a function of the underlying agent utilities, consider the following example:

Example 6.6. Consider a market where is a single customer i and two providers j_1 and j_2. Suppose their utility functions are given by $u_i(j_1) = \varepsilon$, $u_i(j_2) = 2\varepsilon$, $u_{j_1}(i) = 1$, and $u_{j_2}(i) = 0.5$. Then the unique stable matching $\{(i, j_2)\}$ has total utility $0.5 + 2\varepsilon$. Now, consider the perturbed utility function $\tilde{u}$ such that $\tilde{u}_i(j_1) = 2\varepsilon$, $\tilde{u}_i(j_2) = \varepsilon$, $\tilde{u}_{j_1}(i) = 1$, and $\tilde{u}_{j_2}(i) = 0.5$. For this perturbed utility function, the unique stable matching is $\{(i, j_1)\}$, which has total utility $1 + 2\varepsilon$. The utility difference (either optimal or pessimal) for matching $\{(i, j_2)\}$ is 0 for u and $0.5 + \varepsilon$ for $\tilde{u}$. Since this holds for any $\varepsilon > 0$, taking $\varepsilon \to 0$ shows that utility difference is not continuous in the utility function.

That utility difference is discontinuous in agent utilities rules out the existence of bandit algorithms that achieve sublinear instance-independent regret when using utility difference as the regret measure. In particular, the analyses in previous work [LMJ20; SBS21; CS21; LRM+20] focus entirely on *instance-dependent* regret bounds. They show centralized UCB achieves logarithmic instance-dependent regret with respect to the utility difference relative to the customer-pessimal stable matching (but does not achieve sublinear regret with respect to the customer-optimal stable matching). Our insight here is that a new measure of instability can present a more appealing evaluation metric and paint a clearer picture of an algorithm's convergence to the *set* of stable matchings as a whole.

6.4 In what settings are equilibria learnable?

A core insight of our work is that, in a stochastic environment, "optimism in the face of uncertainty" can be effectively leveraged for the problem of learning stable matchings. This motivates us to ask: in what other settings, and with what other algorithmic methods, can equilibria be learned?

One interesting open direction is to understand when equilibria can be learned in *adversarial* environments where the utility functions can change between rounds. From an economic perspective, adversarial environments could capture evolving market conditions. In adversarial bandits, instead of UCB-based algorithms, most works rely on gradient-based algorithms to attain optimal regret bounds (e.g., [ACF+02; AHR08]). Can these gradient-based algorithms similarly be adapted to Subset Instability?

Another interesting open direction is to consider more general market settings, even in stochastic environments. For example, within the context of matching markets, each agent might match to more than one agent on the other side of the market; and outside of matching markets, a buyer might purchase multiple units of multiple goods. In markets with transferable utilities, incentive-aligned outcomes can be captured by *Walrasian equilibria* (see, e.g., [BFS21]). Can Subset Instability and our UCB-based algorithms be adapted to learning Walrasian equilibria in general?

Addressing these questions would provide a richer understanding of when and how large-scale, data-driven marketplaces can efficiently learn market equilibria.

Acknowledgments

We would like to thank Itai Ashlagi, Jiantao Jiao, Scott Kominers (along with the Lab for Economic Design), Cassidy Laidlaw, Celestine Mendler-Dünner, and Banghua Zhu for valuable feedback. Meena Jagadeesan acknowledges support from the Paul and Daisy Soros Fellowship; Alexander Wei acknowledges support from an NSF Graduate Research Fellowship.

References

[ABK+20] Itai Ashlagi, Mark Braverman, Yash Kanoria, and Peng Shi. Clearing matching markets efficiently: informative signals and match recommendations. *Management Science*, 66(5):2163–2193, 2020.

[ACF+02] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002.

[ACF02] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, 2002.

[AH18] Eduardo M. Azevedo and John William Hatfield. Existence of equilibrium in large matching markets with complementarities. *Available at SSRN*, 2018.

[AHR08] Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: an efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 263–274. Omnipress, 2008.

[Als20] Max Alston. On the non-existence of stable matches with incomplete information. *Games and Economic Behavior*, 120:336–344, 2020.

[AMS+20] Guy Aridor, Yishay Mansour, Aleksandrs Slivkins, and Zhiwei Steven Wu. Competing bandits: the perils of exploration under competition. *CoRR*, abs/2007.10144, 2020.

[BFS21] Martin Bichler, Maximilian Fichtl, and Gregor Schwarz. Walrasian equilibria from an optimization perspective: a guide to the literature. *Naval Research Logistics (NRL)*, 68(4):496–513, 2021.

[Bik17] Sushil Bikhchandani. Stability with one-sided incomplete information. *Journal of Economic Theory*, 168:372–399, 2017.

[BKS18] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *J. ACM*, 65(3):13:1–13:55, 2018.

[BSS21] Soumya Basu, Karthik Abinav Sankararaman, and Abishek Sankararaman. Beyond $\log^2(T)$ regret for decentralized bandits in matching markets. *CoRR*, abs/2103.07501, 2021.

[CH87] Chandra R. Chegireddy and Horst W. Hamacher. Algorithms for finding k-best perfect matchings. *Discret. Appl. Math.*, 18(2):155–165, 1987.

[CL12] Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *J. Comput. Syst. Sci.*, 78(5):1404–1422, 2012.

[CS21] Sarah H. Cen and Devavrat Shah. Regret, stability, and fairness in matching markets with bandit learners. *CoRR*, abs/2102.06246, 2021.

[CTP+15] Richard Combes, Mohammad Sadegh Talebi, Alexandre Proutière, and Marc Lelarge. Combinatorial bandits revisited. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 2116–2124, 2015.

[CWY13] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: general framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 151–159. JMLR.org, 2013.

[DK05] Sanmay Das and Emir Kamenica. Two-sided bandits and the dating market. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 947–952. Professional Book Center, 2005.

[DS63] Gerard Debreu and Herbert Scarf. A limit theorem on the core of an economy. *International Economic Review*, 4(3):235–246, 1963.

[EGK20] Ehsan Emamjomeh-Zadeh, Yannai A. Gonczarowski, and David Kempe. The complexity of interactively learning a stable matching by trial and error. In Péter Biró, Jason D. Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC '20: The 21st ACM Conference on Economics and Computation*, page 599. ACM, 2020.

[ELS+13] Federico Echenique, Sangmok Lee, Matthew Shum, and M. Bumin Yenmez. The revealed preference theory of stable and extremal stable matchings. *Econometrica*, 81(1):153–171, 2013.

[FKK+14] Peter I. Frazier, David Kempe, Jon M. Kleinberg, and Robert Kleinberg. Incentivizing exploration. In Moshe Babaioff, Vincent Conitzer, and David A. Easley, editors, *ACM Conference on Economics and Computation, EC '14*, pages 5–22. ACM, 2014.

[GKJ12] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Trans. Netw.*, 20(5):1466–1478, 2012.

[GNO+19] Yannai A. Gonczarowski, Noam Nisan, Rafail Ostrovsky, and Will Rosenbaum. A stable marriage requires communication. *Games Econ. Behav.*, 118:626–647, 2019.

[GS62] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[ISS+19] Nicole Immorlica, Karthik Abinav Sankararaman, Robert E. Schapire, and Aleksandrs Slivkins. Adversarial bandits with knapsacks. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 202–219. IEEE Computer Society, 2019.

[JKK21] Ramesh Johari, Vijay Kamble, and Yash Kanoria. Matching while learning. *Oper. Res.*, 69(2):655–681, 2021.

[KL03] Robert D. Kleinberg and Frank Thomson Leighton. The value of knowing a demand curve: bounds on regret for online posted-price auctions. In *44th Symposium on Foundations of Computer Science (FOCS 2003)*, pages 594–605. IEEE Computer Society, 2003.

[Kuh55] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[KWA+15] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvári. Tight regret bounds for stochastic combinatorial semi-bandits. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015.

[Liu20] Qingmin Liu. Stability and bayesian consistency in two-sided markets. *American Economic Review*, 110(8), August 2020.

[LMJ20] Lydia T. Liu, Horia Mania, and Michael I. Jordan. Competing bandits in matching markets. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1618–1628. PMLR, 2020.

[LMP+] Qingmin Liu, George J. Mailath, Andrew Postlewaite, and Larry Samuelson. Stable matching with incomplete information. *Econometrica*, 82(2):541–587.

[LRM+20] Lydia T. Liu, Feng Ruan, Horia Mania, and Michael I. Jordan. Bandit learning in decentralized matching markets. *CoRR*, abs/2012.07348, 2020.

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[LSY21] Xiaocheng Li, Chunlin Sun, and Yinyu Ye. The symmetry between arms and knapsacks: A primal-dual approach for bandits with knapsacks. *CoRR*, abs/2102.06385, 2021.

[MSS15] Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. Bayesian incentive-compatible bandit exploration. In Tim Roughgarden, Michal Feldman, and Michael Schwarz, editors, *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15*, pages 565–582. ACM, 2015.

[Rot74] Michael Rothschild. A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9(2):185–202, 1974.

[RR13] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pages 2256–2264, 2013.

[SBS21] Abishek Sankararaman, Soumya Basu, and Karthik Abinav Sankararaman. Dominate or delete: decentralized competing bandits in serial dictatorship. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1252–1260. PMLR, 2021.

[Shi20] Peng Shi. Efficient matchmaking in assignment games with application to online platforms. In Péter Biró, Jason D. Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC '20: The 21st ACM Conference on Economics and Computation*, pages 601–602. ACM, 2020.

[SS66] L. S. Shapley and M. Shubik. Quasi-cores in a monetary economy with nonconvex preferences. *Econometrica*, 34(4):805–827, 1966.

[SS71] L. S. Shapley and M. Shubik. The assignment game I: The core. *International Journal of Game Theory*, 1(1):111–130, December 1971.

[TPR+20] Andrea Tirinzoni, Matteo Pirotta, Marcello Restelli, and Alessandro Lazaric. An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

A Classical Results for Matching with Transferable Utilities

To be self-contained, we briefly state and prove the key results from Shapley and Shubik [SS71] we need.

First, we explicitly relate the primal-dual formulation in Section 5.1 to stable matchings.

Theorem A.1 ([SS71]). *If (X, τ) is stable, then (Z, p) is an optimal primal-dual pair to (P) and (D), where $p_a = \tau_a + u_a(X(a))$ and Z is the indicator matrix in $\mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ corresponding to X.*

Moreover, if (Z, p) is an optimal primal-dual pair to (P) and (D) such that Z lies at an extreme point of the feasible set, then (X, τ) is stable where $\tau_a = p_a - u_a(X(a))$ and X is the matching corresponding to the nonzero entries of Z.

Proof. Both statements follow from the complementary slackness conditions and the definition of stability in Definition 2.1. The complementary slackness conditions are:

- If $Z_{i,j} > 0$, then $p_i + p_j = u_i(j) + u_j(i)$.
- If $p_i > 0$, then $\sum_j Z_{i,j} = 1$.
- If $p_j > 0$, then $\sum_i Z_{i,j} = 1$.

Suppose that (X, τ) is stable. Let us first show that (Z, p) is feasible. We see that Z is primal feasible by definition. For dual feasibility, since there are no blocking pairs, we know that

$$\big(u_i(\mu_X(i)) + \tau_i\big) + \big(u_j(\mu_X(j)) + \tau_j\big) \geq u_i(j) + u_j(i),$$

which implies

$$p_i + p_j \geq u_i(j) + u_j(i).$$

The individual rationality condition $u_a(\mu_X(a)) + \tau_a \geq 0$ tells us $p_a \geq 0$. Hence p is dual feasible. Next, we show that (Z, p) is an optimal primal-dual pair by checking the Karush–Kuhn–Tucker conditions. We have already shown primal and dual feasibility, so it suffices to show complementary slackness. The first condition follows from zero-sum transfers. To see the second and third conditions, we show the contrapositive: If $i \in \mathcal{I}$ is such that $\sum_j Z_{i,j} < 1$, then $\sum_j Z_{i,j} = 0$ by our assumption on Z. Hence i is unmatched (i.e., $u_i(\mu_X(i)) = 0$ and $\tau_i = 0$) which implies $p_i = 0$. The analogous argument applies for $j \in \mathcal{J}$.

We now prove the second part of the theorem. Suppose (Z, p) is an optimal solution to (P) and (D) such that Z is at a vertex. By the Birkhoff-von Neumann theorem, since Z is a vertex, it corresponds to a matching. We wish to show that (X, τ) has no blocking pairs, is individually rational, and has zero-sum transfers. Dual feasibility tells us that:

$$p_i + p_j \geq u_i(j) + u_j(i)$$

which means that:

$$\big(u_i(\mu_X(i)) + \tau_i\big) + \big(u_j(\mu_X(j)) + \tau_j\big) \geq u_i(j) + u_j(i),$$

so there are no blocking pairs. Dual feasibility also tells us that $p_a \geq 0$, which means that $u_a(\mu_X(a)) + \tau_a \geq 0$, so individual rationality is satisfied. To show that there are zero-sum transfers, we use complementary slackness. The first complementary slackness condition tells us that if $Z_{i,j} > 0$, then $p_i + p_j = u_i(j) + u_j(i)$. Using the fact that Z corresponds to a matching, this in particular means that if $(i, j) \in X$, we know $\tau_i + \tau_j = 0$. To show that agents who are unnmatched receive 0 transfers, let's use the second and third complementary slackness conditions. The contrapositive tells us that if a is unmatched, then $p_a = 0$, which implies $\tau_a = 0$. □

Since (P) is exactly the maximum weight matching linear program, Theorem A.1 immediately tells us that if (X, p) is stable, then X is a maximum weight matching. This means that stable matchings with transferable utilities maximize social welfare.

B Proofs for Section 4

This section contains further exposition (including proofs) for Section 4.

B.1 Limitations of utility difference as an instability measure

To illustrate why utility difference fails to be a good measure of instability, we describe a matching with transfers that (i) is far from stable and (ii) has 0 utility difference (but large Subset Instability).

Example B.1. Consider the following market with two agents: $\mathcal{I} = \{i\}$ and $\mathcal{J} = \{j\}$. Suppose that $u_i(j) = 2$ and $u_j(i) = -1$. Consider the matching $X = \{(i, j)\}$ with transfers $\tau_i = -\xi$ and $\tau_j = \xi$ for some $\xi > 0$. We will show that this matching with transfers will have the properties stated above when ξ is large.

This matching with transfers has utility difference 0 (for any ξ) since it maximizes the sum of utilities. Indeed, it is stable for any $\xi \in [1, 2]$. However, when $\xi > 2$, this matching with transfers is no longer stable, since the individual rationality condition $u_i(j) + \tau_i \geq 0$ fails. (Intuitively, the larger ξ is, the further we are from stability.) But its utility difference remains at 0.

On the other hand, the Subset Instability of this matching with transfers is $\xi - 2 > 0$ when $\xi > 2$. In particular, Subset Instability increases with ξ in this regime, which is consistent with the intuition that outcomes with larger ξ should be more unstable.

B.2 Proof of Proposition 4.2

Proposition 4.2. *For any market outcome, the minimum stabilizing subsidy equals the Subset Instability.*

Proof of Proposition 4.2. We can take the dual of the linear program (5) to obtain:

$$\max_{\substack{S \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{J}|} \\ Z \in \mathbb{R}^{|\mathcal{A}|}}} \quad \sum_{(i,j) \in \mathcal{I} \times \mathcal{J}} S_{i,j} \Big(\big(u_i(j) - u_i(\mu_X(i)) - \tau_i\big) + \big(u_j(i) - u_j(\mu_X(j)) - \tau_j\big) \Big) - \sum_{a \in \mathcal{A}} Z_a (u_a(\mu_X(a)) + \tau_a) \tag{‡}$$

$$\text{s.t. } Z_i + \sum_{j \in \mathcal{J}} S_{i,j} \leq 1 \qquad \forall i \in \mathcal{I}; \qquad Z_j + \sum_{i \in \mathcal{I}} S_{i,j} \leq 1 \qquad \forall j \in \mathcal{J};$$

$$S_{i,j} \geq 0 \qquad \forall (i,j) \in \mathcal{I} \times \mathcal{J}; \qquad Z_a \geq 0 \qquad \forall a \in \mathcal{A}. \qquad .$$

By strong duality, the optimal values of (5) and (‡) are equal. Thus, it suffices to show that Subset Instability is equal to (‡). By Proposition 4.3, we know that Subset Instability is equal to the maximum unhappiness of any coalition. Thus it suffices to show that (‡) is equal to the maximum unhappiness of any coalition.

To interpret (‡), observe that there exist optimal S^* and Z^* all of whose entries lie in $\{0, 1\}$ because this linear program can be embedded into a maximum weight matching linear program. Take such a choice of optimal S^* and Z^*. Then, S^* is an indicator vector corresponding to a (partial) matching on a subset of the agents such that all pairs in this matching are blocking with respect to (X, τ). Similarly, Z^* is an indicator vector of agents who would rather be unmatched than match according to (X, τ).

We first prove the claim that $I(X, \tau; u, \mathcal{A})$ is at least (‡). Based on the above discussion, the optimal objective of (‡) is obtained through S^* and Z^* that represent a matching and a subset of agents respectively. Let S be the union of agents participating in S^* and Z^*. We see that the objective of (‡) is equal to the utility difference at S, i.e.:

$$\left(\max_{X' \in \mathscr{X}_S} \sum_{a \in S} u_a(\mu_{X'}(a)) \right) - \left(\sum_{a \in S} u_a(\mu_X(a)) + \tau_a \right).$$

This is no larger than Subset Instability by definition.

We next prove the claim that $I(X, \tau; u, \mathcal{A})$ is at most (‡). Let's consider S^* that maximizes:

$$\max_{S \subseteq \mathcal{A}} \left(\max_{X' \in \mathscr{X}_S} \sum_{a \in S} u_a(\mu_{X'}(a)) \right) - \left(\sum_{a \in S} u_a(\mu_X(a)) + \tau_a \right).$$

Let's take the maximum weight matching of S^*. Let S be given by the matched agents in this matching and let Z be given by the unmatched agents in this matching (using the interpretation of (‡) described above). We see that the objective at (‡) for (S, Z) is equal to Subset Instability which proves the desired statement. □

B.3 Proof of Proposition 4.3

We first formally define the *unhappiness of a coalition*, as follows. In particular, the unhappiness with respect to (X, τ) of a coalition $\mathcal{S} \subseteq \mathcal{A}$ is defined to be:

$$\begin{aligned} \sup_{\substack{X' \in \mathscr{X}_{\mathcal{S}} \\ \tau' \in \mathbb{R}^{|\mathcal{S}|}}} & \sum_{a \in \mathcal{S}} \big(u_a(\mu_{X'}(a)) + \tau'_a \big) - \sum_{a \in \mathcal{S}} \big(u_a(\mu_X(a)) + \tau_a \big) \\ \text{s.t. } & u_a(\mu_{X'}(a)) + \tau'_a \geq u_a(\mu_X(a)) + \tau_a \qquad \forall a \in \mathcal{S} \\ & \tau'_a + \tau'_{\mu_{X'}(a)} = 0 \qquad \forall a \in \mathcal{S}, \end{aligned} \tag{7}$$

with unhappiness being 0 if there are no feasible X' and τ'. In the optimization program, (X', τ') represents a matching with transfers over $\mathcal{S}$, with the constraint $\tau'_a + \tau'_{\mu_{X'}(a)} = 0$ ensuring that it is zero-sum. The objective measures the difference between (X, τ) and (X', τ') of the total utility of the agents in $\mathcal{S}$. The constraint $u_a(\mu_{X'}(a)) + \tau'_a \geq u_a(\mu_X(a)) + \tau_a$ encodes the requirement that all agents be at least as well off under (X', τ') as they were under (X, τ). This optimization program therefore captures the objective of $\mathcal{S}$ to maximize their total payoff while ensuring that no member of the coalition is worse off than they were according to (X, τ).

Recall that, in terms of unhappiness, Proposition 4.3 is as follows:

Proposition 4.3. *The maximum unhappiness of any coalition $\mathcal{S} \subseteq \mathcal{A}$ with respect to (X, τ) equals the Subset Instability $I(X, \tau; u, \mathcal{A})$.*

Proof of Proposition 4.3. By Proposition 4.2, we know that Subset Instability is equal to (5). Moreover, by strong duality, we know that Subset Instability is equal to (‡) (the dual linear program of (5)). Thus, it suffices to prove that the maximum unhappiness of any coalition is equal to (‡).

We first prove the claim that (‡) is at most the maximum unhappiness of any coalition with respect to (X, τ). To do this, it suffices to construct a coalition $\mathcal{S} \subseteq \mathcal{A}$ such that (‡) is at most the unhappiness of $\mathcal{S}$. We construct $\mathcal{S}$ as follows: Recall that there exist optimal solutions S^* and Z^* to (‡) such that S^* corresponds to a (partial) matching on $\mathcal{I} \times \mathcal{J}$ and Z^* corresponds to a subset of $\mathcal{A}$. We may take $\mathcal{S}$ to be the union of the agents involved in S^* and in Z^*. Now, we upper bound the unhappiness of $\mathcal{S}$ by constructing X' and τ' that are feasible for (7). We can take X' to be the matching that corresponds to the indicator vector S^*. Because (S^*, Z^*) is optimal for (‡),

$$u_i(j) + u_j(i) \geq (u_i(\mu_X(i)) + \tau_i) + (u_j(\mu_X(j)) + \tau_j)$$

for all $(i, j) \in X'$. Thus, we can find a vector τ' of transfers that is feasible for (7). Then, since $\sum_{a \in \mathcal{S}} \tau'_a = 0$, the objective of (7) at (X', τ') is

$$\sum_{a \in \mathcal{S}} \big(u_a(\mu_{X'}(a)) - u_a(\mu_X(a)) - \tau_a \big).$$

This equals to the objective of (‡) at (S^*, Z^*), which equals (‡), as desired.

We now show the inequality in the other direction, that (‡) is at least the maximum unhappiness of any coalition with respect to (X, τ). It suffices to construct a feasible solution (S, Z) to (‡) that achieves at least the maximum unhappiness of any coalition. Let $\mathcal{S}$ be a coalition with maximum unhappiness, and let (X', τ') be an optimal solution for (7). Moreover, let S be the indicator vector corresponding to agents who are matched in X' and Z be the indicator vector corresponding to agents in $\mathcal{S}$ who are unmatched. The objective of (7) at (X', τ') is

$$\sum_{a \in \mathcal{S}} \big(u_a(\mu_{X'}(a)) - u_a(\mu_X(a)) - \tau_a\big),$$

which equals the objective of (‡) at the (S, Z) that we constructed.

□

B.4 Proof of Proposition 4.4

Proposition 4.4. *Subset Instability satisfies the following properties:*

1. *Subset Instability is always nonnegative and is zero if and only if (X, τ) is stable.*
2. *Subset Instability is Lipschitz continuous with respect to agent utilities. That is, for any possible market outcome (X, τ), and any pair of utility functions u and $\tilde{u}$ it holds that:*

$$|I(X, \tau; u, \mathcal{A}) - I(X, \tau; \tilde{u}, \mathcal{A})| \leq 2 \sum_{a \in \mathcal{A}} \|u_a - \tilde{u}_a\|_\infty.$$

3. *Subset Instability is always at least the utility difference.*

Proof of Proposition 4.4. We first prove the third part of the Proposition statement, then the first part of the Proposition statement, and finally the second part.

Proof of part (c). Because $\sum_{a \in \mathcal{A}} \tau_a = 0$, Subset Instability satisfies the following:

$$\begin{aligned} I(X, \tau; u, \mathcal{A}) &\geq \left(\max_{X' \in \mathscr{X}_\mathcal{A}} \sum_{a \in \mathcal{A}} u_a(\mu_{X'}(a)) \right) - \left(\sum_{a \in \mathcal{A}} u_a(\mu_X(a)) + \tau_a \right) \\ &= \left(\max_{X' \in \mathscr{X}_\mathcal{A}} \sum_{a \in \mathcal{A}} u_a(\mu_{X'}(a)) \right) - \left(\sum_{a \in \mathcal{A}} u_a(\mu_X(a)) \right). \end{aligned}$$

The second line is exactly the utility difference.

Proof of part (a). From above, we have that Subset Instability is lower bounded by the utility difference, which is always nonnegative. Hence Subset Instability is also always nonnegative.

To see that Subset Instability is 0 if and only if (X, τ) is stable, first suppose (X, τ) is unstable. Then, there exists a blocking pair (i, j), in which case

$$I(X, \tau; u, \mathcal{A}) \geq u_i(j) + u_j(i) - (u_i(\mu_X(i)) + u_j(\mu_X(j)) + \tau_i + \tau_j) > 0$$

by the definition of blocking. Now, suppose $I(X, \tau; u, \mathcal{A}) > 0$. Then, there exists a subset $S \subseteq \mathcal{A}$ such that

$$\left(\max_{X' \in \mathscr{X}_S} \sum_{a \in S} u_a(\mu_{X'}(a)) \right) - \left(\sum_{a \in S} u_a(\mu_X(a)) + \tau_a \right) > 0.$$

Let X' be a maximum weight matching on S. We can rewrite the above as

$$\sum_{(i,j)\in X'} \big(u_i(j) + u_j(i) - (u_i(\mu_X(i)) + u_j(\mu_X(j)) + \tau_i + \tau_j\big) > 0.$$

Some term in the sum on the left-hand side must be positive, so there exists a blocking pair $(i,j) \in X'$. In particular, (X,τ) is not stable.

Proof of part (b). We prove that

$$|I(X,\tau;u,\mathcal{A}) - I(X,\tau;\tilde{u},\mathcal{A})| \le 2\sum_{a\in\mathcal{A}} \|u_a - \tilde{u}_a\|_\infty.$$

The supremum of L-Lipschitz functions is L-Lipschitz, so it suffices to show that

$$\left(\max_{X'\in\mathscr{X}_S} \sum_{a\in S} u_a(\mu_{X'}(a))\right) - \sum_{a\in S}(u_a(\mu_X(a)) + \tau_a)$$

satisfies the desired Lipschitz condition for any $S \subseteq \mathcal{A}$. In particular, it suffices to show that

$$\left|\sum_{a\in S}(u_a(\mu_X(a)) + \tau_a) - \sum_{a\in S}(\tilde{u}_a(\mu_X(a)) + \tau_a)\right| \le \sum_{a\in\mathcal{A}} \|u_a - \tilde{u}_a\|_\infty \tag{8}$$

and

$$\left|\left(\max_{X'\in\mathscr{X}_S} \sum_{a\in S} u_a(\mu_{X'}(a))\right) - \left(\max_{X'\in\mathscr{X}_S} \sum_{a\in S} \tilde{u}_a(\mu_{X'}(a))\right)\right| \le \sum_{a\in\mathcal{A}} \|u_a - \tilde{u}_a\|_\infty. \tag{9}$$

For (8), we have

$$\left|\sum_{a\in S}(u_a(\mu_X(a)) + \tau_a) - \sum_{a\in S}(\tilde{u}_a(\mu_X(a)) + \tau_a)\right| = \left|\sum_{a\in S}\big(u_a(\mu_X(a)) - \tilde{u}_a(\mu_X(a))\big)\right| \le \sum_{a\in\mathcal{A}} \|u_a - \tilde{u}_a\|_\infty.$$

For (9), this boils down to showing that the total utility of the maximum weight matching is Lipschitz. Using again the fact that the supremum of Lipschitz functions is Lipschitz, this follows from the total utility of any fixed matching being Lipschitz. □

C Proofs for Section 5

C.1 Proof of Theorem 5.1

Theorem 5.1. *For preference class $\mathcal{U}_{\text{unstructured}}$ (see Section 3),* MatchUCB *(defined in Section 5.3) incurs expected regret $\mathbb{E}(R_T) = O\big(|\mathcal{A}|\sqrt{nT\log(|\mathcal{A}|T)}\big)$, where $n = \max_t |\mathcal{A}_t|$.*

Proof of Theorem 5.1. The starting point for our proof of Theorem 5.1 is the typical approach in multi-armed bandits and combinatorial bandits [GKJ12; CWY13; LS20] of bounding regret in terms of the sizes of the confidence interval of the chosen arms. However, rather than using the sizes of confidence intervals to bound the utility difference (as in the incentive-free maximum weight matching setting), we bound Subset Instability through Lemma 5.4. From here on, our approach composes cleanly with existing bandits analyses; in particular, we can follow the typical combinatorial bandits approach [GKJ12; CWY13] to get the desired upper bound.

For completeness, we present the full proof. We divide into two cases, based on the event E that all of the confidence sets contain their respective true utilities at every time step $t \le T$. That is, $u_i(j) \in C_{i,j}$ and $u_j(i) \in C_{j,i}$ for all $(i,j) \in \mathcal{I}\times\mathcal{J}$ at all t.

Case 1: E holds. By Lemma 5.4, we may bound

$$I(X^t, \tau^t; u, \mathcal{A}^t) \leq \sum_{a \in \mathcal{A}^t} \Big(\max\big(C_{a,\mu_{X^t}(a)}\big) - \min\big(C_{a,\mu_{X^t}(a)}\big) \Big) = O\left(\sum_{(i,j) \in X^t} \sqrt{\frac{\log(|\mathcal{A}|T)}{n_{ij}^t}} \right),$$

where n_{ij}^t is the number of times that the pair (i,j) has been matched at the start of round t. Let $w_{i,j}^t = \frac{1}{\sqrt{n_{ij}^t}}$ be the size of the confidence set (with the log factor scaled out) for (i,j) at the start of round t.

At each time step t, let's consider the list consisting of w_{i_t,j_t}^t for all $(i_t, j_t) \in X^t$. Let's now consider the overall list consisting of the concatenation of all of these lists over all rounds. Let's order this list in decreasing order to obtain a list $\tilde{w}_1, \ldots, \tilde{w}_L$ where $L = \sum_{t=1}^T |X^t| \leq nT$. In this notation, we observe that:

$$\sum_{t=1}^T I(X^t, \tau^t; u, \mathcal{A}) \leq \sum_{t=1}^T \sum_{a \in \mathcal{A}^t} \Big(\max\big(C_{a,\mu_{X^t}(a)}\big) - \min\big(C_{a,\mu_{X^t}(a)}\big) \Big) = \log(|\mathcal{A}|T) \sum_{l=1}^L \tilde{w}_l.$$

We claim that $\tilde{w}_l \leq O\left(\min(1, \frac{1}{\sqrt{(l/|\mathcal{A}|^2) - 1}}) \right)$. The number of rounds that a pair of agents can have their confidence set have size at least $\tilde{w}_l$ is upper bounded by $1 + \frac{1}{\tilde{w}_l^2}$. Thus, the total number of times that any confidence set can have size at least $\tilde{w}_l$ is upper bounded by $(|\mathcal{A}|^2)(1 + \frac{1}{\tilde{w}_l^2})$.

Putting this together, we see that:

$$\begin{aligned}
\log(|\mathcal{A}|T) \sum_{l=1}^L \tilde{w}_l &\leq O\left(\sum_{l=1}^L \min(1, \frac{1}{\sqrt{(l/|\mathcal{A}|^2) - 1}}) \right) \\
&\leq O\left(\log(|\mathcal{A}|T) \sum_{l=1}^{nT} \min(1, \frac{1}{\sqrt{(l/|\mathcal{A}|^2) - 1}}) \right) \\
&\leq O\Big(|\mathcal{A}| \sqrt{nT} \log(|\mathcal{A}|T) \Big).
\end{aligned}$$

Case 2: E does not hold. Since each $n_{ij}(\hat{u}_i(j) - u_i(j))$ is mean-zero and 1-subgaussian, and we have $O(|\mathcal{I}||\mathcal{J}|T)$ such random variables over T rounds, the probability that any of them exceeds

$$2\sqrt{\log(|\mathcal{I}||\mathcal{J}|T/\delta)} \leq 2\sqrt{\log(|\mathcal{A}|^2 T/\delta)}$$

is at most δ by a standard tail bound for the maximum of subgaussian random variables. It follows that E fails to hold with probability at most $|\mathcal{A}|^{-2}T^{-2}$. In the case that E fails to hold, our regret in any given round would be at most $4|\mathcal{A}|$ by the Lipschitz property in Proposition 4.4. (Recall that our upper confidence bound for any utility is wrong by at most 2 due to clipping each confidence interval to lie in $[-1, 1]$.) Thus, the expected regret from this scenario is at most

$$|\mathcal{A}|^{-2}T^{-2} \cdot 4|\mathcal{A}|T \leq 4|\mathcal{A}|^{-1}T^{-1},$$

which is negligible compared to the regret bound from when E does occur. □

C.2 Proof of Theorem 5.2

Theorem 5.2. *For preference class $\mathcal{U}_{\text{typed}}$ (see Section 3),* MatchTypedUCB *(defined in Section 5.3) incurs expected regret $\mathbb{E}(R_T) = O\big(|\mathcal{C}|\sqrt{nT\log(|\mathcal{A}|T)}\big)$, where $n = \max_t |\mathcal{A}_t|$.*

Proof of Theorem 5.2. Like in the proof of Theorem 5.1, we divide into two cases, based on the event E that all of the confidence sets contain their respective true utilities at every time step $t \le T$. That is, $u_a(a') \in C_{a,a'}$ for all pairs of agents at all t.

Case 1: E holds. By Lemma 5.4, we may bound

$$I(X^t, \tau^t; u, \mathcal{A}^t) \le \sum_{a\in\mathcal{A}^t} \Big(\max\big(C_{c_a, c_{\mu_{X^t(a)}}}\big) - \min\big(C_{c_a, c_{\mu_{X^t(a)}}}\big)\Big) = O\left(\sum_{(i,j)\in X^t} \sqrt{\frac{\log(|\mathcal{A}|T)}{n^t_{c_i c_j}}}\right),$$

where $n^t_{c_1 c_2}$ is the number of times that the an agent of type c_1 has been matched with an agent of context c_2 at the start of round t. (We define $n^0_{c_1,c_2} = 0$ by default.) Let $w^t_{c_1,c_2} = \frac{1}{\sqrt{n^t_{c_1,c_2}}}$ be the size of the confidence set (with the log factor scaled out) for (c_1, c_2) at the start of round t.

At each time step t, let's consider the list consisting of $w^t_{c_{i_t}, c_{j_t}}$ for all $(i_t, j_t) \in X^t$. Let's now consider the overall list consisting of the concatenation of all of these lists over all rounds. Let's order this list in decreasing order to obtain a list $\tilde{w}_1, \ldots, \tilde{w}_L$ where $L = \sum_{t=1}^T |X^t| \le nT$. In this notation, we observe that:

$$\sum_{t=1}^T I(X^t, \tau^t; u, \mathcal{A}^t) \le \sum_{t=1}^T \sum_{a\in\mathcal{A}^t} \Big(\max\big(C_{c_a, c_{\mu_{X^t(a)}}}\big) - \min\big(C_{c_a, c_{\mu_{X^t(a)}}}\big)\Big) = \log(|\mathcal{A}|T) \sum_{l=1}^L \tilde{w}_l.$$

We claim that $\tilde{w}_l \le O\left(\min(1, \frac{1}{\sqrt{(l/|\mathcal{C}|^2)-1}})\right)$. The number of instances that a pair of contexts can have their confidence set have size at least $\tilde{w}_l$ is upper bounded by $2n + \frac{1}{\tilde{w}_l^2}$. Thus, the total number of times that any confidence set can have size at least $\tilde{w}_l$ is upper bounded by $(|\mathcal{C}|)(2n + \frac{1}{\tilde{w}_l^2})$.

Putting this together, we see that:

$$\begin{aligned}
\log(|\mathcal{A}|T) \sum_{l=1}^L \tilde{w}_l &\le O\left(\sum_{l=1}^L \min(1, \frac{1}{\sqrt{(l/|\mathcal{A}|^2) - 1}})\right) \\
&\le O\left(\log(|\mathcal{A}|T) \sum_{l=1}^{nT} \min(1, \frac{1}{\sqrt{(l/|\mathcal{C}|^2) - 1}})\right) \\
&\le O\Big(|\mathcal{C}|\sqrt{nT}\log(|\mathcal{C}|^2 T)\Big).
\end{aligned}$$

Case 2: E does not hold. Since each $n_{ij}(\hat{u}_i(j) - u_i(j))$ is mean-zero and 1-subgaussian, and we have $O(|\mathcal{I}||\mathcal{J}|T)$ such random variables over T rounds, the probability that any of them exceeds

$$2\sqrt{\log(|\mathcal{I}||\mathcal{J}|T/\delta)} \le 2\sqrt{\log(|\mathcal{A}|^2 T/\delta)}$$

is at most δ by a standard tail bound for the maximum of subgaussian random variables. It follows that E fails to hold with probability at most $|\mathcal{A}|^{-2}T^{-2}$. In the case that E fails to hold, our regret in any given

round would be at most $4|\mathcal{A}|$ by the Lipschitz property in Proposition 4.4. (Recall that our upper confidence bound for any utility is wrong by at most 2 due to clipping each confidence interval to lie in $[-1, 1]$.) Thus, the expected regret from this scenario is at most

$$|\mathcal{A}|^{-2}T^{-2} \cdot 4|\mathcal{A}|T \leq 4|\mathcal{A}|^{-1}T^{-1},$$

which is negligible compared to the regret bound from when E does occur. □

C.3 Proof of Theorem 5.3

Theorem 5.3. *For preference class $\mathcal{U}_{\text{linear}}$ (see Section 3), MatchLinUCB (defined in Section 5.3) incurs expected regret $\mathbb{E}(R_T) = O\big(d\sqrt{|\mathcal{A}|}\sqrt{nT\log(|\mathcal{A}|T)}\big)$, where $n = \max_t |\mathcal{A}_t|$.*

To prove Theorem 5.3, it suffices to (a) show that the confidence sets contain the true utilities with high probability, and (b) bound the sum of the sizes of the confidence sets.

Part (a) follows from fact established in existing analysis of LinUCB in the classical linear contextual bandits setting [RR13].

Lemma C.1 ([RR13, Proposition 2])**.** *Let the confidence sets be defined as above (and in MatchLinUCB). For each $a \in \mathcal{A}$, it holds that:*

$$\mathbb{P}[\phi(a) \in C_{\phi(a)} \quad \forall 1 \leq t \leq T] \geq 1 - 1/(|\mathcal{A}|^3T^2).$$

Lemma C.2. *Let the confidence sets be defined as above (and in MatchLinUCB). For each $a \in \mathcal{A}$ and for any $\varepsilon > 0$, it holds that:*

$$\sum_{t|a\in\mathcal{A}^t,\mu_{X^t}(a)\neq a} \mathbf{1}\left[\max\big(C_{a,\mu_{X^t}(a)}\big) - \min\big(C_{a,\mu_{X^t}(a)}\big) > \varepsilon\right] \leq O\left(\left(\frac{4\beta_T}{\varepsilon^2} + 1\right) d\log(1/\varepsilon)\right).$$

Proof. We follow the same argument as the proof of Proposition 3 in [RR13].

We first recall the definition of ε-dependence and ε-eluder dimension: We say that an agent a' is *ε-dependent* on $a'_1, \ldots, a'_s$ if for all $\phi(a), \tilde{\phi}(a) \in \mathcal{B}^d$ such that

$$\sum_{k=1}^{s} \langle c_{a'_k}, \tilde{\phi}(a) - \phi(a)\rangle^2 \leq \varepsilon^2,$$

we also have $\langle c_{a'}, \tilde{\phi}(a) - \phi(a)\rangle^2 \leq \varepsilon^2$. The ε-eluder dimension $d_{\varepsilon\text{-eluder}}$ of $\mathcal{B}^d$ is the maximum length of a sequence $a'_1, \ldots, a'_s$ such that no element is ε-dependent on a prefix.

Consider the subset S_a of $\{t \mid a \in \mathcal{A}^t, \mu_{X^t}(a) \neq a\}$ such that

$$\mathbf{1}\left[\max\big(C_{a,\mu_{X^t}(a)}\big) - \min\big(C_{a,\mu_{X^t}(a)}\big) > \varepsilon\right].$$

Suppose for the sake of contradiction that

$$|S_a| > \left(\frac{4\beta_T}{\varepsilon^2} + 1\right) d_{\varepsilon\text{-eluder}}.$$

Then, there exists an element t^* that is ε-dependent on $\frac{4\beta_T}{\varepsilon^2} + 1$ disjoint subsets of S_a: One can repeatedly remove sequences $a'_{\mu_{X^{t_1}}(a)}, \ldots, a'_{\mu_{X^{t_s}}(a)}$ of maximal length such that no element is ε-dependent on a prefix;

note that $s \leq d_{\varepsilon\text{-eluder}}$ always. Let the subsets be $S_a^{(q)}$ for $q = 1, \ldots, \frac{4\beta_T}{\varepsilon^2} + 1$, and let $\phi(a), \tilde{\phi}(a)$ be such that $\langle c_{\mu_{X^{t^*}(a)}}, \tilde{\phi}(a) - \phi(a) \rangle > \varepsilon$. The above implies that

$$\sum_{q=1}^{\frac{4\beta_T}{\varepsilon^2}+1} \sum_{t \in S_a^{(q)}} \langle c\mu_{X^t}(a), \tilde{\phi}(a) - \phi(a) \rangle^2 > 4\beta_T$$

by the definition of ε-dependence. But this is impossible, since the left-hand side is upper bounded by

$$\sum_{t=1}^{T} \langle c\mu_{X^t}(a), \tilde{\phi}(a) - \phi(a) \rangle^2 \leq 4\beta_T$$

by the definition of the confidence sets. Hence it must hold that

$$|S_a| \leq \left(\frac{4\beta_T}{\varepsilon^2} + 1 \right) d_{\varepsilon\text{-eluder}}.$$

Now, it follows from the bound on the eluder dimension for linear bandits (Proposition 6 in [RR13]) that the bound of $\tilde{O}\left(\left(\frac{4\beta_T}{\varepsilon^2} + 1 \right) d \log(1/\varepsilon). \right)$ holds. □

Lemma C.3. *Let the confidence sets be defined as above (and in MatchLinUCB). For any $a \in \mathcal{A}$, it holds that:*

$$\sum_{t | a \in \mathcal{A}^t, \mu_{X^t}(a) \neq a} \left(\max\left(C_{a,\mu_{X^t}(a)}\right) - \min\left(C_{a,\mu_{X^t}(a)}\right) \right) \leq O(d(\log(T|\mathcal{A}|))\sqrt{T_a}),$$

where T_a is the number of times that agents is matched.

Proof. Let's consider the set of confidence set sizes $\left(\max\left(C_{a,\mu_{X^t}(a)}\right) - \min\left(C_{a,\mu_{X^t}(a)}\right) \right)$ for t such that $a \in \mathcal{A}^t, \mu_{X^t}$. Let's sort these confidence set sizes in decreasing order and label them $w_1 \geq \ldots \geq w_{T_a}$. Restating Lemma C.2, we see that

$$\sum_{t=1}^{T_a} w_t \mathbf{1}[w_t > \varepsilon] \leq O\left(\left(\frac{4\beta_T}{\varepsilon^2} + 1 \right) d \log(1/\varepsilon) \right). \tag{10}$$

for all $\varepsilon > 0$.

We see that:

$$\begin{aligned}
\sum_{t | a \in \mathcal{A}^t, \mu_{X^t}(a) \neq a} \left(\max\left(C_{a,\mu_{X^t}(a)}\right) - \min\left(C_{a,\mu_{X^t}(a)}\right) \right) &= \sum_{t=1}^{T_a} w_t \\
&\leq \sum_{t=1}^{T_a} w_t \mathbf{1}[w_t > 1/T_a^2] + \sum_{t=1}^{T_a} w_t \mathbf{1}[w_t \leq 1/T_a^2] \\
&\leq \frac{1}{T_a} + \sum_{t=1}^{T_a} w_t \mathbf{1}[w_t > 1/T_a^2].
\end{aligned}$$

We claim that $w_i \leq 2$ if $i \geq d \log(T_a)$ and $w_i \leq \min(2, \frac{4\beta_T(d \log T_a)}{i - d \log T_a})$ if $i > d \log T_a$. The first part follows from the fact that we truncate the confidence sets to be within $[-1, 1]$. It thus suffices to show that $w_i \leq \frac{4\beta_T(d \log T_a)}{i - d \log T_a}$ for $t \leq d \log T$. If $w_i \geq \varepsilon > 1/T_a^2$, then we see that $\sum_{t=1}^{T_a} \mathbf{1}[w_t > \varepsilon] \geq i$, which means by

(10) that $i \leq O\left(\left(\frac{4\beta_T}{\varepsilon^2}+1\right) d\log(1/\varepsilon)\right) \leq O\left(\left(\frac{4\beta_T}{\varepsilon^2}+1\right) d\log(T_a)\right)$ which means that $\varepsilon \leq \frac{4\beta_T(d\log T_a)}{i - d\log T_a}$. This proves the desired statement.

Now, we can plug this into the above expression to obtain:

$$\begin{aligned}
&\sum_{t|a\in\mathcal{A}^t,\mu_{X^t}(a)\neq a}\Big(\max\big(C_{a,\mu_{X^t}(a)}\big) - \min\big(C_{a,\mu_{X^t}(a)}\big)\Big)\\
&\leq \frac{1}{T_a} + \sum_{t=1}^{T_a} w_t \mathbf{1}[w_t > 1/T_a^2]\\
&\leq \frac{1}{T_a} + 2d\log(T_a) + \sum_{i>d\log T_a}^{T_a} \min\left(2, \frac{4\beta_T(d\log T_a)}{i - d\log T_a}\right)\\
&\leq \frac{1}{T_a} + 2d\log(T_a) + 2\sqrt{d\log T_a \beta_T}\int_{t=0}^{T_a} t^{-1/2}dt\\
&= \frac{1}{T_a} + 2d\log(T_a) + 4\sqrt{dT_a\log T_a\beta_T}
\end{aligned}$$

We now use that:

$$\beta_T = O(d\log T + \frac{1}{T}\sqrt{\log(T^2|A|)}).$$

Plugging this into the above expression, we obtain the desired result. □

With these facts, we are ready to prove Theorem 5.3.

Proof of Theorem 5.3. Like in the proof of Theorem 5.1, we divide into two cases, based on the event E that all of the confidence sets contain their respective true utilities at every time step $t \leq T$. That is, $u_{c_1}(c_2) \in C_{c_1,c_2}$ for all $c_1, c_2 \in \mathcal{C}$ at all t.

Case 1: E holds. By Lemma 5.4, we know that the cumulative regret is upper bounded by

$$\begin{aligned}
R_T &\leq \sum_{t=1}^{T}\sum_{a\in\mathcal{A}^t}\Big(\max\big(C_{a,\mu_{X^t}(a)}\big) - \min\big(C_{a,\mu_{X^t}(a)}\big)\Big)\\
&= \sum_{a\in\mathcal{A}}\sum_{t|a\in\mathcal{A}^t,\mu_{X^t}(a)\neq a}\Big(\max\big(C_{a,\mu_{X^t}(a)}\big) - \min\big(C_{a,\mu_{X^t}(a)}\big)\Big)\\
&\leq \sum_{a\in\mathcal{A}} O(d\log(T|\mathcal{A}|)\sqrt{T_a}),
\end{aligned}$$

where the last inequality applies Lemma C.3 to the inner summand. We see that $\sum_{a\in\mathcal{A}} T_a = \sum_t |\mathcal{A}_t| \leq nT$ by definition, since at most n agents show up at every round. Let's now observe that:

$$\sum_{a\in\mathcal{A}}\sqrt{T_a} \leq \sqrt{|\mathcal{A}|}\sqrt{\sum_{a\in\mathcal{A}} T_a} \leq \sqrt{|\mathcal{A}|nT},$$

as desired.

Case 2: E does not hold. From Lemma C.1, it follows that:

$$\mathbb{P}[\phi(a) \in C_{\phi(a)} \quad \forall 1 \leq t \leq T] \geq 1 - 1/(|\mathcal{A}|^3 T^2).$$

Union bounding, we see that

$$\mathbb{P}[\phi(a) \in C_{\phi(a)} \quad \forall 1 \leq t \leq T \forall a \in \mathcal{A}] \geq 1 - 1/(|\mathcal{A}|^2 T^2).$$

By the definition of the confidence sets for the utilities, we see that:

$$\mathbb{P}[u(a, a') \in C_{a,a'} \quad \forall 1 \leq t \leq T, \forall a, a' \in \mathcal{A}] \geq 1/(|\mathcal{A}|^2 T^2). \tag{11}$$

Thus, the probability that event E does not hold is at most $|\mathcal{A}|^{-2} T^{-2}$. In the case that E fails to hold, our regret in any given round would be at most $4|\mathcal{A}|$ by the Lipschitz property in Proposition 4.4. Thus, the expected regret is at most $4|\mathcal{A}|^{-1} T^{-1}$ which is negligible compared to the regret bound from when E does occur.

□

C.4 Proof of Lemma 5.5

Lemma 5.5. *For any algorithm that learns a stable matching with respect to unstructured preferences, there exists an instance on which it has expected regret $\widetilde{\Omega}(|A|^{3/2}\sqrt{T})$ (where regret is given by Subset Instability).*

Proof of Lemma 5.5. Recall that, by Proposition 4.4, the problem of learning a maximum weight matching with respect to utility difference is no harder than that of learning a stable matching with respect to Subset Instability. In the remainder of our proof, we reduce a standard "hard instance" for stochastic multi-armed bandits to our setting of learning a maximum weight matching.

Step 1: Constructing the hard instance for stochastic MAB. Consider the following family of stochastic multi-armed bandits instances: for a fixed K, let $\mathcal{I}_\alpha$ for $\alpha \in \{1, \ldots, K\}$ denote the stochastic multi-armed bandits problem where all arms have 0-1 rewards, and the k-th arm has mean reward $\frac{1}{2} + \rho$ if $k = \alpha$ and $\frac{1}{2}$ otherwise, where $\rho > 0$ will be set later. A classical lower bound for stochastic multi-armed bandits is the following:

Lemma C.4 ([ACF+02])**.** *The expected regret of any stochastic multi-armed bandit algorithm on an instance $\mathcal{I}_\alpha$ for α selected uniformly at random from $\{1, \ldots, K\}$ is $\Omega(\sqrt{KT})$.*

Step 2: Constructing a (random) instance for the maximum weight matching problem. We will reduce solving the above distribution over stochastic multi-armed bandits problems to a distribution over instances of learning a maximum weight matching. Let us now construct this random instance of the maximum weight matching problem. Let $|\mathcal{I}| = K$ and $|\mathcal{J}| = 10K \log(KT)$. Specifically, we sample inputs for learning a maximum weight matching as follows: For each man $i \in \mathcal{I}$, select $\alpha_i \in \{1, \ldots, K\}$ uniformly at random, and define $u_i(j)$ to be $\frac{1}{2} + \rho$ if $\lfloor (j-1)/\log K \rfloor = \alpha_i$ and $\frac{1}{2}$ otherwise. Furthermore, let $u_j(i) = 0$ for all $(i, j) \in \mathcal{I} \times \mathcal{J}$. Finally, suppose observations are always in $\{0, 1\}$ (but are unbiased).

The key property of the above setup that we will exploit for our reduction is the fact that, due to the imbalance in the market, the maximum weight matching for these utilities has with high probability each i matched with some j whom they value at $\frac{1}{2} + \rho$. Indeed, by a union bound, the probability that more than $10 \log(KT)$ different i have the same α_i is at most

$$K \cdot \binom{K}{10 \log(KT)} K^{-10 \log(KT)} = O\big(K^{-4} T^{-4}\big).$$

Thus, with probability $1-O(K^{-4}T^{-4})$, this event holds. The case where this event does not hold contributes negligibly to regret, so we do not consider it further.

Step 3: Establishing the reduction. Now, suppose for the sake of contradiction that some algorithm could solve our random instance of learning a maximum weight matching problem with expected regret $o(K^{3/2}\sqrt{T})$. We can obtain a stochastic multi-armed bandits that solves the instances in Lemma C.4 as follows: Choose a random $i^* \in \mathcal{I}$ and set $\alpha_{i^*} = \alpha$. Simulate the remaining i by choosing α_i for all $i \neq i^*$ uniformly at random. Run the algorithm on this instance of learning a maximum weight matching, "forwarding" arm pulls to the true instance when matching i^*.

To analyze the regret of this algorithm when faced with the distribution from Lemma C.4, we first note that with high probability, all the agents $i \in \mathcal{I}$ can simultaneously be matched to a set of $j \in \mathcal{J}$ such that each i is matched to some j whom they value at $\frac{1}{2}+\rho$. Then, the regret of any matching is ρ times the number of $i \in \mathcal{I}$ who are not matched to a j whom they value at $\frac{1}{2}+\rho$. Thus, we can define the cumulative regret for an agent $i \in \mathcal{I}$ as ρ times the number of rounds they were not matched to someone whom they value at $\frac{1}{2}+\rho$. For i^*, this regret is just the regret for the distribution from Lemma C.4. Since i^* was chosen uniformly at random, their expected cumulative regret is at most

$$\frac{1}{K} \cdot o(K^{3/2}\sqrt{T}) = o(\sqrt{KT}),$$

in violation of Lemma C.4.

Step 4: Concluding the lower bound. This contradiction implies that no algorithm can hope to obtain $o(K^{3/2}\sqrt{T})$ expected regret on this distribution over instances of learning a maximum weight matching. Since there are $O(K\log(KT)) = \widetilde{O}(K)$ agents in the market total, the desired lower bound follows. □

D Proof of Theorem 6.1

Theorem 6.1 (Instance-Dependent Regret). *Suppose that $\mathcal{A}_t = \mathcal{A}$ for all t. Let $u \in \mathcal{U}_{\text{unstructured}}$ be any utility function, and put*

$$\Delta = \inf_{X \neq X^*} \left\{ \sum_{a \in \mathcal{A}} u_a(\mu_{X^*}(a)) - \sum_{a \in \mathcal{A}} u_a(\mu_X(a)) \right\}.$$

Then MatchUCB′ *incurs expected regret* $\mathbb{E}(R_T) = O(|\mathcal{A}|^5 \cdot \log(|\mathcal{A}|T)/\Delta^2)$.

D.1 MatchUCB′

MatchUCB′ is the same as MatchUCB, except we call ComputeMatch′ instead of ComputeMatch. The idea behind ComputeMatch′ is that we compute an optimal primal-dual solution for both the original confidence sets C as well as expanded confidence sets C', which we define to be twice the width of the original confidence sets. More formally, we define

$$C'_{a,a'} := \left[\min(C_{a,a'}) - \frac{\max(C_{a,a'}) - \min(C_{a,a'})}{2}, \max(C_{a,a'}) + \frac{\max(C_{a,a'}) - \min(C_{a,a'})}{2} \right].$$

We will adaptively explore (following UCB) according to both C and C'. Doing extra exploration according to the more pessimistic confidence sets C' is necessary for us to be able to find "robust" dual solutions for setting transfers.

We define (X^*, p^*), which will be an optimal primal-dual solution for the upper confidence bounds of C as follows. Let X^* be a maximum weight matching with respect to u^{UCB}. We next compute the gap

$$\Delta^{\mathrm{UCB}} = \min_{X \neq X^*} \left\{ \sum_{a \in \mathcal{A}} u_a^{\mathrm{UCB}}(\mu_{X^*}(a)) - \sum_{a \in \mathcal{A}} u_a^{\mathrm{UCB}}(\mu_X(a)) \right\}$$

with respect to u^{UCB}. We can compute this gap by computing the maximum weight matching and the second-best matching with respect to u^{UCB}.[15] Next, define utility functions u'_a such that

$$u'_a(a') = \begin{cases} u_a^{\mathrm{UCB}}(a') - \frac{\Delta^{\mathrm{UCB}}}{|\mathcal{A}|} & \text{if } \mu_{X^*}(a) = a' \text{ and } a \neq a' \\ u_a^{\mathrm{UCB}}(a') & \text{otherwise} \end{cases}$$

for all $a \in \mathcal{A}$. (We show in Lemma D.1 that X^* is still a maximum weight matching for u'.) Now, compute an optimal dual solution p' for utility function u'. To get p^*, we add $\Delta^{\mathrm{UCB}}/|\mathcal{A}|$ to p'_a for each matched agent a in X^*. (See Lemma D.3 for a proof that (X^*, p^*) is an optimal primal-dual pair with respect to u^{UCB}.)

Finally, let $(X^{*,2}, p^{*,2})$ be any optimal primal-dual pair for the utility function $u^{\mathrm{UCB},2}$ given by the upper confidence bounds $\max(C'_{a,a'})$ of C'.

With this setup, we define ComputeMatch$'$ as follows: If $X^* \neq X^{*,2}$, return $(X^{*,2}, \tau^{*,2})$, where $\tau^{*,2}$ is given by $\tau_a^{*,2} = p_a^{*,2} - u_a^{\mathrm{UCB},2}(\mu_{X^{*,2}}(a))$ if a is matched and $\tau_a^{*,2} = 0$ if a is unmatched. Otherwise, return (X^*, τ^*), where τ^* is given by $\tau_a^* = p_a^* - u_a^{\mathrm{UCB}}(\mu_X(a))$ if a is matched and $\tau_a^* = 0$ if a is unmatched.

D.2 Proof of Theorem 6.1

We first verify (as claimed above) that X^* is a maximum weight matching with respect to u'.

Lemma D.1. *Matching X^* is a maximum weight matching with respect to u'.*

Proof. Consider any matching $X \neq X^*$. Since $\sum_{a \in \mathcal{A}} u_a^{\mathrm{UCB}}(\mu_X(a)) \leq -\Delta^{\mathrm{UCB}} + \sum_{a \in \mathcal{A}} u_a^{\mathrm{UCB}}(\mu_{X^*}(a))$ by the definition of Δ^{UCB}, we have

$$\sum_{a \in \mathcal{A}} u'_a(\mu_X(a)) \leq \sum_{a \in \mathcal{A}} u_a^{\mathrm{UCB}}(\mu_X(a)) \leq \sum_{a \in \mathcal{A}} \left(u_a^{\mathrm{UCB}}(\mu_{X^*}(a)) - \frac{\Delta^{\mathrm{UCB}}}{|\mathcal{A}|} \right) \leq \sum_{a \in \mathcal{A}} u'_a(\mu_{X^*}(a)). \qquad \square$$

We now prove the main lemma for this analysis: if the confidence sets are small enough, then the selected matching will be stable with respect to the true utilities.

Lemma D.2. *Suppose* ComputeMatch$'$ *is run on confidence sets C satisfying*

$$\max(C_{i,j}) - \min(C_{i,j}) \leq 0.05 \frac{\Delta}{|\mathcal{A}|} \quad \textit{and} \quad \max(C_{j,i}) - \min(C_{j,i}) \leq 0.05 \frac{\Delta}{|\mathcal{A}|}$$

for all (i, j) in the matching returned by ComputeMatch$'$*. Suppose also that the confidence sets C contain the true utilities for all pairs of agents. Then the market outcome returned by* ComputeMatch$'$ *is stable with respect to the true utilities u.*

Remark. Lemma D.2 does *not* hold for ComputeMatch. We rely on the particular specification of the optimal primal-dual pair and tweaks discussed above.

[15] See Chegireddy and Hamacher [CH87] for efficient algorithms for to compute the second-best matching.

Proof of Lemma D.2. The proof proceeds in five steps, which we now outline. We first show the matching returned by ComputeMatch′ is the maximum weight matching X^{opt} with respect to u. We next show that X^* as defined in ComputeMatch′ also equals X^{opt}. These facts let us conclude that ComputeMatch′ returns (X^*, τ^*). We then show Δ^{UCB} is at least 0.1Δ. We then show that (X^*, τ^*) is stable with respect to u'. We finish by showing that this implies (X^*, τ^*) is a stable with respect to u.

Throughout the proof, we will use the following observation about the expanded confidence sets:

$$\max\big(C'_{i,j}\big) - \min\big(C'_{i,j}\big) \le 0.1\frac{\Delta}{|\mathcal{A}|} \quad \text{and} \quad \max\big(C'_{j,i}\big) - \min\big(C'_{j,i}\big) \le 0.1\frac{\Delta}{|\mathcal{A}|} \tag{12}$$

for all (i, j) in the matching returned by ComputeMatching′. This follows from the assumptions in the lemma statement.

Proving ComputeMatch′ returns X^{opt} as the matching. ComputeMatch′ by definition returns $X^{*,2}$ always, so it suffices to show that $X^{*,2} = X^{\text{opt}}$. Note that $X^{*,2}$ is a maximum weight matching with respect to $u^{\text{UCB},2}$. This means that

$$\begin{aligned}
\sum_{a\in\mathcal{A}} u_a(\mu_{X^{*,2}}(a)) &\ge -\sum_{a\in\mathcal{A}}\big(\max\big(C'_{a,\mu_{X^{*,2}}(a)}\big) - \min\big(C'_{a,\mu_{X^{*,2}}(a)}\big)\big) + \sum_{a\in\mathcal{A}} u_a^{\text{UCB},2}(\mu_{X^{*,2}}(a)) \\
&\ge -0.1\Delta + \sum_{a\in\mathcal{A}} u_a^{\text{UCB},2}(\mu_{X^{*,2}}(a)) \\
&\ge -0.1\Delta + \sum_{a\in\mathcal{A}} u_a^{\text{UCB},2}(\mu_{X^{\text{opt}}}(a)) \\
&\ge -0.1\Delta + \sum_{a\in\mathcal{A}} u_a(\mu_{X^{\text{opt}}}(a)).
\end{aligned}$$

By the definition of the gap Δ, we conclude that $X^{*,2} = X^{\text{opt}}$.

Proving $X^* = X^{\text{opt}}$. Suppose for sake of contradiction that $X^* \ne X^{\text{opt}}$. Then

$$\sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_{X^*}(a)) \ge \sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_{X^{\text{opt}}}(a)) \ge \sum_{a\in\mathcal{A}} u_a(\mu_{X^{\text{opt}}}(a)),$$

since X^* is a maximum weight matching with respect to u^{UCB}. Moreover, by the definition of the gap, we know that $\sum_{a\in\mathcal{A}} u_a(\mu_{X^*}(a)) \le \sum_{a\in\mathcal{A}} u_a(\mu_{X^{\text{opt}}}(a)) - \Delta$. Putting this all together, we see that

$$\begin{aligned}
\sum_{a\in\mathcal{A}}\big(\max\big(C_{a,\mu_{X^*}(a)}\big) - \min\big(C_{a,\mu_{X^*}(a)}\big)\big) &\ge \sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_{X^*}(a)) - \sum_{a\in\mathcal{A}} u_a(\mu_{X^*}(a)) \\
&\ge \Delta.
\end{aligned}$$

We now use this to lower bound the utility of X^* on $u^{\text{UCB},2}$. By the definition of the confidence sets, we see that

$$\begin{aligned}
\sum_{a\in\mathcal{A}} u_a^{\text{UCB},2}(\mu_{X^*}(a)) &\ge \sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_{X^*}(a)) + \frac{1}{2}\sum_{a\in\mathcal{A}}\big(\max\big(C_{a,\mu_{X^*}(a)}\big) - \min\big(C_{a,\mu_{X^*}(a)}\big)\big) \\
&\ge \sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_{X^*}(a)) + 0.5\Delta.
\end{aligned}$$

However, X^{opt} only achieves a utility of

$$\begin{aligned}\sum_{a\in\mathcal{A}} u_a^{\text{UCB},2}(\mu_{X^{\text{opt}}}(a)) &\le \sum_{a\in\mathcal{A}} u_a(\mu_{X^{\text{opt}}}(a)) + \sum_{a\in\mathcal{A}} \Big(\max\big(C'_{a,\mu_{X^{\text{opt}}}(a)}\big) - \min\big(C'_{a,\mu_{X^{\text{opt}}}(a)}\big)\Big) \\ &\le \sum_{a\in\mathcal{A}} u_a(\mu_{X^{\text{opt}}}(a)) + 0.1\Delta.\end{aligned}$$

But this contradicts the fact (from above) that $X^{\text{opt}} = X^{*,2}$ is a maximum weight matching with respect to $u^{\text{UCB},2}$. Therefore, it must be that $X^* = X^{\text{opt}}$.

Putting the above two arguments together, we conclude ComputeMatch′ returns (X^*, τ^*) in this case.

Bounding the gap Δ^{UCB}. We next show that $\Delta^{\text{UCB}} \ge 0.1\Delta$. We proceed by assuming

$$\sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_X(a)) \ge -0.1\Delta + \sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_{X^*}(a)) \tag{13}$$

for some $X \ne X^*$ and deriving a contradiction.

We first show that (13) implies a lower bound on

$$S = \sum_{a\in\mathcal{A}} \Big(\max\big(C_{a,\mu_X(a)}\big) - \min\big(C_{a,\mu_X(a)}\big)\Big)$$

in terms of Δ. Because the confidence sets contain the true utilities and u_a^{UCB} upper bounds u_a pointwise, (13) implies

$$S + \sum_{a\in\mathcal{A}} u_a(\mu_X(a)) \ge \sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_X(a)) \ge -0.1\Delta + \sum_{a\in\mathcal{A}} u_a(\mu_{X^*}(a)).$$

Applying the definition of Δ, we obtain the lower bound

$$S \ge -0.1\Delta + \sum_{a\in\mathcal{A}} u_a(\mu_{X^*}(a)) - \sum_{a\in\mathcal{A}} u_a(\mu_X(a)) \ge (1-0.1)\Delta.$$

Now, we apply the fact that $X^* = X^{*,2} = X^{\text{opt}}$. We establish the following contradiction:

$$\begin{aligned}0.1\Delta + \sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_{X^*}(a)) &\ge 0.1\Delta + \sum_{a\in\mathcal{A}} u_a(\mu_{X^*}(a)) \\ &= \sum_{a\in\mathcal{A}} (u_a(\mu_{X^*}(a)) + 0.1\Delta/|\mathcal{A}|) \\ &\overset{\text{(i)}}{\ge} \sum_{a\in\mathcal{A}} u_a^{\text{UCB},2}(\mu_{X^*}(a)) \\ &\overset{\text{(ii)}}{\ge} \sum_{a\in\mathcal{A}} u_a^{\text{UCB},2}(\mu_X(a)) \\ &\overset{\text{(iii)}}{\ge} \frac{S}{2} + \sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_X(a)) \\ &\overset{\text{(iv)}}{\ge} \left(\frac{1}{2}(1-0.1)\right)\Delta + \sum_{a\in\mathcal{A}} u_a^{\text{UCB}}(\mu_X(a))\end{aligned}$$

$$\overset{\text{(v)}}{\geq} \left(\frac{1}{2}(1-0.1)-0.1\right)\Delta + \sum_{a\in\mathcal{A}} u_a^{\mathrm{UCB}}(\mu_{X^*}(a)).$$

Here, (i) comes from (12) in the lemma statement; (ii) holds because $X^* = X^{*,2}$ is a maximum weight matching with respect to $u^{\mathrm{UCB},2}$; (iii) is by the definition of $u^{\mathrm{UCB},2}$; (iv) follows from our lower bound on S; and (v) follows from (13).

Proving that (X^*, τ^*) is stable with respect to u'. By Lemma D.1, (X^*, p') is an optimal primal-dual pair with respect to u'. Now, it suffices to show that the primal-dual solution corresponds to the market outcome (X^*, τ^*) for u'. To see this, notice that $p'_a = 0$ for unmatched agents and

$$p'_a = p^*_a - \frac{\Delta^{\mathrm{UCB}}}{2|\mathcal{A}|} = \tau^*_a + u'_a(\mu_{X^*}(a))$$

for matched agents.

Proving that (X^*, τ^*) is stable with respect to u. We show the stability (X^*, τ^*) with respect to u by checking that individual rationality holds and that there are no blocking pairs.

The main fact that we will use is that

$$u_a(\mu_{X^*}(a)) \geq u'_a(\mu_{X^*}(a)).$$

To prove this, we split into two cases: (i) agent a is matched in X^* (i.e., $\mu_{X^*}(a) \neq a$), and (ii) agent a is not matched by X^*. For (i), if a is matched by X^*, then

$$u_a(\mu_{X^*}(a)) \geq u_a^{\mathrm{UCB}}(\mu_{X^*}(a)) - 0.1\frac{\Delta}{|\mathcal{A}|} \geq u_a^{\mathrm{UCB}}(\mu_{X^*}(a)) - \frac{\Delta^{\mathrm{UCB}}}{|\mathcal{A}|} = u'_a(\mu_{X^*}(a)).$$

For (ii), if a is not matched by X^*, then $u_a(\mu_{X^*}(a)) \geq u'_a(\mu_{X^*}(a))$ because both sides are 0.

For individual rationality, we thus have

$$u_a(\mu_{X^*}(a)) + \tau^*_a \geq u'_a(\mu_{X^*}(a)) + \tau^*_a \geq 0,$$

where the second inequality comes from the individual rationality of (X^*, τ^*) with respect to u'.

Let's next show that there are no blocking pairs. If $(i,j) \in X^*$, then we see that:

$$u_i(\mu_{X^*}(i)) + \tau^*_i + u_j(\mu_{X^*}(j)) + \tau^*_j = u_i(\mu_{X^*}(i)) + u_j(\mu_{X^*}(j)),$$

as desired. Next, consider any pair $(i,j) \notin X^*$. Then,

$$u_i(j) + u_j(i) \leq u_i^{\mathrm{UCB}}(j) + u_j^{\mathrm{UCB}}(i) = u'_i(j) + u'_j(i).$$

It follows that

$$\begin{aligned} u_i(\mu_{X^*}(i)) + \tau^*_i + u_j(\mu_{X^*}(j)) + \tau^*_j &\geq u'_i(\mu_{X^*}(i)) + \tau^*_i + u_j(\mu_{X^*}(j)) + \tau^*_j \\ &\geq u'_i(j) + u'_j(i) \\ &\geq u_i(j) + u_j(i), \end{aligned}$$

where the second inequality comes from the fact that (X^*, τ^*) has no blocking pairs with respect to u'.

This completes our proof that (X^*, τ^*) is stable with respect to u. □

Now, we are ready to prove Theorem 6.1.

Proof of Theorem 6.1. As in the proof of Theorem 5.1, the starting point for our proof is the typical approach in multi-armed bandits and combinatorial bandits [GKJ12; CWY13; LS20] of bounding regret in terms of the sizes of the confidence interval of the chosen arms. Our approach does not quite compose cleanly with these proofs, since we need to handle the transfers in addition to the matching.

We divide in two cases, based on the event E that all of the confidence sets contain their respective true utilities at every time step $t \leq T$. That is, $u_i(j) \in C_{i,j}$ and $u_j(i) \in C_{j,i}$ for all $(i,j) \in \mathcal{I} \times \mathcal{J}$ at all t.

Case 1: E holds. Let n_{ij}^t be the number of times that the pair (i,j) has been matched by round t. For each pair (i,j), we maintain a "blame" counter b_{ij}^t that will always be at most n_{ij}^t.

First, suppose that

$$\max\left(C_{a,\mu_{X^t}(a)}\right) - \min\left(C_{a,\mu_{X^t}(a)}\right) \leq 0.1\frac{\Delta}{|\mathcal{A}|}$$

for every matched agent $a \in \mathcal{A}$. By Lemma D.2 we know the chosen matching is stable and thus incurs 0 regret.

Now, suppose that

$$\max\left(C_{a,\mu_{X^t}(a)}\right) - \min\left(C_{a,\mu_{X^t}(a)}\right) > 0.1\frac{\Delta}{|\mathcal{A}|}$$

for some matched agent a. We increment the counter of the least-blamed pair $(i,j) \in X^t$. By the definition of the confidence sets and the assumption that E holds, we know that each blame counter is bounded as

$$b_{ij}^T = O\left(\frac{|\mathcal{A}|^2 \log(|\mathcal{A}|T))}{\Delta^2}\right).$$

The maximum regret incurred by any matching is at most $12|\mathcal{A}|$ which means that the regret incurred by this case is at most:

$$12|\mathcal{A}| \sum_{(i,j)} O\left(\frac{|\mathcal{A}|^2 \log(|\mathcal{A}|T))}{\Delta^2}\right) = O\left(\frac{|\mathcal{A}|^5 \log(|\mathcal{A}|T))}{\Delta^2}\right).$$

Case 2: E does not hold. Since each $n_{ij}(\hat{u}_i(j) - u_i(j))$ is mean-zero and 1-subgaussian and we have $O(|\mathcal{I}||\mathcal{J}|T)$ such random variables over T rounds, the probability that any of them exceeds

$$2\sqrt{\log(|\mathcal{I}||\mathcal{J}|T/\delta)} \leq 2\sqrt{\log(|\mathcal{A}|^2 T/\delta)}$$

is at most δ by a standard tail bound for the maximum of subgaussian random variables. It follows that E fails to hold with probability at most $|\mathcal{A}|^{-2}T^{-2}$. In the case that E fails to hold, our regret in any given round would be at most $12|\mathcal{A}|$ by the Lipschitz property in Proposition 4.4. (Recall that our upper confidence bound is off by at most 6 due to clipping the confidence interval to lie in $[-1,1]$, so that the expanded confidence sets also necessarily lie in $[-3,3]$.) Thus, the expected regret from this scenario is at most

$$|\mathcal{A}|^{-2}T^{-2} \cdot 12|\mathcal{A}|T \leq 12|\mathcal{A}|^{-1}T^{-1},$$

which is negligible compared to the regret bound from when E does occur. □

D.3 Instance-independent regret bounds for MatchUCB′

To establish *instance-independent* regret bounds for MatchUCB′, we show that (X^*, p^*) is indeed optimal with respect to u^{UCB}; the remainder then follows the same argument as Theorem 5.1.

Lemma D.3. *The pair (X^*, p^*) is an optimal primal-dual pair with respect to u^{UCB}.*

Proof. It suffices to verify feasibility and, by weak duality, check that X^* and p^* achieve the same objective value. It is clear that X^* is primal feasible. For dual feasibility, if $(i,j) \notin X^*$, then

$$p_i^* + p_j^* \geq p_i' + p_j' \geq u_i'(j) + u_j'(i) = u_i^{\mathrm{UCB}}(j) + u_j^{\mathrm{UCB}}(i);$$

and if $(i,j) \in X^*$, then

$$p_i^* + p_j^* = p_i' + p_j' + 2\frac{\Delta^{\mathrm{UCB}}}{|\mathcal{A}|} \geq u_i'(j) + u_j'(i) + 2\frac{\Delta^{\mathrm{UCB}}}{|\mathcal{A}|} = u_i^{\mathrm{UCB}}(j) + u_j^{\mathrm{UCB}}(i).$$

Finally, we check that they achieve the same objective value with respect to u^{UCB}. By Lemma D.1 and strong duality, X^* achieves the same objective value as p' with respect to u'. Hence

$$\sum_{a\in\mathcal{A}} u_a^{\mathrm{UCB}}(\mu_{X^*}(a)) = 2|X^*|\frac{\Delta^{\mathrm{UCB}}}{|\mathcal{A}|} + \sum_{a\in\mathcal{A}} u_a'(\mu_{X^*}(a)) = 2|X^*|\frac{\Delta^{\mathrm{UCB}}}{|\mathcal{A}|} + \sum_{a\in\mathcal{A}} p_a' = \sum_{a\in\mathcal{A}} p_a^*. \qquad \square$$

E Proofs for Section 6.2

Theorem 6.2. *For preference class $\mathcal{U}_{\mathrm{unstructured}}$ (see Section 3), there exists an algorithm giving the platform*

$$\varepsilon\Big(\sum_{t=1}^{T}|\mathcal{A}_t|\Big)T - O\big(|\mathcal{A}|\sqrt{nT}\sqrt{\log(|\mathcal{A}||T|)}\big)$$

revenue in the presence of search frictions while maintaining stability with high probability.

Proof of Theorem 6.2. The algorithm is defined as follows. We set confidence sets according to MatchUCB and run essentially that algorithm, but with a modified ComputeMatch. Instead of ComputeMatch, we use the following algorithm. The platform first computes a matching with transfers (X^*, τ^*) according to the UCB estimates u^{UCB}, like before. Then, the platform chooses X^* to be the selected matching, and sets the transfers according to:

$$\tau_a = \tau_a^* - \varepsilon + \max\big(C_{a,\mu_X(a))}\big) - \min\big(C_{a,\mu_X(a)}\big).$$

This choice of transfers has a clean economic intuition: agents should be compensated based on the platform's uncertainty about their utilities with ε of their transfer shaved off as revenue for the platform.

First, we show that if the confidence sets contain the true utilities, then (X^*, τ) is ε-stable. It suffices to show that (X^*, τ') where:

$$\tau_a' = \tau_a^* + \max\big(C_{a,\mu_X(a))}\big) - \min\big(C_{a,\mu_X(a)}\big)$$

is stable. First, we see that

$$u_a(\mu_{X^{\mathrm{UCB}}}(a)) + \tau_a' = u_a^{\mathrm{UCB}}(\mu_{X^{\mathrm{UCB}}}(a)) + \tau_a^* \geq 0,$$

since (X, τ^*) is stable with respect to u^{UCB}. Furthermore, we see that:

$$\begin{aligned}\big(u_i(\mu_X(i)) + \tau_i'\big) + \big(u_j(\mu_X(j)) + \tau_j'\big) &\geq \big(u_i^{\text{UCB}}(\mu_X(i)) + \tau_i^*\big) + \big(u_j^{\text{UCB}}(\mu_X(j)) + \tau_j^*\big) \\ &\geq u_i^{\text{UCB}}(j) + u_j^{\text{UCB}}(i) \\ &\geq u_i(j) + u_j(i),\end{aligned}$$

where the second to last line follows from the fact that (X, τ^*) is stable with respect to u^{UCB}.

We first show that s is a feasible solution to (†).

$$\begin{aligned}&\min\big(u_i(j) - u_i(\mu_{X^{\text{UCB}}}(i)) - s_i, \tilde{u}_j(i) - u_j(\mu_{X^{\text{UCB}}}(j)) - s_j\big) \\ &= \min\big(u_i(j) - u_i^{\text{UCB}}(\mu_{X^{\text{UCB}}}(i)), \tilde{u}_j(i) - u_j^{\text{UCB}}(\mu_{X^{\text{UCB}}}(j))\big) \\ &\leq \min\big(u_i^{\text{UCB}}(j) - u_i^{\text{UCB}}(\mu_{X^{\text{UCB}}}(i)), u_j^{\text{UCB}}(i) - u_j^{\text{UCB}}(\mu_{X^{\text{UCB}}}(j))\big) \\ &\leq 0,\end{aligned}$$

where the last step uses the fact that $\mu_{X^{\text{UCB}}}$ is stable with respect to u^{UCB} by definition. Moreover, we see that

$$u_a(\mu_{X^{\text{UCB}}}(a)) + s_a = u_a^{\text{UCB}}(\mu_{X^{\text{UCB}}}(a)) \geq 0,$$

where the last inequality uses that $\mu_{X^{\text{UCB}}}$ is stable with respect to u^{UCB} by definition. This implies that s is feasible.

We see that the platform's revenue is equal to:

$$\begin{aligned}-\sum_{t=1}^{T}\sum_{a\in\mathcal{A}_t}\tau_a &= -\sum_{t=1}^{T}\sum_{a\in\mathcal{A}_t}\tau_a^* + \sum_{t=1}^{T}\sum_{a\in\mathcal{A}_t}\varepsilon + \sum_{t=1}^{T}\sum_{a\in\mathcal{A}_t}\big(\max\big(C_{a,\mu_X(a))}\big) - \min\big(C_{a,\mu_X(a)}\big)\big) \\ &= \varepsilon\sum_{t=1}^{T}|\mathcal{A}_t| - \sum_{t=1}^{T}\sum_{a\in\mathcal{A}_t}\big(\max\big(C_{a,\mu_X(a))}\big) - \min\big(C_{a,\mu_X(a)}\big)\big)\,.\end{aligned}$$

Using the proof of Theorem 5.1, we see that

$$\sum_{t=1}^{T}\sum_{a\in\mathcal{A}_t}\big(\max\big(C_{a,\mu_X(a))}\big) - \min\big(C_{a,\mu_X(a)}\big)\big) \leq O(|\mathcal{A}|\sqrt{nT}\log(|\mathcal{A}|T)),$$

as desired.

□

F Proofs for Section 6.3

F.1 Proof of Proposition 6.4

Proof of Proposition 6.4. We first prove the first part of the statement, and then the second part of the statement.

Proof of part (a). We note that it follows immediately from Definition 6.3 that NTU Subset Instability is nonnegative. Let's now show that $I(X; u, \mathcal{A})$ is 0 if and only if (X, τ) is stable. It is not difficult to see that the infimum of (†) is attained at some s^*.

If $I(X; u, \mathcal{A}) = 0$, then we know that $s_a^* = 0$ for all $a \in \mathcal{A}$. The constraints in the optimization problem imply that X has no blocking pairs and individually rationality is satisfied, as desired.

If X is stable, then we see that $s = \vec{0}$ is a feasible solution to (†), which means that the optimum of (†) is at most 0. This coupled with the fact that $I(X; u, \mathcal{A})$ is always nonnegative means that $I(X; u, \mathcal{A}) = 0$ as desired.

Proof of part (b). Consider two utility functions u and $\tilde{u}$. To show Lipchitz continuity, it suffices to show that for any matching X:

$$|I(X; u, \mathcal{A}) - I(X; \tilde{u}, \mathcal{A})| \leq 2 \sum_{a \in \mathcal{A}} \|u_a - \tilde{u}_a\|_\infty.$$

We show that:

$$I(X; \tilde{u}, \mathcal{A}) \leq I(X; u, \mathcal{A}) + 2 \sum_{a \in \mathcal{A}} \|u_a - \tilde{u}_a\|_\infty,$$

noting that the other direction follows from an analogous argument. Let s^* be an optimal solution to (†) for the utilities u. Consider the solution $s_a = s^*_a + 2\|u_a - u_a\|_\infty$. We first verify that s is a feasible solution to (†) for $\tilde{u}$. We see that:

$$\begin{aligned}
&\min\big(\tilde{u}_i(j) - \tilde{u}_i(\mu_X(i)) - s_i, \tilde{u}_j(i) - \tilde{u}_j(\mu_X(j)) - s_j\big) \\
&= \min\big(\tilde{u}_i(j) - \tilde{u}_i(\mu_X(i)) - s^*_i - 2\|u_i - \tilde{u}_i\|_\infty, \tilde{u}_j(i) - \tilde{u}_j(\mu_X(j)) - s^*_j - 2\|u_j - \tilde{u}_j\|_\infty\big) \\
&\leq \min\big(u_i(j) - u_i(\mu_X(i)) - s^*_i, u_j(i) - u_j(\mu_X(j)) - s^*_j\big) \\
&\leq 0,
\end{aligned}$$

as desired. Moreover, we see that

$$\tilde{u}_a(\mu_X(a)) + s_a = \tilde{u}_a(\mu_X(a)) + s^*_a + 2\|u_a - u_a\|_\infty \leq u_a(\mu_X(a)) + s^*_a \geq 0.$$

Thus we have demonstrated that s is feasible. This means that:

$$I(X; \tilde{u}, \mathcal{A}) \leq \sum_{a \in \mathcal{A}} s_a = \sum_{a \in \mathcal{A}} s^*_a + 2 \sum_{a \in \mathcal{A}} \|u_a - \tilde{u}_a\|_\infty = [I(X; u, \mathcal{A}) + 2 \sum_{a \in \mathcal{A}} \|u_a - \tilde{u}_a\|_\infty,$$

as desired.

□

F.2 Proof of Theorem 6.5

We show that the algorithmic approach from Section 5 can be adapted to the setting of matching with non-transferable utilities.

Drawing intuition from Section 5, at each round, we compute a stable matching for utilities given by the upper confidence bounds. More precisely, suppose we have a collection $\mathscr{C}$ of confidence sets $C_{i,j}, C_{j,i} \subseteq \mathbb{R}$ such that $u_i(j) \in C_{i,j}$ and $u_j(i) \in C_{j,i}$ for all $(i, j) \in \mathcal{I} \times \mathcal{J}$. Our algorithm uses $\mathscr{C}$ to get an upper confidence bound for each agent's utility function and then computes a stable matching with transfers as if these upper confidence bounds were the true utilities (see ComputeMatchNTU). This can be implemented efficiently if we use, e.g., the Gale-Shapley algorithm (either the customer-proposing algorithm or the provider-proposing algorithm will work).

The core property of ComputeMatchNTU is that we can upper bound NTU Subset Instability by the sum of the sizes of the relevant confidence sets, assuming that the confidence sets contain the true utilities.

Algorithm 5 ComputeMatchNTU: Compute matching with transfers from confidence sets

procedure ComputeMatchNTU($\mathscr{C}$)
  for $(i,j) \in \mathcal{I} \times \mathcal{J}$ **do**
    $u_i^{\mathrm{UCB}}(j) \leftarrow \max(C_{i,j})$; $u_j^{\mathrm{UCB}}(i) \leftarrow \max(C_{j,i})$ ▷ UCB estimates of utilities.
  Run any version of the Gale-Shapley algorithm [GS62] on u^{UCB} to obtain a matching X^*.
  return X^*

Proposition F.1. *Consider a collection confidence sets $\mathscr{C}$ such that $u_i(j) \in C_{i,j}$ and $u_j(i) \in C_{j,i}$ for all $(i,j) \in \mathcal{I} \times \mathcal{J}$. The instability of the output X^{UCB} of ComputeMatch satisfies*

$$I(X^{\mathrm{UCB}}; u, \mathcal{A}) \leq \sum_{a \in \mathcal{A}^t} \Big(\max\big(C_{a,\mu_{X^{\mathrm{UCB}}}(a)}\big) - \min\big(C_{a,\mu_{X^{\mathrm{UCB}}}(a)}\big) \Big). \tag{14}$$

Proof. We construct subsidies for this setting to be:

$$s_a = \max\big(C_{a,\mu_X(a)}\big) - u_a(\mu_X(a)) \leq \max\big(C_{a,\mu_X(a)}\big) - \min\big(C_{a,\mu_X(a)}\big).$$

Step 1: Verifying feasibility. We first show that s is a feasible solution to (†).

$$\begin{aligned}
&\min\big(u_i(j) - u_i(\mu_{X^{\mathrm{UCB}}}(i)) - s_i, \tilde{u}_j(i) - u_j(\mu_{X^{\mathrm{UCB}}}(j)) - s_j\big) \\
&= \min\big(u_i(j) - u_i^{\mathrm{UCB}}(\mu_{X^{\mathrm{UCB}}}(i)), \tilde{u}_j(i) - u_j^{\mathrm{UCB}}(\mu_{X^{\mathrm{UCB}}}(j))\big) \\
&\leq \min\big(u_i^{\mathrm{UCB}}(j) - u_i^{\mathrm{UCB}}(\mu_{X^{\mathrm{UCB}}}(i)), u_j^{\mathrm{UCB}}(i) - u_j^{\mathrm{UCB}}(\mu_{X^{\mathrm{UCB}}}(j))\big) \\
&\leq 0,
\end{aligned}$$

where the last step uses the fact that $\mu_{X^{\mathrm{UCB}}}$ is stable with respect to u^{UCB} by definition. Moreover, we see that

$$u_a(\mu_{X^{\mathrm{UCB}}}(a)) + s_a = u_a^{\mathrm{UCB}}(\mu_{X^{\mathrm{UCB}}}(a)) \geq 0,$$

where the last inequality uses that $\mu_{X^{\mathrm{UCB}}}$ is stable with respect to u^{UCB} by definition. This implies that s is feasible.

Step 2: Computing the objective. We next compute the objective of (†) at s and use this to bound $I(X^*; u, \mathcal{A})$. A simple calculation shows that:

$$I(X^*; u, \mathcal{A}) \leq \sum_a s_a = \sum_{a \in \mathcal{A}} \Big(\max\big(C_{a,\mu_{X^{\mathrm{UCB}}}(a)}\big) - \min\big(C_{a,\mu_{X^{\mathrm{UCB}}}(a)}\big) \Big),$$

as desired.

□

F.2.1 Explicit algorithm and regret bounds

Using the same intuition as Section 5, the regret bound of Proposition F.1 hints at an algorithm: each round, select the matching with transfers returned by ComputeMatchNTU and update confidence sets accordingly. To instantiate this approach, it remains to construct confidence intervals that contain the true utilities with high probability.

We showcase this algorithm in the simple setting of unstructured preferences. For this setting, we can construct our confidence intervals following the classical UCB approach. That is, for each utility value

Algorithm 6 MatchNTUUCB: A bandit algorithm for matching with non-transferable utilities.

procedure MatchNTUUCB(T)
  for $(i,j) \in \mathcal{I} \times \mathcal{J}$ **do** ▷ Initialize confidence intervals and empirical mean.
    $C_{i,j} \leftarrow [-1,1]$; $C_{j,i} \leftarrow [-1,1]$; $\hat{u}_i(j) \leftarrow 0$; $\hat{u}_j(i) \leftarrow 0$
  for $1 \le t \le T$ **do**
    $X^t \leftarrow$ ComputeMatchNTU(C)
    for $(i,j) \in X^t$ **do** ▷ Set confidence intervals and update means.
      Update $\hat{u}_i(j)$ and $\hat{u}_j(i)$ from feedback; increment counter n_{ij}
      $C_{i,j} \leftarrow \left[\hat{u}_i(j) - 8\sqrt{\log(|\mathcal{A}|T)/n_{ij}}, \hat{u}_i(j) + 8\sqrt{\log(|\mathcal{A}|T)/n_{i,j}}\right] \cap [-1,1]$
      $C_{j,i} \leftarrow \left[\hat{u}_j(i) - 8\sqrt{\log(|\mathcal{A}|T)/n_{ij}}, \hat{u}_j(i) + 8\sqrt{\log(|\mathcal{A}|T)/n_{i,j}}\right] \cap [-1,1]$

involving the pair (i,j), we take a length $O(\sqrt{\log(|\mathcal{A}|T)/n_{ij}})$ confidence interval centered around the empirical mean, where n_{ij} is the number of times the pair has been matched before. We describe this construction precisely in Algorithm 2 (MatchNTUUCB).

To analyze MatchNTUUCB, recall that Lemma 5.4 bounds the regret at each step by the lengths of the confidence intervals of each pair in the selected matching. Like in Section 5, this yields the following instance-independent regret bound:

Theorem F.2. *MatchNTUUCB incurs expected regret* $\mathbb{E}(R_T) \le O\big(|\mathcal{A}|^{3/2}\sqrt{T}\sqrt{\log(|\mathcal{A}|T)}\big)$.

Proof. This proof proceeds very similarly to the proof of Theorem 5.1. We consider the event E that all of the confidence sets contain their respective true utilities at every time step $t \le T$. That is, $u_i(j) \in C_{i,j}$ and $u_j(i) \in C_{j,i}$ for all $(i,j) \in \mathcal{I} \times \mathcal{J}$ at all t.

Case 1: E holds. By Lemma 5.4, we may bound

$$I(X^t; u, \mathcal{A}^t) \le \sum_{a \in \mathcal{A}^t} \Big(\max\big(C_{a,\mu_{X^t}(a)}\big) - \min\big(C_{a,\mu_{X^t}(a)}\big)\Big) = O\left(\sum_{(i,j)\in X^t} \sqrt{\frac{\log(|\mathcal{A}|T)}{n_{ij}^t}}\right),$$

where n_{ij}^t is the number of times that the pair (i,j) has been matched at the start of round t. Let $w_{i,j}^t = \frac{1}{\sqrt{n_{ij}^t}}$ be the size of the confidence set (with the log factor scaled out) for (i,j) at the start of round t.

At each time step t, let's consider the list consisting of w_{i_t,j_t}^t for all $(i_t,j_t) \in X^t$. Let's now consider the overall list consisting of the concatenation of all of these lists over all rounds. Let's order this list in decreasing order to obtain a list $\tilde{w}_1, \ldots, \tilde{w}_L$ where $L = \sum_{t=1}^T |X^t| \le nT$. In this notation, we observe that:

$$\sum_{t=1}^T I(X^t; u, \mathcal{A}^t) \le \sum_{t=1}^T \sum_{a \in \mathcal{A}^t} \Big(\max\big(C_{a,\mu_{X^t}(a)}\big) - \min\big(C_{a,\mu_{X^t}(a)}\big)\Big) = \log(|\mathcal{A}|T) \sum_{l=1}^L \tilde{w}_l.$$

We claim that $\tilde{w}_l \le O\left(\min(1, \frac{1}{\sqrt{(l/|\mathcal{A}|^2)-1}})\right)$. The number of rounds that a pair of agents can have their confidence set have size at least $\tilde{w}_l$ is upper bounded by $1 + \frac{1}{\tilde{w}_l^2}$. Thus, the total number of times that any confidence set can have size at least $\tilde{w}_l$ is upper bounded by $(|\mathcal{A}|^2)(1 + \frac{1}{\tilde{w}_l^2})$.

Putting this together, we see that:

$$\begin{aligned}\log(|\mathcal{A}|T)\sum_{l=1}^{L}\tilde{w}_l &\leq O\left(\sum_{l=1}^{L}\min(1,\frac{1}{\sqrt{(l/|\mathcal{A}|^2)-1}})\right)\\ &\leq O\left(\log(|\mathcal{A}|T)\sum_{l=1}^{nT}\min(1,\frac{1}{\sqrt{(l/|\mathcal{A}|^2)-1}})\right)\\ &\leq O\Big(|\mathcal{A}|\sqrt{nT}\log(|\mathcal{A}|T)\Big).\end{aligned}$$

Case 2: E does not hold. Since each $n_{ij}(\hat{u}_i(j)-u_i(j))$ is mean-zero and 1-subgaussian, and we have $O(|\mathcal{I}||\mathcal{J}|T)$ such random variables over T rounds, the probability that any of them exceeds

$$2\sqrt{\log(|\mathcal{I}||\mathcal{J}|T/\delta)} \leq 2\sqrt{\log(|\mathcal{A}|^2T/\delta)}$$

is at most δ by a standard tail bound for the maximum of subgaussian random variables. It follows that E fails to hold with probability at most $|\mathcal{A}|^{-2}T^{-2}$. In the case that E fails to hold, our regret in any given round would be at most $4|\mathcal{A}|$ by the Lipschitz property in Proposition 6.4. (Recall that our upper confidence bound for any utility is wrong by at most 2 due to clipping each confidence interval to lie in $[-1,1]$.) Thus, the expected regret from this scenario is at most

$$|\mathcal{A}|^{-2}T^{-2}\cdot 4|\mathcal{A}|T \leq 4|\mathcal{A}|^{-1}T^{-1},$$

which is negligible compared to the regret bound from when E does occur. □