Competing Bandits in Decentralized Contextual Matching Markets

Learning Environments and Stable Matchings with Logarithmic Regret

Paper by: Satush Parikh, Soumya Basu, Avishek Ghosh, Abishek Sankararaman

Presented by: Team 12 CS6007
Sarthak Mishra & Abhimanyu Singh Rathore

arXiv:2411.11794v2

November 7, 2025

Table of Contents

Matching Markets: Classical Context

- **Definition**: Two-sided markets with N agents (workers) and K arms (tasks/items).
- **Classical Model** (Gale–Shapley 1962): Agents and arms have fixed, known preference rankings.
- **Goal**: Achieve *stable matching* where no agent–arm pair prefers each other over current matches.
- **Real-world Applications**:
  - School admissions, organ transplants, job matching
  - Amazon Mechanical Turk, TaskRabbit, UpWork, Jobble
- **Challenge in Modern Platforms**: Agents do not know preferences a priori; they must learn through interaction.

Modern Learning Markets

One-Sided Learning:

- Agents learn preferences
- Arms have fixed preferences
- Typical in gig economy

Challenges Addressed:

- Decentralized decisions
- No central coordinator
- Collision resolution
- Time-varying preferences

Motivating Example: Amazon Mechanical Turk

- Task rewards vary with product releases or demand
- Workers need to detect temporal variability
- Should adapt preferences to maximise reward

Technical Foundations: Linear Contextual Bandits

Definition (Linear Contextual Bandit Model)

For each agent i:

- Latent parameter $\theta_i \in \mathbb{R}^d$ (agent-specific)
- Feature vector $\mathbf{x}_{i,j}(t) \in \mathbb{R}^d$ (agent–arm pair at time t)
- Expected reward: $\mu_{i,j}(t) = \langle \mathbf{x}_{i,j}(t), \theta_i \rangle$
- Observed reward: $r_i(t) = \mu_{i,j}(t) + \eta_i(t)$, $\eta_i(t)$ sub-Gaussian (e.g. $\mathcal{N}(0,1)$)

Advantages

- Regret independent of the number of arms K (crucial for large markets)
- Only need to learn the low-dimensional θ_i
- Enables feature-based preference modelling

Non-Stationarity via Latent Environments

The Challenge

If feature vectors change arbitrarily, agents need N^2 rounds to relearn stable matchings (via Gale–Shapley), leading to linear regret.

Solution (Latent Environment Structure)

Introduce a finite set $\mathcal{E}$ of size E representing latent environments:

- *Each environment characterises preference rankings.*
- *Feature vectors change within and across environments.*
- *Agents infer the active environment from observed features and learned θ_i.*
- *Small feature perturbations within an environment $\Rightarrow$ consistent rankings.*

Numerical Example: World Setup

- **Agents (N=2):** Alice, Bob
- **Arms (K=3):** Task A, Task B, Task C
- **Agent Parameters ($d = 2$):**
  - Alice $\theta_A = [10, -2]$ (likes relevance, dislikes difficulty)
  - Bob $\theta_B = [5, 5]$ (generalist)
- **Environments (E=2):**

Env 1: "Weekday"

Arm	μ_{Alice}	μ_{Bob}
A	**7.8**	**4.5**
B	7.7	4.45
C	4.6	3.5

Alice Rank: A > B > C

Stable Match: {(A, A), (B, B)}

Env 2: "Weekend"

Arm	μ_{Alice}	μ_{Bob}
A	0.8	1.0
B	6.4	**8.0**
C	**6.8**	4.0

Alice Rank: C > B > A

Stable Match: {(A, C), (B, B)}

Two-Sided Matching Market Setup

Definition (Contextual Matching Market)

- N agents, K arms ($K \geq N$), horizon T.
- At each round t, an *active environment* $e(t) \in \mathcal{E}$ (latent to agents).
- Each agent i proposes one arm $m_i(t)$.
- Each arm has a fixed preference ranking over agents.
- On a collision the arm selects its most-preferred agent.
- Unmatched agents receive zero reward.
- Matched agent i with arm j receives $r_i(t) = \langle \mathbf{x}_{i,j}(t), \theta_i \rangle + \eta_i(t)$.

Decentralised Setting

Agents make independent decisions without a central coordinator. Communication is only via a shared information board (standard in recent literature).

Stable Matching Concept

Definition (Stable Matching)

A matching is *stable* if no agent–arm pair mutually prefers each other over their current matches (no blocking pairs).

Theorem (Gale–Shapley, 1962)

The Gale–Shapley algorithm with deferred acceptance produces:

1. *A stable matching for any market.*
2. *The unique agent-optimal stable matching.*
3. *Convergence in at most* $N^2 - 2N + 2$ *rounds.*

Key Challenge: Agents do not know true preferences until learning θ_i; preferences change across environments.

Environment Definition and Assumption 1

Definition (Environment)

An environment $e \in \mathcal{E}$ specifies:

- Ranking $\rho_i^e \in \mathbb{R}^K$ for each agent i over all arms.
- Fixed ranking of all agents for each arm.
- The environment $e(t)$ active at time t is unknown to agents.

Assumption (Environment Structure)

For any two distinct environments $e, e' \in \mathcal{E}$,

$$\rho_i^e[1:N] \neq \rho_i^{e'}[1:N] \qquad \forall i \in [N].$$

The top-N preferences of each agent must be distinct across environments.

Justification: Without distinct rankings, cycles prevent convergence (example with serial dictatorship in the paper).

Regret Definition

Definition (Cumulative Expected Regret)

For agent i,

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

where

- $e(t)$ = active environment at time t,
- $m_i^*(e(t))$ = agent i's arm in the *agent-optimal* stable matching for $e(t)$,
- $m_i(t)$ = arm actually matched with i at time t.

Objective: Minimise cumulative regret for all agents over T rounds.

Minimum Reward Gap

Definition (Minimum Gap)

For agent i at time t,

$$\Delta_{\min,i}(t) = \min_{j,j' \in \mathrm{Top}_i(N+1)} \left| \langle \mathbf{x}_{i,j}(t) - \mathbf{x}_{i,j'}(t), \theta_i \rangle \right|.$$

Globally,

$$\Delta_{\min} = \min_{i,t} \Delta_{\min,i}(t).$$

$\mathrm{Top}_i(N+1)$ denotes the top $N+1$ arms for agent i in the current ranking.

Interpretation: Minimum margin between the $(N+1)$-st best arm and the $(N+2)$-nd best arm; determines identifiability of the top-N arms.

Key Algorithmic Ideas

1. **Round-Robin Exploration**: Avoid collisions during the learning phase.
2. **Least-Squares Estimation**: Estimate θ_i from the exploration phase.
3. **Confidence Intervals**: UCB/LCB for top-N arm identification.
4. **Environment Detection**: Detect when the top-N arms change (new environment).
5. **Gale-Shapley Matching**: Execute GS once the top-N arms are identified.
6. **Re-triggering**: Restart exploration if the environment changes.

Least Squares Estimation

Definition (LS Estimate and Design Matrix)

$$V_i(t) = \sum_{s=1}^{t} \mathbf{x}_{i,m_i(s)}(s)\mathbf{x}_{i,m_i(s)}^\top(s) \quad \text{(design matrix)} \tag{1}$$

$$\hat{\theta}_i(t) = V_i(t)^{-1} \sum_{s=1}^{t} r_i(s)\, \mathbf{x}_{i,m_i(s)}(s) \quad \text{(LS estimate)} \tag{2}$$

$$\hat{\mu}_{i,j}(t) = \langle \hat{\theta}_i(t), \mathbf{x}_{i,j}(t) \rangle \quad \text{(estimated reward).} \tag{3}$$

Confidence Intervals

Definition (Upper/Lower Confidence Bounds)

$$\mathrm{UCB}_{i,j}(t) = \hat{\mu}_{i,j}(t) + w_i\big(t, \mathbf{x}_{i,j}(t)\big), \tag{4}$$

$$\mathrm{LCB}_{i,j}(t) = \hat{\mu}_{i,j}(t) - w_i\big(t, \mathbf{x}_{i,j}(t)\big), \tag{5}$$

$$w_i(t, \mathbf{x}) = \sum_{\ell=1}^{d} \left|\langle \mathbf{x}, v_\ell \rangle\right| \sqrt{2 \, \|v_\ell\|^2_{V_i(t)^{-1}} \, \log t^2}, \tag{6}$$

where $\{v_\ell\}_{\ell=1}^{d}$ is any orthonormal basis of $\mathbb{R}^d$.

Top-N Arms Identification Condition

Definition (Top-N Arms Identification)

Agent i identifies its top-N arms at time t if

$$\forall a \in [N] : \ \mathrm{LCB}_{i,\sigma(a)}(t) > \max_{c:\, \sigma(a+1) \leq c \leq \sigma(K)} \mathrm{UCB}_{i,c}(t),$$

where $\sigma : [K] \to [K]$ is the permutation that orders the arms according to the true ranking.

Key Insight: Once the confidence intervals for the top-N arms do not overlap with any lower-ranked arm, the ranking is known with high probability.

Algorithm 1: ETP-GS (Pseudo-code)

Algorithm 1 ETP-GS: Environment-Triggered Phased Gale-Shapley

Initialize: $D[\cdot] \leftarrow \{\}$, $\tau_{\text{end}} \leftarrow 0$, $l \leftarrow 0$, $B \leftarrow 1$
for $t \geq 1$ **do**
  $B \leftarrow 1$ ▷ Environment Recovery
  if $\sigma_i(t) \neq \emptyset$ **then**
    if $\sigma_i(t) \notin \{\sigma_i[e] : e \in D\}$ **then**
      Add new environment: $D[e^*] \leftarrow (1, \sigma_i(t))$
    end if
    $e_i(t) \leftarrow e$ such that $\sigma_i[e] = \sigma_i(t)$
  else
    $B \leftarrow B \wedge 0$
  end if
  if $B = 0 \wedge t > \tau_{\text{end}}$ **then** ▷ Exploration Triggering
    $\tau_{\text{end}} \leftarrow t + 2^l$, $l \leftarrow l + 1$
  end if
  if $t \leq \tau_{\text{end}}$ **then** ▷ Exploration Phase
    $m_i(t) \leftarrow ((i + t) \bmod K) + 1$ ▷ Round-robin
    Observe $r_i(t)$, update $\hat{\theta}_i(t)$, UCB, LCB
  else ▷ Exploitation: GS Matching
    Retrieve $(s, \sigma) = D[e_i(t)]$
    $m_i(t) \leftarrow \sigma[s]$
    if $m_i(t) = \phi$ **then**
      Update $D[e_i(t)] \leftarrow (s + 1, \sigma)$
    end if
  end if
end for

Assumption 2: Full-Rank Feature Vectors

Assumption (Full-Rank Feature Vectors)

For each agent i, the arms $[K]$ can be partitioned into non-overlapping groups of d distinct arms $\mathcal{G} = \{G_1, \ldots, G_{\lfloor K/d \rfloor}\}$ such that for any group $G \in \mathcal{G}$ and any $\{t_1, \ldots, t_d\} \subseteq [T]$,

$$\lambda_{\min}\Big(\sum_{j=1}^{d} \mathbf{x}_{i,G(j)}(t_j)\mathbf{x}_{i,G(j)}^{\top}(t_j)\Big) \geq \kappa > 0.$$

Intuition

- Guarantees that the design matrix is well-conditioned.
- Ensures confidence intervals shrink at the usual $1/\sqrt{t}$ rate.
- Weaker than requiring all K feature vectors to be linearly independent.

Lemma 1: Concentration Bound

Lemma (Concentration for Least-Squares Estimate)

Under Assumption 2, with round-robin exploration and $\|\mathbf{x}\| \leq L$*, for* $t \geq d$*,*

$$\|\mathbf{x}\|_{V_i(t)^{-1}} \leq \sqrt{L \frac{d}{\kappa \mathcal{T}(t)}},$$

where $\mathcal{T}(t)$ *is the cumulative number of exploration pulls and* $\|\mathbf{x}\|_A = \sqrt{\mathbf{x}^\top A \mathbf{x}}$*.*

Proof Idea

- Apply a self-normalised concentration inequality (e.g. Abbasi-Yadkori et al., 2011).
- Round-robin guarantees each group G is sampled regularly, giving the eigenvalue lower bound κ.

Lemma 2: Confidence Width Shrinkage

Lemma (Confidence Width)

Conditional on the good event,

$$w_{i,j}(t) \leq 2dL\sqrt{\frac{\log t}{\kappa \mathcal{T}(t)}}.$$

Proof Sketch

- Use Lemma 1 to bound $\|\mathbf{x}_{i,j}(t)\|_{V_i(t)^{-1}}$.
- Plug the bound into the definition of $w_i(\cdot)$.
- The width shrinks as $1/\sqrt{\mathcal{T}(t)}$.

Lemma 3: UCB/LCB Consistency

Lemma (Upper/Lower Confidence Bound Consistency)

Conditional on the good event,

$$UCB_{i,j}(t) < LCB_{i,j'}(t) \implies \mu_{i,j}(t) < \mu_{i,j'}(t)$$

for all arms j, j'.

Proof Idea

- The good event guarantees $|\mu_{i,j}(t) - \hat{\mu}_{i,j}(t)| \leq w_{i,j}(t)$.
- If $\mathrm{UCB}_{i,j} < \mathrm{LCB}_{i,j'}$, then $\hat{\mu}_{i,j} + w_{i,j} < \hat{\mu}_{i,j'} - w_{i,j'}$, which implies $\mu_{i,j} < \mu_{i,j'}$.

Corollary: Once the top-N arms satisfy the identification condition, the agent's ranking is correct with high probability.

Lemma 4: Top-N Arms Identification Time

Lemma (Top-N Arms Identification)

Conditional on the good event, agent i identifies its top-N arms once

$$\Delta_{\min,i}(t) \geq 8Ld\sqrt{\frac{\log t}{\kappa \mathcal{T}(t)}}.$$

Equivalently, after at most

$$\mathcal{T}_{identify} = O\left(\frac{d^2 L^2 \log T}{\kappa \Delta_{\min,i}^2}\right)$$

exploration pulls.

Proof Sketch

- The identification condition requires $\mathrm{LCB}_{i,\sigma(N)} > \mathrm{UCB}_{i,\sigma(N+1)}$.
- Using Lemma 2, this holds when the confidence width is smaller than the minimum gap $\Delta_{\min,i}(t)$.

Lemma 5: Gale-Shapley Convergence

Lemma (Gale-Shapley Steps)

Once every agent has correctly identified its top-N arms for a given environment, the Gale-Shapley deferred-acceptance procedure reaches a stable matching in at most $N^2 - 2N + 2$ proposals.

Proof Reference: Classical analysis of the Gale-Shapley algorithm (Gale & Shapley, 1962).

Lemma 6: Bad-Event Probability

Lemma (High-Probability Bound)

The total expected number of rounds in which the concentration bounds fail satisfies

$$\mathbb{E}\Big[\sum_{t=1}^{T} \mathbf{1}\{\textit{bad event at } t\}\Big] \le \frac{Nd\pi^2}{3}.$$

Proof Idea

- Apply the self-normalised concentration inequality with a $1/t^2$ tail.
- Union-bound over N agents and d dimensions.
- $\sum_{t=1}^{\infty} 1/t^2 = \pi^2/6$ gives the constant.

Implication: Regret incurred due to bad events is $O(1)$, not $O(T)$.

Lemma 7: Dominant Regret Term

Lemma (Event $\mathcal{E}_3(t)$ Timing)

Conditional on the good event, the event $\neg\mathcal{E}_3(t)$ (top-N arms not yet identified) can only occur while

$$\mathcal{T}(t) < \frac{64 d^2 L^2 \log T}{\kappa\, \Delta_{\min}^2}.$$

Interpretation

- After $\tau_* = 64 d^2 L^2 \log T / (\kappa \Delta_{\min}^2)$ exploration pulls, every agent has identified its top-N arms with high probability.
- The algorithm then switches to the Gale-Shapley exploitation phase.

Theorem 1: ETP-GS Regret Bound

Theorem (ETP-GS Regret)

Under Assumptions 1 and 2, the cumulative expected regret for agent i satisfies

$$\boxed{\mathbb{E}\big[R_T^{(i)}\big] \leq \left(\frac{64 d^2 L^2 \log T}{\kappa\,\Delta_{\min}^2} + EN^2 + \frac{N d \pi^2}{3}\right)\mu_{i,\max}}.$$

Here $\mu_{i,\max} = \max_{j,t} \mu_{i,j}(t)$.

Proof Sketch

1. **Exploration**: Lemma 7 gives $O\big(d^2 L^2 \log T/(\kappa\Delta_{\min}^2)\big)$ rounds of sub-optimal play.
2. **Exploitation (GS)**: Lemma 5 contributes at most EN^2 rounds of regret (one GS run per environment).
3. **Bad events**: Lemma 6 contributes $O(1)$ regret.

Regret Bound Scaling

Term 1: Exploration

$$\frac{64d^2L^2\log T}{\kappa\,\Delta_{\min}^2}$$

- Dominant for large horizons.
- Logarithmic in T.
- Independent of the number of arms K.

Term 2: Gale-Shapley

$$EN^2$$

- Finite, environment-dependent term.
- Independent of T.

Overall order

$$\mathbb{E}[R_T^{(i)}] = \tilde{O}\Big(\frac{d^2}{\Delta_{\min}^2} + EN^2\Big)$$

(ignoring logarithmic factors and constants).

Numerical Example: The Problem with ETP-GS

The "Small Gap" Problem

In **Environment 1 ("Weekday")**, Alice's preferences are:

- $\mu_{A,A} = 7.8$
- $\mu_{A,B} = 7.7$

The gap between her #1 and #2 arm is tiny: $\Delta = 0.1$.

The Sticking Point

ETP-GS *must* wait until it satisfies the Top-N ID Condition:

$$\mathrm{LCB}_{A,A} > \mathrm{UCB}_{A,B}$$

To separate a true gap of 0.1, the confidence width $w_i(t, x)$ must be even smaller.

Contnd...

Result (from Lemma 4)

The required exploration rounds $\mathcal{T}_{\text{identify}}$ depends on $1/\Delta^2_{\text{min}}$:

$$\mathcal{T}_{\text{identify}} \propto \frac{1}{(0.1)^2} \propto \frac{1}{0.01} \propto 100\text{x}$$

The algorithm is forced into a very long exploration phase to resolve a tiny gap that doesn't even change the stable match for Alice.

Comparison with Single-Agent Lower Bounds

Theorem (Linear Contextual Bandit Lower Bound)

For any algorithm in the single-agent linear contextual bandit setting,

$$\Omega\left(\frac{d}{\Delta_{\min}} \log T\right)$$

regret is unavoidable.

Interpretation

- Our multi-agent bound matches the $\log T$ dependence but incurs an extra factor $d/\Delta_{\min}$ due to the need to identify the top-N arms jointly.
- The EN^2 term is unavoidable in a changing-environment setting (each environment may require a new stable matching).

IETP-GS – Key Improvements

1. **Partial-Rank Matching**: Use Kendall-τ distance to detect environment changes earlier.
2. **Reward-Gap Period**: Exploit periods where the reward gap is larger than Δ_{min}.
3. **Rank-Based Gap $\Delta_{\text{min}}^{\text{rank}}$**: Often larger than the raw reward gap, leading to fewer exploration rounds.
4. **Improved Constants**: Tighter analysis yields smaller multiplicative factors.

Motivation: In many practical settings the ordering of arms stabilises long before the exact reward values do.

Kendall-τ Distance

Definition (Inverted Pairs and Kendall-τ)

For two (partial) rankings pr and pr$'$,

$$\mathrm{Inv}(\mathrm{pr}, \mathrm{pr}') = \{(k, j) : (k \succ_{\mathrm{pr}} j \wedge k \prec_{\mathrm{pr}'} j) \vee (k \prec_{\mathrm{pr}} j \wedge k \succ_{\mathrm{pr}'} j)\}.$$

The Kendall-τ distance is

$$\mathrm{KT}(\mathrm{pr}, \mathrm{pr}') = |\mathrm{Inv}(\mathrm{pr}, \mathrm{pr}')|.$$

Use in IETP-GS

- When the partial ranking inferred from the current confidence intervals matches an existing environment (i.e. $\mathrm{KT} = 0$), we can immediately switch to the corresponding Gale-Shapley phase.
- This avoids waiting for the full reward gap to become large.

Reward-Gap Period

Definition (Reward-Gap Period $P_e(\Delta)$)

For environment e,

$$P_e(\Delta) = \max_i \max_{\nu_e \geq 1} \min\left\{\delta\nu_e : \delta\nu_e \geq 0,\ \Delta_{\min,i}\big(t_e(\nu_e + \delta\nu_e)\big) \geq \Delta\right\}.$$

Intuitively, $P_e(\Delta)$ is the longest stretch (in rounds) needed for the minimum reward gap to reach at least Δ after the environment e becomes active.

Key Insight: By choosing a larger Δ we may reduce the number of required exploration rounds, at the cost of waiting a (potentially short) gap period.

Algorithm 2: IETP-GS (Pseudo-code)

Algorithm 2 IETP-GS

```
Initialize: $D[\cdot] \leftarrow \{\}$, $\tau_{\text{end}} \leftarrow 0$, $l \leftarrow 0$, $B \leftarrow 1$
for $t \geq 1$ do
    $B \leftarrow 1$                                                        ▷ Environment Recovery
    Compute partial ranking $\text{pr}_i(t)$ from $\hat{\mu}_{i,\cdot}(t)$
    if $\sigma_i(t) \neq \emptyset$ or $(|D| = E$ and $\exists! e : \text{KT}(\text{pr}_i(t), \sigma_i[e]) = 0)$ then
        if $\sigma_i(t) \neq \emptyset$ and $\sigma_i(t) \notin D$ then
            Add new environment: $D[e^*] \leftarrow (1, \sigma_i(t))$
        end if
        Retrieve or set $e_i(t)$ (environment index)
    else
        $B \leftarrow B \wedge 0$
    end if
    if $B = 0 \wedge t > \tau_{\text{end}}$ then
        $\tau_{\text{end}} \leftarrow t + 2^l$, $l \leftarrow l + 1$
    end if
    if $t \leq \tau_{\text{end}}$ then                                     ▷ Exploration Phase
        $m_i(t) \leftarrow ((i + t) \bmod K) + 1$                          ▷ Round-robin
        Observe $r_i(t)$, update $\hat{\theta}_i(t)$, UCB, LCB
    else                                                                   ▷ Exploitation: GS Matching
        Retrieve $(s, \sigma) = D[e_i(t)]$
        $m_i(t) \leftarrow \sigma[s]$
        if $m_i(t) = \phi$ then
            Update $D[e_i(t)] \leftarrow (s + 1, \sigma)$
        end if
    end if
end for
```

Key Lemmas for IETP-GS

Lemma (First Exploration Phase Length)

Conditional on the good event,

$$\neg\mathcal{E}_0(t) \cap \neg\mathcal{E}_1(t) \text{ occurs for at most } \min_{\Delta>0}\Big(\tau(\Delta) + \sum_e P_e(\Delta)\Big)$$

rounds, where $\tau(\Delta) = \frac{64 d^2 L^2 \log T}{\kappa \Delta^2}$.

Lemma (Additional Exploration Phases)

Conditional on the good event,

$$\mathcal{E}_1(t) \cap \neg\mathcal{E}_2(t) \text{ occurs for at most } g\Big(\frac{64 L^2 d^2 \log T}{\Delta_{\min}^{rank2}}\Big)$$

rounds, where $g(\cdot)$ *is a polynomial.*

Key Lemmas for IETP-GS Contd...

Lemma (Environment Identification Error)

Conditional on the good event, the total number of rounds in which the algorithm mis-identifies the active environment is bounded by EN^2.

Theorem 2: IETP-GS Regret Bound

Theorem (IETP-GS Regret)

Under Assumptions 1 and 2,

$$\boxed{\mathbb{E}[R_T^{(i)}] \leq \left(\min_{\Delta>0}\left(\frac{64d^2L^2\log T}{\kappa\Delta^2} + \sum_e P_e(\Delta)\right) + g\left(\frac{64L^2d^2\log T}{\Delta_{\min}^{rank2}}\right) + EN^2 + \frac{Nd\pi^2}{3}\right)\mu}$$

Improvements over Theorem 1

- The minimisation over Δ allows the algorithm to exploit larger reward gaps when they appear.
- The rank-based gap $\Delta_{\min}^{\text{rank}}$ can be substantially larger than $\Delta_{\min}$, reducing the dominant exploration term.
- The $P_e(\Delta)$ term captures the (often short) waiting time needed for a gap of size Δ to materialise.

Numerical Example: How IETP-GS Wins

The "Smart" Check (Partial Rank)

After a *short* exploration, Alice's bounds are still overlapping:

- $\mu_{A,A}$ (True 7.8) $\rightarrow$ Estimate [7.6, 8.0]
- $\mu_{A,B}$ (True 7.7) $\rightarrow$ Estimate [7.5, 7.9]
- $\mu_{A,C}$ (True 4.6) $\rightarrow$ Estimate [4.4, 4.8]

Alice's Partial Rank: $\mathrm{pr}_A(t) = (A = B) > C$

The Kendall-τ (KT) Match

The algorithm checks $\mathrm{pr}_A(t)$ against its memory D:

- KT($\mathrm{pr}_A(t)$, ρ^{e1} ("A>B")) = KT("(A=B)>C" , "A>B") = 0 (Match!)
- KT($\mathrm{pr}_A(t)$, ρ^{e2} ("C>B")) = KT("(A=B)>C" , "C>B") = 2 (No Match)

Contnd...

Result: Uses a Bigger Gap

IETP-GS finds a unique match and **stops exploring**. It doesn't use $\Delta_{\text{min}} = 0.1$. It uses the *rank-based gap* (to tell $e1$ from $e2$):

$$\Delta_{\text{min}}^{\text{rank}} = \min(|\mu_{A,A} - \mu_{A,C}|, |\mu_{A,B} - \mu_{A,C}|) = \min(3.2, 3.1) = \mathbf{3.1}$$

Exploration is now $O(1/\mathbf{3.1}^2) \approx O(0.1)$, which is much faster than $O(100)$.

Beyond Fixed Parameters: Piecewise-Stationary Model

Challenge

Agent parameters θ_i may evolve over time (e.g. career changes, seasonal effects).

Solution (Piecewise-Stationary Assumption)

- *There exist change-points* $0 < \tau_1 < \tau_2 < \cdots < \tau_{\gamma_T} < T$.
- *Within each interval* $[\tau_k, \tau_{k+1})$, *the vector* θ_i *is fixed.*
- *The number of changes* γ_T *may grow with* T *(but slowly).*

Motivation: Workers' skills, task rewards, or platform policies often shift in a piecewise-constant fashion.

Algorithm 3: CD-ETP-GS (Change-Detection)

Idea

Combine IETP-GS with a statistical change-detection test (e.g. CUSUM) on the residuals of the LS estimator.

Definition (CUSUM Statistic)

$$S_t = \max\bigl(0,\ S_{t-1} + \ell_t - \mu - \beta\bigr),$$

where ℓ_t is the log-likelihood ratio of the new observation, μ a reference mean, and $\beta > 0$ a threshold. A change is declared when S_t exceeds a pre-specified level.

Result: When a change is detected, the algorithm resets its exploration/exploitation state and starts learning the new θ_i.

Algorithm 3: CD-ETP-GS (Pseudo-code)

Algorithm 3 CD-ETP-GS: Change-Detection aided ETP-GS

Initialize: CUSUM detector, $\hat{\tau} \leftarrow 0$, IETP-GS state.
for $t' = 1$ **to** T **do**
  $t \leftarrow t' - \hat{\tau}$ ▷ Local time in current segment
  $\mathcal{CD} \leftarrow 1$
  if CD.IsForcedExploration(t) **then**
    Play round-robin exploration, observe $r_i(t)$.
    Update LS estimate and CUSUM statistic.
    if S_t exceeds threshold **then**
      $\mathcal{CD} \leftarrow 0$ ▷ Change detected
    end if
    if $\mathcal{CD} = 0$ **then**
      $\hat{\tau} \leftarrow t'$; clear D, reset θ_i, UCB/LCB, etc.
    end if
  else
    Run IETP-GS (business as usual).
  end if
end for

Theorem (CD-ETP-GS Regret)

Let γ_T be the number of change-points. Then for each agent i,

$$\mathbb{E}[R_T^{(i)}] = \tilde{O}\Big(L\sqrt{\gamma_T T \log \frac{NT}{\gamma_T}} + 2\gamma_T \, Regret^{(i)}(T/\gamma_T; IETP\text{-}GS)\Big).$$

Interpretation

- The first term is the cost of detecting and adapting to changes (sub-linear in T as long as $\gamma_T = o(T)$).
- The second term is the sum of the regrets incurred in each stationary segment (each segment behaves like the static case analysed for IETP-GS).
- If $\gamma_T = O(1)$, the overall regret matches the static bound up to logarithmic factors.

Numerical Example: How CD-ETP-GS Works

The Change (at $t = 201$)

The system has learned $e1$ and $e2$. Now, Alice's own parameter *changes*:

$$\text{Old } \theta_A = [10, -2] \quad \rightarrow \quad \text{New } \theta'_A = [10, 10]$$

(Alice suddenly *likes* difficult tasks)

The Error (at $t = 205$, Env 2 is active)

- **Algorithm's Belief (Old θ_A):** In $e2$, Task C is best for Alice. It matches her to Task C, expecting $\mu_{A,C} \approx 6.8$.
- **Reality (New θ'_A):** Alice's *true* reward for Task C is now:

$$\mu'_{A,C} = \langle [10, 10], [0.7, 0.1] \rangle = 7.0 + 1.0 = \mathbf{8.0}$$

Contnd...

The Detection

1. The CUSUM detector calculates the *residual* (error):

$$\text{residual} = \text{observed reward} - \text{expected reward} \approx 8.0 - 6.8 = +\mathbf{1.2}$$

2. This large, consistent error accumulates in the CUSUM statistic S_t.
3. S_t crosses the threshold $\rightarrow$ **Change Detected!**
4. The algorithm RESETS all estimates and re-enters exploration to learn the new θ'_A.

Computational Complexity

Component	Complexity per round	Remarks
Round-robin selection	$O(1)$	Simple arithmetic
LS update	$O(d^2)$ (Sherman–Morrison)	Inverse update can be don
UCB/LCB computation	$O(d^2)$ per arm	Cached V_i^{-1} helps
Top-N sorting	$O(K \log K)$	Only needed when checkir
Environment lookup	$O(\log E)$	Hash table / map
Gale-Shapley step	$O(N)$	One proposal per active ag
Dominant cost: $O(d^2 + K \log K)$ (usually $d \ll K$)		

Speed-up tricks

- Incremental matrix inversion (Sherman–Morrison) reduces LS update to $O(d^2)$.
- Update UCB/LCB only for arms whose features changed.
- Run Gale-Shapley only when a new environment is detected.

Relaxing Assumption 2

Full-Rank Requirement

Assumption 2 guarantees a uniform lower bound κ on the smallest eigenvalue of the design matrix.

- **Random features**: If $\mathbf{x}_{i,j}(t)$ are i.i.d. from a distribution with full-rank covariance, the condition holds w.h.p. (by concentration of random matrices).
- **Block designs**: It suffices that each *group* of d arms is sampled regularly (as already enforced by the round-robin schedule).
- **Heteroskedastic noise**: Recent work (e.g. Lumbreras & Tomamichel, 2024) shows that logarithmic regret can be achieved without a strict spectral gap, at the price of larger constants.

Practical tip: Verify the condition empirically on a pilot dataset; if violated, increase the exploration frequency for poorly-conditioned arms.

Related Work and Open Questions

Related Areas

- Gale–Shapley with bandit feedback (Liu et al., 2020; Kong & Li, 2023).
- Latent bandits and contextual matching (Hong et al., 2020).
- Linear contextual bandits (OFUL, Abbasi-Yadkori et al., 2011).
- Multi-agent learning and decentralized decision making.

Open Questions

- Two-sided learning (both agents and arms learn).
- Continuous (non-finite) environment spaces.
- Removing the spectral assumption completely.
- Communication-efficient decentralised protocols.
- Complementary (rather than identical) preferences across agents.

Appendix – Proof Sketch of Theorem 1

Regret decomposition.

$$\mathbb{E}[R_T^{(i)}] = \underbrace{\text{Exploration regret}}_{\text{Rounds } t \le \tau_{\text{end}}} + \underbrace{\text{Exploitation regret}}_{\text{Rounds } t > \tau_{\text{end}}}.$$

- **Exploration**: By Lemma 7 the algorithm stays in exploration for at most $\frac{64 d^2 L^2 \log T}{\kappa \Delta_{\min}^2}$ rounds; each round incurs at most $\mu_{i,\max}$ regret.
- **Exploitation**: Lemma 5 guarantees at most EN^2 rounds of sub-optimal proposals (one Gale-Shapley run per environment).
- **Bad events**: Lemma 6 contributes at most $\frac{Nd\pi^2}{3}$ regret.

Adding the three contributions yields the bound stated in the theorem. □

Notation Summary

Symbol	Meaning
N, K, T	Number of agents, arms, horizon
$\theta_i \in \mathbb{R}^d$	Latent parameter of agent i
$\mathbf{x}_{i,j}(t) \in \mathbb{R}^d$	Feature vector for (i, j) at time t
$\mu_{i,j}(t) = \langle \mathbf{x}_{i,j}(t), \theta_i \rangle$	Expected reward
$m_i(t)$	Arm matched to agent i at time t
$e(t)$	Active environment at time t (latent)
$\mathcal{E}$	Set of all environments, $\lvert\mathcal{E}\rvert = E$
ρ_i^e	Preference ranking of agent i in environment e
$\hat{\theta}_i(t)$	LS estimate of θ_i
$V_i(t)$	Design matrix $\sum_{s \le t} \mathbf{x}_{i,m_i(s)} \mathbf{x}_{i,m_i(s)}^\top$
$\Delta_{\min}$	Minimum reward gap (Definition 4)
$P_e(\Delta)$	Reward-gap period (Definition 6)
$\mathrm{KT}(\cdot, \cdot)$	Kendall-τ distance (Definition 5)

References

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

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

Liu, M., et al. (2020). Learning in auctions: Regret of dominant-strategy mechanisms. *Proceedings of the ACM EC*.

Kong, D., & Li, Y. (2023). Incentive-compatible learning for two-sided matching markets. *Proceedings of the ACM EC*.

Sankararaman, A., et al. (2021). A unified framework for two-agent matching markets. *International Conference on Machine Learning*.

Hong, L., et al. (2020). Latent bandits. *International Conference on Machine Learning*.

Lumbreras, A., & Tomamichel, M. (2024). Heteroskedastic linear

Thank you!

Questions?