Combinatorial Pure Exploration for Dueling Bandits

Wei Chen [*1] **Yihan Du** [2] **Longbo Huang** [2] **Haoyu Zhao** [2]

Abstract

In this paper, we study combinatorial pure exploration for dueling bandits (CPE-DB): we have multiple candidates for multiple positions as modeled by a bipartite graph, and in each round we sample a duel of two candidates on one position and observe who wins in the duel, with the goal of finding the best candidate-position matching with high probability after multiple rounds of samples. CPE-DB is an adaptation of the original combinatorial pure exploration for multi-armed bandit (CPE-MAB) problem to the dueling bandit setting. We consider both the Borda winner and the Condorcet winner cases. For Borda winner, we establish a reduction of the problem to the original CPE-MAB setting and design PAC and exact algorithms that achieve both the sample complexity similar to that in the CPE-MAB setting (which is nearly optimal for a subclass of problems) and polynomial running time per round. For Condorcet winner, we first design a fully polynomial time approximation scheme (FPTAS) for the offline problem of finding the Condorcet winner with known winning probabilities, and then use the FPTAS as an oracle to design a novel pure exploration algorithm CAR-Cond with sample complexity analysis. CAR-Cond is the first algorithm with polynomial running time per round for identifying the Condorcet winner in CPE-DB.

1. Introduction

Multi-Armed Bandit (MAB) (Lai & Robbins, 1985; Thompson, 1933; Auer et al., 2002; Agrawal & Goyal, 2012) is a classic model that characterizes the exploration-exploitation tradeoff in online learning. The pure exploration task (Even-Dar et al., 2006; Chen & Li, 2016; Sabato, 2019) is an important variant of the MAB problems, where the objective is to identify the best arm with high confidence, using as few samples as possible. A rich class of pure exploration problems have been extensively studied, e.g., best K-arm identification (Kalyanakrishnan et al., 2012) and multi-bandit best arm identification (Bubeck et al., 2013). Recently, Chen et al. (2014) proposes a general combinatorial pure exploration for multi-armed bandit (CPE-MAB) framework, which encompasses previous pure exploration problems. In the CPE-MAB problem, a learner is given a set of arms and a collection of arm subsets with certain combinatorial structures. At each time step, the learner plays an arm and observes the random reward, with the objective of identifying the best combinatorial subset of arms. Gabillon et al. (2016); Chen et al. (2017) follow this setting and further improve the sample complexity.

However, in many real-world applications involving implicit (human) feedback including social surveys (Alwin & Krosnick, 1985), market research (Ben-Akiva et al., 1994) and recommendation systems (Radlinski et al., 2008), the information observed by the learner is intrinsically relative. For example, in voting and elections, it is more natural for the electors to offer preference choices than numerical evaluations on candidates. For this scenario, the dueling bandit formulation (Yue et al., 2012; Ramamohan et al., 2016; Sui et al., 2018) provides a promising model for online decision making with relative feedback.

In this paper, we contribute a model adapting the original CPE-MAB problem to the dueling bandit setting. Specifically, we formulate the *combinatorial pure exploration for dueling bandit (CPE-DB)* problem as follows. A CPE-DB instance consists of a bipartite graph G modeling multiple candidates that could fit into multiple positions, and an *unknown preference probability matrix* specifying when we play a duel between two candidates for one position, the probability that the first would win over the second. At each time step, a learner samples a duel of two candidates on one position and observes a random outcome of which candidate wins in this duel sampled according to the preference probability matrix. The objective is to use as few duel samples as possible to identify the best candidate-position matching with high confidence, for two popular optimality metrics

[*]Alphabetical order [1]Microsoft Research, Beijing, China [2]IIIS, Tsinghua University, Beijing, China. Correspondence to: Wei Chen <weic@microsoft.com>, Yihan Du <duyh18@mails.tsinghua.edu.cn>, Longbo Huang <longbohuang@mail.tsinghua.edu.cn>, Haoyu Zhao <zhaohy16@tsinghua.org.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

in the dueling bandit literature, i.e., Condorcet winner and Borda winner.

The CPE-DB model represents a novel preference-based version of the common candidate-position matching problems, which occurs in various real-world scenarios, including social choice (McLean, 1990), multi-player game (Graepel & Herbrich, 2006) and online advertising (Joachims et al., 2017). For instance, a committee selection procedure (Gehrlein, 1985) may want to choose among multiple candidates one candidate for each position to form a committee. For any two candidates on one position, we can play a duel on them, e.g., by surveying a bystander, to learn a sample of which candidate would win on this position, and the sample follows an unknown preference probability. We hope to play as few duels as possible (or by surveying as few people as possible) to identify the best performing committee.

The CPE-DB problem raises interesting challenges on exponentially large decision space and relative feedback. The key issue here is how to exploit the problem structure and design algorithms that guarantee both high computational efficiency and low sample complexity. Therefore, the design and analysis of algorithms for CPE-DB demand novel computational acceleration techniques. The contributions of this work are summarized as follows:

(1). We formulate the combinatorial pure exploration for dueling bandit (CPE-DB) problem, adapted from the original combinatorial pure exploration for multi-armed bandit (CPE-MAB) problem to the dueling bandit setting, and associate it with various real-world applications involving preference-based bipartite matching selection.

(2). For the Borda winner metric, we reduce CPE-DB to the original CPE-MAB problem, and design algorithms CLUCB-Borda-PAC and CLUCB-Borda-Exact with polynomial running time per round. We provide their sample complexity upper bounds and a problem-dependent lower bound for CPE-DB with Borda winner. Our upper and lower bound results together show that CLUCB-Borda-Exact achieves near-optimal sample complexity for a subclass of problems.

(3). For the Condorcet winner metric, we design a fully polynomial time approximation scheme (FPTAS) for a proper extended version of the offline problem, and then adopt the FPTAS to design a novel online algorithm CAR-Cond with sample complexity analysis. To our best knowledge, CAR-Cond is the first algorithm with polynomial running time per round for identifying the Condorcet winner in CPE-DB.

1.1. Related Works

Combinatorial pure exploration The combinatorial pure exploration for multi-armed bandit (CPE-MAB) problem is first formulated by Chen et al. (2014) and generalizes the multi-armed bandit pure exploration task to general combinatorial structures. Gabillon et al. (2016) follow the setting of (Chen et al., 2014) and propose algorithms with improved sample complexity but a loss of computational efficiency. Chen et al. (2017) further design algorithms for this problem that have tighter sample complexity and pseudo-polynomial running time. Wu et al. (2015) study another combinatorial pure exploration case in which given a graph, at each time step, a learner samples a path with the objective of identifying the optimal edge.

Dueling bandit The dueling bandit problem (Yue et al., 2012; Ramamohan et al., 2016; Sui et al., 2018), first proposed by (Yue et al., 2012), is an important variation of the multi-armed bandit setting. According to the assumptions on preference structures and definitions of the optimal arm (winner), previous methods can be categorized as methods on Condorcet winner (Komiyama et al., 2015; Xu et al., 2019), methods on Borda winner (Jamieson et al., 2015; Xu et al., 2019), methods on Copeland winner (Wu & Liu, 2016; Agrawal & Chaporkar, 2019), etc. Recently, Saha & Gopalan (2019) propose a variant of combinatorial bandits with relative feedback. In their setting, a learner plays a subset of arms (assuming each arm has an unknown positive value) in a time step and observes the ranking feedback, and the goal is to minimize the cumulative regret. Therefore, their model is quite different from ours.

2. Problem Formulation

In this section, we formally define the combinatorial pure exploration problem for dueling bandits. Suppose that there are n candidates $C = \{c_1, \ldots, c_n\}$ and ℓ positions $S = \{s_1, \ldots, s_\ell\}$ with $n \geq \ell$. Each candidate is available for several positions, and we use bipartite graph $G(C, S, E)$ to denote this relation, where each edge $e = (c_i, s_j) \in E$ denotes that candidate c_i is capable for position s_j. We define $m = |E|$. We use E_j to denote the set of edges connected to position j, i.e., $E_j = \{e = (c, s_j) \in E : c \in C\}$ and we also use $s(e)$ to denote the position index of e.

Two edges e and e' are comparable if they have the same position indices, i.e. $s(e) = s(e')$. For any two comparable edges $e = (c, s_j)$ and $e' = (c', s_j)$, there is an unknown preference probability $p_{e,e'}$, which means that with probability $p_{e,e'}$, e wins e', or c wins c' on position j. We have $p_{e,e'} = 1 - p_{e',e}$. For any $e \in E$, we define $p_{e,e} = \frac{1}{2}$.

Given the graph $G(C, S, E)$, we define an order of edges in E by first ranking them by their position indices from smallest to the largest and then ranking them by their candidate index from the smallest to the largest. Given the order of the edges, we use e_i to denote the i-th edge in the order, and define $\chi_M \in \{0, 1\}^m$ as the vector representation of

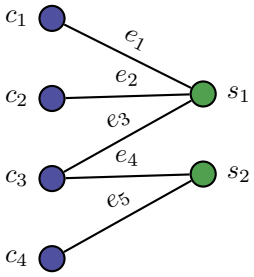

Figure 1. Graph

	e_1	e_2	e_3	e_4	e_5
e_1	0.5	0.45	1	0	0
e_2	0.55	0.5	0.55	0	0
e_3	0	0.45	0.5	0	0
e_4	0	0	0	0.5	0
e_5	0	0	0	1	0.5

Figure 2. Preference Matrix

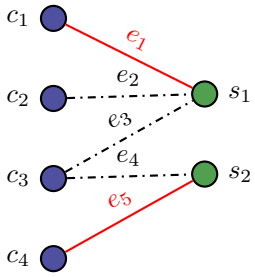

Figure 3. Borda Winner

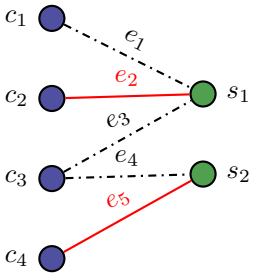

Figure 4. Condorcet Winner

the edges $M \subset E$, where $(\chi_M)_i = 1$ if and only if $e_i \in M$. We also use a preference matrix $P \in ([0,1])^{m \times m}$ to record all preference probabilities. Specifically, for any two comparable edges e_i, e_j, $P_{i,j} = p_{e_i,e_j}$ is the preference probability of e_i over e_j. For two incomparable edges $e_{i'}, e_{j'}$, $P_{i',j'}$ is set to 0 for the convenience of later computations. Figure 1 show an example bipartite graph and Figure 2 shows its corresponding preference matrix.

Note that for each position s_j, any two edges connecting to s_j can be compared with a preference probability. This is similar to the dueling bandit setting (Yue et al., 2012), where each edge is an arm, and we can compare the arms (edges) to find the best arm (edge), i.e., finding the best candidate for a position. Thus, from now on, we will use arms and edges interchangeably. We define $K = \sum_{j=1}^{\ell} \frac{|E_j|(|E_j|-1)}{2}$, which is the number of all possible duels between any two comparable arms.

We assume that there is at least one matching with cardinality ℓ in G, meaning that we can find at least one candidate for each position without a conflict. The *decision class* $\mathcal{M} \subset 2^E$ is the set of all maximum matchings in G. We can also view a matching as a team that specifies which candidate shall play which position for all the positions. Given a matching M and a position s_j, we use $e(M,j)$ to represent the edge in M that connects to position s_j. For any two matchings $M_1, M_2 \in \mathcal{M}$, we define the preference probability of M_1 over M_2 as follows:

$$f(M_1, M_2, P) := \frac{1}{\ell} \sum_{j=1}^{\ell} p_{e(M_1,j),e(M_2,j)}. \tag{1}$$

It is easy to show that $f(M_1, M_2, P) = 1 - f(M_2, M_1, P)$. Written in vector representation, we have $f(M_1, M_2, P) = \frac{1}{\ell} \chi_{M_1}^T \cdot P \cdot \chi_{M_2}$.

Now, we define the "best" matching in the decision class $\mathcal{M}$. There are several different definitions, e.g., Borda winner (Emerson, 2013; 2016), Condorcet winner (Black, 1948), and Copeland winner (Copeland, 1951; Saari & Merlin, 1996). In this paper, we focus on the Borda winner and the Condorcet winner, the definitions of which are given below.

Borda winner The Borda winner refers to the winner that maximizes the average preference probability over the decision class, which we call "Borda score". Mathematically, in our framework, the Borda score of any matching $M_x \in \mathcal{M}$ and the Borda winner are defined as:

$$B(M_x) = \frac{1}{|\mathcal{M}|} \sum_{M_y \in \mathcal{M}} f(M_x, M_y, P), \tag{2}$$

$$M_*^B = \operatorname*{argmax}_{M_x \in \mathcal{M}} B(M_x). \tag{3}$$

For the pure exploration task, we assume that there is a unique Borda winner, similar to the assumption in other pure exploration tasks (Even-Dar et al., 2006; Bubeck et al., 2013; Chen et al., 2014; 2017). Figure 3 shows the Borda winner as the matching with the red edges, because according to the preference matrix in Figure 2, it has the largest Borda score of 0.64.

Condorcet winner The Condorcet winner is the matching that always wins when compared to others. In our framework, the Condorcet winner is defined as the matching M_*^C such that $f(M_*^C, M, P) \geq \frac{1}{2}$ for any matching $M \in \mathcal{M}$. We assume that the Condorcet winner exists as several previous works (Zoghi et al., 2014; Komiyama et al., 2015; Chen & Frazier, 2017) do, and the Condorcet winner wins over any other matching with probability strictly better than $\frac{1}{2}$, i.e. $f(M_*^C, M, P) > \frac{1}{2}$ for any $M \in \mathcal{M} \setminus \{M_*^C\}$. Figure 4 shows the Condorcet winner as the matching with the red edges. It is different from the Borda winner in this example, since this matching wins over all other matchings, but its average winning score (the Borda score) 0.615 is not as good as the Borda winner.

Our goal is to find the best matching (Borda winner or the Condorcet winner) by exploring the duels at all the positions, and we want the number of duels that we need to explore as small as possible. This is the problem of combinatorial pure exploration for dueling bandits (CPE-DB). More precisely, at the beginning, the graph $G(C, S, E)$ is given to the learner, but the preference matrix P is un-

known. Because the learner does not know the preference probability for arms connected to the same position, she needs to sample the duel between edges. In each round the learner samples one duel pair (e, e') for some position, and she observes a Bernoulli random variable $X_{e,e'}$ with $\Pr\{X_{e,e'} = 1\} = p_{e,e'}$. The observed feedback could be used to help to select future pairs to sample. Our objective is to find the Borda winner M_*^B or the Condorcet winner M_*^C with as few samples as possible.

3. Efficient Exploration for Borda Winner

In this section, we first show the reduction of the Borda winner identification problem to the combinatorial pure exploration for multi-armed bandit (CPE-MAB) problem, originally proposed and studied in (Chen et al., 2014). Next, we introduce an efficient PAC pure exploration algorithm CLUCB-Borda-PAC for Borda winner, and show that with an almost uniform sampler for perfect matchings (Jerrum et al., 2004), CLUCB-Borda-PAC has both tight sample complexity and fully-polynomial time complexity. Then, based on the PAC algorithm CLUCB-Borda-PAC, we further propose an exact pure exploration algorithm CLUCB-Borda-Exact for Borda Winner, and provide its sample complexity upper bound. Finally, we present the sample complexity lower bound for identifying the Borda winner.

3.1. Reduction to Conventional Combinatorial Pure Exploration

In order to show the reduction of CPE-DB for Borda winner to the conventional CPE-MAB (Chen et al., 2014) problem, we first define the rewards for edges. Then, we define the reward of a matching to be the sum of its edge rewards. Based on the reward definitions, it can be shown that the problem of identifying the Borda winner is equivalent to identifying the matching with the maximum reward. Specifically, for any edge $e = (c_i, s_j) \in E$ and matching $M \in \mathcal{M}$, we define their rewards and the reduction relationship between the two problems as follows:

$$\begin{aligned} w(e) &= \frac{1}{|\mathcal{M}|} \sum_{M \in \mathcal{M}} p_{e,e(M,j)}, \\ w(M) &= \sum_{e \in M} w(e) \overset{(a)}{=} \ell \cdot B(M), \\ M_*^B &= \operatorname*{argmax}_{M \in \mathcal{M}} B(M) = \operatorname*{argmax}_{M \in \mathcal{M}} w(M), \end{aligned} \tag{4}$$

where the equality (a) is due to the definitions of the Borda score (Eq. (2)) and preference probability between two matchings (Eq. (1)) (see the supplementary material for a detailed proof of equality (a)).

It remains to show how to efficiently learn the reward $w(e)$ for edge e, by sampling arm pairs in CPE-DB.

First, we can see that for any edge $e = (c_i, s_j)$, $w(e)$ is exactly the *expected* preference probability of e over $e(\bar{M}, j)$, where $\bar{M}$ is a uniformly sampled matching from $\mathcal{M}$. In other words, we could treat e as a base arm in the CPE-MAB setting with mean reward $w(e)$, and we could obtain an unbiased sample for e if we can uniformly sample $\bar{M}$ from $\mathcal{M}$ and then play the duel $(e, e(\bar{M}, j))$ to observe the outcome. However, a naive sampling method on $\mathcal{M}$ would take exponential time. To resolve this issue, we employ a fully-polynomial almost uniform sampler for perfect matchings (Jerrum et al., 2004) $\mathcal{S}(\eta)$ to obtain an almost uniformly sampled matching M' from $\mathcal{M}$. Below we give the formal definition of $\mathcal{S}(\eta)$.

Definition 1. *An almost uniform sampler for perfect matchings is a randomized algorithm $\mathcal{S}(\eta)$ that, if given any bipartite graph G and bias parameter η, it returns a random perfect matching from a distribution π' that satisfies*

$$d_{\text{tv}}(\pi', \pi) = \frac{1}{2} \sum_{x \in \Theta} |\pi'(x) - \pi(x)| \leq \eta,$$

where d_{tv} is the total variation, Θ is the set of all perfect matchings in G and π is the uniform distribution on Θ.

Next, we show how to obtain M' using $\mathcal{S}(\eta)$. We add some ficticious vertices in S and ficticious edges in E to construct a new bipartite graph $G'(C, S', E')$ where $|C| = |S'|$. There is a one-to-n relationship between a maximum matchings in G and a perfect matchings in G'. Then, with $\mathcal{S}(\eta)$, we can almost uniformly sample a maximum matching M' from G in fully-polynomial time. We defer the details for sampling with $\mathcal{S}(\eta)$ to the supplementary material.

3.2. Efficient PAC Pure Exploration Algorithm

In the previous subsection, we present a reduction of CPE-DB for Borda winner to the conventional CPE-MAB (Chen et al., 2014) problem. However, directly applying the existing CLUCB algorithm in (Chen et al., 2014) cannot obtain an efficient algorithm for our problem. The main obstacle is that there is currently no efficient algorithm to sample from an exact uniform distribution over all the maximum matchings in a general bipartite graph, and thus the original CLUCB algorithm is not directly applicable. To tackle this problem, we need to use an approximate sampler and modify the original CLUCB algorithm to handle the bias introduced by the approximate sampler.

Algorithm 1 illustrates an efficient PAC pure exploration algorithm CLUCB-Borda-PAC for the Borda winner case. Given a confidence level δ and an accuracy requirement ε, CLUCB-Borda-PAC returns an approximate Borda winner Out such that $B(\text{Out}) \geq B(M_*^B) - \varepsilon$ with probability at least $1 - \delta$.

CLUCB-Borda-PAC is built on the CLUCB (Chen et al., 2014) algorithm designed for the conventional CPE-MAB problem, and CLUCB-Borda-PAC efficiently transforms the original numerical observations to the equivalent relative observations. In particular, the maximization oracle $\mathsf{MWMC}(\cdot)$ called in CLUCB-Borda-PAC is exactly the maximum-weighted maximum-cardinality matching algorithm, performed in fully-polynomial time. The main structure follows the CLUCB algorithm: in each round, we first use the empirical mean $\bar{\boldsymbol{w}}_t$ as the input to the oracle $\mathsf{MWMC}(\cdot)$ to find a matching M_t. Then we use the lower confidence bounds for all edges in M_t and upper confidence bounds for all edges outside M_t as the input and call $\mathsf{MWMC}(\cdot)$ again to find an adjusted matching $\tilde{M}_t$. If the difference in weights of the adjusted and non-adjusted matchings are small (line 15), the algorithm stops and returns M_t as the final matching. If not, the algorithm finds the edge z_t in the symmetric difference of $\tilde{M}_t$ and M_t. Then, the algorithm samples a matching M' using sampler $\mathcal{S}(\eta)$, and plays a duel between z_t and the corresponding edge in M' with the same position as z_t. After playing the duel, the algorithm observes the result and updates empirical mean $\bar{w}_{t+1}(z_t)$. With the fast maximization oracle $\mathsf{MWMC}(\cdot)$ and sampler $\mathcal{S}(\eta)$, the CLUCB-Borda-PAC algorithm can be performed in fully-polynomial time.

To formally state the sample complexity upper bound of the CLUCB-Borda-PAC algorithm, we need to first define the width of G, the Borda gap and the Borda hardness.

Definition 2 (Width)**.** *For a bipartite graph G, let $\mathcal{M}(G)$ denote the set of all its maximum matchings. For any $M_1, M_2 \in \mathcal{M}(G)$ such that $M_1 \neq M_2$, we define* $\mathrm{width}(M_1, M_2)$ *as the number of edges of the maximum connected component in their union graph. Then, we define the width of bipartite graph G as*

$$\mathrm{width}(G) = \max_{\substack{M_1, M_2 \in \mathcal{M}(G) \\ M_1 \neq M_2}} \mathrm{width}(M_1, M_2).$$

This width definition for bipartite maximum matching is inline with the general width definition in (Chen et al., 2014).

Definition 3 (Borda gap)**.** *We define the Borda gap Δ_e^B for any edge $e \in E$ as*

$$\Delta_e^B = \begin{cases} w(M_*^B) - \max\limits_{M \in \mathcal{M}: e \in M} w(M) & \text{if } e \notin M_*, \\ w(M_*^B) - \max\limits_{M \in \mathcal{M}: e \notin M} w(M) & \text{if } e \in M_*, \end{cases}$$

where we make the convention that the maximum value of an empty set is $-\infty$.

Definition 4 (Borda hardness)**.** *We define the hardness H^B for identifying Borda winner in CPE-DB as*

$$H^B := \sum_{e \in E} \frac{1}{(\Delta_e^B)^2}.$$

Algorithm 1 CLUCB-Borda-PAC

1: **Input:** confidence δ, accuracy ε, bipartite graph G, maximization oracle $\mathsf{MWMC}(\cdot)$: $\mathbb{R}^m \to \mathcal{M}$ and almost uniform sampler for perfect matchings $\mathcal{S}(\eta)$
2: Set bias parameter $\eta \leftarrow \frac{1}{8}\varepsilon$
3: Initialize $T_1(e) \leftarrow 0$ and $\bar{w}_1(e) \leftarrow 0$ for all $e \in E$
4: **for** $t = 1, 2, \ldots$ **do**
5: $\quad M_t \leftarrow \mathsf{MWMC}(\bar{\boldsymbol{w}}_t)$
6: $\quad$ Compute confidence radius $c_t(e) \leftarrow \sqrt{\frac{\ln(\frac{4Kt^3}{\delta})}{2T_t(e)}}$ for all $e \in E$ // $\frac{x}{0} := 1$ for any x
7: $\quad$ **for** all $e \in E$ **do**
8: $\quad\quad$ **if** $e \in M_t$ **then**
9: $\quad\quad\quad \tilde{w}_t(e) \leftarrow \bar{w}_t(e) - c_t(e) - \frac{1}{4}\varepsilon$
10: $\quad\quad$ **else**
11: $\quad\quad\quad \tilde{w}_t(e) \leftarrow \bar{w}_t(e) + c_t(e) + \frac{1}{4}\varepsilon$
12: $\quad\quad$ **end if** // $\bar{w}_t(e) := 0$ if $T_t(e) = 0$
13: $\quad$ **end for**
14: $\quad \tilde{M}_t \leftarrow \mathsf{MWMC}(\tilde{\boldsymbol{w}}_t)$
15: $\quad$ **if** $\tilde{w}_t(\tilde{M}_t) - \tilde{w}_t(M_t) \leq \ell\varepsilon$ **then**
16: $\quad\quad \mathsf{Out} \leftarrow M_t$
17: $\quad\quad$ **return** Out
18: $\quad$ **end if**
19: $\quad z_t \leftarrow \arg\max_{e \in (\tilde{M}_t \setminus M_t) \cup (M_t \setminus \tilde{M}_t)} c_t(e)$
20: $\quad$ Sample a matching M' from $\mathcal{M}$ using $\mathcal{S}(\eta)$
21: $\quad$ Pull the duel (z_t, e'), where $e' = e(M', s(z_t))$
22: $\quad$ Update $\bar{w}_{t+1}(z_t) \leftarrow \frac{\bar{w}_t(z_t) \cdot T_t(z_t) + X_t(z_t)}{T_t(z_t)+1}$ where $X_t(z_t)$ takes value 1 if z_t wins, 0 otherwise, and $T_{t+1}(z_t) \leftarrow T_t(z_t) + 1$
23: **end for**

The Borda gap and Borda hardness definitions are naturally inherited from those in (Chen et al., 2014). For each edge $e \notin M_*^B$, the Borda gap Δ_e^B is the sub-optimality of the best matching that includes edge e, while for each edge $e \in M_*^B$ the Borda gap Δ_e^B is the sub-optimality of the best matching that does not include edge e. The Borda hardness H^B is the sum of inverse squared Borda gaps, which represents the problem hardness for identifying the Borda winner.

Now we present a problem-dependent upper bound of the sample complexity for the CLUCB-Borda-PAC algorithm.

Theorem 1 (CLUCB-Borda-PAC)**.** *With probability at least $1 - \delta$, the* CLUCB-Borda-PAC *algorithm (Algorithm 1) returns an approximate Borda winner* Out *such that $B(\mathsf{Out}) \geq B(M_*^B) - \varepsilon$ with sample complexity*

$$O\left(H_\varepsilon^B \ln\left(\frac{H_\varepsilon^B}{\delta}\right)\right),$$

where $H_\varepsilon^B := \sum_{e \in E} \min\left\{\frac{\mathrm{width}(G)^2}{(\Delta_e^B)^2}, \frac{1}{\varepsilon^2}\right\}$.

We can see that when the accuracy parameter ε is small enough, H_ε^B coincides with the hardness metric H^B.

3.3. Efficient Exact Pure Exploration Algorithm

Based on the PAC algorithm CLUCB-Borda-PAC, we further design an efficient exact pure exploration algorithm CLUCB-Borda-Exact for Borda winner and analyze its sample complexity upper bound. Generally speaking, CLUCB-Borda-Exact performs CLUCB-Borda-PAC as a sub-procedure, and guesses the smallest Borda gap $\Delta^B_{\min} := \min_{e\in E} \Delta^B_e$. Iterating epoch $q = 1, 2, \ldots$, we set accuracy $\varepsilon_q = \frac{1}{2^q}$ and confidence $\delta_q = \frac{\delta}{2q^2}$. CLUCB-Borda-Exact will guess $\Delta^B_{\min} > \ell\varepsilon_q$, and call CLUCB-Borda-PAC as a sub-procedure with parameters ε_q, δ_q. If the adjusted matching $\tilde{M}_t$ has exactly the same weight as the non-adjusted matching M_t ($\tilde{w}_t(\tilde{M}_t) = \tilde{w}_t(M_t)$, similar as in line 15 of Algorithm 1), then the algorithm stops and returns M_t as the final matching. If $\tilde{w}_t(\tilde{M}_t) \neq \tilde{w}_t(M_t)$ but they differ within $\ell\varepsilon_q$, then the current epoch stops and CLUCB-Borda-Exact will enter the next epoch and cut the guess in half ($\varepsilon_{q+1} = \varepsilon_q/2$). (See the supplementary materials for more details.) Using this technique, we can obtain an algorithm to identify the exact Borda winner with a loss of logarithmic factors in its sample complexity upper bound.

Below we present a problem-dependent upper bound of the sample complexity for the CLUCB-Borda-Exact algorithm and defer the detailed algorithm and proof to the supplementary material.

Theorem 2 (CLUCB-Borda-Exact)**.** *With probability at least $1-\delta$, the* CLUCB-Borda-Exact *algorithm returns the Borda winner with sample complexity*

$$O\left(\text{width}(G)^2 H^B \cdot \ln\left(\frac{\ell}{\Delta^B_{\min}}\right) \cdot \left(\ln\left(\frac{\text{width}(G) H^B}{\delta}\right) + \ln\ln\left(\frac{\ell}{\Delta^B_{\min}}\right)\right)\right),$$

where $\Delta^B_{\min} := \min_{e\in E} \Delta^B_e$.

3.4. Lower Bound

To formally state our result for lower bound, we first introduce the definition of δ-correct algorithm as follows. For any $\delta \in (0,1)$, we call an algorithm $\mathbb{A}$ a δ-correct algorithm if, for any problem instance of CPE-DB with Borda winner, algorithm $\mathbb{A}$ identifies the Borda winner with probability at least $1-\delta$.

Now we give a problem-dependent lower bound on the sample complexity for CPE-DB with Borda winner.

Theorem 3 (Borda lower bound)**.** *Consider the problem of combinatorial pure exploration for identifying the Borda winner. Suppose that, for some constant $\gamma \in (0, \frac{1}{4})$, $\frac{1}{2} - \gamma \le p_{e_i,e_j} \le \frac{1}{2} + \gamma$, $\forall e_i, e_j \in E$ and $\frac{|\mathcal{M}|}{|\mathcal{M}|-|\mathcal{M}_e|} \le \frac{1-4\gamma}{4\gamma\ell}$, $\forall e \in E$. Then, for any $\delta \in (0, 0.1)$, any δ-correct algorithm has sample complexity $\Omega\left(H^B \ln\left(\frac{1}{\delta}\right)\right)$, where $\mathcal{M}_e := \{M \in \mathcal{M} : e \in M\}$.*

We defer the detailed proof of Theorem 3 to the supplementary material.

From the upper bounds (Theorems 1,2) and lower bound (Theorem 3), we see that when ignoring the logarithmic factors, our algorithms are tight on the hardness metric H^B. However, whether the width(G) factor is tight or not remains unclear and we leave it for future investigation.

4. Efficient Exploration for Condorcet Winner

In this section, we introduce the efficient pure exploration algorithm CAR-Cond to find a Condorcet winner. We first introduce the efficient pure exploration part assuming there exists "an oracle" that performs like a black-box, and we show the correctness and the sample complexity of CAR-Cond given the oracle. Next, we present the details of the oracle and show that the time complexity of the oracle is polynomial. Then, we apply the verification technique (Karnin, 2016) to improve our sample complexity further. Finally, we give the sample complexity lower bound for finding the Condorcet winner.

4.1. Efficient Pure Exploration Algorithm: CAR-Cond

We first introduce our algorithm CAR-Cond for CPE-DB for the Condorcet winner assuming that there is a proper "oracle". Note that finding the Condorcet winner if existed is equivalent to the following optimization problem,

$$\max_{x=\chi_{M_1}} \min_{y=\chi_{M_2}} \frac{1}{\ell} x^T P y,$$

where $M_1, M_2 \in \mathcal{M}$ are feasible matchings and the value is optimal when $x = y = \chi_{M^C_*}$. This is because if M_1 is not the Condorcet winner M^C_*, it will lose to M^C_* with score $\chi^T_{M_1} P \chi_{M^C_*} < 1/2$, and only when $x = \chi_{M^C_*}$, $\min_{y=\chi_{M_2}} \frac{1}{\ell} x^T P y$ reaches $1/2$ when $y = \chi_{M^C_*}$. However, the optimization problem is "discrete" and we first use the continuous relaxation technique to solve the following optimization problem

$$\max_{x\in\mathcal{P}(\mathcal{M})} \min_{y\in\mathcal{P}(\mathcal{M})} \frac{1}{\ell} x^T P y, \tag{5}$$

where $\mathcal{P}(\mathcal{M}) = \{\sum_i \lambda_i \chi_{M_i} : M_i \in \mathcal{M}, \sum_i \lambda_i = 1, \lambda_i \ge 0\}$ is the convex hull of the vectors $\chi_M, M \in \mathcal{M}$. There is an algorithm that can solve x, y approximately in polynomial time, but solving the optimization problem of Eq. (5) is not enough for our CPE-DB problem. Therefore, we need the following more powerful oracle.

We assume that there is an oracle O_ε that takes the inputs $\varepsilon, A_1, R_1, A_2, R_2, Q$, where ε is the error of the oracle, $A_1, R_1, A_2, R_2 \subset E$ and $Q \in [0,1]^{m\times m}$. The oracle can approximately solve the following optimization

$$\max_{x\in\mathcal{P}(\mathcal{M},A_1,R_1)} \min_{y\in\mathcal{P}(\mathcal{M},A_2,R_2)} \frac{1}{\ell} x^T Q y, \tag{6}$$

where $\mathcal{P}(\mathcal{M},A,R) = \{\sum_i \lambda_i \chi_{M_i} : M_i \in \mathcal{M}, A \subset M_i, R \subset (M_i)^c, \sum_i \lambda_i = 1, \lambda_i \geq 0\}$ is the convex hull of the vector representations of the matchings, such that all edges in A are included in the matching and none of the edges in R is included in the matching. More specifically, we assume that the oracle O_ε will compute a solution x_0 that satisfies both the constraint and the following guarantee:

$$\min_{y\in\mathcal{P}(\mathcal{M},A_2,R_2)} \frac{1}{\ell} x_0^T Q y \geq \max_{x\in\mathcal{P}(\mathcal{M},A_1,R_1)} \min_{y\in\mathcal{P}(\mathcal{M},A_2,R_2)} \frac{1}{\ell} x^T Q y - \varepsilon.$$

In the algorithm, we only require that the oracle O_ε returns the value $\min_{y\in\mathcal{P}(\mathcal{M},A_2,R_2)} \frac{1}{\ell} x_0^T Q y$, not the x_0.

Given the oracle O_ε, the high level idea of CAR-Cond (Algorithm 2) is as follows: If we know how to set the approximation parameter properly, then in every round we partition the edge set E into A, R, and U, where A is the set of the edges that should be included in the Condorcet winner, R is the set of edges that should be excluded, and U are the remaining undecided edges. In each round, we only sample the duel between two comparable edges in the set U (Line 6). Then, we use the upper and lower confidence bounds to estimate the real preference matrix P (Line 7). After that, for every undecided edge e, we enforce it to be included in the optimal solution or to be excluded in the solution, and use the oracle to see if the included and excluded cases vary much. If so, we classify edge e into A or R in the next round (Line 9). Since we do not know how to set the approximation parameter properly, we use the "doubling trick" to shrink the approximation parameter ε_q by a factor of 2 in each epoch q (Line 4).

For the value of the confidence radius and the upper and lower confidence bound for the matrix P, we use the following quantity for the confidence radius of the winning probability of the duel between any two comparable arms.

$$c_t(e_i,e_j) = \sqrt{\frac{\ln(4Kt^3/\delta)}{2T_t(e_i,e_j)}}, \tag{7}$$

where $T_t(e_i,e_j)$ is the number of duels between two comparable arms e_i, e_j at the beginning of round t. Now given some duels (at least one) between e_i, e_j, we define $\hat{p}_t(e_i,e_j)$ as the empirical winning probability of e_i over e_j *up to* round t's exploration phase, and we define

$$\bar{p}_t(e_i,e_j) := \min\{1, \hat{p}_t(e_i,e_j) + c_t(e_i,e_j)\}, \tag{8}$$

Algorithm 2 CAR-Cond

1: **Input:** Bipartite graph G, Oracle O_ε with accuracy ε
2: $A_0 \leftarrow \phi, R_0 \leftarrow \phi, U_0 \leftarrow E, e_0 = 0$.
3: **for** $q = 1, 2, \ldots$ **do**
4: $\quad \varepsilon_q \leftarrow \frac{1}{2^q}, e_q \leftarrow \frac{1}{\varepsilon_q^2}$
5: $\quad$ **for** $t = e_{q-1}+1, e_{q-1}+2, \ldots, e_q$ **do**
6: $\quad\quad$ For every $e_1 \neq e_2$ and $e_1, e_2 \in E_j$ for some j and $e_1, e_2 \in U_{t-1}$, sample duel between e_1, e_2
7: $\quad\quad$ Compute $\bar{P}_t, \underline{P}_t$
8: $\quad\quad$ $A_t \leftarrow A_{t-1}, R_t \leftarrow R_{t-1}, U_t \leftarrow U_{t-1}$
9: $\quad\quad$ **for** $e \in U_{t-1}$ **do**
10: $\quad\quad\quad$ // We use A, R as shorthands for A_{t-1}, R_{t-1}
11: $\quad\quad\quad$ $\mathrm{InU} = \mathsf{O}_{\varepsilon_q}(A\cup\{e\}, R, A, R, \bar{P}_t)$
12: $\quad\quad\quad$ $\mathrm{InL} = \mathsf{O}_{\varepsilon_q}(A\cup\{e\}, R, A, R, \underline{P}_t)$
13: $\quad\quad\quad$ $\mathrm{ExU} = \mathsf{O}_{\varepsilon_q}(A, R\cup\{e\}, A, R, \bar{P}_t)$
14: $\quad\quad\quad$ $\mathrm{ExL} = \mathsf{O}_{\varepsilon_q}(A, R\cup\{e\}, A, R, \underline{P}_t)$
15: $\quad\quad\quad$ **if** $\mathrm{InL} > \mathrm{ExU} + \varepsilon_q$ **then**
16: $\quad\quad\quad\quad$ $A_t \leftarrow A_t \cup \{e\}, U_t \leftarrow U_t \setminus \{e\}$
17: $\quad\quad\quad$ **else if** $\mathrm{ExL} > \mathrm{InU} + \varepsilon_q$ **then**
18: $\quad\quad\quad\quad$ $R_t \leftarrow R_{t-1} \cup \{e\}, U_t \leftarrow U_t \setminus \{e\}$
19: $\quad\quad\quad$ **end if**
20: $\quad\quad\quad$ **if** $|A_t| = \ell$ **then** $\mathsf{Out} \leftarrow A$, **return** Out
21: $\quad\quad$ **end for**
22: $\quad$ **end for**
23: **end for**

$$\underline{p}_t(e_i,e_j) := \max\{0, \hat{p}_t(e_i,e_j) - c_t(e_i,e_j)\}.$$

$\bar{p}_t(e_i,e_j)$ and $\underline{p}_t(e_i,e_j)$ can be interpreted as the upper and lower confidence bounds of the winning probability of e_1 over e_2. Then we denote $\bar{P}_t$ as the matrix where $\bar{P}_{t,ij} := \bar{p}_t(e_i,e_j)$ where i,j are edge indices, $\bar{P}_{t,ii} := 0.5$, and $\bar{P}_{t,ij} = 0$ for any 2 incomparable indices. Similarly, we define $\underline{P}_t$ as the matrix where $\underline{P}_{t,ij} := \underline{p}_t(e_i,e_j)$, $\underline{P}_{t,ii} := 0.5$, and $\underline{P}_{t,ij} = 0$ for any two incomparable indices.

Sample complexity for CAR-Cond To present our main result on the sample complexity of CAR-Cond, we need to first introduce the notion of *gap* for each edge and each comparable pair under the Condorcet setting.

Definition 5 (Condorcet gap)**.** *We define the Condorcet gap Δ_e^C of an edge e as the following quantity.*

$$\Delta_e^C = \begin{cases} 1/2 - \max_{\chi_M, e\in M} \frac{1}{\ell}\chi_M^T P \chi_{M_*^C}, & \text{if } e \notin M_*^C \\ 1/2 - \max_{\chi_M, e\notin M} \frac{1}{\ell}\chi_M^T P \chi_{M_*^C}, & \text{if } e \in M_*^C \end{cases}$$

Then we define the gap $\Delta_{e,e'}^C$ for a pair of arms $e \neq e'$ and $e, e' \in E_j$ as the following quantity $\Delta_{e,e'}^C = \max\{\Delta_e^C, \Delta_{e'}^C\}$.

The definition of gap is very similar to the gap defined in (Chen et al., 2014). Intuitively speaking, the definition of

the gap of each edge e is a measurement of how easily e will be classified into the accepted set A or the rejected set R. Given the definition of the gap, we have the following main theorem for the Condorcet setting.

Theorem 4 (CAR-Cond). *With probability at least $1-\delta$, algorithm* CAR-Cond *returns the correct Condorcet winner with a sample complexity bounded by*

$$O\left(\sum_{j=1}^{\ell}\sum_{e\neq e',e,e'\in E_j}\frac{1}{(\Delta^C_{e,e'})^2}\ln\left(\frac{K}{\delta(\Delta^C_{e,e'})^2}\right)\right).$$

Generally speaking, our algorithm sequentially classifies each edge into M^C_* or $(M^C_*)^c$. The definition of the gap shows the sub-optimality of wrongly classifying each edge, and $\frac{1}{(\Delta^C_e)^2}$ is roughly the number of times to correctly classify the edge e. Because each query is a sample between two edges e, e', the number of query between e, e' is roughly $1/(\Delta^C_{e,e'})^2$, this is so as when we correctly classify an edge, we will not need to query any pair that contains this edge. Summing over all comparable pairs of edges, we get our upper bound when omitting all logarithm terms.

When there is only one position, our problem reduces to the original dueling bandit problem. In special cases when the Condorcet winner beat every arm with the largest margin (formally, for all arm $i \in [m]$, $i^C = \arg\max_{j\in[m]}\Pr\{j \text{ wins } i\}$), our sample complexity bound is at the same order as the state-of-the-art (Karnin, 2016) when omitting the logarithmic terms.

4.2. Implementation of Oracle

In this part, we present the high level idea of our method to solve the optimization problem (Eq. (6)). If we define

$$g(x) = \min_{y\in\mathcal{P}(\mathcal{M},A_2,R_2)}\frac{1}{\ell}x^TQy,$$

then g is concave in x, since x^TQy is linear in x and the minimum of linear functions is a concave function. Also note that the constraint set $\mathcal{P}(\mathcal{M}, A_2, R_2)$ is a convex set since it is defined as the convex hull of the vector representations. Thus, using the projected sub-gradient ascent method, we can solve the optimization problem by an error of ε in $O(\frac{1}{\varepsilon})$ number of iterations. To do so, we need to address two problems: how to compute the gradient at a given point, and how to compute the projection efficiently.

The first problem is rather easy to solve, because if we want to compute the sub-gradient at a given point x_0, it suffices to compute the parameter $y_0 = \arg\min_{y\in\mathcal{P}(\mathcal{M},A_2,R_2)} x_0^TQy$, and the sub-gradient will be $\frac{1}{\ell}Qy_0 \in \partial_x g(x_0)$. Computing the parameter y_0 can be done in polynomial time, since the minimum cost maximum matching can be solved in polynomial time.

Algorithm 3 CAR-Parallel

Input: confidence $\delta < 0.01$, algorithm CAR-Verify
Define CAR-Verify$_k$, $k \in \mathbb{N}$ as the CAR-Verify algorithm with confidence $\frac{\delta}{2^{k+1}}$
Simulate $\{\text{CAR-Verify}_k\}_{k\in\mathbb{N}}$ in parallel
for $t = 1, 2, \ldots$ **do**
  for each $k \in \mathbb{N}$ $s.t.$ $t \bmod 2^k = 0$ **do**
    Start or resume CAR-Verify$_k$, allowing only one sample, and then suspend CAR-Verify$_k$
    if CAR-Verify$_k$ returns an answer Out$_k$ **then**
      Out $\leftarrow$ Out$_k$
      return Out
    end if
  end for
end for

The second problem is the main challenge. Note that there may be an exponentially large number of vertices in the polytope $\mathcal{P}(\mathcal{M}, A_2, R_2)$ because the number of feasible matchings may be exponential, and we cannot solve the projection step in general. However, if we can tolerate some error in the projection step, we may solve the approximate projection in polynomial time by the Frank-Wolfe algorithm. Then, we can set the approximate projection error to be relatively small, so the cumulative error due to the projection can also be bounded. In this way, we can solve the optimization problem Eq. (6) with $poly(1/\varepsilon, m, K, \ell)$ time complexity.

Please see the supplementary material for more backgrounds on convex optimization and the detailed implementation of the oracle.

4.3. Further Improvements through Verification

Based on the CAR-Cond algorithm, we further design an algorithm CAR-Parallel for identifying Condorcet winner, which uses the parallel simulation technique (Chen & Li, 2015; Chen et al., 2017) and achieves a tighter expected sample complexity for small confidence. CAR-Parallel calls a variant of CAR-Cond, named CAR-Verify, which applies the verification technique (Karnin, 2016) to improve the sample complexity of the original CAR-Cond. Specifically, CAR-Verify calls CAR-Cond(δ_0) to obtain a hypothesized Condorcet winner $\hat{M}$ using a constant confidence $\delta_0 > \delta$. Then, CAR-Verify verifies the correctness of $\hat{M}$ using confidence δ. While CAR-Verify loses a part of confidence in order to obtain better sample complexity for small confidence, CAR-Parallel boosts the confidence to δ by simulating a sequence of CAR-Verify in parallel and keeps the obtained better sample complexity in expectation.

Algorithm 3 illustrates the detailed algorithm CAR-Parallel that applies the parallel simulation technique (Chen & Li,

Algorithm 4 CAR-Verify

Input: confidence $\delta < 0.01$, algorithm CAR-Cond
$\delta_0 \leftarrow 0.01$
$\hat{M} = \text{CAR-Cond}(\delta_0)$
for $t = 1, 2, \ldots$ **do**
  Compute $\bar{P}_t, \underline{P}_t$
  if $\max_{M \in \mathcal{M} \setminus \{\hat{M}\}} f(M, \hat{M}, \underline{P}_t) \geq \frac{1}{2}$ **then**
    return error
  end if
  $M_t = \operatorname{argmax}_{M \in \mathcal{M} \setminus \{\hat{M}\}} f(M, \hat{M}, \bar{P}_t)$
  if $f(M_t, \hat{M}, \bar{P}_t) \leq \frac{1}{2}$ **then**
    Out $\leftarrow \hat{M}$
    return Out
  else
    $(e_t, f_t) \leftarrow \operatorname{argmax}_{\substack{e_t \in M_t \setminus \hat{M}, f_t \in \hat{M} \setminus M_t \\ s(e_t) = s(f_t)}} c_t(e_t, f_t)$
    Pull the duel (e_t, f_t) and update empirical means
  end if
end for

2015; Chen et al., 2017) and achieves a tighter expected sample complexity for small confidence. Algorithm 4 illustrates the sub-procedure CAR-Verify called in CAR-Parallel. CAR-Verify is based on the original algorithm CAR-Cond and employs the verification technique to improve the sample complexity for small confidence.

In order to formally state our result for the CAR-Parallel algorithm, we first introduce the following definitions.

For any $e \notin M_*^C$, we define the verification gap $\tilde{\Delta}_e^C$ as

$$\min_{M \in \mathcal{M} \setminus \{M_*^C\} : e \in M} \left\{ \frac{\ell}{d_{M_*^C, M}} \cdot \left(\frac{1}{2} - \frac{1}{\ell} \chi_M^T P \chi_{M_*^C} \right) \right\},$$

where d_{M_x, M_y} denotes the number of positions with different edges between M_x and M_y, *i.e.*, $d_{M_x, M_y} := \sum_{j=1}^{\ell} \mathbb{I}\{e(M_x, j) \neq e(M_y, j)\}$.

For ease of notation, we define the following quantity

$$H_{\text{ver}}^C := \sum_{e \notin M_*^C} \frac{1}{(\tilde{\Delta}_e^C)^2}.$$

Then, we have the main theorem of the sample complexity of algorithm CAR-Parallel.

Theorem 5 (CAR-Parallel)**.** *Assume the existence of Condorcet winner. Then, given $\delta < 0.01$, with probability at least $1 - \delta$, the* CAR-Parallel *algorithm (Algorithm 3) will return the Condorcet winner with an expected sample complexity*

$$O\left(\sum_{j=1}^{\ell} \sum_{\substack{e \neq e' \\ e, e' \in E_j}} \frac{\ln\left(K/(\Delta_{e,e'}^C)^2\right)}{(\Delta_{e,e'}^C)^2} + H_{\text{ver}}^C \ln\left(\frac{H_{\text{ver}}^C}{\delta}\right) \right).$$

To the best of our knowledge, the best sample complexity for pure exploration of Condorcet dueling bandit is $O(n^2/\Delta^2 + n/\Delta^2 \log(1/\delta))$ by (Karnin, 2016) using the verification technique. When reducing our setting to the simple Condorcet dueling bandit ($\ell = 1$), Theorem 5 recovers this result.

We defer the detailed results and proofs to the supplementary material.

5. Conclusion and Future Work

In this paper, we formulate the combinatorial pure exploration for dueling bandit (CPE-DB) problem. We consider two optimality metrics, Borda winner and Condorcet winner. For Borda winner, we first reduce the problem to CPE-MAB, and then propose efficient PAC and exact algorithms. We provide sample complexity upper and lower bounds for these algorithms. For a subclass of problems the upper bound of the exact algorithm matches the lower bound when ignoring the logarithmic factor. For Condorcet winner, we first design an FPTAS for a properly extended offline problem, and then employ this FPTAS to design a novel online algorithm CAR-Cond. To our best knowledge, CAR-Cond is the first algorithm with polynomial running time per round for identifying the Condorcet winner in CPE-DB.

There are several promising directions worth further investigation for CPE-DB. One direction is to improve the sample complexity of the CAR-Cond algorithm without compromising its computational efficiency, and try to find a lower bound in this case that matches the upper bound. Other directions of interest include studying a more general CPE-DB model than the current candidate-position matching version, or a family of practical preference functions $f(M_1, M_2, P)$ other than linear functions.

Acknowledgement

The work of Yihan Du and Longbo Huang is supported in part by the National Natural Science Foundation of China Grant 61672316, the Zhongguancun Haihua Institute for Frontier Information Technology and the Turing AI Institute of Nanjing.

References

Agrawal, N. and Chaporkar, P. Klucb approach to copeland bandits. *arXiv preprint arXiv:1902.02778*, 2019.

Agrawal, S. and Goyal, N. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pp. 39–1, 2012.

Alwin, D. F. and Krosnick, J. A. The measurement of values

in surveys: A comparison of ratings and rankings. *Public Opinion Quarterly*, 49(4):535–552, 1985.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

Ben-Akiva, M., Bradley, M., Morikawa, T., Benjamin, J., Novak, T., Oppewal, H., and Rao, V. Combining revealed and stated preferences data. *Marketing Letters*, 5(4):335–349, 1994.

Black, D. On the rationale of group decision-making. *Journal of Political Economy*, 56(1):23–34, 1948.

Bubeck, S., Wang, T., and Viswanathan, N. Multiple identifications in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 258–265, 2013.

Chen, B. and Frazier, P. I. Dueling bandits with weak regret. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 731–739. JMLR. org, 2017.

Chen, L. and Li, J. On the optimal sample complexity for best arm identification. *arXiv preprint arXiv:1511.03774*, 2015.

Chen, L. and Li, J. Open problem: Best arm identification: Almost instance-wise optimality and the gap entropy conjecture. In *Conference on Learning Theory*, pp. 1643–1646, 2016.

Chen, L., Gupta, A., Li, J., Qiao, M., and Wang, R. Nearly optimal sampling algorithms for combinatorial pure exploration. In *Conference on Learning Theory*, pp. 482–534, 2017.

Chen, S., Lin, T., King, I., Lyu, M. R., and Chen, W. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems*, pp. 379–387, 2014.

Copeland, A. H. A reasonable social welfare function. Technical report, mimeo, 1951. University of Michigan, 1951.

Emerson, P. The original borda count and partial voting. *Social Choice and Welfare*, 40(2):353–358, 2013.

Emerson, P. *From Majority Rule to Inclusive Politics*. Springer, 2016.

Even-Dar, E., Mannor, S., and Mansour, Y. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7(Jun):1079–1105, 2006.

Gabillon, V., Lazaric, A., Ghavamzadeh, M., Ortner, R., and Bartlett, P. Improved learning complexity in combinatorial pure exploration bandits. In *Artificial Intelligence and Statistics*, pp. 1004–1012, 2016.

Gehrlein, W. V. The condorcet criterion and committee selection. *Mathematical Social Sciences*, 10(3):199–209, 1985.

Graepel, T. and Herbrich, R. Ranking and matchmaking. *Game Developer Magazine*, 25:34, 2006.

Jamieson, K., Katariya, S., Deshpande, A., and Nowak, R. Sparse dueling bandits. In *Artificial Intelligence and Statistics*, pp. 416–424, 2015.

Jerrum, M., Sinclair, A., and Vigoda, E. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM (JACM)*, 51(4):671–697, 2004.

Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Forum*, volume 51, pp. 4–11. Acm New York, NY, USA, 2017.

Kalyanakrishnan, S., Tewari, A., Auer, P., and Stone, P. Pac subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning*, volume 12, pp. 655–662, 2012.

Karnin, Z. S. Verification based solution for structured mab problems. In *Advances in Neural Information Processing Systems*, pp. 145–153, 2016.

Komiyama, J., Honda, J., Kashima, H., and Nakagawa, H. Regret lower bound and optimal algorithm in dueling bandit problem. In *Conference on Learning Theory*, pp. 1141–1154, 2015.

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

McLean, I. The borda and condorcet principles: three medieval applications. *Social Choice and Welfare*, 7 (2):99–108, 1990.

Radlinski, F., Kurup, M., and Joachims, T. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 43–52, 2008.

Ramamohan, S. Y., Rajkumar, A., and Agarwal, S. Dueling bandits: Beyond condorcet winners to general tournament solutions. In *Advances in Neural Information Processing Systems*, pp. 1253–1261, 2016.

Saari, D. G. and Merlin, V. R. The copeland method. *Economic Theory*, 8(1):51–76, 1996.

Sabato, S. Epsilon-best-arm identification in pay-per-reward multi-armed bandits. In *Advances in Neural Information Processing Systems*, pp. 2876–2886, 2019.

Saha, A. and Gopalan, A. Combinatorial bandits with relative feedback. In *Advances in Neural Information Processing Systems*, pp. 983–993, 2019.

Sui, Y., Zoghi, M., Hofmann, K., and Yue, Y. Advancements in dueling bandits. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 5502–5510, 2018.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Wu, H. and Liu, X. Double thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems*, pp. 649–657, 2016.

Wu, Y., Gyorgy, A., and Szepesvari, C. On identifying good options under combinatorially structured feedback in finite noisy environments. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1283–1291, 2015.

Xu, L., Honda, J., and Sugiyama, M. Dueling bandits with qualitative feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5549–5556, 2019.

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Zoghi, M., Whiteson, S., Munos, R., and De Rijke, M. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proceedings of the 31st International Conference on Machine Learning*, pp. II–10, 2014.