ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-363

How Hard Is Bribery with Distance Restrictions?

Yongjie Yang[1,2] and **Yash Raj Shrestha**[3] and **Jiong Guo**[1]

Abstract. We study the complexity of the bribery problem with distance restrictions. In particular, in the bribery problem, we are given an election and a distinguished candidate p, and are asked whether we can make p win/not win the election by bribing at most k voters to recast their votes. In the bribery problem with distance restrictions, we require that the votes recast by the bribed voters are close to their original votes. To measure the closeness between two votes, we adopt the prevalent Kendall-Tau distance and the Hamming distance. We achieve a wide range of complexity results for this problem under a variety of voting correspondences, including the Borda, Condorcet, Copeland$^{\alpha}$ for every $0 \leq \alpha \leq 1$ and Maximin.

1 Introduction

Voting is a common method for preference aggregation and collective decision-making, and has applications in many areas such as political elections, multi-agent systems, web spam reduction and pattern recognition [14, 15, 32, 37]. In real-world applications, there exist many potential factors that may affect the result of voting. For instance, a strategic individual may alter some of the already submitted votes, or the votes that the voters intend to submit. An example of scenario is when a candidate attempts to change the preferences of voters by running a campaign, or in more extreme cases where this strategy involves paying voters to change their votes, or bribing election officials to get access to already submitted votes in order to modify them. A prominent method to address such issues concerning strategic behavior is to use complexity as a barrier [23, 31, 39]. The key point is that if it is computationally difficult for the strategic individual to figure out how to successfully change the result, he may refrain from attacking the voting.

In this paper, we study a voting model in which an external agent attempts in switching the voters' preferences in order to make a distinguished candidate win the election (*constructive*), or lose the election (*destructive*). The external agent's capacity is bounded by a budget constraint. We observe that, while the voter is willing to recast a new vote persuaded by an external agent, he may nevertheless prefer to submit a preference that deviates as little as possible from his true preference. Indeed, if voting is public, he may be worried that switching his preference completely may harm his reputation, yet he will not be caught out if his final preference is sufficiently similar to his true preference. We call this model *distance restricted bribery*. To quantify the amount of deviation of the new recast vote and the original vote of a bribed voter, we use two *distance* measures. Particularly, we consider what are arguably the most prominent distances on votes, namely, the *Hamming distance* (see, e.g., [8, 16, 34, 35, 40] for discussions of Hamming distance in the context of voting) and *Kendall-Tau distance* (KT-distance for short. See [2, 5, 6, 9] for further discussions). The definitions of these two distances are in Section 2. We study the complexity of the voting model for various voting systems, including the Borda, Condorcet, Maximin and Copeland$^{\alpha}$. We obtain a broad range of results showing that the complexity of bribery depends closely on the settings. A primary conclusion from our results is that the distance restricted bribery problem remains NP-hard for some voting systems even when the distance is bounded by a very small constant. On the other hand, there exist voting systems for which the distance restricted bribery problem is polynomial-time solvable, when the distance is bounded by 1 or 2, and voting systems for which the distance restricted bribery problem is polynomial-time solvable regardless of the values of the distance bound. In particular, we achieve several dichotomy results with respect to the values of the distance bound. For instance, for Condorcet, the constructive restricted bribery problem with KT-distance restriction is polynomial time solvable if the distance is bounded by at most 2; and NP-hard otherwise. See Table 1 for further details on our results. Due to space limitation, several proofs are deferred to a full version of the paper.

Our model is closely related to the bribery problem which has been widely studied in computational social choice. Faliszewski, Hemaspaandra and Hemaspaandra [22] introduced the *bribery* problem, where one is to decide whether a distinguished candidate can become a winner (constructive) or be prevented from being a winner (destructive) by recasting at most $\mathcal{R}$ (a given integer) votes. Clearly, the bribery problem studied in [22] can be considered as a distance restricted bribery problem with the distance bound being considerably large (depends on which distance concept we adopt). The complexity of the bribery problem proposed in [22] has been extensively studied in the literature for various voting systems. In particular, it is known that, for Borda and Condorcet, the constructive bribery problem is NP-hard, while the destructive counterpart turned out to be polynomial-time solvable [11, 24]. For Maximin and Copeland$^{\alpha}$, both the constructive and the destructive bribery problems are NP-hard [24, 26]. Our study clearly complements these complexity results. Of particular interest is that our study shed significant light on the complexity border between polynomial-time solvability and NP-hardness of the bribery problem, with respect to the distance bound. Recently, exploring the complexity border for various strategic voting problems, with respect to diverse structural parameters, has received a considerable amount of attention of researchers from both theoretical computer science and computational social choice communities [10, 18, 25, 43, 44, 45, 46]. The reason is that in many real-world applications, the votes of the voters are subject to some natural combinatorial restrictions.

Our model is also related to some other variants of the bribery problem. In [22], the authors also considered the $bribery version where each voter has a price to change his vote. Later,

[1] Shandong University, Jinan, China, email:jguo@sdu.edu.cn
[2] Saarland University, Germany, email:yyongjie@mmci.uni-saarland.de
[3] ETH Zürich, Zürich, Switzerland, email:yshrestha@ethz.ch

	General		Kendall-Tau distance								Hamming distance		
	Const	Dest	Const (ℓ)				Dest (ℓ)				Const (ℓ)	Dest (ℓ)	
			1	2	3	≥ 4	1	≥ 2	3	≥ 4	2	2	≥ 3
Borda	NP-h$^{\diamondsuit}$	P	P		NP-h (Thm. 2)		P	P	P	P		P	P
Condorcet	NP-h$^{\clubsuit}$	P$^{\clubsuit}$	P	P	NP-h (Thm. 4)		P	P	P	P	NP-h (Thm. 10)	P	P
Copeland$^{\alpha}$ $0 \leq \alpha \leq 1$	NP-h$^{\heartsuit}$	NP-h$^{\heartsuit}$			NP-h (Thm. 5)				NP-h (Thm. 5) UNI: $\ell \geq 5$		NP-h (Thm. 9)		
Maximin	NP-h$^{\clubsuit}$	NP-h$^{\clubsuit}$				NP-h (Thm. 7)	P			NP-h	NP-h (Thm. 11)		

Table 1. A summary of the complexity of the bribery problems. Here, the general case refers to the bribery problem studied in [22]. In the table, "Const" stands for "Constructive" and "Dest" stands for "Destructive". Moreover, "ℓ" is the distance upper bound. Furthermore, "P" stands for "polynomial-time solvable" and "NP-h" stands for "NP-hard". Unless stated otherwise, all results shown in this table apply to both the unique-winner model and the nonunique-winner model. The complexity of the problems whose distance bound ℓ is not shown in the table remains open. The polynomial-time solvability results for the Kendall-Tau distance restriction are from Theorem 1, and the polynomial-time solvability results for the Hamming distance restriction are from Theorem 8. The result marked by $\diamondsuit$ is from [11], by $\clubsuit$ from [24], and by $\heartsuit$ from [26].

Faliszewski [21] proposed a new notion of bribery, called *nonuniform bribery*, where a voter's price may depend on the nature of changes he is asked to implement. A similar notion called *microbribery* was considered in [26]. Elkind, Faliszewski and Slinko [17] introduced the framework of *swap bribery* where the briber can ask a voter to perform a sequence of swaps; each swap changes the relative order of two candidates that are currently adjacent in this voter's preference list. Moreover, each swap may have a different price; and the price of a bribery is the sum of the prices of all swaps that it involves. In the same paper [17], the authors also studied the *shift bribery* problem, which is a restricted variant of swap bribery. In particular, in the shift bribery problem, only swaps involving the distinguished candidate are allowed. Recently, Pini, Rossi and Venable [42] investigated the complexity of bribery in voting with soft constraints, where each candidate is an element of the Cartesian product of the domains of some variables, and voters express their preferences over the candidates via soft constraints. Mattei et al. [38] studied the complexity of bribery in CP-nets.

In addition, our study is related to Obraztsova and Elkind's work [41] where a manipulator aims to make a distinguished candidate win or lose the election by casting an untruthful vote. Here, the untruthful vote should be as close as possible to the truthful vote of the manipulator. They examined this problem for several voting systems with the adoption of three prominent distances, namely, the KT-distance, the footrule distance, and the maximum displacement distance. Our model differs from theirs in the following aspects. First, in our settings, at most $\mathcal{R}$ voters might be bribed, however, they considered only one such voter. Second, their problems ask the manipulator to cast an untruthful vote which is as close as possible to the truthful vote. We mainly focus on the settings where the bribed voters must cast their votes which have a small constant discrepancy from their original votes.

2 Preliminaries

Voting system. A *voting system* can be specified by a set $\mathcal{C}$ of *candidates*, a multiset $\Pi_{\mathcal{V}} = \{\pi_{v_1}, \pi_{v_2}, ..., \pi_{v_n}\}$ of *votes* cast by a corresponding set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ of *voters* (π_{v_i} is cast by v_i), and a *voting correspondence* τ which maps the *election* $\mathcal{E} = (\mathcal{C}, \Pi_{\mathcal{V}}, \mathcal{V})$ to a subset of candidates $\tau(\mathcal{E})$, the *winners*. We often discard $\mathcal{V}$ from the above notation for election $\mathcal{E}$ since $\Pi_{\mathcal{V}}$ is sufficient to determine the winners. If there is only one winner, we call it a *unique winner*; otherwise we call them *co-winners*. Moreover, each vote $\pi_v \in \Pi_{\mathcal{V}}$ is defined as a linear order over the candidates. Throughout this paper, we interchangeably use the terms "vote" and "voter". The linear order of a vote is also called the *preference* of the vote over the candidates. For convenience, we use $\succ_v$ to denote the preference of the vote cast by the voter v. Therefore, for a voter v who prefers the candidate a to b to c, the vote will be written as $\pi_v : a \succ_v b \succ_v c$. In context where $\succ_v$ is clearly known to be whose preference, we drop v from $\succ_v$. We say a is *ranked above* b in a vote π_v if $a \succ_v b$. The *position* of a candidate $a \in \mathcal{C}$ in a vote π_v, denoted as $pos_{\succ_v}(a)$ (or simply $pos_v(a)$), is defined as $|\{b \in \mathcal{C} \mid b \succ_v a\}| + 1$, i.e., the number of candidates ranked above a in the vote plus 1.

For two candidates c and c' in an election $\mathcal{E} = (\mathcal{C}, \Pi_{\mathcal{V}})$, let $N_{\mathcal{E}}(c, c')$ denote the number of votes which prefer c to c'. We drop the index $\mathcal{E}$ when it is clear from context. If $N_{\mathcal{E}}(c, c') > N_{\mathcal{E}}(c', c)$, we say c *beats* c' by $N_{\mathcal{E}}(c, c')$ in $\mathcal{E}$; otherwise if $N_{\mathcal{E}}(c, c') = N_{\mathcal{E}}(c', c)$ we say c *ties* c' in $\mathcal{E}$.

Voting Correspondences. We mainly study the following voting correspondences.

Borda: Every voter gives 0 points to his last-ranked candidate, 1 point to his second-last ranked candidate and so on. A candidate with the highest score is a winner.

Copeland$^{\alpha}$ *($0 \leq \alpha \leq 1$):* For a candidate c, let $B(c)$ be the set of candidates who are beaten by c, and $T(c)$ the set of candidates who tie with c. The Copeland$^{\alpha}$ score of c is then defined as $|B(c)| + \alpha \cdot |T(c)|$. A Copeland$^{\alpha}$ winner is a candidate with the highest score.

Maximin: For a candidate c, the Maximin score of c is defined as $\min_{c' \in \mathcal{C} \setminus \{c\}} N(c, c')$. A Maximin winner is a candidate with the highest Maximin score.

Condorcet: A Condorcet winner is a candidate with Copeland^0 score $m - 1$, and a weak Condorcet winner is a candidate with Copeland^1 score $m - 1$, where m denotes the number of candidates. It is known that Condorcet winner (weak Condorcet winner) may not exist in an election. However, if there is a Condorcet winner in an election, it must be unique. Given an election, the voting correspondence returns all weak Condorcet winners if there is a weak Condorcet winner; otherwise, it returns the empty set.

Distance. We mainly consider the Hamming distance and the KT-distance in this paper. The Hamming distance, named after Richard Hamming, was initially defined on strings [28]. In particular, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. For example, the Hamming distance between the string "a 1 b b" and the string "a b 1 b" is two since there are two positions (the second and the third positions) where the symbols are different. In the context of Hamming distance in this paper, we regard each vote as a string with each element being (the name of) a candidate. For example, the vote defined as $a \succ b \succ c \succ d$ will be considered as the string "a b c d". Hence, the *Hamming distance* between every two votes with preferences $\succ_1, \succ_2$, denoted as $D_{HAM}(\succ_1, \succ_2)$, is the Hamming distance between the two strings corresponding to the two votes.

The KT-distance was coined by Maurice Kendall [33]. In a formal way, the *KT-distance* between two votes with preferences $\succ_1$ and $\succ_2$, respectively, denoted as $D_{KT}(\succ_1, \succ_2)$, is equal to $|\{(a,b)|a,b \in \mathcal{C}, a \succ_1 b \text{ and } b \succ_2 a\}|$. Equivalently, the KT-distance between two votes can be defined as the minimum number of swaps of adjacent candidates needed to transform one into the other [4]. In addition, the KT-distance also turns out to be equal to the number of exchanges needed in a bubble sort (see [1] for an introduction to bubble sort) to convert one full ranking to the other [19]. Due to this fact, the KT-distance is also referred to as *bubble-sort distance* in the literature [7, 12, 19, 20].

Problem Definitions. We mainly study the following problems for different voting correspondences. In what follows, let τ be a voting correspondence and "DIST" a distance function. In this paper, "DIST" can be "KT" for the KT-distance or "HAM" for the Hamming distance. Moreover, ℓ is a positive integer. For two votes with preferences $\succ_1, \succ_2$ and a distance "DIST", we say these two votes are DIST(ℓ)-close if $D_{DIST}(\succ_1, \succ_2) \leq \ell$. For each problem, we study both the *unique-winner model* and the *nonunique-winner model*. In the unique-winner model (resp. nonunique-winner model) for τ not being Condorcet, a candidate $c \in \mathcal{C}$ *wins* an election $\mathcal{E} = (\mathcal{C}, \Pi_\mathcal{V})$ if and only if $\tau(\mathcal{E}) = \{c\}$ (resp. $c \in \tau(\mathcal{E})$), i.e., c is the unique winner (resp. c is either the unique winner or a co-winner). For Condorcet, a candidate c *wins* an election $(\mathcal{C}, \Pi_\mathcal{V})$ in the unique-winner model (resp. nonunique-winner model) if and only if c is the Condorcet winner (resp. a weak Condorcet winner). In the following, let "MOD" be either "UNI" standing for "unique-winner model", or "NON" standing for "nonunique-winner model".

> Constructive Distance Restricted Bribery under τ (C-DIST(ℓ)-τ-MOD)
>
> *Input:* An election $(\mathcal{C}, \Pi_\mathcal{V})$, a distinguished candidate $p \in \mathcal{C}$, and a positive integer $\mathcal{R} \leq |\Pi_\mathcal{V}|$. Here, p does not win the election $(\mathcal{C}, \Pi_\mathcal{V})$ under τ.
>
> *Question*: Is it possible to make p win the election by replacing (recasting) at most $\mathcal{R}$ votes, under τ? Here, a vote can only be replaced with a DIST(ℓ)-close vote.

> Destructive Distance Restricted Bribery under τ (D-DIST(ℓ)-τ-MOD)
>
> *Input:* An election $(\mathcal{C}, \Pi_\mathcal{V})$, a distinguished candidate $p \in \mathcal{C}$, and a positive integer $\mathcal{R} \leq |\Pi_\mathcal{V}|$. Here, p wins $(\mathcal{C}, \Pi_\mathcal{V})$ under τ.
>
> *Question*: Is it possible to prevent p from winning the election by replacing (recasting) at most $\mathcal{R}$ votes, under τ? Here, a vote can only be replaced with a DIST(ℓ)-close vote.

We give either polynomial-time algorithms or NP-hardness reductions for the above problems. Our hardness proofs are based on reductions from the X3C problem and the X4C problem defined as follows. Let ℓ be 3 or 4.

> Exact ℓ-Set Cover (XℓC)
> *Input:* A universal set $U = \{c_1, c_2, ..., c_{\ell \cdot \kappa}\}$ and a collection $S = \{s_1, s_2, ..., s_n\}$ of ℓ-subsets of U.
> *Question:* Is there an $S' \subseteq S$ such that $|S'| = \kappa$ and each $c_i \in U$ appears in exactly one set of S'?

It is known that both the X3C and the X4C problems are NP-hard [3, 27]. In particular, the NP-hardness of both problems holds even when each element $c_i \in U$ occurs in exactly 3 subsets of S [3, 27]. Notice that under this assumption, $n = 3\kappa$ in X3C.

The words "promote" and "degrade" are often used in NP-hardness reductions and description of polynomial-time algorithms with specific meanings in this paper. In particular, for a vote π and a candidate c, *promoting* the candidate c by ℓ positions for some $\ell < pos_\pi(c)$ means recasting the vote π as follows: (1) rank c in the $(pos_\pi(c) - \ell)$-th position; (2) rank every candidate c' with $pos_\pi(c) > pos_\pi(c') \geq pos_\pi(c) - \ell$ in the $(pos_\pi(c') + 1)$-th position; and (3) rank all the remaining candidates in their original positions. *Degrading* the candidate c by ℓ positions means recasting the vote π as follows: (1) rank c in the $(pos_\pi(c) + \ell)$-th position of π; (2) rank every candidate c' with $pos_\pi(c) < pos_\pi(c') \leq pos_\pi(c) + \ell$ in the $(pos_\pi(c') - 1)$-th position; and (3) rank all the remaining candidates in their original positions. See Figure 1 for an illustration.

$a \succ b \succ c \succ d \succ e \succ f \qquad a \succ c \succ d \succ b \succ e \succ f$

Degrading b by two positions.

$a \succ b \succ c \succ d \succ e \succ f \qquad a \succ e \succ b \succ c \succ d \succ f$

Promoting e by three positions.

Figure 1. Illustrations of promoting and degrading candidates.

3 Kendall-Tau Distance Restricted Bribery

In this section, we investigate the bribery problem with KT-distance restrictions. In the following, we summarize our results in several theorems. We begin with some polynomial-time solvability results.

Theorem 1 *The following problems are polynomial-time solvable: C-KT(1)-Borda-UNI/NON, C-KT(ℓ)-Condorcet-UNI/NON for $\ell = 1, 2$, D-KT(ℓ)-Borda-UNI/NON for every possible ℓ, D-KT(ℓ)-Condorcet-UNI/NON for every possible ℓ, D-KT(1)-Maximin-UNI/NON.*

PROOF. Due to space limitation, we give only the polynomial-time algorithms for the C-KT(2)-Condorcet-UNI problem and the D-KT(1)-Maximin-UNI problem. Let $((\mathcal{C}, \Pi_\mathcal{V}), p \in \mathcal{C}, \mathcal{R})$ be a given instance. Let m be the number of candidates and n the number of votes, i.e., $m = |\mathcal{C}|$ and $n = |\Pi_\mathcal{V}|$.

C-KT(2)-Condorcet-UNI. We reduce the problem to the SIMPLE B-EDGE COVER OF MULTIGRAPHS problem which is polynomial-time solvable [36].

SIMPLE B-EDGE COVER OF MULTIGRAPHS (B-ECM)
Input: An undirected multigraph $G = (U, E)$ where U is the set of vertices and E is the set of edges, a function $f : U \to \mathbb{Z}^+$ and a positive integer κ.
Question: Does there exist an $E' \subseteq E$ such that $|E'| \leq \kappa$ and every vertex $u \in U$ is incident to at least $f(u)$ edges in E'?

Now we show how to reduce the C-KT(2)-Condorcet-UNI problem to the B-ECM problem. For each candidate $c \in \mathcal{C} \setminus \{p\}$ which is not beaten by p, we create a vertex. For simplicity, we still use c to denote the corresponding vertex. We define $\overleftarrow{p}_v$ for a vote π_v where p is not ranked in the top as follows: if p is not ranked in the second highest position in π_v, then $\overleftarrow{p}_v$ is the set consisting of the two candidates immediately ranked above p in π_v; if p is ranked in the second-highest position in π_v, then $\overleftarrow{p}_v$ is the set consisting of the candidate ranked in the highest position in π_v. For example, for a vote π_v with preference $a \succ b \succ c \succ p \succ d$, $\overleftarrow{p}_v = \{b, c\}$, while for a vote π_u with preference $a \succ p \succ c \succ b \succ d$, $\overleftarrow{p}_u = \{a\}$. The edges are created according to the votes. Precisely, for each vote π_v with $|\overleftarrow{p}_v| = 2$, if both candidates of $\overleftarrow{p}_v = \{c, c'\}$ are not beaten by p, we create an edge between c and c'. On the other hand, if only one of $\overleftarrow{p}_v$ is not beaten by p, we introduce a new degree-1 vertex adjacent to the vertex in $\overleftarrow{p}_v$ that is not beaten by p. For each vote π_v with $|\overleftarrow{p}_v| = 1$, if the candidate in $\overleftarrow{p}_v$ is not beaten by p, we introduce a new degree-1 vertex adjacent to the candidate in $\overleftarrow{p}_v$. Now we come to the capacities of the vertices. Each vertex corresponding to a candidate c has a capacity $f(c) = (N(c,p) - N(p,c))/2 + 1$ whenever $N(c,p) - N(p,c) \equiv 0 \mod 2$, and has a capacity $f(c) = (N(c,p) + 1 - N(p,c))/2$ otherwise. Moreover, each newly introduced degree-1 vertex has capacity 0. The value of the capacity $f(c)$ indicates the minimum number of votes which rank c above p, that are needed to be replaced with votes which rank p above c in order to make p beat c. Finally, we set $\kappa = \mathcal{R}$.

Now we get an instance of the B-ECM problem. Moreover, given a solution E' of the instance of the B-ECM problem, we can get a solution for C-KT(2)-Condorcet-UNI in polynomial time. In particular, for each edge $(c, c') \in E'$, if none of $\{c, c'\}$ is a newly introduced degree-1 vertex, we recast the corresponding vote by promoting p by two positions; otherwise, we recast the corresponding vote by promoting p by one position.

D-KT(1)-Maximin-UNI: The algorithm first carries out a polynomial number of guesses. In particular, the algorithm guesses a candidate p' which prevents p from being the unique winner, an integer s which plays the role as an upper bound of the Maximin score of p in the final election and a lower bound of the Maximin score of p' in the final election, and a candidate q with $N(p, q) \leq s$ in the final election. These lead to at most $(m-1)^2 \times n$ subinstances where m is the number of candidates and n the number of votes. To make it clear, we give the formal definition of the subproblem.

Sub-D-KT(1)-Maximin-UNI
Input: An election $\mathcal{E} = (\mathcal{C}, \Pi_{\mathcal{V}})$, a distinguished candidate $p \in \mathcal{C}$, two further candidates $p', q \in \mathcal{C} \setminus \{p\}$ (it may be that $p' = q$), and two integers $0 \leq s, \mathcal{R} \leq |\Pi_{\mathcal{V}}|$.
Question: Is there a submultiset $\Pi_{\mathcal{T}} \subseteq \Pi_{\mathcal{V}}$ of votes such that
(1) $\Pi_{\mathcal{T}}$ contains at most $\mathcal{R}$ votes; and
(2) we can replace every vote $\pi_v \in \Pi_{\mathcal{T}}$ with a new vote obtained from π_v by swapping two consecutively ranked candidates so that $N(p, q) \leq s$ in the final election, and the Maximin score of p' is at least s in the final election?

Now we focus on solving the subproblem. Let Π_p be the multiset of votes which rank p immediately above q. Let $A = \{c \in \mathcal{C} \setminus \{p'\} \mid N_{\mathcal{E}}(p', c) < s\}$. For each $c \in A$, let $\Pi_c \subseteq \Pi_{\mathcal{V}} \setminus \Pi_p$ be the multiset of votes that rank c immediately above p'. Clearly, for every distinct two candidates $c, c' \in A$, $\Pi_c \cap \Pi_{c'} = \emptyset$. Moreover, for every $c \in A$, let $f(c) = s - N_{\mathcal{E}}(p', c)$. The algorithm works as follows. For each $c \in A$, arbitrarily choose $\min\{f(c), |\Pi_c|\}$ votes in Π_c, and replace each of them with a new vote obtained from the original vote by swapping c and p'; then, set $f(c) := f(c) - \min\{f(c), |\Pi_c|\}$ and $\mathcal{R} := \mathcal{R} - \min\{f(c), |\Pi_c|\}$. If $\mathcal{R} < 0$ after doing so, we cannot make p' have a Maximin score at least s by replaying at most $\mathcal{R}$ votes; and thus, the algorithm returns "No". Otherwise, let $B = \{c \in A \mid f(c) > 0\}$. Then, for each $c \in B$, let $\bar{\Pi}_c$ be the multiset of votes in Π_p that rank c immediately above p'. If $|\bar{\Pi}_c| < f(c)$, the given instance is a No-instance (since we cannot make p' have a Maximin score at least s in the final election); and thus, we return "No". Otherwise, we arbitrarily choose $\min\{f(c), |\bar{\Pi}_c|\}$ votes in $\bar{\Pi}_c$, and (1) replace each of them with a new vote obtained from the original vote by swapping c and p'; (2) remove them from the multiset Π_p; and (3) set $\mathcal{R} := \mathcal{R} - \min\{f(c), |\bar{\Pi}_c|\}$. If $\mathcal{R} < 0$ or $\min\{|\Pi_p|, \mathcal{R}\} < N_{\mathcal{E}}(p, q) - s$ after doing so, we return "No". Otherwise, we return "Yes" since we can get a solution by replacing arbitrary $N_{\mathcal{E}}(p, q) - s$ votes in Π_p by new votes obtained from the original votes by swapping p and q. □

Now we discuss the NP-hardness results. We begin with constructive distance restricted bribery for Borda. We have seen from Theorem 1 that the destructive counterpart turned out to be polynomial-time solvable for every possible value of ℓ. The following theorem shows, however, that constructive distance restricted bribery for Borda is NP-hard, even when the distance is bounded by a small constant. We first define some useful notations.

For an order $X = (x_1, x_2, ..., x_i)$ over a set $\{x_1, x_2, ..., x_i\}$, we denote by $\overleftarrow{X}$ the reverse order of X, i.e., $\overleftarrow{X} = (x_i, ..., x_2, x_1)$. For a subset $Y \subseteq \{x_1, x_2, ..., x_i\}$, $X \setminus Y$ is the order obtained from X by deleting all the elements in Y. For example, for $X = (1, 4, 3, 8, 5)$ and $Y = \{4, 8\}$, $X \setminus Y = (1, 3, 5)$. For two subsets X and Y of candidates such that $X \cap Y = \emptyset$, and a vote with preference $\succ$, $X \succ Y$ means that every candidate in X is ranked above every candidate in Y in the vote.

Theorem 2 *C-KT(ℓ)-Borda-UNI/NON are NP-hard for all $\ell \geq 3$.*

PROOF. We give only proofs for the case $\ell = 3$ here. We first consider C-KT(3)-Borda-NON. The reduction is from X3C. Given an instance $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S = \{s_1, s_2, ..., s_m\})$ of X3C, we create an instance $\mathcal{E}$ for C-KT(3)-Borda-NON as follows.

Candidates: For each $c \in U$, we create a corresponding candidate. For convenience, we still use c to denote the corresponding candidate. In addition, we create a set $Y = \{y_1, y_2, ..., y_{6m-6}\}$ of $6m-6$ dummy candidates, each of which has a considerably less Borda score than that of any other candidate not in Y. For ease of exposition, we partition the dummy candidates into subsets $Z_1, Z_2, ..., Z_m$. To be precise, for each $i = 1, 2, ..., m-2$, $Z_i = \{y_{6i-5}, y_{6i-4}, y_{6i-3}, y_{6i-2}, y_{6i-1}, y_{6i}\}$. Moreover, $Z_{m-1} = \{y_{6m-11}, y_{6m-10}, y_{6m-9}\}$ and $Z_m = \{y_{6m-8}, y_{6m-7}, y_{6m-6}\}$. Finally, we create a distinguished candidate p.

Votes: We create $2m+2$ votes in total. In the following, we do not distinguish between the terms "set" and "order". In particular, U is considered as an order $(c_1, c_2, ..., c_{3\kappa})$, and every $s = \{c_x, c_y, c_z\} \in S$ is considered as an order (c_x, c_y, c_z) with $x < y < z$. Moreover, each Z_i where $i \in \{1, ..., m\}$ is considered as an arbitrary but fixed order of its elements.

For each $s_j \in S$ with $j = 1, 2, ..., m-2$, we create two votes as follows.

$\pi_{s_j} : s_j \succ p \succ Z_j \succ U \setminus s_j \succ Y \setminus Z_j$

$\pi'_{s_j} : \overleftarrow{U \setminus s_j} \succ Z_j \succ p \succ \overleftarrow{s_j} \succ Y \setminus Z_j$

Note that with the above $2(m-2)$ votes, all candidates in $U \cup \{p\}$ have the same Borda score. The following four votes are created according to the last two 3-subsets $s_{m-1}, s_m \in S$.

$\pi_{s_{m-1}} : s_{m-1} \succ p \succ Z_m \cup Z_{m-1} \succ U \setminus s_{m-1} \succ Y \setminus (Z_m \cup Z_{m-1})$

$\pi'_{s_{m-1}} : \overleftarrow{U \setminus s_{m-1}} \succ Z_{m-1} \succ p \succ Z_m \succ \overleftarrow{s_{m-1}} \succ Y \setminus (Z_m \cup Z_{m-1})$

$\pi_{s_m} : s_m \succ p \succ Z_m \cup Z_{m-1} \succ U \setminus s_m \succ Y \setminus (Z_m \cup Z_{m-1})$

$\pi'_{s_m} : \overleftarrow{U \setminus s_m} \succ Z_{m-1} \succ p \succ Z_m \succ \overleftarrow{s_m} \succ Y \setminus (Z_m \cup Z_{m-1})$

With the above four votes and the previously created $2(m-2)$ votes, p has exactly 6 more points than every candidate $c \in U$.

Finally, we have two votes with preferences $U \succ Z_m \succ p \succ Y \setminus Z_m$; and $\overleftarrow{U} \succ Z_m \succ p \succ Y \setminus Z_m$, respectively.

With all the $2m+2$ votes created as above, p has exactly $3\kappa + 1$ less points than every candidate $c \in U$, and all candidates in U have the same Borda score.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

In the following, let $A = \{\pi'_{s_j} \mid s_j \in S\}$ and B the set of the last two created votes. Now we discuss the correctness of the reduction.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a Yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_{s_j} \mid s_j \in S'\}$ be the multiset of the votes of the first type corresponding to S'. Every vote π_{s_j} in $\Pi_{S'}$ ranks the three candidates in s_j above p. Consider the election $\mathcal{E}'$ obtained from the original election $\mathcal{E}$ by replacing each $\pi_{s_j} \in \Pi_{S'}$ with a vote obtained from π_{s_j} by promoting p to the highest position. Precisely, for each $\pi_{s_j} \in \Pi_{S'}$ defined as $s_j \succ p \succ Z_j \succ U \setminus s_j \succ Y \setminus Z_j$, we replace it with a vote defined as $p \succ s_j \succ Z_j \succ U \setminus s_j \succ Y \setminus Z_j$. Clearly, each replacement increases the score of p by 3, and decreases the score of every candidate in s_j by 1. Since there are exactly κ votes in $\Pi_{S'}$, the score of p is finally increased by 3κ. Since S' is an exact 3-set cover, for every $c \in U$, there is only one vote in $\Pi_{S'}$ which ranks c above p. Therefore, all replacements decrease the score of each candidate in U by 1. Since p has exactly $3\kappa + 1$ less points than every candidate $c \in U$ in the original election $\mathcal{E}$, p has exactly the same score as every candidate $c \in U$ in the final election $\mathcal{E}'$. Therefore, p becomes a winner in $\mathcal{E}'$.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a Yes-instance and $\Pi_{S'}$ is the multiset of votes which are replaced. We assume that $\Pi_{S'}$ does not contain any vote in $A \cup B$. This assumption is sound due to the following lemma.

Lemma 3 *If $\mathcal{E}$ is a Yes-instance, there must be a solution wherein no vote in $A \cup B$ is replaced.*

PROOF. We prove the lemma by showing that it is always better to replace a vote not in $A \cup B$ than to replace a vote in $A \cup B$. Suppose that π is a vote in $A \cup B$ that is replaced. Observe that promoting p is always better than degrading candidates in U, since promoting p by one position decreases the score gap between every candidate in U and p by one, while degrading some candidate $c \in U$ by one position only decreases the score gap between c and p by one (sometimes even increases the score gap between some other candidate $c' \in U$ and p). Moreover, the amount of points that can be decreased in the score gap between every candidate in U and p by promoting p in π, can be also achieved by promoting p in any vote that is not in $A \cup B$. In fact, since in every vote in $A \cup B$ there are at least three dummy candidates ranked below some candidates in U but ranked above p, replacing votes which are not in $A \cup B$ can always do better: replacing a vote $\pi_s \notin A \cup B$ where $s = \{c_x, c_y, c_z\}$ with preference $c_x \succ c_y \succ c_z \succ p...$ with a vote with preference $p \succ c_x \succ c_y \succ c_z...$ does not only decrease the score gap between every candidate in $U \setminus s$ and p by 3, but better yet, decreases the score gap between every candidate in s and p by 4. □

Due to the above analysis, we assume that $\Pi_{S'}$ contains only the votes in $\{\pi_{s_j} \mid s_j \in S\}$, where S is the collection of 3-subsets in $\mathcal{F}$. Let $S' = \{s_j \mid \pi_{s_j} \in \Pi_{S'}\}$ be the subcollection corresponding to $\Pi_{S'}$. First observe that for any vote $\pi_s \in \Pi_{S'}$ where $s \in S$, promoting p by three positions is always better than any other combinations: by doing so, the score gap between every candidate in U and p is decreased by at least 3 (for candidates in s, the score gaps are decreased by 4). Therefore, we can assume that in the solution, every vote in $\Pi_{S'}$ is replaced with a new vote obtained from the original vote by promoting p by three positions. Since p has $3\kappa + 1$ less points than every candidate in U in the original election $\mathcal{E}$, and we can replace at most κ votes, every candidate in U must be degraded by one position at least once. This implies that for every $c \in U$, there must be a vote $\pi_s \in \Pi_{S'}$ with $c \in s$, further implying that S' is an exact 3-set cover of $\mathcal{F}$.

The reduction for C-KT(3)-Borda-UNI is similar to the above reduction, with the difference in the last created vote. Precisely, we remove the last vote created in the reduction for C-KT(3)-Borda-NON, and create a vote with preference $\overleftarrow{U} \succ Z_m \cup \{y_{6m-12}\} \succ p \succ Y \setminus Z_m \cup \{y_{6m-12}\}$.

By ranking the candidate y_{6m-12} between Z_m and p, the score gap between every candidate in U and p decreases to 3κ, one point less than that in the reduction for C-KT(3)-Borda-NON. □

Now we come to Condorcet. The C-KT(ℓ)-Condorcet-UNI problem is related to the Dodgson voting [13], where each candidate has a Dodgson score defined as the minimum number of swaps of adjacent candidates needed to make the candidate the Condorcet winner. Calculating the Dodgson score of a candidate is NP-hard [29, 30]. Recall that the KT-distance between two votes is equal to the minimum number of swaps of adjacent candidates needed to transform one into the other. Therefore, if a candidate can become the Condorcet winner by recasting at most $\mathcal{R}$ votes with respect to KT-distance upper bound ℓ, then the Dodgson score of the candidate is at most $\mathcal{R} \cdot \ell$. In Theorem 1, we have shown that both C-KT(1)-Condorcet-UNI/NON and C-KT(2)-Condorcet-UNI/NON are polynomial-time solvable. In the following, we show that the polynomial-time solvability does not hold for C-KT(ℓ)-Condorcet-UNI/NON for every $\ell \geq 3$. Recall that in the general case, the constructive bribery for Condorcet is NP-hard [26, 30].

Hereinafter, we assume that in both the X3C problem and the X4C problem, each $c_i \in U$ occurs in exactly three subsets of S.

Theorem 4 *C-$KT(\ell)$-Condorcet-UNI/NON are NP-hard for every $\ell \geq 3$.*

PROOF. We first consider C-KT(3)-Condorcet-UNI. The reduction is from the X3C problem. Given an instance $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S = \{s_1, s_2, ..., s_{3\kappa}\})$ of X3C, we create an instance $\mathcal{E}$ for C-KT(3)-Condorcet-UNI as follows.

Candidates: For each $c \in U$, we create a corresponding candidate. For simplicity, we still use the same notation c to denote this candidate. In addition, we have a distinguished candidate p and a set $Y = \{y_1, y_2, y_3\}$ of three dummy candidates.

Votes: For each $s = \{c_x, c_y, c_z\} \in S$, we create a vote π_s defined as $s \succ p \succ U \setminus s \succ Y$. Here, the candidates in s, in $U \setminus s$ and in Y are ranked according to the increasing order of the indices, respectively. In addition, we create $3\kappa - 5$ votes defined as $U \succ Y \succ p$. Here, the candidates in U and in Y are ranked according to the increasing order of the indices, respectively. In total, we have $6\kappa - 5$ votes.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

Now we discuss the correctness. Observe that c_1 is the current Condorcet winner, and no candidate in Y can become the Condorcet winner by replacing at most κ votes with respect to the distance restriction.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a Yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_{s_j} \mid s_j \in S'\}$ be the multiset of votes corresponding to S'. Consider replacing each vote $\pi_s \in \Pi_{S'}$ by another vote obtained from π_s by promoting p to the highest position, that is, replacing each vote $\pi_s \in \Pi_{S'}$ defined as $s \succ p \succ U \setminus s \succ Y$ with a vote defined as $p \succ s \succ U \setminus s \succ Y$. Since s is a 3-subset, the KT-distance between the original vote and the new vote is 3. Since S' is an exact 3-set cover, for every $c \in U$ there is exactly one vote in $\Pi_{S'}$ which ranks c above p (and p is ranked above c after the replacement). Therefore, after κ replacements as discussed above, for every $c \in U$, there are exactly $3\kappa - 2$ votes ranking p above c, implying that p is the Condorcet winner in the final election.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a Yes-instance and $\Pi_{S'}$ is the multiset of votes which are replaced. Since $|Y| = 3$ and each vote can be replaced only with a vote which has KT-distance at most 3 from it, replacing any of the last $3\kappa - 5$ votes is not helpful in improving the wining status of p (In other words, replacing a vote in the last $3\kappa - 5$ votes is not helpful for p to beat any candidate in U, since the dummy candidates in Y are ranked between U and p; and thus, according to the distance restriction, p cannot be ranked above any candidate in U via a replacement of a vote in the last $3\kappa - 5$ votes.). Therefore, we know that $\Pi_{S'}$ contains only votes corresponding to S. Let $S' = \{s \in S \mid \pi_s \in \Pi_{S'}\}$ be the subcollection of S corresponding to $\Pi_{S'}$. In order to make p the Condorcet winner, for every $c \in U$ there must be at least one vote, corresponding to some s with $c \in s$, which is replaced with a vote ranking p above c. This implies that S' is an exact 3-set cover of $\mathcal{F}$.

The above reduction directly applies to C-KT(3)-Condorcet-NON. The NP-hardness of C-KT(4)-Condorcet-UNI/NON can be proved via reductions from the X4C problem. The reductions are analogous to the ones for C-KT(3)-Condorcet-UNI/NON (we need to create one more dummy candidate y_4 and add it to Y). The NP-hardness of C-KT(ℓ)-Condorcet-UNI/NON for every $\ell \geq 5$ is implied by the NP-hardness reductions in Theorem 3.2 in [26]. □

Now we come to Copeland$^\alpha$. It is known that both the constructive and the destructive bribery problems without distance restrictions for Copeland$^\alpha$ are NP-hard [26], for both the unique-winner model and the nonunique-winner model. We show that these NP-hardness results hold for the distance restricted bribery problem, even when the distance bound is demanded to be a very small constant.

Theorem 5 *The C-KT(ℓ)-Copeland$^\alpha$-UNI/NON problem, the D-KT(ℓ)-Copeland$^\alpha$-NON problem for every $\ell \geq 3$, and the D-KT(ℓ)-Copeland$^\alpha$-UNI problem for every $\ell \geq 5$ are NP-hard. All these results hold for every $0 \leq \alpha \leq 1$.*

PROOF. We first consider the C-KT(3)-Copeland$^\alpha$-UNI problem. The reduction is from X3C. Let $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S = \{s_1, s_2, ..., s_{3\kappa}\})$ be an instance of X3C. We create an instance $\mathcal{E}$ for C-KT(3)-Copeland$^\alpha$-UNI as follows.

Candidates: We have $|U| + 8$ candidates in total. In particular, for each $c_i \in U$, we create a candidate. For simplicity, we still use c_i to denote this candidate. In addition, we have 8 candidates $p, y, z_1, z_2, z_3, z_1', z_2', z_3'$, where p is the distinguished candidate. For ease of exposition, let $Z = \{z_1, z_2, z_3\}$ and $Z' = \{z_1', z_2', z_3'\}$.

Votes: Let $n = |S| = 3\kappa$. We create $2n + 1$ votes in total. In particular, for each $s = \{c_i, c_j, c_k\} \in S$, we create one vote π_s with preference $y \succ Z' \succ s \succ p \succ U \setminus s \succ Z$. Here, the candidates in $Z, Z', s, U \setminus s$ are ranked according to the increasing order of the indices, respectively. In addition, we create $n - 2$ votes each with preference $U \succ Z \succ p \succ y \succ Z'$. Finally, we create 3 votes each with preference $p \succ y \succ Z' \succ U \succ Z$. In the above $n + 1$ votes, the candidates in U, Z and Z' are ranked according to the increasing order of the indices, respectively. It is easy to verify that the candidate y is the current (unique) winner.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

Now we prove that $\mathcal{F}$ is a Yes-instance if and only if $\mathcal{E}$ is a Yes-instance.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a Yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_s \mid s \in S'\}$ be the set of votes corresponding to S'. Consider the election after replacing all votes in $\Pi_{S'}$ in the following way: each $\pi_s \in \Pi_{S'}$ with $s \in S'$ is replaced with a vote defined as $y \succ Z' \succ p \succ s \succ U \setminus s \succ Z$. Clearly, the KT-distance between these two votes is 3. Since S' is an exact 3-set cover, for each $c_i \in U$ there is exactly one vote $\pi_s \in \Pi_{S'}$ with $c_i \in s$. Due to the construction, c_i is ranked above p in π_s, while ranked below p in the new vote which replaces π_s. Therefore, after κ replacements as discussed above, for every $c_i \in U$ there are $n + 1$ votes which rank p above c_i, implying that p beats every candidate $c_i \in U$, further implying that p is the unique Copeland$^\alpha$ winner (holds for every $0 \leq \alpha \leq 1$).

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a Yes-instance and $\Pi_{S'}$ is the multiset of votes which are replaced. Let $\mathcal{E}'$ be the final election obtained form $\mathcal{E}$ by replacing the votes in $\Pi_{S'}$ with κ many new votes (we discuss later what are the new votes). Observe that the candidate y beats every other candidate except p in $\mathcal{E}$. A deeper observation is that y still beats these candidates in the final election $\mathcal{E}'$.

Lemma 6 *The candidate y beats everyone in $U \cup Z \cup Z'$ in $\mathcal{E}'$.*

PROOF. Clearly, y beats every candidate in Z' in the final election $\mathcal{E}'$ since all votes rank y above Z'. Now we consider the candidates in $U \cup Z$. Observe first that every vote in $\mathcal{E}$ either ranks y above all candidates in $U \cup Z$, or ranks all candidates in $U \cup Z$ above y. Moreover, the votes that rank y above all candidates in $U \cup Z$ are those corresponding to S, and the last three created votes. However, in these votes, the candidates in Z' ($|Z'| = 3$) are ranked between y and every candidate in $U \cup Z$; thus, we cannot replace a vote which ranks y above a candidate $a \in U \cup Z$ by a KT(3)-close vote which, however, ranks a above y. Therefore, the votes which rank y above a candidate $a \in U \cup Z$ will still rank y above a in the final election $\mathcal{E}'$. The lemma follows. □

Due to the above lemma and the fact that p is the unique winner in the final election $\mathcal{E}'$, we know that p beats every other candidate in $\mathcal{E}'$. Observe that in the original election $\mathcal{E}$, p is beaten by every candidate in U. Then, due to the distance restriction, $\Pi_{S'}$ must be from the votes corresponding to S. Let $S' = \{s \mid \pi_s \in \Pi_{S'}\}$ be the subcollection of 3-subsets corresponding to $\Pi_{S'}$. Since p beats all candidates in U in the final election $\mathcal{E}'$ and we can replace at most $\mathcal{R} = \kappa$ votes, for each $c_i \in U$ there must be a vote $\pi_s \in \Pi_{S'}$ with $c_i \in s$, which is replaced with a new vote obtained from π_s by promoting p by three positions. Since $|S| = 3\kappa$, it follows that S' is an exact 3-set cover.

The above reduction applies to D-KT(3)-Copeland$^\alpha$-NON if we set y as the distinguished candidate.

Now we consider the C-KT(3)-Copeland$^\alpha$-NON problem. The above reduction does not apply to this case, since p does not need to beat every other candidate in the final election to become a winner (p could also become a winner even there is no exact 3-set cover). In order to overcome this situation, we introduce a new dummy candidate y' which beats p, but is beaten by y in the original election. Precisely, we adopt the votes constructed as above, and rank y' in the votes as follows: (1) rank y' immediately after y in all votes corresponding to S and all the $n-2$ votes following; and (2) rank y' above p in all the three votes created in the last.

The NP-hardness of the C-KT(4)-Copeland$^\alpha$-UNI/NON problem and the D-KT(4)-Copeland$^\alpha$-NON problem can be proved via reductions from the X4C problem analogously. The NP-hardness of the C-KT(ℓ)-Copeland$^\alpha$-UNI/NON problem and the D-KT(ℓ)-Copeland$^\alpha$-UNI/NON problem for every $\ell \geq 5$ is implied by the NP-hardness reductions in Theorem 3.2 in [26]. $\square$

We have just investigated the NP-hardness of the C-KT(ℓ)-Copeland$^\alpha$-UNI/NON and the D-KT(ℓ)-Copeland$^\alpha$-NON problems for every $\ell \geq 3$, and the D-KT(ℓ)-Copeland$^\alpha$-UNI problem for every $\ell \geq 5$, but left the complexity of the D-KT(ℓ)-Copeland$^\alpha$-UNI problem for each integer $\ell \in \{3, 4\}$ open. We cannot straightforwardly adopt the reductions for the C-KT$(3, 4)$-Copeland$^\alpha$-NON problem to prove the NP-hardness of the D-KT$(3, 4)$-Copeland$^\alpha$-UNI problem, since both candidates y and y' win the election, and thus, no candidate is valid to be the distinguished candidate.

Now we investigate the complexity of the distance restricted bribery problems for Maximin. It is known that both the constructive and the destructive bribery problems for Maximin without distance restrictions are NP-hard [24]. We prove that the NP-hardness holds even when each bribed voter only wants to recast a new vote which has a small constant discrepancy from his original vote.

Theorem 7 *The D-KT(ℓ)-Maximin-UNI/NON problem and the C-KT(ℓ)-Maximin-UNI/NON problem are NP-hard for every $\ell \geq 4$.*

4 Hamming Distance Restricted Bribery

In this section, we study the bribery problem with Hamming distance restrictions. It should be noted that the Hamming distance between every two votes is at least 2. We begin with several polynomial-time solvability results.

Theorem 8 *The D-HAM(ℓ)-Condorcet-UNI/NON problem and the D-HAM(ℓ)-Borda-UNI/NON problem are polynomial-time solvable for every positive integer ℓ.*

PROOF. We prove the theorem by deriving polynomial-time algorithms for the problems stated in the theorem. We only describe the algorithms for the unique-winner model in detail. The algorithms for the nonunique-winner model are similar. Let m be the number of candidates, n the number of votes, $\mathcal{R}$ the number of votes that can be replaced, and p the distinguished candidate.

D-HAM(ℓ)-Condorcet. We first consider D-HAM(2)-Condorcet. The algorithm first guesses a candidate p' which is not beaten by p in the final election. This leads to $m-1$ subinstances, each asking whether we can make p' not be beaten by p by replacing $\mathcal{R}' \leq \mathcal{R}$ votes with $\mathcal{R}'$ many HAM(2)-close votes. To solve each subinstance, we need only to arbitrarily choose up to $\mathcal{R}$ votes which rank p above p', and replace each of them with a new vote obtained from the original vote by swapping p and p'. After this, if p' is not beaten by p, the subinstance is a Yes-instance; otherwise, the subinstance is a No-instance. It is clear that the original instance is a Yes-instance if and only if at least one of the subinstances is a Yes-instance. The above algorithm directly applies to D-HAM(ℓ)-Condorcet for every possible $\ell \geq 2$.

D-HAM(2)-Borda. The algorithm first guesses a candidate p' which prevents p from being the unique-winner in the final election. This leads to $m-1$ subinstances, each asking whether we can make p' have an equal or greater Borda score than that of p by replacing $\mathcal{R}' \leq \mathcal{R}$ votes with $\mathcal{R}'$ many HAM(2)-close votes. To solve each subinstance, we order all votes π_v according to a nonincreasing order of $\max\{pos_v(p') - 1, m - pos_v(p), 2 \cdot (pos_v(p') - pos_v(p))\}$. Let Π be the multiset of the first $\min\{n, \mathcal{R}\}$ votes according to this order. Then, we replace every vote in Π in the following way. For each $\pi_v \in \Pi$, if $pos_v(p') - 1 \geq m - pos_v(p)$ and $pos_v(p') - 1 \geq 2 \cdot (pos_v(p') - pos_v(p))$, then replace π_v with a new vote obtained from π_v by swapping p' and the first ranked candidate in π_v; otherwise, if $m - pos_v(p) \geq pos_v(p') - 1$ and $m - pos_v(p) \geq 2 \cdot (pos_v(p') - pos_v(p))$, replace π_v with a vote obtained from π_v by swapping p and the last ranked candidate in π_v; finally, if $2 \cdot (pos_v(p') - pos_v(p)) \geq pos_v(p') - 1$ and $2 \cdot (pos_v(p') - pos_v(p)) \geq m - pos_v(p)$, replace π_v with a vote obtained from π_v by swapping p and p'. After doing this for every vote in Π, if p' has an equal or greater Borda score than that of p, the subinstance is a Yes-instance; otherwise, the subinstance is a No-instance. It is clear that the original instance is a Yes-instance if and only if at least one of the subinstances is a Yes-instance.

D-HAM(3)-Borda. The algorithm carries out $m-1$ guesses as in the above algorithm for D-HAM(2)-Borda. So, we need only to focus on the subinstances. Notice that to prevent p from being the unique winner, we have more choices to do in this case than in D-HAM(2)-Borda. In particular, for a vote π_v, to decrease the score gap between p and the guessed candidate p', we can either place p' in the first position, p in the $\max\{pos_v(p), pos_v(p')\}$-th position, the first ranked candidate in π_v in the $\min\{pos_v(p), pos(p')\}$-th position, or we can place p in the last position, p' in the $\min\{pos_v(p), pos_v(p')\}$-th position, the last ranked candidate in π_v in the $\max\{pos_v(p), pos_v(p')\}$-th position. Let $sga(\pi_v)$ be the amount of the decrease of the score gap between p and p' if we perform the first operation, and $sgb(\pi_v)$ the amount of decrease of the score gap between p and p' if we perform the second operation. Then, to solve each subinstance, we order the votes according to a nonincreasing order of $\max\{sga(\pi_v), sgb(\pi_v)\}$ and recast the votes accordingly, as discussed above.

D-HAM(4)-Borda. Similar to the above algorithms, the algorithm for D-HAM(4)-Borda first carries out $m-1$ guesses, leading to $m-1$ subinstances. Now we restrict our attention to these subinstances. For each subinstance, we partition the votes into two multisubsets Π_1 and Π_2, where Π_1 includes all votes that rank p above p', and Π_2 includes all votes that rank p' above p. Then, we order the votes in Π_1 according to a nonincreasing order of $pos(p') - pos(p)$. Moreover, we choose the first $\min\{|\Pi_1|, \mathcal{R}\}$ votes, and replace each of them with a vote obtained from the original vote by swapping p' and the first ranked candidate, and swapping p and the last ranked candidate. After doing so, if p' has an equal or greater score than that of p, we return "Yes". Otherwise, if p' still has a less score than that of p, we distinguish between two cases. If $|\Pi_1| \geq \mathcal{R}$, we return "No" immediately. In the case that $|\Pi_1| < \mathcal{R}$, we order the votes in Π_2 according to a nondecreasing order of $pos(p) - pos(p')$. Then, we choose the first $\mathcal{R} - |\Pi_1|$ votes, and replace each of them with a vote obtained from the original vote by swapping p' and the first ranked candidate, and swapping p with the last ranked candidate. After doing this, if p' has an equal or greater score than that of p, the subinstance is a Yes-instance; otherwise it is a No-instance.

D-HAM(ℓ)-Borda. The algorithm for D-HAM(ℓ)-Borda with $\ell > 4$ is exactly the same as for D-HAM(4)-Borda. □

Now we show our hardness results. We begin with the distance restricted bribery problem for Copeland$^\alpha$.

Theorem 9 *The C-HAM(2)-Copeland$^\alpha$-UNI/NON problem and the D-HAM(2)-Copeland$^\alpha$-UNI/NON problem are NP-hard for every $0 \leq \alpha \leq 1$.*

PROOF. Due to space limitation, we give only the NP-hardness proof for C-HAM(2)-Copeland$^\alpha$-UNI. Our reduction is from the X3C problem. Let $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S = \{s_1, s_2, ..., s_{3\kappa}\})$ be a given instance of the X3C problem. We create an instance $\mathcal{E}$ for C-HAM(2)-Copeland$^\alpha$-UNI as follows.

Candidates: We create $3\kappa + 2$ candidates in total. In particular, for each element $c_i \in U$, we create one candidate. For convenience, we still use c_i to denote the corresponding candidate. In addition, we have two candidates p and q with p being the distinguished candidate.

Votes: For each $s \in S$, we create a vote π_s with preference $q \succ U \setminus s \succ p \succ s$. In addition, we create $\kappa - 1$ votes with preference $p \succ q \succ U$, and two votes with preference $U \succ p \succ q$. In total, we have $4\kappa + 1$ votes. The comparisons between every two candidates are summarized in Table 2. It is easy to verify that q beats every other candidate; and thus, q is the current unique winner.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

	q	p	c_j
q	-	3κ	$4\kappa - 1$
p	$\kappa + 1$	-	$\kappa + 2$
c_i	2	$3\kappa - 1$	$\cdots$

Table 2. Comparisons between every two candidates in the NP-hardness reduction for C-HAM(2)-Copeland$^\alpha$-UNI in Theorem 9. The comparisons between c_i and c_j for $i \neq j$ do not play any role in the correctness argument.

Now we prove the correctness of the reduction.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a Yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_s \mid s \in S'\}$ be the set of votes corresponding to S'. Consider the final election $\mathcal{E}'$ obtained from $\mathcal{E}$ by replacing each vote π_s where $s \in S$ with a vote obtained from π_s by swapping p and q. More precisely, each $\pi_s \in \Pi_{S'}$ with preference $q \succ U \setminus s \succ p \succ s$ is replaced with a vote with preference $p \succ U \setminus s \succ q \succ s$. Clearly, the Hamming distance between these two votes is two. Moreover, we have that $N_{\mathcal{E}'}(p, q) = 2\kappa + 1$. Now we consider the comparison between p and every $c_i \in U$. Since S' is an exact 3-set cover, for every c_i there are exactly $\kappa - 1$ votes $\pi_s \in \Pi_{S'}$ with $c_i \notin s$. All these votes rank c_i above p in $\mathcal{E}$. However, these votes are replaced with $\kappa - 1$ votes which rank p above c_i as discussed above, in the final election $\mathcal{E}'$. Therefore, for every $c_i \in U$, there are $(\kappa+2)+(\kappa-1) = 2\kappa + 1$ votes which rank p above c_i, implying that p beats every $c_i \in U$ in $\mathcal{E}'$. Therefore, p becomes the unique winner in $\mathcal{E}'$.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a Yes-instance. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by replacing at most κ votes. Since $N_{\mathcal{E}}(q, c_i) = 4\kappa - 1$, we know that q beats every candidate $c_i \in U$ in $\mathcal{E}'$. As a result, q is beaten by p in $\mathcal{E}'$, since otherwise, q would beat every other candidate in the final election $\mathcal{E}'$, and thus, remains the unique winner. Moreover, since p is the unique winner in $\mathcal{E}'$, p must beat every other candidate in the final election $\mathcal{E}'$. Since $N_{\mathcal{E}}(p, q) = \kappa+1$, in order to make p beat q, there has to be κ votes ranking q above p that are replaced by κ new votes ranking p above q. Due to this, we know that the replaced votes are from the votes corresponding to S, since any other vote has already ranked p above q. Let $\Pi_{S'}$ be the replaced votes, and $S' = \{s \mid \pi_s \in \Pi_{S'}\}$ the subcollection of 3-subsets corresponding to $\Pi_{S'}$. As discussed above, p beats every candidate $c_i \in U$ in $\mathcal{E}'$. Since $N_{\mathcal{E}}(p, c_i) = \kappa + 2$, for every $c_i \in U$, there must be at least $\kappa - 1$ votes in $\Pi_{S'}$ ranking c_i above p that are replaced by $\kappa - 1$ votes ranking p above c_i. This happens only if S' is an exact 3-set cover. □

For Condorcet and Maximin, we have the following results.

Theorem 10 *C-HAM(2)-Condorcet-UNI/NON are NP-hard.*

Theorem 11 *The D-HAM(2)-Maximin-UNI/NON problem and the D-HAM(2)-Maximin-UNI/NON problem are NP-hard.*

5 Conclusion

We have studied the complexity of the distance restricted bribery problem which differs from the traditional bribery problem [22] in that the bribed voters only recast new votes which are "close" to their original votes. In particular, we adopted the Hamming distance and the KT-distance to measure the closeness between two votes. We achieved both polynomial-time solvability results and NP-hardness results for Borda, Condorcet, Maximin and Copeland$^\alpha$. In particular, we achieved dichotomy results for Condorcet. A primary conclusion of our findings is that the constructive distance restricted bribery problem is generally NP-hard even when the distance is bounded by a very small integer (this holds for all cases studied in this paper except the Hamming distance restricted bribery problem for Borda, whose complexity remains open for every distance $\ell \geq 2$). On the other hand, there exist voting systems where the constructive distance restricted bribery problem is polynomial-time solvable, when the distance is bounded by 1 or 2. For the destructive distance restricted bribery problem, it turned out that for Maximin and Copeland$^\alpha$ it has become NP-hard even when the distance is bounded by 2. On the other hand, for many other cases, it is polynomial-time solvable for every possible distance bound. As the bribery problem proposed in [22] can be considered as a distance restricted bribery problem with the distance bound being considerably large ($m(m-1)/2$ in KT-distance restriction and m in Hamming distance restriction, where m is the number of candidates), our work complements the complexity results of the bribery problem obtained in [11, 24, 26]. Of particular importance is that our work pinpoints the complexity border between polynomial-time solvability and NP-hardness of distance restricted bribery problem, with respect to the distance bound. See Table 1 for a summary of our results.

There remain several open problems as shown in Table 1. For example, we do not know the complexity of the D-KT(1)-Copeland$^\alpha$-UNI/NON problem. Another avenue of research would be to explore these problems from the parameterized complexity viewpoint. Furthermore, exploring the same problems with respect to further distance measurements (see [14, 16, 33] for several distance measurements on linear orders) is also an interesting direction for future research.

REFERENCES

[1] O. L. Astrachan, 'Bubble sort: An archaeological algorithmic analysis', in *SIGCSE*, pp. 1–5, (2003).

[2] M. S. Bansal and D. Fernàndez-Baca, 'Computing distances between partial rankings', *Inf. Process. Lett.*, **109**(4), 238–241, (2009).

[3] R. Berghammer and H. Schnoor, 'Control of Condorcet voting: Complexity and a relation-algebraic approach', *Eur. J. Oper. Res.*, **246**(2), 505–516, (2015).

[4] N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier, 'Studies in computational aspects of voting – A parameterized complexity perspective', in *The Multivariate Algorithmic Revolution and Beyond*, pp. 318–363, (2012).

[5] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond, 'Fixed-parameter algorithms for Kemeny rankings', *Theor. Comput. Sci.*, **410**(45), 4554–4570, (2009).

[6] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond, 'How similarity helps to efficiently compute Kemeny rankings', in *AAMAS (1)*, pp. 657–664, (2009).

[7] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, 'Link analysis ranking: Algorithms, theory, and experiments', *ACM Trans. Internet Techn.*, **5**(1), 231–297, (2005).

[8] S. J. Brams, D. M. Kilgour, and M. R. Sanver, 'A minimax procedure for electing committees', *Public. Choice.*, **132**(3-4), 401–420, (2007).

[9] F. J. Brandenburg, A. Gleißner, and A. Hofmeier, 'Comparing and aggregating partial orders with Kendall-Tau distances', *Discrete Math., Alg. and Appl.*, **5**(2), (2013).

[10] R. Bredereck, J. Chen, and G. J. Woeginger, 'Are there any nicely structured preference profiles nearby?', *Math. Soc. Sci.*, **79**, 61–73, (2016).

[11] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, and I. Schnoor, 'Approximability of manipulating elections', in *AAAI*, pp. 44–49, (2008).

[12] E. Chávez, K. Figueroa, and G. Navarro, 'Effective proximity retrieval by ordering permutations', *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(9), 1647–1658, (2008).

[13] C. L. Dodgson, *A Method for Taking Votes on More than Two Issues*, Clarendon Press, 1876.

[14] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, 'Rank aggregation methods for the web', in *WWW*, pp. 613–622, (2001).

[15] P. J. Egan, 'Do something politics and double-peaked policy preferences', *J. Polit.*, **76**(2), 333–349, (2014).

[16] E. Elkind, P. Faliszewski, and A. Slinko, 'Distance rationalization of voting rules', *Soc. Choice. Welfare.*, **45**(2), 345–377, (2015).

[17] E. Elkind, P. Faliszewski, and A. M. Slinko, 'Swap bribery', in *SAGT*, pp. 299–310, (2009).

[18] G. Erdélyi, M. Lackner, and A. Pfandler, 'Computational aspects of nearly single-peaked electorates', in *AAAI*, pp. 283–289, (2013).

[19] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee, 'Comparing and aggregating rankings with ties', in *PODS*, pp. 47–58, (2004).

[20] R. Fagin, R. Kumar, and D. Sivakumar, 'Comparing top k lists', *SIAM J. Discrete Math.*, **17**(1), 134–160, (2003).

[21] P. Faliszewski, 'Nonuniform bribery', in *AAMAS (3)*, pp. 1569–1572, (2008).

[22] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra, 'The complexity of bribery in elections', in *AAAI*, pp. 641–646, (2006).

[23] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra, 'How hard is bribery in elections?', *J. Artif. Intell. Res. (JAIR)*, **35**, 485–532, (2009).

[24] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra, 'Multimode control attacks on elections', *J. Artif. Intell. Res. (JAIR)*, **40**, 305–351, (2011).

[25] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra, 'The complexity of manipulative attacks in nearly single-peaked electorates', *Artif. Intell.*, **207**, 69–99, (2014).

[26] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe, 'Llull and Copeland voting computationally resist bribery and constructive control', *J. Artif. Intell. Res. (JAIR)*, **35**, 275–341, (2009).

[27] T. F. Gonzalez, 'Clustering to minimize the maximum intercluster distance', *Theor. Comput. Sci.*, **38**, 293–306, (1985).

[28] R. W. Hamming, 'Error detecting and error correcting codes', *AT* & *T. Tech. J.*, **26**(2), 147–160, (1950).

[29] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe, 'Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP', *J. ACM.*, **44**(6), 806–825, (1997).

[30] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick, 'Voting schemes for which it can be difficult to tell who won the election', *Soc. Choice. Welfare.*, **6**(2), 157–165, (1989).

[31] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick, 'How hard is it to control an election?', *Math. Comput. Model.*, **16**(8-9), 27–40, (1992).

[32] M. Kalech, S. Kraus, G. A. Kaminka, and C. V. Goldman, 'Practical voting rules with partial information', *Auton. Agent. Multi-AG.*, **22**, 151–182, (2011).

[33] M. G. Kendall, 'A new measure of rank correlation', *Biometrika.*, **30**(1/2), pp. 81–93, (1938).

[34] S. Konieczny, J. Lang, and P. Marquis, 'DA^2 merging operators', *Artif. Intell.*, **157**(1-2), 49–79, (2004).

[35] J. Lang, G. Pigozzi, M. Slavkovik, and L. van der Torre, 'Judgment aggregation rules based on minimization', in *TARK*, pp. 238–246, (2011).

[36] A. P. Lin, 'The complexity of manipulating k-Approval elections', in *ICAART (2)*, pp. 212–218, (2011). http://arxiv.org/abs/1005.4159.

[37] A. Lumini and L. Nanni, 'Detector of image orientation based on Borda count', *Pattern. Recongn. Lett.*, **27**(3), 180–186, (2006).

[38] N. Mattei, M. S. Pini, F. Rossi, and K. B. Venable, 'Bribery in voting with CP-nets', *Ann. Math. Artif. Intell.*, **68**(1-3), 135–160, (2013).

[39] R. Meir, A. D. Procaccia, J. S. Rosenschein, and A. Zohar, 'Complexity of strategic behavior in multi-winner elections', *J. Artif. Intell. Res. (JAIR)*, **33**, 149–178, (2008).

[40] M. K. Miller and D. N. Osherson, 'Methods for distance-based judgment aggregation', *Soc. Choice. Welfare.*, 575–601, (2009).

[41] S. Obraztsova and E. Elkind, 'Optimal manipulation of voting rules', in *AAMAS*, pp. 619–626, (2012).

[42] M. S. Pini, F. Rossi, and K. B. Venable, 'Bribery in voting with soft constraints', in *AAAI*, pp. 803–809, (2013).

[43] Y. Yang, 'Manipulation with bounded single-peaked width: A parameterized study', in *AAMAS*, pp. 77–85, (2015).

[44] Y. Yang and J. Guo, 'The control complexity of r-Approval: from the single-peaked case to the general case', in *AAMAS*, pp. 621–628, (2014).

[45] Y. Yang and J. Guo, 'Controlling elections with bounded single-peaked width', in *AAMAS*, pp. 629–636, (2014).

[46] Y. Yang and J. Guo, 'How hard is control in multi-peaked elections: A parameterized study', in *AAMAS*, pp. 1729–1730, (2015).