Manipulation with Bounded Single-Peaked Width: A Parameterized Study

Yongjie Yang
Universität des Saarlandes
Campus E 1.7 (MMCI), D-66123 Saarbrücken, Germany
yyongjie@mmci.uni-saarland.de

ABSTRACT

We study the manipulation problem in elections with bounded single-peaked width from the parameterized complexity point of view. In particular, we focus on the Borda, Copeland$^\alpha$ and Maximin voting correspondences. For Borda, we prove that the unweighted manipulation problem with two manipulators is fixed-parameter tractable with respect to single-peaked width. For Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$, we prove that the unweighted manipulation problem is fixed-parameter tractable with respect to the combined parameter (k, t), where k denotes the single-peaked width and t denotes the number of manipulators. In addition, we study the weighted manipulation problem for Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ in single-peaked elections and achieve several polynomial-time solvability results.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; G.2.1 [**Combinatorics**]: Combinatorial algorithms; J.4 [**Computer Applications**]: Social Choice and Behavioral Sciences

General Terms

Algorithms

Keywords

single-peaked width; fixed-parameter tractable; parameterized complexity; Borda; Maximin; Copeland; weighted election

1. INTRODUCTION

Voting is a common method for preference aggregation and collective decision-making, and has applications in multi-agent systems [11], political elections, web spam reduction, pattern recognition, etc. For instance, in multiagent systems, it is often necessary for a group of agents to make a collective decision by means of voting in order to reach a joint goal. Unfortunately, by Arrow's impossibility theorem [1], there is no (rank-based) voting system which satisfies a certain set of desirable criteria (see [1] for the details) when more than two candidates are involved. One possible way to bypass Arrow's impossibility theorem is to restrict the domain of the preferences, for instance, the single-peaked domain introduced by Black [3]. Intuitively, in a single-peaked election, one can order the candidates from left to right such that every voter's preference increases first and then decreases after some point as the candidates are considered from left to right.

Recently, the complexity of various voting problems in single-peaked elections has been attracting attention of many researchers from both theoretical computer science and social choice communities [4, 13, 15]. It turned out that many voting problems being $\mathcal{NP}$-hard in general become polynomial-time solvable when restricted to single-peaked elections [4, 15]. However, most elections in practice are not purely single-peaked, which motivates researchers to study more general models of elections. We refer to [5, 8, 10, 12, 14] for some variants of the single-peaked model.

In this paper, we consider a newly introduced generalization of single-peaked elections, the so-called elections with bounded single-peaked width [7]. Other nearly single-peakedness concepts like κ-maverick, κ-global swaps, κ-candidate deletion, and multi-peaked elections have also been considered to cope with voting problems [5, 12, 14, 23, 25]. Cornaz et al. [7] first introduced single-peaked width into the context of complexity studies of voting problems. In particular, they considered a multi-winner determination problem and proved that this problem is fixed-parameter tractable ($\mathcal{FPT}$) with single-peaked width as parameter. Recall that a *parameterized problem* consists of instances of the form (I, κ), where I denotes the *main part* and κ is an integer *parameter*. A parameterized problem is $\mathcal{FPT}$ if it can be solved in $f(\kappa) \cdot |I|^{O(1)}$ time, where f is a computable function in the parameter κ. Later, Cornaz et al. [8] showed that the Kemeny winner determination is $\mathcal{FPT}$ with single-peaked width as parameter. Recently, Yang and Guo [24] studied control problems under Condorcet, Maximin and Copeland in elections with bounded single-peaked width. They showed that the destructive control problems (making someone not win the election by adding/deleting votes) are generally $\mathcal{FPT}$ with respect to single-peaked width, while the constructive control problems (making someone win the election by adding/deleting votes) are generally $\mathcal{NP}$-hard even when the single-peaked width is bounded by a small constant.

We mainly focus on the manipulation problem for Maximin, Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ and Borda. In the following, unless stated otherwise, manipulation refers to unweighted manipulation. In the *manipulation* problem, we are given a set of candidates including a distinguished candidate, a multiset of votes cast by the voters (nonmanipulators), and a set of manipulators who have not cast their votes yet. The question is whether the manipulators can cast their votes in a way so that the distinguished candidate becomes the winner. In the general case (the domain of the votes is not restricted), the manipulation problem for Maximin, Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ and Borda is polynomial-

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

		number of manipulators t		
		$t=1$	$t=2$	$t\geq 3$
Borda	Gen	$\mathcal{P}$	$\mathcal{NP}$-hard	
	SPW		$\boldsymbol{\mathcal{FPT}}$	?
Maximin	Gen	$\mathcal{P}$	$\mathcal{NP}$-hard	
	SPWNM		$\boldsymbol{\mathcal{FPT}}$	
Copeland$^\alpha$ $\alpha \in [0, 0.5) \cup (0.5, 1]$	Gen	$\mathcal{P}$	$\mathcal{NP}$-hard	
	SPWNM		$\boldsymbol{\mathcal{FPT}}$	
Copeland$^{0.5}$	Gen	$\mathcal{P}$	?	
	SPWNM		$\boldsymbol{\mathcal{FPT}}$	

Table 1: A summary of the complexity of the unweighted manipulation problem. Let k denote the single-peaked width. Here, 'Gen' should be read as 'the general case', 'SPW' should be read as "with respect to k", and 'SPWNM' should be read as "with respect to the combined parameter (k, t)". Moreover, '$\mathcal{P}$' stands for polynomial-time solvable. Our results are shown in bold. The polynomial-time solvability results are from [20], and the $\mathcal{NP}$-hardness results are from [2, 9, 16, 17, 20]. All the results shown in this table apply to both the unique-winner model and the non-unique winner model. Entries with '?' means that the corresponding problems are open.

	number of candidates m				
	General			Single-Peaked	
	$m=2$	$m=3$	$m\geq 4$	$m=3$	$m\geq 4$
Borda	$\mathcal{P}$	$\mathcal{NP}$-h	$\mathcal{NP}$-h	$\mathcal{P}$	$\mathcal{NP}$-h
Maximin	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{NP}$-h	$\mathcal{P}$	$\mathcal{P}$
Copeland^0	$\mathcal{P}$	$\mathcal{NP}$-h	$\mathcal{NP}$-h	$\boldsymbol{\mathcal{P}}$	$\boldsymbol{\mathcal{P}}$
Copeland^1	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$: $m=4$? : $m>4$	$\mathcal{P}$	$\mathcal{P}$
Copeland$^\alpha$ $0<\alpha<1$	$\mathcal{P}$	$\mathcal{NP}$-h: NON $\mathcal{P}$: UNI	$\mathcal{NP}$-h	$\boldsymbol{\mathcal{P}}$	$\boldsymbol{\mathcal{P}}$

Table 2: A summary of the complexity of the weighted manipulation problem. Here '$\mathcal{NP}$-h' stands for $\mathcal{NP}$-hard, and $\mathcal{P}$ stands for polynomial-time solvable. Moreover, 'NON' and 'UNI' in the entry in the last row means the nonunique-winner model and the unique-winner model, respectively. All the other results apply to both the unique-winner model and the nonunique-winner model. Our results are in bold. The polynomial-time solvability results for Maximin in single-peaked elections follow from several lemmas in [4]. Other results are from [4, 6, 15, 16, 18]. The entry with "?" means that the corresponding problem is open.

time solvable if there is only one manipulator [20]. However, if there are two manipulators all these problems, except the manipulation for Copeland$^{0.5}$, turned out to be $\mathcal{NP}$-hard [2, 9, 16, 17, 20]. To the best of our knowledge, the complexity of the Copeland$^{0.5}$ manipulation problem with two manipulators is still open. In the special case, Yang and Guo [22] proved that the Borda manipulation problem with two manipulators is polynomial-time solvable in single-peaked elections. In this paper, we explore the parameterized complexity of these problems. In particular, we prove that the manipulation problem with two manipulators for Borda is $\mathcal{FPT}$ with respect to single-peaked width. For Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$, we prove that the manipulation problem is $\mathcal{FPT}$, when parameterized by the combined parameter (k, t), where k denotes the single-peaked width and t the number of manipulators[1]. To this end, we derive several properties of elections with bounded single-peaked width. We believe that these properties are also helpful in solving further voting problems. Our results imply that the manipulation problem with any constant number of manipulators is polynomial-time solvable for Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$, in single-peaked elections, in contrast to the $\mathcal{NP}$-hardness of the problem in the general case. We remark in our analysis, the single-peaked width is based on all the votes; that is, the votes by the nonmanipulators union the votes by the manipulators. Moreover, all the above mentioned results apply to both the unique-winner model and the non-unique winner model (definition is in Section 2). Our results concerning the above problems are summarized in Table 1.

In addition, we study the weighted manipulation problem, where each voter (manipulator or nonmanipulator) has a non-negative integer weight, in single-peaked elections. Conitzer et al. [6] proved that the weighted manipulation problem with four or more candidates for Maximin is $\mathcal{NP}$-hard in general. Faliszewski et al. [16] studied the weighted manipulation for Copeland$^\alpha$. They proved that the problem with three candidates is $\mathcal{NP}$-hard for Copeland$^\alpha$ with $0 \leq \alpha < 1$, except for the unique-winner model for Copeland$^\alpha$ for every $0 < \alpha < 1$ which is polynomial-time solvable. Moreover, the weighted manipulation problem with three candidates is polynomial-time solvable for Copeland^1 [16]. We discuss the unique-winner model and the nonunique-winner model in detail latter. Unless stated otherwise, all the results mentioned throughout this paper apply to both the unique-winner and the nonunique-winner models. In this paper, we show that the weighted manipulation for both Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ in single-peaked elections is polynomial-time solvable even when the number of candidates is not bounded by a constant. Table 2 summarizes the complexity of the above problems.

2. PRELIMINARIES

Elections: An *election* is a tuple $\mathcal{E} = (\mathcal{C}, \mathcal{V})$, where $\mathcal{C}$ is a set of candidates and $\mathcal{V}$ is a multiset of votes (for convenience, the terminologies "vote" and "voter" are used interchangeably throughout this paper), each of which is defined as a linear order $\succ$ over $\mathcal{C}$. For two candidates c and c' and a vote $\succ$, we say c is *ranked above* c' or $\succ$ prefers c to c' if $c \succ c'$. We use $N_{\mathcal{E}}(c, c')$ to denote the number of votes ranking c above c' in $\mathcal{E}$. We drop the index $\mathcal{E}$ if it is clear from the context. We say c *beats* c' if $N(c, c') > N(c', c)$, and c ties c' if $N(c, c') = N(c', c)$. For two subsets C and C' of candidates, $C \succ C'$ means that every candidate in C is ranked above every candidate in C' in the vote $\succ$. A *voting correspondence*[2] φ is

[1] An instance of a parameterized problem with combined parameter (κ_1, κ_2) can be considered as an instance of the same problem with the single parameter $\kappa = \kappa_1 + \kappa_2$.

[2] A related terminology is *voting rule* which is defined as a function mapping an election to a single candidate. A voting correspondence can be modified to a voting rule using a tie-breaking method.

a function that maps an election $\mathcal{E} = (\mathcal{C}, \mathcal{V})$ to a nonempty subset $\varphi(\mathcal{E})$ of $\mathcal{C}$. We call the elements in $\varphi(\mathcal{E})$ the *winners* of $\mathcal{E}$. If $\varphi(\mathcal{E})$ contains only one winner, we call it a *unique winner*; otherwise, we call them *co-winners*. For an election $\mathcal{E} = (\mathcal{C}, \mathcal{V})$ and a subset $C \subseteq \mathcal{C}$, we use $\mathcal{E}|_C$ to denote the election restricted to C. Precisely, the restricted election $\mathcal{E}|_C$ has C as the candidate set, and the votes of $\mathcal{E}|_C$ are obtained from $\mathcal{E}$ by replacing each vote $\succ$ of $\mathcal{E}$ by a new vote $\succ'$ such that for every two candidates $a, b \in C$, $a \succ' b$ whenever $a \succ b$.

Single-Peaked Width: An election $(\mathcal{C}, \mathcal{V})$ is *single-peaked* if there is an order $\mathcal{L}$ of $\mathcal{C}$, from left to right, such that for every vote $\succ$ and every three candidates $a, b, c \in \mathcal{C}$ with $a \mathcal{L} b \mathcal{L} c$ or $c \mathcal{L} b \mathcal{L} a$, $c \succ b$ implies $b \succ a$, where $a \mathcal{L} b$ means a lies on the left-side of b in $\mathcal{L}$. We call $\mathcal{L}$ a *harmonious order*. See Figure 1 for an example.

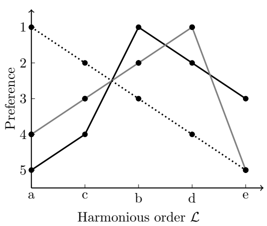

Figure 1: A single-peaked election with three votes (1) $b \succ_u d \succ_u e \succ_u c \succ_u a$; (2) $d \succ_v b \succ_v c \succ_v a \succ_v e$; and (3) $a \succ_w c \succ_w b \succ_w d \succ_w e$. The votes $\succ_u$, $\succ_v$ and $\succ_w$ are illustrated by the dark line, the gray line, and the dotted line, respectively.

A subset $C \subseteq \mathcal{C}$ is called an *interval* if all candidates in C are ranked contiguously in every vote. For example, for the election with candidates $\{a, b, c, d, e\}$ and votes $\{a \succ_1 b \succ_1 c \succ_1 d \succ_1 e,\ d \succ_2 c \succ_2 b \succ_2 e \succ_2 a,\ a \succ_3 e \succ_3 b \succ_3 d \succ_3 c\}$, $\{b, c, d\}$ is an interval. *Contracting* an interval C is the operation that first adds a new candidate c' to the election such that $C \cup \{c'\}$ forms a new interval and the preference between any two candidates of $\mathcal{C}$ in each vote preserves the same as before, and then deletes all candidates in C. For example, after contracting the interval $\{b, c, d\}$ in the above example, we get the new election with candidates a, c', e and votes $\{a \succ_1 c' \succ_1 e,\ c' \succ_2 e \succ_2 a,\ a \succ_3 e \succ_3 c'\}$, where c' is the newly introduced candidate.

Let $P = (C_1, C_2, ..., C_\omega)$ be an ordered partition of $\mathcal{C}$ with each C_i being an interval. We say P is a *single-peaked partition* if contracting all intervals in P results in a single-peaked election with respect to the harmonious order $(c_1, c_2, ..., c_\omega)$, where each c_i is the newly introduced candidate for the interval C_i. We say a vote *has its peak at C_i with respect to P* if the interval C_i is ranked above every other interval by the vote. The *width* of P is defined as $\max_{1 \le i \le \omega}\{|C_i|\}$[3]. The *single-peaked width* of an election is the minimum width among all its single-peaked partitions. Cornaz et al. [8] proved that calculating the single-peaked width of an election and constructing an optimal single-peaked partition can be done in polynomial time.

Median Group: Let $P = (C_1, C_2, ..., C_\omega)$ be a single-peaked partition of the election $(\mathcal{C}, \mathcal{V})$, and let $(\succ_1, \succ_2, ..., \succ_n)$ be an order of $\mathcal{V}$ such that for every i, j with $1 \le i < j \le n$ the peak of $\succ_i$ does not lie on the right-side of the peak of $\succ_j$ in P. The set of all intervals lying between the peak C_l of $\succ_{\lceil n/2 \rceil}$ and the peak C_r of $\succ_{\lfloor n/2+1 \rfloor}$, together with C_l and C_r, denoted $\mathcal{G}[C_l, C_r]$, is called the *median group*. If there is only one interval in the median group, we call it a *median interval*. See Figure 2 for an example.

Voting Correspondences: We mainly study the following voting correspondences.

[3]Cornaz et al. [7] defined the width of the partition P as $\max_{1 \le i \le \omega}\{|C_i|\} - 1$, the size of the maximum group minors one. However, this does not affect the results of this paper.

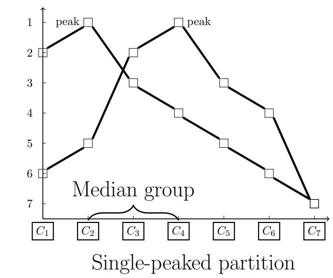

Figure 2: An illustration of median group. There are two votes, where the first vote has preference $C_2 \succ C_1 \succ C_3, ..., \succ C_7$ over the intervals, and the second vote has the preference $C_4 \succ C_3 \succ C_5 \succ C_6 \succ C_2 \succ C_1 \succ C_7$. The peak C_2 of the first vote is on the left side of the peak C_4 of the second vote.

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

Copeland$^\alpha$ $(0 \le \alpha \le 1)$**:** 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.

Problem Definitions. In the *unweighted manipulation problem* studied in this paper, we are given an election $\mathcal{E} = (\mathcal{C} \cup \{p\}, \mathcal{V})$ with single-peaked width k, an optimal single-peaked partition P and a set of voters who have not cast their votes yet. Here, p is the distinguished candidate. We call the set of voters who have not cast their votes the *manipulators*. The question is whether the manipulators can cast their votes according to the single-peaked partition P so that the distinguished candidate p becomes the winner under a specific voting correspondence, e.g., Maximin, Copeland$^\alpha$ and Borda. In the *weighted manipulation problem*, each voter (manipulator or nonmanipulator) has a nonnegative integer weight. Each vote defined as $\succ$ and with weight w is regarded as w individual votes each of which is defined as $\succ$. The assumption that the single-peaked partition is given in the input is based on the observation that in many real-world applications, the single-peaked partition is known in advance. This is actually one of the reasons why domain restricted elections can arise in practice. For example, in real-world single-peaked political elections, the voters are thought to agree upon that the candidates are ordered on a common known left-right dimension. See [3] for related discussion. Moreover, in this scenario, if the manipulators do not cast their votes according to the given single-peaked partition, they will be easily recognized as manipulators. See also [4, 14, 15, 19] for further study of manipulation problems where the domain of the manipulative votes is restricted.

Remark. All our results apply to both the unique-winner and the nonunique-winner models. In the unique-winner model, the objective is to make the distinguished candidate the unique winner, while in the nonunique-winner model the objective is to make the distinguished candidate a winner (that is, either a unique winner or a co-winner). For simplicity, all our proofs and algorithms are solely based on the unique-winner model.

3. MAXIMIN

In this section, we explore the parameterized complexity of the manipulation problem for Maximin. In particular, we prove that the manipulation problem for Maximin is $\mathcal{FPT}$, when parameterized by the combined parameter (k, t), where k is the single-peaked width and t is the number of manipulators. To this end, we introduce some properties of Maximin elections with bounded single-peaked width. These properties are also helpful in understanding the behavior of the Maximin correspondence. The first property is formally stated in Lemma 1. In an informal way, it states that for each candidate c, the closer another candidate c' lies to c according to the single-peaked partition, the less is the number of voters who prefer c to c'. For a positive integer n, let $[n]$ be the set $\{1, 2, ..., n\}$.

LEMMA 1. *Let $(C_1, C_2, ..., C_\omega)$ be the single-peaked partition and c be a candidate in a certain interval C_i. Then, $N(c, b_1) \leq N(c, b_2)$ for all $b_1 \in C_{x_1}$ and $b_2 \in C_{x_2}$ with $i < x_1 < x_2$, and $N(c, a_1) \leq N(c, a_2)$ for all $a_1 \in C_{z_1}$ and $a_2 \in C_{z_2}$ with $z_2 < z_1 < i$.*

PROOF. We first prove the first part of the claim. Let b_1 and b_2 be the two candidates as stated in the claim. For all $j \in [\omega]$, we denote the set of votes with peaks at C_j or on the right-side of C_j by $\mathcal{V}_j^r$, and denote the set of votes with peaks at C_j or on the left-side of C_j by $\mathcal{V}_j^l$. It is obvious that all votes in $\mathcal{V}_i^l$ prefer c to b_1 to b_2 and all votes in $\mathcal{V}_{x_1}^r$ prefer b_1 to c. Let $\mathcal{V}_{i,x_1}^{c \succ b_1}$ be the set of votes with peaks between C_i and C_{x_1} and prefer c to b_1. Thus, $N(c, b_1) = |\mathcal{V}_i^l| + |\mathcal{V}_{i,x_1}^{c \succ b_1}|$. Due to the definition of single-peaked partition, all votes in $\mathcal{V}_{i,x_1}^{c \succ b_1}$ prefer c to b_2. Therefore, $N(c, b_2) \geq |\mathcal{V}_i^l| + |\mathcal{V}_{i,x_1}^{c \succ b_1}| = N(c, b_1)$.

Due to symmetry, the second part is also correct. □

Recall that the Maximin score of a candidate c is equal to $N(c, c')$ where c' achieves the minimum value of $N(c, \cdot)$. Let c be a candidate from a certain interval C_i. Let $MIN(c)$ be the set of candidates that achieve the minimum value of $N(c, \cdot)$; hence, we have that $Maximin(c) = N(c, c')$ for every $c' \in MIN(c)$. According to Lemma 1, we have that $(C_{i-1} \cup C_i \cup C_{i+1}) \cap MIN(c) \neq \emptyset$. Therefore, to determine the Maximin score of c, it is sufficient to consider the election restricted to $C_{i-1} \cup C_i \cup C_{i+1}$ whose size is bounded by $3k$, where k is the single-peaked width. In the following, we introduce another property which helps to improve the upper bound.

LEMMA 2. *Let c be a candidate and C' be an interval with $c \notin C'$. Then, $N(c, a) = N(c, b)$ for every two candidates $a, b \in C'$.*

PROOF. Since C' is an interval, all votes rank the candidates in C' contiguously. Therefore, each vote either prefers c to all candidates in C' or prefers all candidates in C' to c, implying that for every two candidates $a, b \in C'$, $N(c, a) = N(c, b)$. □

According to Lemmas 1 and 2, the Maximin score of a candidate c is determined by all candidates in the interval including c, together with any two arbitrary candidates from the two neighbor intervals of the interval including c, one from each.

LEMMA 3. *Let $\mathcal{E} = (\mathcal{C}, \mathcal{V})$ be an election with single-peaked partition $P = (C_1, C_2, ..., C_\omega)$ and c be a candidate in an interval C_i. Then the Maximin score of c in $\mathcal{E}$, denoted by Maximin$_\mathcal{E}(c)$, is*

$$\textit{Maximin}_\mathcal{E}(c) = \textit{Maximin}_{\mathcal{E}|C_i \cup \{a,b\}}(c)$$

Here, a and b are any two arbitrary candidates in C_{i-1} and C_{i+1}, respectively (only b appears if $i = 1$ and only a appears if $i = \omega$).

It is clearly true that in the general case the optimal choice for the manipulators is to rank the distinguished candidate in the top. This is because of the monotonicity of Maximin. Recall that a voting correspondence τ is *monotonic* if in every τ election (that is, election where winners are selected according to τ), ranking a winner higher in some vote does not exclude the winner from the winning set [20]. However, when the election has a bounded single-peaked width, the correctness of the statement is not straightforward anymore, since improving one's position in a vote may destroy the single-peakedness. In the following, we provide a formal proof to support the above claim in elections with bounded single-peaked width.

LEMMA 4. *Every Yes-instance of the manipulation problem for Maximin in elections with bounded single-peaked width has a solution where all manipulators rank the interval C_p including the distinguished candidate p above every other interval. Moreover, p is ranked above every other candidate.*

PROOF. We prove this lemma by showing that if there is a solution which does not satisfy the lemma, we can construct another solution which satisfies the lemma. Observe first that it is always optimal to rank p above every other candidate in C_p, since there is no single-peaked restriction inside C_p. Therefore, it is sufficient to prove that ranking C_p in the top is the optimal choice for all the manipulators.

Assume that v is a manipulator who did not rank the interval C_p in the top. Let $(L_a, L_{a-1}, ..., L_1, C_p, R_1, ..., R_b)$ be the single-peaked partition. Let $\mathcal{C}_l = \{L_a, L_{a-1}, ..., L_1\}$ be the set of intervals on the left-side of C_p according to the single-peaked partition and $\mathcal{C}_r = \{R_1, ..., R_b\}$ be the set of intervals on the right-side of C_p. Without loss of generality, assume that $a, b \geq 1$, that is, $\mathcal{C}_l, \mathcal{C}_r \neq \emptyset$. Furthermore, assume that the manipulator v ranked some interval $L \in \mathcal{C}_l$ in the top. Due to the single-peakedness, the manipulator v has the following preference over the intervals in $C_p \cup \mathcal{C}_r$: $C_p \succ R_1 \succ R_2 \succ, ..., \succ R_b$. We consider two cases.

The first case is that for each $L \in \mathcal{C}_l$, $L \succ C_p$. In this case, we can create a new solution by recasting the vote of v with $\succ'$ defined as $C_p \succ' L_1, ..., \succ' L_a \succ' R_1 \succ', ..., \succ' R_b$. Here, the preference between every two candidates in the same interval preserves the same as before. That is, for every two candidates c and c' in the same interval, c is ranked above c' in the new vote whenever c is ranked above c' in the original vote.

The second case is that there is an $L \in \mathcal{C}_l$ with $L \prec C_p$. In this case, there must be a $z \in [a]$ such that $L_j \succ C_p$ for all $j \in [z-1]$ and $L_j \prec C_p$ for all $a \geq j \geq z$. Let $\widetilde{\mathcal{C}} = \mathcal{C}_r \cup \{L_z, L_{z+1}, ..., L_a\}$. We can get a new solution by recasting the vote $\succ$ of v as $\succ'$ with preference $C_p \succ' L_1 \succ', ..., \succ' L_{z-1} \succ' \widetilde{\mathcal{C}}$, where among $\widetilde{\mathcal{C}}$, we have $C \succ' C'$ if and only if $C \succ C'$ for every $C, C' \in \widetilde{\mathcal{C}}$. Moreover, the preference between every two candidates in the same interval preserves the same as before. See Figure 3 for an illustration.

Now we prove the correctness. Let $\mathcal{E} = (\mathcal{C} \cup \{p\}, \mathcal{V})$ be the original election and $\mathcal{E}' = (\mathcal{C} \cup \{p\}, \mathcal{V}')$ be the election obtained from $\mathcal{E}$ by replacing the vote $\succ \in \mathcal{V}$ by $\succ'$, as discussed above. We need to prove $Maximin_{\mathcal{E}'}(p) > Maximin_{\mathcal{E}'}(c)$ for every $c \in \mathcal{C}$, given that $Maximin_\mathcal{E}(p) > Maximin_\mathcal{E}(c)$. For a candidate c, let $MIN(c)$ be the set of candidates c' such that $Maximin_\mathcal{E}(c) = N_\mathcal{E}(c, c')$. Let $peak(c, right)$ be the number of votes with peaks not on the left side of the interval including c. We prove for the second case (the proof for the first case is analogous). Let

$$\mathcal{C}' = \mathcal{C}_l \setminus \widetilde{\mathcal{C}} = \{L_1, ..., L_{z-1}\}.$$

Observe first that with recasting $\succ$, only the candidates in $\mathcal{C}' \cup C_p$ have chance to increase their scores. Moreover, each candidate can increase his score by at most one. Therefore, it is sufficient

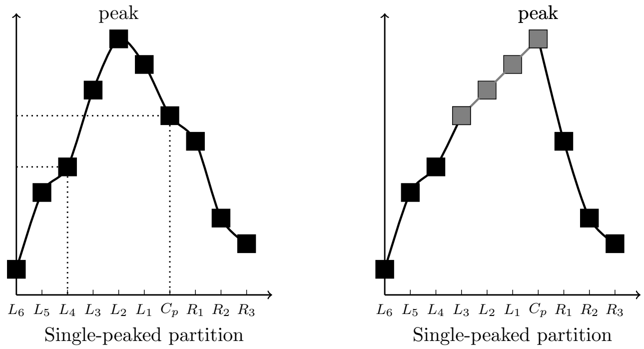

Figure 3: Case 2 in the proof of Lemma 4. The figure on the left side shows the original vote $\succ$ with $L_2 \succ L_1 \succ L_3 \succ C_p \succ R_1 \succ L_4 \succ L_5 \succ R_2 \succ R_3 \succ L_6$. The figure on the right side shows the new recast vote $\succ'$ with $C_p \succ' L_1 \succ' L_2 \succ' L_3 \succ' R_1 \succ' L_4 \succ' L_5 \succ' R_2 \succ' R_3 \succ' L_6$. Here we have $\widetilde{\mathcal{C}} = \{L_4, L_5, L_6, R_1, R_2, R_3\}$.

to prove that $Maximin_{\mathcal{E}'}(p) > Maximin_{\mathcal{E}'}(c)$ for every $c \in \mathcal{C}' \cup C_p$. This clearly holds for every candidate in C_p, since the new vote preserves the preference between every two candidates in C_p. We prove the correctness for every candidate in $\mathcal{C}'$ by contradiction. Suppose that $c \in L_i$ with $L_i \in \mathcal{C}'$ is a candidate with $Maximin_{\mathcal{E}}(p) > Maximin_{\mathcal{E}}(c)$ but $Maximin_{\mathcal{E}'}(p) \leq Maximin_{\mathcal{E}'}(c)$. Since recasting the vote $\succ$ does not decrease the score of p, we know that the score of c is increased by one after recasting the vote $\succ$. This happens only if $L_i \cap MIN(c) = \emptyset$ and $L_{i+1} \cap MIN(c) \neq \emptyset$. In this case, $Maximin_{\mathcal{E}}(c) = N_{\mathcal{E}}(c, c') = peak(c, right)$, where c' is any arbitrary candidate in L_{i+1} (the first equation is due to Lemma 2 and the second is due to the single-peakedness). Let c'' be any arbitrary candidate in L_1. Then, due to the definition of the Maximin correspondence, $Maximin_{\mathcal{E}}(p) \leq N_{\mathcal{E}}(p, c'') = peak(p, right) \leq peak(c, right) = Maximin_{\mathcal{E}}(c)$, contradicting with the fact that $Maximin_{\mathcal{E}}(p) > Maximin_{\mathcal{E}}(c)$ for every $c \in \mathcal{C}$. The lemma is proved. $\square$

Now we are ready to show the main result of this section.

THEOREM 1. *The manipulation problem for Maximin is $\mathcal{FPT}$ with respect to the combined parameter (k, t), where k is the single-peaked width and t is the number of manipulators.*

PROOF. We prove the theorem by proposing an $\mathcal{FPT}$ algorithm. The algorithm ranks all the intervals first and then ranks the candidates in each interval.

Let $(C_1, C_2, ..., C_i, ..., C_\omega)$ be the single-peaked partition with width k, and C_i be the interval including the distinguished candidate p. Due to Lemma 4, all manipulators can safely rank C_i in the top. Let c be any arbitrary candidate. Without loss of generality, assume c is in the interval C_j. Then, for every candidate $c' \in C_{j-1} \cup C_{j+1}$, $N(c, c')$ is known. Precisely, $N(c, c') = |\mathcal{V}_j^l|$ if $c' \in C_{j+1}$, and $N(c, c') = |\mathcal{V}_j^r|$ otherwise. Here, $\mathcal{V}_j^l$ (resp. $\mathcal{V}_j^r$) is the set of votes with peaks at C_j or on the left-side (resp. right-side) of C_j. Due to Lemma 3 and the above analysis, the Maximin score of c is $\min\{|\mathcal{V}_j^l|, |\mathcal{V}_j^r|, \text{Maximin}_{\mathcal{E}|C_j}(c)\}$. Since $\text{Maximin}_{\mathcal{E}|C_j}(c)$ does not depend on how the manipulators rank the intervals, all manipulators can safely rank the intervals in any way which is consistent with the single-peaked partition. For example all the manipulators can rank the intervals as follows.

$$C_i \succ C_{i-1} \succ C_{i-2} \succ, ..., \succ C_1 \succ C_{i+1} \succ C_{i+2} \succ, ..., \succ C_\omega$$

It remains to rank the candidates in each interval. Let t be the number of manipulators. We begin with the interval C_i including p. We enumerate all the possible combinations of t linear orders over the candidates in C_i, each linear order is assumed to be the partial vote over C_i cast by a manipulator. Since $|C_i| \leq k$, there are at most $k!^t$ combinations. Moreover, each combination gives p a Maximin score by adding the partial votes corresponding to the t linear orders of the combination to the election. The algorithm chooses one combination which gives p the maximum Maximin score in the election restricted to C_i. Then, the manipulators rank the candidates in C_i according to the t linear orders of the chosen combination. Now the final Maximin score of p is known. It remains to rank the candidates in other intervals. We use a similar method. In particular, for each remaining interval C, we enumerate all possibilities of ranking the candidates in C until we find one which does not prevent p from being the winner. If for every possibility there is a candidate in C which has an equal or greater score than that of p, we immediately return "No"; otherwise, we proceed with the next interval. The algorithm runs in $O(\omega \cdot k!^t)$ time since ranking the candidates in each interval takes $k!^t$ time and we have ω intervals to consider. $\square$

4. COPELAND

In this section, we study the manipulation problem for Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$. In particular, we prove that the manipulation problem for Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ is $\mathcal{FPT}$, when parameterized by the combined parameter (k, t), where k is the single-peaked width and t is the number of manipulators. We start with some useful properties.

LEMMA 5. *Every Yes-instance of the manipulation problem for Copeland$^\alpha$ in elections with bounded single-peaked width has a solution where all manipulators rank the interval including the distinguished candidate in the top.*

The proof for the above lemma is similar to the one for Lemma 4. We omit the proof, due to space limitations.

The following lemma states that the Copeland$^\alpha$ scores of the candidates in different intervals strictly increase when the intervals are considered from either side to the median group.

LEMMA 6. *Let $\mathcal{G}[C_l, C_r]$ be the median group of an election with candidates set $\mathcal{C}$, with respect to the single-peaked partition $(C_1, C_2, ..., C_\omega)$. Let $a_1 \in C_{z_1}, a_2 \in C_{z_2}, b_1 \in C_{x_1}, b_2 \in C_{x_2}$ be any four arbitrary candidates with $z_2 < z_1 \leq l$ and $r \leq x_1 \leq x_2$. Then, the Copeland$^\alpha$ score of b_1 (resp. a_1) is strictly greater than that of b_2 (resp. a_2), for every $0 \leq \alpha \leq 1$.*

PROOF. Due to symmetry, we prove only for b_1 and b_2. Let $\mathcal{C}_1$ be the set of candidates included in intervals on the right-side of C_{x_1}. Clearly, $b_2 \in \mathcal{C}_1$. Since all votes with peaks at C_r or on the left-side of C_r, which amount to more than half of the votes, rank every candidate in C_{x_1} above every candidate in $\mathcal{C}_1$, we know that every candidate in C_{x_1} beats every candidate in $\mathcal{C}_1$. Thus, b_1 beats every candidate in $\mathcal{C}_1$, implying that the candidates in $\mathcal{C}_1$ contribute at least one more point (from b_2) to b_1 than to b_2. Now consider the candidates in $\mathcal{C}_2 = \mathcal{C} \setminus (\mathcal{C}_1 \cup C_{x_1})$. That is, the candidates included in intervals on the left-side of C_{x_1}. Due to the definitions of single-peaked election and single-peaked partition, for every candidate $c \in \mathcal{C}_2$, every vote which prefers b_2 to c also prefers b_1 to c. Thus, if b_2 beats (resp. ties) a candidate $c \in \mathcal{C}_2$, so does b_1 (resp.

b_1 beats c or ties c). Thus, the candidates in $\mathcal{C}_2$ contribute to b_1 at least the same points as to b_2. Since every candidate in C_{x_1} beats b_2, the lemma follows. $\square$

Due to Lemma 6, we know that for every candidate c which is not in the median group, there exists at least one candidate who has a strictly greater Copeland$^\alpha$ score than that of c. This implies that the Copeland$^\alpha$ winners must be included in the median group.

LEMMA 7. *Every Copeland$^\alpha$ winner for all $0 \leq \alpha \leq 1$ is from the median group.*

Now we come to the main result of this section.

THEOREM 2. *The manipulation problem for Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ is $\mathcal{FPT}$ with respect to the combined parameter (k, t), where k denotes the single-peaked width and t the number of manipulators.*

PROOF. To prove the theorem, we derive an $\mathcal{FPT}$-algorithm. Let t be the number of manipulators and k be the single-peaked width of the given election. The algorithm first ranks the interval C including the distinguished candidate p in the top in all the manipulative votes. If p is not in the median group, we can immediately return "No", due to Lemma 7. Otherwise, we enumerate all possible combinations of t linear orders over the candidates in C. Since C has at most k candidates, there are at most $k!^t$ combinations. Moreover, each combination of t linear orders gives a Copeland$^\alpha$ score of p in the election restricted to C by asking the t manipulators to rank the candidates in C according to the t linear orders of the combinations. The algorithm chooses one combination which gives p the maximum Copeland$^\alpha$ score in the election restricted to C. Then, the manipulators rank the candidates in C according to the linear orders in the combination. Without loss of generality, assume that the single-peaked partition is $(C_1, C_2, ..., C_l, ..., C_r, C_{r+1}, ..., C_\omega)$, and the median group $\mathcal{G}[C_l, C_r]$ contains C_l, C_r and all intervals between C_l and C_r (C_l and C_r may be identical. In this case the algorithm becomes easier). Since all manipulators rank C in the top and C is in the median group, we have that either $p \in C_l$ or $p \in C_r$ (that is, $C = C_l$ or $C = C_r$). Without loss of generality, assume that $p \in C_l$. Due to Lemma 7, to make p the winner, we need only to make sure that every candidate included in $\mathcal{G}[C_l, C_r]$ has no equal or greater score than that of p. Hence, the optimal solution for all manipulators is to rank the intervals as follows.

$$C_l \succ C_{l-1} \succ, ..., C_1 \succ C_{l+1} \succ, ..., \succ C_\omega$$

In this case, every candidate in $\mathcal{G}[C_l, C_r] \setminus C_l$ gets the least points from the candidates in $\bigcup_{i=1}^{i=l-1} C_i$. It remains to rank the candidates in each interval. Due to Lemmas 6 and 7, for each interval which is not in $\mathcal{G}[C_l, C_r]$, no matter how the candidates in this interval are ranked, none of the candidates can have an equal or greater score than that of p. Hence, we rank them arbitrarily. For each interval in $\mathcal{G}[C_l, C_r] \setminus \{C\}$, we rank the candidates with the similar method as for Maximin. That is, for each interval $C' \in \mathcal{G}[C_l, C_r] \setminus \{C\}$, we enumerate all the combinations of t linear orders over the candidates in C' until we find one which does not prevent p from being the winner. On the other hand, if no such combinatorial exists, we return "No". The running time of the algorithm is $O(\omega \cdot k!^t)$. $\square$

5. BORDA

In this section, we study the Borda manipulation with two manipulators in elections with bounded single-peaked width. Recall that this problem is $\mathcal{NP}$-hard in general [2, 9] but polynomial-time solvable in single-peaked elections [22].

THEOREM 3. *The manipulation problem with two manipulators for Borda is $\mathcal{FPT}$ when parameterized by single-peaked width.*

Our $\mathcal{FPT}$ algorithm is based on the polynomial-time algorithm for the same problem in single-peaked elections [22]. The algorithm ranks the intervals according to the single-peaked partition, beginning with the one including the distinguished candidate and ending with a one on either side. To rank each interval, the algorithm first assigns respective contiguous positions for the interval, then ranks the candidates in the interval in a brute-force way. The procedure of ranking the intervals mimics the algorithm for Borda manipulation in single-peaked elections in [22], and thus takes polynomial time. Since each interval contains at most k candidates (k is the single-peaked width of the given election), ranking candidates in each interval takes $O(k!^2)$ time (each manipulator has at most $k!$ choices and there are two manipulators). The whole running time of the algorithm will be $O(k!^2 \cdot poly(|\mathcal{E}|))$, where $|\mathcal{E}|$ is the size of the given election[4].

Main Idea. We illustrate the algorithm according to Figure 4. In the first step, the interval C_p including p is ranked in the top of the two manipulative votes, and p is ranked in the top within C_p. The final score of p is known now. Assume that p is the current winner. Then we check all possibilities (at most $(k-1)!^2$) of ranking the candidates in $C_p \backslash \{p\}$ until we find one case which does not prevent p from being the winner. After this, due to the single-peakedness, only R_1 or L_1 can be ranked in the next free positions. We check whether at least one of them can be ranked in the next free positions of the two manipulative votes simultaneously, without preventing p from being the winner. This can be done in $k!^2$ time by enumerating all possibilities. Suppose that R_1 can be ranked in this way as shown in Figure 4. Then, again due to the single-peakedness, only R_2 or L_1 can be ranked in the next free positions. We do the same thing for these two intervals as discussed above for R_1 and L_1. Differently, suppose that at this time none of R_2 and L_1 can be ranked simultaneously without preventing p from being the winner. Then, if the given instance is a Yes-instance, the only possible case is that each of R_2 and L_1 is ranked in the next free positions of different manipulative votes, as shown in Figure 4. Note that at this moment we do not know how the manipulators rank the candidates in R_2 and L_1. We will rank them as follows. In fact, our algorithm will always do the following once there is an interval which has been ranked by one manipulator but not by the other one. Let's take R_2 as an example. In this case, we are going to rank R_2 in the highest possible free contiguous positions of the second manipulative vote. To this end, for all free contiguous positions, from the highest to the lowest, we check whether R_2 can be ranked in these positions so that p is still the winner. Each case can be checked in $k!^2$ time as discussed above. If no such case exists, the instance must be a No-instance. Suppose that R_2 is ranked as shown in Figure 4 without preventing p from being the winner. Then due to the single-peakedness, the free positions between L_1 and R_2 can only be occupied by L_2, L_3 and so forth (the interval R_2 may need to be moved to lower contiguous positions if there are no enough positions for L_4. This does not change the solvability.). We do the same thing for each interval which has been ranked by exactly one manipulator until none of them exists. Then, either we find a solution or the next free positions of the two manipulative votes are "neat" (the set of intervals that have ranked in the first manipulative vote is

[4]By employing a similar dynamic programming technique as in [22], ranking candidates in each interval can be done in $O(8^k)$ time. Hence, the whole running time can be improved to $O(8^k \cdot poly(|\mathcal{E}|))$.

the same as that in the second manipulative vote). If it is the latter case, we go back to the step where we shall consider whether one interval can be ranked in the next free positions simultaneously as discussed in the beginning of the algorithm. A formal description of the algorithm is given in Algorithm 1.

Algorithm 1: The $\mathcal{FPT}$ algorithm for the unweighted Borda manipulation with two manipulators in elections with bounded single-peaked width.

1 Both manipulators rank p in their highest positions;
2 **if** $C_p \setminus \{p\} \rightarrow \{(\pi_1, 2, |C_p|), (\pi_2, 2, |C_p|)\}$ **then**
3   extend(S_1, C_p) and extend(S_2, C_p);
4 **else**
5   Return "NO";
6 **end**
7 **while** $\bigcup_{s \in S_1} s = \bigcup_{s \in S_2} s \neq \mathcal{C} \cup \{p\}$ **do**
8     $S := S_1$;
9   **if** $\exists B \in N(S)$ *with* $B \rightarrow \{(\pi_1, |S|+1, |S|+|B|), (\pi_2, |S|+1, |S|+|B|)\}$ **then**
10   extend(S_1, B) and extend(S_2, B);
11   **end**
12   **else if** $|N(S)| = 1$ **then**
13   Return "No";
14   **end**
15   **else**
16   Let $N(S) = \{B, B'\}$;
17   **if** $B \rightarrow (\pi_1, |S_1|+1, |S_1|+|B|)$ *and* $B' \rightarrow (\pi_2, |S_2|+1, |S_2|+|B'|)$ **then**
18   extend(S_1, B) and extend(S_2, B');
19   **else**
20   Return "No";
21   **end**
22   **end**
23   **while** $S_1 \neq S_2$ **do**
24   Let B be any interval in $N(S_1 \cap S_2)$;
25   Let $z = 1$ if $B \in S_2$ and $z = 2$ if $B \in S_1$;
26   **while** $B \not\rightarrow (\pi_z, |S_z|+1, |S_z|+|B|)$ **do**
27   **if** $N(S_z) \setminus B = \emptyset$ **then**
28   Return "No";
29   **end**
30   **else**
31                 $B' := N(S_z) \setminus B$;
32   **if** $B' \rightarrow (\pi_z, |S_z|+1, |S_z|+|B'|)$ **then**
33   extend(S_z, B');
34   **else**
35   Return "NO";
36   **end**
37   **end**
38   **end**
39   extend(S_z, B);
40   **end**
41 **end**
42 Return "Yes";

We use functions $\pi : \mathcal{C} \cup \{p\} \mapsto [|\mathcal{C} \cup \{p\}|]$ that map candidates to positions to represent a vote. In particular, π_1 and π_2 will be the first and the second manipulative votes, respectively. A candidate c with $\pi(c) = 1$ has the highest position and thus gets the maximum $|\mathcal{C}|$ points. Initially, we set $\pi_i(c) = 0$ for every candidate c.

For an interval C and two integers k_l and k_r with $1 \leq k_l \leq k_r \leq |\mathcal{C}|$ and $|C| = k_r - k_l + 1$, we use $C \rightarrow (\pi, k_l, k_r)$ (resp. $C \not\rightarrow (\pi, k_l, k_r)$) to denote that the candidates in C can (resp. cannot) be safely ranked in the positions $\{k_l, k_l + 1, ..., k_r\}$ of π. Here, "safely" means that it is possible to rank the candidates in these positions of π without preventing p from being the winner. Similarly, we use $C \rightarrow \{(\pi_1, k_l^1, k_r^1), (\pi_2, k_l^2, k_r^2)\}$ (resp. $C \not\rightarrow \{(\pi_1, k_l^1, k_r^1), (\pi_2, k_l^2, k_r^2)\}$) to denote that it is possible (resp. not possible) to rank the candidates in C in the positions $\{k_l^1, k_l^1 + 1, ..., k_r^1\}$ of π_1 and in the positions $\{k_l^2, k_l^2 + 1, ..., k_r^2\}$ of π_2 simultaneously, without preventing p from being the winner. Checking whether $C \rightarrow (\pi, k_l, k_r)$ can be done in $|C|!$ time by enumerating all the linear orders over C, and checking whether $C \rightarrow \{(\pi_1, k_l^1, k_r^1), (\pi_2, k_l^2, k_r^2)\}$ can be done in $|C|!^2$ by enumerating all the two linear orders over C.

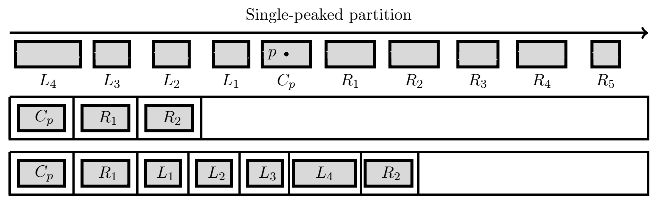

Figure 4: An illustration of the $\mathcal{FPT}$-algorithm for Borda. Inside C_p, the distinguished candidate is ranked in the top.

A *block* is a collection of intervals lying contiguously in the single-peaked partition. For example in Figure 4, $\{C_p, R_1, R_2\}$ is a block, but $\{C_p, R_2\}$ is not since there is an interval R_1 between C_p and R_2. For a block S, let $N(S)$ be the set of intervals lying directly on the left-side or on the right-side of S. For example in Figure 4, setting $S = \{C_p, R_1, R_2\}$, we have $N(S) = \{L_1, R_3\}$. Clearly, $|N(S)| \leq 2$ for every block S. In our algorithm, each manipulator maintains a block which initially is empty. Let S_i be the block maintained by the i-th manipulator, where $i = 1, 2$. For an interval $C \in N(S_i)$ and the block S_i, we use extend(S_i, C) to denote the operation $S_i := S_i \cup C$, where ':=' is the assignment operator that sets the left-hand operand equal to the right-hand expression value.

In Line 25, one of $B \in S_1$ or $B \in S_2$ must hold. We remark that the above algorithm can be adapted to handle the corresponding optimization problem, where instead of answering 'Yes' or 'No', the algorithm finds a solution for the given instance. For this purpose, extend(S_i, C) will denote both the operation $S_i := S_i \cup C$ and the following operation: ranks the candidates in C in the next contiguous positions of the i-th manipulative vote in a way that does not prevent p from being the winner. Due to space limitations, we omit further details.

6. WEIGHTED MANIPULATION

In this section, we study the weighted manipulation problem in single-peaked elections.

Faliszewski et al. [15] examined the weighted manipulation problem in single-peaked elections with three candidates for positional scoring correspondences, and proved that the problem is $\mathcal{NP}$-hard if and only if $a_1 - a_3 > 2(a_2 - a_3) > 0$, where (a_1, a_2, a_3) is the scoring vector with $a_1 \geq a_2 \geq a_3$. Recall that each positional scoring correspondence is defined by a non-negative integer scoring vector $(a_1, a_2, ..., a_m)$ with $a_1 \geq a_2 \geq, ..., \geq a_m$, where m is the number of candidates. Then every candidate c gets a_i points from each vote that ranks c in the i-th position. The winners are the candidates who have the maximum total score. Their result implies that the weighted Borda manipulation with three candidates is polynomial-time solvable in single-peaked elections. However, when the number of candidates increases to four, the weighted Borda manipulation problem becomes $\mathcal{NP}$-hard [15]. Brandt et al. [4] took the result in [15] a further step by deriving a dichotomy for the weighted manipulation problem for positional scoring correspondences with no restriction on the number of candidates.

In this section, we complement their results by exploring the weighted manipulation problem for Copeland$^\alpha$ and Maximin in single-peaked elections. A voting correspondence is *weakCondorcet-consistent* if the winners are exactly the weak Condorcet winners whenever there exists weak Condorcet winners [4]. First recall that both Maximin and Copeland^1 are weakCondorcet-consistent in single-peaked elections, and thus, the weighted manipulation problem for Maximin and Copeland^1 is polynomial-time solvable in single-peaked elections (we refer to [4] for the detailed arguments why the polynomial-time solvability holds). However, the Copeland$^\alpha$ voting for every $0 \leq \alpha < 1$ is not weakCondorcet-consistent even in single-peaked elections [4].

Our result of this section is summarized in the following theorem.

THEOREM 4. *The weighted manipulation problem for Copeland$^\alpha$ is polynomial-time solvable in single-peaked elections, for every $0 \leq \alpha < 1$.*

PROOF. We first give the proof for the unique-winner model with three candidates. To this end, we derive a polynomial-time algorithm. First observe that ranking the distinguished candidate p in the top is always the optimal choice. Hence, if p is not in the middle in the harmonious order, all the manipulators have only one way to cast their votes, and thus, the problem can be solved. Assume now that p is in the middle of the harmonious order. Without loss of generality, let a, b, p be the three candidates and (a, p, b) be the harmonious order. We consider the following cases to rank a and b. First observe that p beats at least one of a and b in the final election, no matter how the manipulators rank a and b. To check this, let x_a, x_p and x_b be the total weight of the votes (both manipulators and nonmanipulators) with their peaks at a, at p and at b, respectively. Since all manipulators have their peaks at p, we have that $x_p > 0$. Thus, one of $x_a + x_p > \frac{x_a+x_b+x_p}{2}$ and $x_b + x_p > \frac{x_a+x_b+x_p}{2}$ must hold, implying that p beats at least one of a and b. It remains to consider the following two cases. Note that since all the manipulators rank the distinguished candidate p in the top, the comparison between p and every a and b is known.

Case 1. p beats both a and b. In this case, p must be the unique winner no matter the comparison between a and b. Thus, we can immediately return 'Yes'.

Case 2. p beats exactly one of $\{a, b\}$. Without loss of generality, assume that p beats only a. Then, casting their votes as $p \succ a \succ b$ must be the optimal choice for the manipulators, since otherwise, b will beat a, implying p cannot be the unique winner.

The above algorithm directly applies to the nonunique-winner model. Due to space limitations, we only give the main idea of the algorithm for the weighted manipulation when the number of candidates is not bounded. Let $c_1, c_2, ..., c_i, p, c_{i+1}, ..., c_m$ be the harmonious order. The manipulators first rank the distinguished candidate in the top. Then, if $\{p\}$ is not in the median group (the median group in the weighted case is the median group in the unweighted case with each voter with weight w being considered as w individual unweighted voters), return "No". Otherwise, all the manipulators cast their votes as follows. If $\{p\}$ is on the left-side of the median group, then all the manipulators cast their votes as $p \succ c_i \succ ..., \succ c_1 \succ c_{i+1} \succ, ..., \succ c_m$. Otherwise, all the manipulators cast their votes as $p \succ c_{i+1} \succ ..., \succ c_m \succ c_i \succ, ..., \succ c_1$. The correctness of the algorithm is based on the weighted versions of Lemmas 5, 6 and 7. □

Note that the weighted manipulation with three candidates for Copeland$^\alpha$ for every $0 \leq \alpha < 1$ is $\mathcal{NP}$-hard in general, for the nonunique-winner model [6, 16]. Our polynomial-time algorithm in the above theorem does not apply to the general case since our algorithm relies heavily on the single-peaked restriction. For example, when the distinguished candidate is not in the middle of the harmonious order, all the manipulators have only one way to cast their votes. However, in the $\mathcal{NP}$-hardness reduction of the problem in the general case, the manipulators can cast their votes freely to balance the scores of a and b.

7. CONCLUDING REMARKS

We have studied the parameterized complexity of the unweighted manipulation problem under the Borda, Maximin and Copeland$^\alpha$ voting correspondences, and achieved several $\mathcal{FPT}$ results, with respect to the parameters "single-peaked width" and "number of manipulators". Moreover, we proposed several properties (Lemmas 1-7) of Maximin and Copeland$^\alpha$ elections with bounded single-peaked width. We believe that these properties are helpful in solving further voting problems. In addition, we proved that the weighted manipulation problem for Copeland$^\alpha$ for every $0 \leq \alpha < 1$ is polynomial-time solvable in single-peaked elections, regardless of the number of manipulators and the number of candidates. Tables 1 and 2 summarize our results.

The two $\mathcal{FPT}$-algorithms for Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ run in $O(k!^t \cdot Poly)$ time, where k is the single-peaked width and t is the number of manipulators. The factorial $k!$ corresponds to the number of enumerations over a (candidate) set of size k. Several algorithms for generating all the permutations over a set of size k have been proved practical when k is a small number (see, e.g., http://theory.cs.uvic.ca/). It is plausible that our algorithms are implementable when both k and t are small (e.g., $k \leq 5$ and $t = 2, 3$). The number of candidates, however, can be very large. For lager k and t, more heuristic methods should be derived to speed up the algorithms.

We remark that our $\mathcal{FPT}$-algorithms for the unweighted manipulation problem for Maximin and Copeland$^\alpha$ can be extended to solve the weighted manipulation problem. Nevertheless, the extension for Copeland$^\alpha$ does not cover Theorem 4, since the polynomial-time solvability result stated in Theorem 4 holds regardless of the number of manipulators. The $\mathcal{FPT}$-algorithm for the unweighted manipulation for Borda, however, cannot be extended to the weighted manipulation since the polynomial-time algorithm in [22] does not apply to the weighted case.

We end with several open problems. First, it would be interesting to know whether the fixed-parameter tractability of the manipulation problem remains when parameterized by the single parameter "single-peaked width". Second, it would be interesting to explore the parameterized complexity of the manipulation problem for further voting correspondences with respect to the parameters "single-peaked width" and "number of manipulators", such as the second-order Copeland, STV and Ranked pair. In general, the unweighted manipulation problem under these voting correspondences is $\mathcal{NP}$-hard even when there is only two manipulators [21].

Acknowledgments

The author thanks the AAMAS-15 reviewers for their helpful comments. The author was supported by the DFG Cluster of Excellence (MMCI) and the China Scholarship Council (CSC).

REFERENCES

[1] K. J. Arrow. A difficulty in the concept of social welfare. *J. Polit. Econ.*, 58(4):328–346, 1950.

[2] N. Betzler, R. Niedermeier, and G. J. Woeginger. Unweighted coalitional manipulation under the Borda rule is NP-hard. In *IJCAI*, pages 55–60, 2011.

[3] D. Black. On the rationable of group decition-making. *J. Polit. Econ.*, 56:23–34, 1948.

[4] F. Brandt, M. Brill, E. Hemaspaandra, and L. A. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *AAAI*, pages 715–722, 2010.

[5] R. Bredereck, J. Chen, and G. J. Woeginger. Are there any nicely structured preference profiles nearby? In *IJCAI*, pages 62–68, 2013.

[6] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *J. ACM*, 54(3):1–33, 2007.

[7] D. Cornaz, L. Galand, and O. Spanjaard. Bounded single-peaked width and proportional representation. In *ECAI*, pages 270–275, 2012.

[8] D. Cornaz, L. Galand, and O. Spanjaard. Kemeny elections with bounded single-peaked or single-crossing width. In *IJCAI*, pages 76–82, 2013.

[9] J. Davies, G. Katsirelos, N. Narodytska, and T. Walsh. Complexity of and algorithms for Borda manipulation. In *AAAI*, pages 657–662, 2011.

[10] G. Demange. Single-peaked orders on a tree. *Math. Soc. Sci.*, 3(3):389–396, 1983.

[11] E. Ephrati and J. S. Rosenschein. A heuristic technique for multi-agent planning. *Ann. Math. Artif. Intell.*, 20(1-4):13–67, 1997.

[12] G. Erdélyi, M. Lackner, and A. Pfandler. Computational aspects of nearly single-peaked electorates. In *AAAI*, pages 283–289, 2013.

[13] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *ECAI*, pages 366–370, 2008.

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

[15] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Inf. Comput.*, 209(2):89–107, 2011.

[16] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: ties matter. In *AAMAS (2)*, pages 983–990, 2008.

[17] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Manipulation of copeland elections. In *AAMAS*, pages 367–374, 2010.

[18] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Weighted manipulation for four-candidate Llull is easy. In *ECAI*, pages 318–323, 2012.

[19] U. Grandi, A. Loreggia, F. Rossi, K. B. Venable, and T. Walsh. Restricted manipulation in iterative voting: Condorcet efficiency and Borda score. In *ADT*, pages 181–192, 2013.

[20] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Soc. Choice. Welfare.*, 6(3):227–241, 1989.

[21] L. Xia, M. Zuckerman, A. D. Procaccia, V. Conitzer, and J. S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *IJCAI*, pages 348–353, 2009.

[22] Y. Yang and J. Guo. Exact algorithms for weighted and unweighted Borda manipulation problems. In *AAMAS*, pages 1327–1328, 2013. http://arxiv.org/abs/1304.3145.

[23] Y. Yang and J. Guo. The control complexity of r-approval: from the single-peaked case to the general case. In *AAMAS*, pages 621–628, 2014.

[24] Y. Yang and J. Guo. Election controls with small single-peaked width. In *AAMAS*, pages 629–636, 2014.

[25] Y. Yang and J. Guo. Election control in multi-peaked elections: a parameterized study. In *AAMAS*, 2015. extended abstract.