EPiC Series in Computing
Volume 50, 2017, Pages 162–174
GCAI 2017. 3rd Global Conference on Artificial Intelligence

Improved Heuristic for Manipulation of Second-order Copeland Elections

Ramoni O. Lasisi[1] and Abibat A. Lasisi[2]

[1] Virginia Military Institute, Lexington, Virginia, U.S.A.
LasisiRO@vmi.edu
[2] Utah State University, Logan, Utah, U.S.A.
abibat.lasisi@aggiemail.usu.edu

Abstract

The *second-order Copeland* voting scheme is NP-complete to manipulate even if a manipulator has perfect information about the preferences of other voters in an election. A recent work proposes a *branch-and-bound* heuristic for manipulation of second-order Copeland elections. The work shows that there are instances of the elections that may be manipulated using the branch-and-bound heuristic. However, the performance of the heuristic degraded for fairly large number of candidates in elections. We show that this heuristic is *exponential* in the number of candidates in an election, and propose an improved heuristic that extends this previous work. Our improved heuristic is based on *randomization technique* and is shown to be *polynomial* in the number of candidates in an election. We also account for the number of samples required for a given accuracy and the probability of missing the accurate value of the number of manipulations in an election.

1 Introduction

Preference aggregation is used in a variety of applications, including artificial intelligence and multiagent systems - to ease group decision-making when agents are faced with a number of alternatives to make a single choice. For example, Ephrati and Rosenschein [5] use virtual elections for preference aggregation in multiagent systems planning where agents vote on the next step of a plan under consideration. Also, according to Bartholdi, Tovey, and Trick [2], "The Federation Internationale Des Echecs and the United States Federation (USCF) implement tie-breaking rules that are either identical to, or are minor variants of the second-order Copeland scheme" to determine winners in competitions.

Voting protocols, such as the *Copeland* and *second-order Copeland* schemes are appropriate candidates, among others, for modeling such preference aggregation. Bartholdi, Tovey, and Trick define a voting scheme as an algorithm that takes as input a set C of candidates and a set P of preference orders that are *strict* (irreflexive and antisymmetric), *transitive*, and *complete* on C. The algorithm outputs a subset of C, who are the winners (allowing for ties). The ideal of a society is that a candidate emerging as a winner in an election be as widely and socially acceptable as possible.

Strategic manipulation of elections by agents remains a bane of voting protocols. Thus, the inability to limit or understand the effects of this menace may undermine the confidence agents have in decisions made via such protocols. While the Copeland voting protocol can be efficiently manipulated in polynomial time, the second-order Copeland voting scheme is NP-complete to manipulate *even if a* *manipulator has complete information about the preferences of other voters in an election* [2]. Although this complexity result is daunting to deter a strategic voter from manipulation, NP-completeness is a worst case measure. And only shows that at least one instance of the problem requires such complexity [3]. Thus, real-life instances of an election that we care about may be easy to manipulate.

Conitzer and Sandholm [3] show that when the number of candidates in an election is small, manipulation algorithms that are exponential only in the number of candidates might be available. In line with this, and in an attempt to determine how few the candidates of elections can be for voting procedures to be hard, they show that *at least four candidates* are needed in any Copeland election for the manipulation to become hard in the second-order Copeland scheme [4].

Lasisi [13] in a recent work and using the hypothesis that "There are instances of second-order Copeland elections that may be efficiently manipulated using heuristics," proposes a branch-and-bound heuristic for manipulation of second-order Copeland elections. The performance of the heuristic was experimentally evaluated using randomly generated data based on three distributions, including uniform, normal, and Poisson. Results of experiments from the work suggest that there are instances of the second-order Copeland elections that may be manipulated using the proposed branch-and-bound heuristic when a voter has perfect information about the preferences of other voters. However, the performance of the heuristic degraded for fairly large number, $k \geq 8$, of candidates. This degradation in performance is expected, as our analysis of the branch-and-bound heuristic shows that the heuristic is *exponential* in the number of candidates in the elections. Considering the exponential running time of this branch-and-bound heuristic, we propose in this paper an improved heuristic that extends this previous work. Our main results are as follows:

- The textual description of the proposed branch-and-bound heuristic in [13] is unclear where it needs to be precise. So, we present a pseudocode to clarify the heuristic branch-and-bound algorithm.
- We provide an analysis of the running time of the branch-and-bound heuristic. Our analysis shows that the heuristic is exponential in the number of candidates in an election.
- We propose a randomized method for manipulation of second-order Copeland elections. We use a sampling procedure approach to generate samples of voters' preferences and employ a specialized version of the well-known Hoeffding's inequality [12], referred to as the Chernoff's bound to account for the number of samples required for a given accuracy as well as the probability of missing the accurate value of the number of manipulations in an election.
- Finally, based on our proposed randomized method for manipulation of second-order Copeland election above, we propose an improved randomized heuristic that extends the branch-and-bound heuristic. Our heuristic is shown to be polynomial in the number of candidates in an election.

2 Preliminaries

We present preliminaries covering definitions, notation, and examples of voting and manipulations in first and second-order Copeland elections.

2.1 Definitions and Notation

Definition 1. *First-order Copeland Voting*

The Copeland voting scheme (also known as the *first-order Copeland* method) is a protocol in which all candidates in an election engage in the same number of pairwise contests. A winner is a candidate that maximizes her *Copeland score*: the difference between her number of victories and defeats in all pairwise contests [17, 18].

Definition 2. *Second-order Copeland Voting*

In the case of a tie in the first-order Copeland voting, the *eventual* winner is the candidate whose defeated competitors have the largest sum of Copeland score. This tie-breaking rule is the second-order Copeland voting [2].

Let $k, n \in \mathbb{Z}^+$. Let $C = \{c_1, \ldots, c_k\}$ be a set of candidates in an election, with $k \geq 4$. Let $P = \{p_1, \ldots, p_n\}$, respectively, be the preference orders of voters, $V = \{v_1, \ldots, v_n\}$, over C. We define a relation, $\succ$, for each $p_i \in P$ over C. We say that a voter $v_m \in V$ ranks candidate c_i over candidate c_j denoted, $c_i \succ c_j$, if v_m prefers c_i to c_j in her preference order $p_m \in P$. Since all candidates must engage in the same number of contests, the preference orders are required to be complete on C. Finally, denote by $C_S(c_i)$, the Copeland score of a candidate $c_i \in C$ in a first-order Copeland election.

2.2 Voting and Manipulation in First-order Copeland Elections

Example 1. *First-order Copeland Voting*

Consider the following preferences by seven agents for three candidates, a, b, and c, in an election:

3 agents : $a \succ c \succ b$

2 agents : $c \succ b \succ a$

1 agent : $c \succ a \succ b$

1 agent : $b \succ a \succ c$

In the pairwise contests:

a vs b : a received 4 votes while b received 3 votes $\Rightarrow$ a wins and b loses

a vs c : a received 4 votes while c received 3 votes $\Rightarrow$ a wins and c loses

b vs c : b received 1 vote while c received 6 votes $\Rightarrow$ b loses and c wins

The Copeland scores for the three candidates are:

$C_S(a) = 2 - 0 = 2$

$C_S(b) = 0 - 2 = -2$

$C_S(c) = 1 - 1 = 0$

Thus, candidate a is the overall winner in this election.

Example 2. *Manipulation of First-order Copeland Voting*

We consider the same set of candidates and voters as in Example 1. However, one of the voters from that example is using a different preference order, i.e., not reporting her preference order truthfully. Let s be this strategic voter with the truthful preference $b \succ a \succ c$ (as given in Example 1). Let c be the candidate that s would like to manipulate her preference for. Suppose s employs the polynomial manipulation algorithm of [2], designed to place favored candidate at the top of strategic agent's preference, to uncover a manipulative preference order, $c \succ a \succ b$, and then participates in the election. The new Copeland scores for the three candidates are:

$C_S(a) = 1 - 1 = 0$

$C_S(b) = 0 - 2 = -2$

$C_S(c) = 2 - 0 = 2$

Thus, candidate c is the overall winner in this election where the strategic agent fails to report her preference order truthfully. Note that the preference orders of the remaining six agents remain the same as before.

2.3 Voting and Manipulation in Second-order Copeland Elections

Example 3. *Second-order Copeland Voting*

Consider the following preferences by seven agents for four candidates, a, b, c, and d, in an election:

3 agents : $a \succ b \succ c \succ d$

2 agents : $d \succ b \succ c \succ a$

1 agent : $d \succ b \succ a \succ c$

1 agent : $c \succ d \succ a \succ b$

In the pairwise contests:

a vs b : a received 4 votes while b received 3 votes $\Rightarrow$ a wins and b loses

a vs c : a received 4 votes while c received 3 votes $\Rightarrow$ a wins and c loses

a vs d : a received 3 votes while d received 4 votes $\Rightarrow$ a loses and d wins

b vs c : b received 6 votes while c received 1 vote $\Rightarrow$ b wins and c loses

b vs d : b received 3 votes while d received 4 votes $\Rightarrow$ b loses and d wins

c vs d : c received 4 votes while d received 3 votes $\Rightarrow$ c wins and d loses

The Copeland scores for the four candidates are:

$C_S(a) = 2 - 1 = 1$

$C_S(b) = 1 - 2 = -1$

$C_S(c) = 1 - 2 = -1$

$C_S(d) = 2 - 1 = 1$

Thus, candidates a and d tied in this first-order Copeland voting. This tie is broken by computing the sum of the Copeland scores for the defeated competitors for both a and d. The sum of the Copeland scores for the defeated competitors of a is $C_S(b) + C_S(c) = -2$ and that of the defeated competitors of d is $C_S(a) + C_S(b) = 0$. Thus, candidate d emerges as the overall winner in this election after the application of the tie-breaking rule.

Example 4. *Manipulation of Second-order Copeland Voting*

We consider the same set of candidates and voters as in Example 3. However, one of the voters from that example is using a different preference order, i.e., not reporting her preference order truthfully. Let s be this strategic voter with the truthful preference $c \succ d \succ a \succ b$ (as given in Example 3). Let a be the candidate that s would like to manipulate her preference for. Suppose s uses the manipulative preference order, $a \succ c \succ d \succ b$, and then participates in the election. The pairwise contests are:

a vs b : a received 4 votes while b received 3 votes $\Rightarrow$ a wins and b loses

a vs c : a received 5 votes while c received 2 votes $\Rightarrow$ a wins and c loses

a vs d : a received 4 votes while d received 3 votes $\Rightarrow$ a wins and d loses

b vs c : b received 6 votes while c received 1 vote $\Rightarrow$ b wins and c loses

b vs d : b received 3 votes while d received 4 votes $\Rightarrow$ b loses and d wins

c vs d : c received 4 votes while d received 3 votes $\Rightarrow$ c wins and d loses

The Copeland scores for the four candidates are:

$C_S(a) = 3 - 0 = 3$

$C_S(b) = 1 - 2 = -1$

$C_S(c) = 1 - 2 = -1$

$C_S(d) = 1 - 2 = -1$

Thus, candidate a is the overall winner in this election where the strategic agent fails to report her preference order truthfully. Observe that the preference order used by the strategic agent avoids ties and the tie breaking rule employed in Example 3.

Winners in both the Copeland and second-order Copeland votings can be efficiently computed in polynomial time. However, while a strategic voter can efficiently manipulate the Copeland voting in polynomial time, the second-order Copeland voting is NP-complete to manipulate [2]. According to [2]: "Intuitively, it is difficult to construct a manipulative preference under Second-Order Copeland because it is difficult to know where to place candidates in the preference. For example, placing a favored candidate at the top can unintentionally improve the scores of rivals because of second order effects in the scoring."

3 The Branch-and-Bound Heuristic

Heuristics are known to provide preferable solutions to instances of hard problems in practical situations without having to examine all the possible choices. The approach employed in the design of this heuristic is to compute an estimated score of the current problem instance, called the *bound*. This bound is then used to determine how to *branch* or bypass (without examination of) large instances of the problem whose scores cannot be better than the bound found so far. For completeness, we present a pseudocode (Algorithm 1), description, and analysis of the branch-and-bound heuristic proposed in [13].

3.1 Description of the Heuristic

Let C, P, and V, be as defined in Section 2, with $|C| = k$ and $|P| = |V| = n$. Let $v_s \in V$ be a strategic voter with preference order $p_s \in P$. Let $c_d \in C$ be a candidate that v_s would like to manipulate her

preference order for so that c_d wins in an election. Suppose $m \leq k$ of the candidates, including c_d tied under the Copeland voting protocol, then the difficulty is to manipulate the tie-breaking rule (under the second-order Copeland scheme) such that candidate c_d wins. Hence, the interest is to find all such preference orderings p'_s of voter v_s different than p_s, that elicits wins for c_d. The preference orders of all the voters are assumed to be fixed except that of v_s, and these other orderings are known to v_s. Altogether, there are $\binom{k}{2}$ pairwise contests among the k candidates and each of the candidate participates in exactly $k-1$ contests.

Using Algorithm 1, for every new preference order p'_s of v_s, compute the Copeland scores of the k candidates in $\lceil \frac{k-1}{2} \rceil$ contests and set the bound for the heuristic as the maximum score in the contests, denoted by $\max_{C_S}$. Estimate the overall score of c_d, denoted $C_{S_{est}}(c_d)$, in the expected $k-1$ contests as $C_{S_{est}}(c_d) = C_S(c_d) + \lfloor \frac{k-1}{2} \rfloor$. This score gives an estimate of the upper bound on the Copeland score that is attainable by c_d since it is assumed that c_d will win in all of the remaining $\lfloor \frac{k-1}{2} \rfloor$ contests. If this estimated score is less than or equal to the bound value, i.e., $C_{S_{est}}(c_d) \leq \max_{C_S}$, then bypass this preference order since this is the best score attainable by c_d in these $k-1$ contests using this particular preference order. Otherwise, compute the Copeland scores of the k candidates in the remaining $\lfloor \frac{k-1}{2} \rfloor$ contests and determine the winner. If candidate c_d wins, then the heuristic has found an instance of the second-order Copeland voting that is manipulable.

The salient point worthy of note in this heuristic is that it allows c_d to circumvent election scenarios that lead to a tie with other candidates, thus ensuring a win in all situations where the heuristic reports a win.

Algorithm 1: *Branch-and-Bound.* Find manipulable preference orders

input : preferences of n voters (including p_s of the strategic agent v_s) over k candidates and a distinguished candidate, c_d

1 **output:** set of preferences $p'_s \neq p_s$ of the strategic agent v_s that ensures that c_d is a winner

2 $found \longleftarrow \emptyset$

3 **for** *every new preference order* p'_s *of* v_s *different than* p_s **do**

4   **for** $\lceil \frac{k-1}{2} \rceil$ *contests* **do**

5   **for** *each candidate* $c_i \in C$ **do**

6   compute Copeland score $C_S(c_i)$ using the n voters' preferences

7   set $score_i \longleftarrow C_S(c_i)$

8     $\max_{C_S} \longleftarrow \max_{i \in C}\ score_i$

9     $C_{S_{est}}(c_d) \longleftarrow C_S(c_d) + \lfloor \frac{k-1}{2} \rfloor$

10   **if** $C_{S_{est}}(c_d) \leq \max_{C_S}$ **then**

11   continue

12   **for** *the remaining* $\lfloor \frac{k-1}{2} \rfloor$ *contests* **do**

13   **for** *each candidate* $c_i \in C$ **do**

14   compute Copeland score $C_S(c_i)$ using the n voters' preferences

15             $score_i \longleftarrow score_i + C_S(c_i)$

16   **if** $C_{S_{est}}(c_d) == \max_{i \in C}\ score_i$ **then**

17   manipulable preference order found

18         $found \longleftarrow found \cup \{p'_s\}$

19 return $found$

3.2 Analysis of the Heuristic

Clearly, the time expended in lines 8 and $16-18$ of the algorithm is $O(k)$ since there are k candidates in the election. Lines 9 and $10-11$ is all a constant amount of work and can be ignored given the $O(k)$ time. In cases that we bypass some instances of the problem, we will execute only the nested for loop in lines $4-7$. However, in the worst case, both nested loops in lines $4-7$ and lines $12-15$ will be executed. The two nested loops combine to compute all the $\binom{k}{2}$ pairwise contests among the k candidates, for which each of the candidate participates in exactly $k-1$ contests. So, the time expended in lines $4-7$ and lines $12-15$ for the contest is $\binom{k}{2} = \frac{k(k-1)}{2} = O(k^2)$. Since we are using n voters' preferences, the total time to determine the winner (i.e., computes $score_i$) in the contests is $O(k^2 n)$. Thus, the previous $O(k)$ can be ignored compared to the $O(k^2 n)$. Furthermore, line 3 considers every new preference order of the strategic agent. Since there are k candidates, we will consider a total of $k!$ preferences (or permutations). This gives $O(k!)$ work to generate all the permutations. Using Stirling's approximation, the amount of work is about $O(2^{k \log k})$. Finally, since the two nested loops that gave rise to the $O(k^2 n)$ work above are included in this outermost loop, the total running time of the algorithm is $O(k^2 n \cdot 2^{k \log k})$, which is exponential in the number k of the candidates in an election.

4 A Randomized Method for Manipulation of Second-order Copeland Elections

Since it is non-trivial to consider every possible preference orders of a strategic voter to uncover possible manipulations in second-order Copeland elections as demonstrated in the previous section, a scientific approach is to consider *samples* from the population of the preference orders. The information contained within a sample can then be used to investigate properties of the population from which the sample is drawn [[10],page 294]. We represent the set P of preference orders by a corresponding set Π of permutations of the k candidates in an election. Consider an election with three candidates, a, b, and c. If a voter's preference order $p_i \in P$ is, $b \succ a \succ c$, then the corresponding permutation $\pi_i \in \Pi$ of the preference order is given as *bac*. Similarly, the preference order $p_j : a \succ b \succ c$ corresponds to the permutation $\pi_j : abc$.

The idea of our randomized method for manipulation of the second-order Copeland elections that we propose in this section comes from a similar sampling procedure for approximating power indices in [1]. The method involves sampling permutations and checking whether each permutation results in a preference order that is manipulable. Thus, we can approximate the actual number of manipulable preferences in an election by taking large enough samples of the preference orders of the strategic agent that are different from her truthful preference order. The amount of permutations sampled determines the accuracy of the randomized method. Similar to [1], our proposed method determines the number η of samples required for a given approximation accuracy $\epsilon > 0$ and probability ρ of missing the accurate value of the number of manipulable preferences in an election.

4.1 A Manipulable Preference Order

We need an operation in our randomized method to determine if a preference is manipulable. Given preferences P of n voters (including the preference p_s of the strategic agent v_s) over k candidates and a distinguished candidate, c_d. Let c_d ties with some candidates in the Copeland election and loses in the second-order Copeland election when the strategic agent v_s votes truthfully using p_s.

Given a new preference order p'_s different than p_s, we can easily check in polynomial time if p'_s is manipulable, i.e, p'_s results in a win for c_d in a Copeland election. We use permutations $\pi_s, \pi'_s \in \Pi$ to represent $p_s, p'_s \in P$, respectively. Let procedure $Manipulable(\Pi, \pi_s, \pi'_s, C, c_d)$ be a polynomial algorithm for checking if the preference p'_s is manipulable. The procedure returns true if p'_s is manipulable and false otherwise. The running time of the procedure is precisely $O(k^2 n)$. See Algorithm 2.

4.2 Random Sampling of Permutation

We define a sampling procedure. See Algorithm 3. Let the permutation π_s represents the original preference order p_s of the strategic agent v_s. This procedure simply generates a new random permutation π'_s of the candidates C, ensures that it is different than π_s, and has not been found before. A *hashset* data structure named *found*, is maintained and is also used to check for duplicate permutations. Further, the procedure calls the $Manipulable$ procedure of Subsection 4.1 to check if π'_s is manipulable. The procedure returns a 1 if π'_s is manipulable and 0 otherwise. The two operations of generating a permutation[1] and checking that the permutation is different than π_s can be completed in linear time of the number of candidates in the permutation, i.e., $O(k)$. Checking the existence of duplicates in the *found* hashset can be done in constant time. The overall running time of this procedure is $O(k^3 n)$ since it calls the $Manipulable$ procedure each time. Let procedure $RandomSample(\Pi, \pi_s, C, c_d, found)$ be the polynomial algorithm for sampling a permutation as shown in Algorithm 3.

This sampling procedure models the *Bernoulli* distribution. This is a random trial or experiment in which the outcome can be classified into two mutually exclusive ways called success or failure. Let

[1] There are several techniques for generating permutation that are linear in the size of the permutation.

Algorithm 2: *Manipulable(*$\Pi, \pi_s, \pi'_s, C, c_d$*)*. Checks if preference p'_s corresponding to the permutation π'_s is manipulable

input : Π, π_s, π'_s, C, and c_d

output: return true if preference p'_s corresponding to the permutation π'_s is manipulable or false otherwise

```
Π ⟵ Π \ {π_s} ∪ {π'_s}
for m = 1  to  |Π| = n do
    for i = 1  to  |C| − 1 = k − 1 do
        for j = i + 1  to  |C| = k do
            conduct pairwise contest between candidates c_i and c_j in permutation π_m
            update Copeland scores C_S(c_i) and C_S(c_j)
if C_S(c_d) == max_{i∈C} C_S(c_i) then
    return true
return false
```

Algorithm 3: *RandomSample(*$\Pi, \pi_s, C, c_d, found$*)*. Samples a permutation π'_s and returns 1 if it is manipulable or 0 otherwise

input : Π, π_s, C, c_d and found

output: return 1 if preference p'_s corresponding to the sampled permutation π'_s is manipulable or 0 otherwise

```
π'_s ⟵ generatePermutation() of size |C|
ensure that π'_s ≠ π_s and π'_s ∉ found
if Manipulable(Π, π_s, π'_s, C, c_d) then
    p'_s ⟵ getCorrespondingPreference(π'_s)
    found ⟵ found ∪ {p'_s}
    return 1
return 0
```

X_i be Bernoulli random variables associated with different trials in which X_i is 1 if π'_s is manipulable and 0 otherwise. The Bernoulli random variable is defined by the parameter p, $0 \leq p \leq 1$, which is the probability that the outcome is 1, i.e., $P(X_i = 0) = 1 - p$ and $P(X_i = 1) = p$.

Consider η independent repetitions of such trials. Let X be the number of successes in this series of the Bernoulli trials, $X = \sum_{i=1}^{\eta} X_i$ is said to have a *Binomial distribution* with parameters η and p, denoted $X \sim B(\eta, p)$. The Binomial distribution gives *the number of successes* obtained within a fixed number of η trials. Observe that parameter p, the probability of success which is the probability that π'_s is manipulable is unknown. Since $X \sim B(\eta, p)$, then the estimate for p, is $\hat{p} = \frac{X}{\eta}$. This estimator $\hat{p}$, is an unbiased estimate for the probability p [10]. Clearly, $\hat{p}$ is the estimate of the proportion of the number of manipulable preferences p'_s for the original preference p_s in a sequence of η independent Bernoulli trials.

4.3 Estimating Manipulations in a Second-order Copeland Election

We now estimate the amount of p'_s that are manipulable for a given p_s in an election. We employ a specialized version of the well-known Hoeffding's inequality [12], referred to as the Chernoff's bound to obtain a relationship among the number η of samples required for a given approximation accuracy $\epsilon > 0$ and probability ρ of missing the accurate value of the number of manipulable preferences in an election. That is we are willing to accept with probability ρ of having our estimator $\hat{p}$ miss p by more than ϵ.

Theorem 1. *(Hoeffding's inequality). Let $X_1, \ldots, X_\eta$ be independent random variables on $\mathbb{R}$ such that $a_i \leq X_i \leq b_i$ with probability one. If $X = \sum_{i=1}^{\eta} X_i$ then for all $\epsilon > 0$*

$$Pr(|X - E[X]| \geq \epsilon) \leq 2e^{-\frac{2\epsilon^2}{\sum(b_i - a_i)^2}} \tag{1}$$

Hoeffding's inequality specializes to Chernoff's bound as follows. If X_i are independent and identically distributed Bernoulli random variables, then $a_i = 0, b_i = 1$, and $X \sim B(\eta, p)$. Since $E[X] = \eta p$, Chernoff's bound is given as:

$$Pr\left(\left|\frac{1}{\eta}\sum_{i=1}^{\eta} X_i - p\right| \geq \epsilon\right) \leq 2e^{-2\eta\epsilon^2} \tag{2}$$

which simplifies to the following

$$\begin{aligned} Pr\left(\left|\frac{X}{\eta} - p\right| \geq \epsilon\right) &\leq 2e^{-2\eta\epsilon^2} &(3)\\ Pr(|\hat{p} - p| \geq \epsilon) &\leq 2e^{-2\eta\epsilon^2}. &(4)\end{aligned}$$

We ensure that the Chernoff's bound given in Equation 4 does not exceed the probability ρ of missing the accurate value of the number of manipulable preferences in an election and simplify the expression using the *Power Confidence Interval* Theorem of [1]. Thus, we have

$$\begin{aligned} Pr(|\hat{p} - p| \geq \epsilon) \leq 2e^{-2\eta\epsilon^2} &\leq \rho \\ 2e^{-2\eta\epsilon^2} &\leq \rho \\ -2\eta\epsilon^2 &\leq \ln\frac{\rho}{2} \end{aligned}$$

Then,

$$\begin{aligned} \eta &\geq -\frac{1}{2\epsilon^2}\ln\frac{\rho}{2} \\ &\geq \ln\left(\frac{\rho}{2}\right)^{-\frac{1}{2\epsilon^2}} \\ &\geq \ln\left(\frac{2}{\rho}\right)^{\frac{1}{2\epsilon^2}} \\ &\geq \frac{1}{2\epsilon^2}\ln\frac{2}{\rho} \end{aligned}$$

Thus, the number η of samples required for a given accuracy $\epsilon > 0$ and probability ρ of missing the accurate value of the number of manipulable preferences is at least $\frac{\ln\frac{2}{\rho}}{2\epsilon^2}$.

Algorithm 4: *RandomizedHeuristic*($\Pi, \pi_s, C, c_d, \epsilon, \rho$). Find manipulable preferences using randomization

input : $\Pi, \pi_s, C, c_d, \epsilon$, and ρ
1
output: set of preferences p'_s of the strategic agent v_s that ensures that c_d is a winner
2 counter = 1
3 $found \longleftarrow \emptyset$
4 **repeat**
5 **if** $RandomSample(\Pi, \pi_s, C, c_d, found) == 1$ **then**
6 manipulable preference order found
7 counter += 1
8 **until** *counter* $\geq \frac{\ln \frac{2}{\rho}}{2\epsilon^2}$
9 return *found*

5 Randomized Heuristic

We now propose a randomized heuristic for manipulation of second-order Copeland election based on the randomized method of Section 4.

5.1 Description of the Heuristic

Let procedure *RandomizedHeuristic* be our randomized heuristic. *RandomizedHeuristic* accepts as input, permutations Π representing the preferences P of n voters in a second-order Copeland election, a permutation $\pi_s \in \Pi$ representing the truthful preference $p_s \in P$ of the strategic agent $v_s \in V$, the set C of candidates with a distinguished candidate $c_d \in C$, approximation accuracy $\epsilon > 0$, and probability ρ of missing the accurate value of the number of manipulable preferences in the Copeland election.

RandomizedHeuristic repeatedly samples permutations by calling the procedure *RandomSample* of Subsection 4.2, which in turn calls the *Manipulable* procedure of Subsection 4.1. The heuristic starts with an empty value for the hashset, *found*. The data structure is continuously updated in the *RandomSample* procedure when a new manipulable preference is found. The heuristic checks the values return by each call of *RandomSample* procedure. A value of 1 indicates that a manipulable preference has been found, and 0 otherwise. The heuristic continues to sample permutations until enough samples η have been generated, i.e., when $\eta \geq \frac{\ln \frac{2}{\rho}}{2\epsilon^2}$. The pseudocode of the heuristic is given in Algorithm 4.

5.2 Analysis of the Heuristic

It is easy to see that the running time of the randomized heuristic for manipulation of second-order Copeland election is $\Omega\left(k^3 n \cdot \frac{\ln \frac{2}{\rho}}{2\epsilon^2}\right)$, which is polynomial in the number $|C| = k$ of candidates in the election. This is because the running time of the heuristic is entirely due to line 5 of the algorithm which consists of an $O(k^3 n)$ statement as demonstrated in Subsection 4.2. This statement is contained in a loop that is repeated for at least $\frac{\ln \frac{2}{\rho}}{2\epsilon^2}$ times.

6 Related Work

Voting and elections play major roles in artificial intelligence, multiagent systems, and human societies. We use elections to make hiring decisions, nominate representatives to various arms of government, decide winners in competitions, and many more. There are several well-known voting protocols, including, *simple majority, Borda count, plurality, Copeland voting, second-order Copeland, runoff,* and *maximin*, for group decision-making in the literature. See [21, 22, 23], for a general introduction to the theory of voting and elections.

The ideal of a society is that a candidate emerging as a winner in an election be as widely and socially acceptable as possible. However, the problem of manipulation in elections and voting systems are pervasive in human societies and multiagent systems, and has received attention of many researchers in recent years. See, for example, [8, 11, 14, 15, 16, 19]. The famous Gibbard-Satterthwaite theorem states: *Every voting scheme with at least three outcomes is either dictatorial or manipulable* [9, 20]. This implies that in any non-dictatorial voting protocol with at least three candidates, there exist some preferences of the voters such that some voters achieve better outcomes voting strategically i.e., not truthfully representing their preferences.

Previous works on strategic manipulations in Copeland elections have been devoted to computational complexity results. Faliszewski et al. [6] study the complexity of manipulation for a family of election systems derived from Copeland voting via introducing a parameter α that describes how ties in head-to-head contests are valued. They show that the problem of manipulation for unweighted $Copeland^{\alpha}$ elections is NP-complete even if the size of the manipulating coalition is limited to two. In a follow up research, Piotr et al. [7] resolved an open problem regarding the complexity of unweighted coalitional manipulation, namely, the complexity of $Copeland^{\alpha}$-manipulation for $\alpha \in \{0, 1\}$ posed in [6]. They show that the problem remains NP-complete for $\alpha \in \{0, 1\}$. As stated in the introduction, Bartholdi et al. [2] show that the second-order Copeland voting scheme is NP-complete to manipulate *even if a manipulator has complete information about the preferences of other voters in an election.* This result is a clear demonstration of how to use computational complexity in the real-world to deter would-be strategic voter from engaging in manipulation.

However, it is possible that real instances of elections that we care about are easy to manipulate, and for elections with fewer number of candidates and voters, exponential-increasing work may not be a deterrent to manipulators. Bartholdi et al. further raised an interesting concern about election manipulation that "It might be that there are effective heuristics to manipulate election even though manipulation is NP-complete." Lasisi [13] in a recent work proposes a branch-and-bound heuristic and examines the empirical behavior of the second-order Copeland election under assumptions that are realistic in familiar kinds of voting, e.g., many voters, but a relatively small candidates. The empirical investigation into this question takes the form of a large number of experiments where a voter manipulates the outcome of an election. There is definitely a place in the literature for the empirical understanding of this type of model when used to abstract some real-world scenarios.

In this paper, we show that the proposed branch-and-bound heuristic of [13] is exponential in the number of candidates in an election. Thus, giving partial explanation for the second-order Copeland scheme being resistant to manipulation for fairly large number of candidates in Lasisi's experiments. Furthermore, we propose a randomized heuristic that extends this previously known branch-and-bound heuristic for manipulation of second-order Copeland elections.

7 Conclusions and Future Work

The *second-order Copeland* voting has been shown to be NP-complete to manipulate even if a manipulator has perfect information about the preferences of other voters in an election. Lasisi [13] in a recent experimental work and using the hypothesis that "There are instances of second-order Copeland elections that may be efficiently manipulated using heuristics," proposes a branch-and-bound heuristic for manipulation of second-order Copeland elections. The performance of the heuristic was experimentally evaluated using randomly generated data based on three distributions, including uniform, normal, and Poisson. Results of experiments from the work suggest that there are instances of the second-order Copeland elections that may be manipulated using the proposed branch-and-bound heuristic when a voter has perfect information about the preferences of other voters.

In this paper, we first note that the textual description of the proposed branch-and-bound heuristic of Lasisi [13] is unclear where it needs to be precise. So, we present a pseudocode to clarify the branch-and-bound heuristic. Second, we provide an analysis of the running time of the heuristic. Our analysis shows that the heuristic is exponential in the number of candidates in an election. The exponential running time of this heuristic that we demonstrate in the present paper complements the performance degradation of the heuristic for fairly large number of candidates as reported in [13]. Third, we propose a randomized method for manipulation of second-order Copeland elections. We use a sampling procedure approach to generate samples of voters' preferences and employ a specialized version of the well-known Hoeffding's inequality [12], referred to as the Chernoff's bound, to account for the number of samples required for a given accuracy as well as the probability of missing the accurate value of the number of manipulations in an election. Finally, we propose an improved heuristic that extends the branch-and-bound heuristic based on our proposed randomized method for manipulation of second-order Copeland election. Our heuristic is also shown to be polynomial in the number of candidates in an election.

The following are some ideas for future work on this area of research. We plan to implement the proposed randomized heuristic in this work and empirically evaluate it using similar experimental setup of [13]. Also, we seek to find a theoretical bound on the approximation factor by which our heuristic is from the optimal value of the number of manipulable preferences that may be present in a second-order Copeland election. Furthermore, there are still several other interesting open problems on manipulation of second-order Copeland elections from [13]. Finally, developing methods to reduce the effects of this problem in second-order Copeland voting scheme is an interesting research problem.

Acknowledgement

This research is partially supported by the Virginia Military Institute's Professional Travel Funds.

References

[1] Yoram Bachrach, Evangelos Markakis, Ariel D. Procaccia, Jeffrey S. Rosenschein, and Amin Saberi. Approximating power indices. In *7th International Conference on Autonomous Agents and Multiagent Systems*, pages 943–950, Estoril, Portugal, May 2008.

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

[3] Vincent Conitzer and Tuomas Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the 21st National Conference on Artificial Intelligence.*, August 2006.

[4] Vincent Conitzer, Tuomas Sandholm, and Jerome Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3), 2007.

[5] Eithan Ephrati and Jeffrey S. Rosenschein. A heuristic technique for multiagent planning. *Annals of Mathematics and Intelligence*, 20:13–67, 1997.

[6] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Copeland voting: ties matter. In *7th International Conference on Autonomous Agents and Multiagent Systems*, pages 983–990, May 2008.

[7] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Manipulation of Copeland elections. In *9th International Conference on Autonomous Agents and Multiagent Systems*, pages 367–374, May 2010.

[8] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Weighted manipulation for four-candidate llull is easy. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 318–323, 2012.

[9] Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41(4):587–601, July 1973.

[10] Anthony Hayter. *Probability and Statistics for Engineers and Scientists*. Thomson Brooks/Cole, 2007.

[11] Edith Hemaspaandra, Lane A. Hemaspaandra, and Joerg Rothe. The complexity of online manipulation of sequential elections. In *TARK 2013*, pages 111–120, 2013.

[12] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[13] Ramoni O. Lasisi. Manipulation of second-order Copeland elections: Heuristic and experiment. In *29th FLAIRS Conference*, pages 68–73, Key Largo, Florida, USA, May 2016.

[14] Ramoni O. Lasisi and Vicki H. Allan. New bounds on false-name manipulation in weighted voting games. In *27th FLAIRS Conference*, pages 57–62, Pensacola Beach, Florida, USA, May 2014.

[15] Elchanan Mossel and Milkos Z. Racz. Election manipulation: the average case. *ACM SIGecom Exchanges*, 11(2):22–24, 2012.

[16] Nina Narodytska and Toby Walsh. The computational impact of partial votes on strategic voting. In *Proceedings of the Twenty-first European Conference on Artificial*, pages 657–662, 2014.

[17] Richard G. Niemi and William H. Riker. The choice of voting systems. *Scientific American*, 234:21–27, 1976.

[18] Hannu Nurmi. Voting procedures: A summary analysis. *British Journal of Political Science*, 13:181–208, 1983.

[19] Jorg Rothe and Lena Schend. Challenges to complexity shields that are supposed to protect elections against manipulation and control. *Annals of Mathematics and Artificial Intelligence*, 68:161–193, 2013.

[20] Mark A. Satterthwaite. Strategy-proofness and arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, April 1975.

[21] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundation*. Cambridge, 2009.

[22] W. D. Wallis. *A Beginner's Guide to Discrete Mathematics*. Birkhauser, 2012.

[23] Michael Woodridge. *An Introduction to Multiagent Systems*. John & Wiley, 2009.