

Kemeny Consensus Complexity

Zack Fitzsimmons¹ and Edith Hemaspaandra²

¹College of the Holy Cross

²Rochester Institute of Technology
zfitzsim@holycross.edu, eh@cs.rit.edu

Abstract

The computational study of election problems generally focuses on questions related to the winner or set of winners of an election. But social preference functions such as Kemeny rule output a full ranking of the candidates (a consensus). We study the complexity of consensus-related questions, with a particular focus on Kemeny and its qualitative version Slater. The simplest of these questions is the problem of determining whether a ranking is a consensus, and we show that this problem is coNP-complete. We also study the natural question of the complexity of manipulative actions that have a specific consensus as a goal. Though determining whether a ranking is a Kemeny consensus is hard, the optimal action for manipulators is to simply vote their desired consensus. We provide evidence that this simplicity is caused by the combination of election system (Kemeny), manipulative action (manipulation), and manipulative goal (consensus). In the process we provide the first completeness results at the second level of the polynomial hierarchy for electoral manipulation and for optimal solution recognition.

1 Introduction

Elections are a widely used tool for aggregating the preferences of several agents into a collective decision. Often the goal is to determine a single winner or set of winners from among a set of candidates. However, in other cases, such as constructing a meta-search engine [Dwork *et al.*, 2001] or genetic maps [Jackson *et al.*, 2008], the natural desired outcome is a ranking.

One of the most compelling ways of aggregating preferences is the Kemeny rule. It is known that computing a Kemeny consensus (i.e., a ranking closest to the electorate) is a computationally difficult problem. We show that even simply checking if a given ranking is a consensus is coNP-complete. This problem is naturally motivated by an agent wanting to verify the claimed outcome of an election.

One of the most important lines of research in the computational study of elections (see, e.g., Faliszewski and Rothe [2016]) is the study of different manipulative actions

such as manipulation and control [Bartholdi *et al.*, 1989a; Bartholdi *et al.*, 1992], where an agent (or agents) seek to ensure their preferred outcome by either voting strategically or modifying the structure of the election. In each of these models, the goal is typically to ensure a preferred candidate wins. For scenarios where the collective decision is a consensus, it is natural to consider manipulative actions where the goal of the agent(s) is to reach a preferred consensus.

Even though the problem of determining whether a ranking is a Kemeny consensus is hard, the optimal action for the manipulators is to simply vote their desired consensus. We provide evidence that this simplicity is caused by the combination of the manipulative action (manipulation), the manipulative goal (consensus), and the election system (Kemeny). In particular:

- Determining if a given ranking is a Kemeny consensus is coNP-complete. (Section 3)
- Control by deleting candidates for Kemeny with the goal of a particular consensus is Σ_2^P -complete (and thus, unlike manipulation, the optimal control action is not polynomial-time computable). (Section 5)
- We provide evidence that manipulation (to winner) for Kemeny is also much harder than manipulation to consensus, by showing that manipulation (to winner) for a natural variant of Slater (the qualitative version of Kemeny) is Σ_2^P -complete. (Section 6)
- The choice of system matters as well. For example the optimal action for the manipulators to reach a consensus is not polynomial-time computable for Borda. (Section 7)

2 Preliminaries

An election consists of a set of candidates C and a collection of voters V where each voter has a ranking (total order preference) over the set of candidates. For example, $a > b > c$, where $>$ denotes strict preference, is a vote over $\{a, b, c\}$.

We consider voting rules that are social preference functions, which map an election to a set of one or more rankings (total orderings) of the candidates. One of the most important social preference functions is the Kemeny rule [Kemeny, 1959].

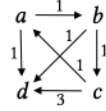
A ranking $>$ is a Kemeny consensus if the sum of the Kendall tau distances to the voters is minimal, i.e.,

$\sum_{a>b} N(b, a)$ is minimal, where for candidates a and b , $N(b, a)$ denotes the number of voters that state $b > a$.

It is often useful to refer to the induced weighted majority graph of the election when working with the Kemeny rule. The weighted majority graph of an election (C, V) has a vertex for each candidate and for each pair of candidates $a, b \in C$ if $N(a, b) > N(b, a)$ there is an arc (a, b) labeled with $N(a, b) - N(b, a)$.

Example 1. Consider an election with candidates $\{a, b, c, d\}$ and three voters with their votes, and the corresponding induced weighted majority graph below.

- $a > b > c > d$
- $c > a > d > b$
- $b > c > d > a$



Consensuses: $a > b > c > d$, $b > c > a > d$, $c > a > b > d$. Kendall tau distance: 6.

We consider several different computational problems relating to the Kemeny rule. For readability, the formal definitions of these problems are deferred to where the results appear.

We assume that the reader is familiar with the complexity classes P, NP, and coNP. Our complexity results also concern the class $\Sigma_2^P = \text{NP}^{\text{NP}}$, a class at the second level of the polynomial hierarchy, which is the class of problems solvable by an NP-machine with access to an NP oracle [Meyer and Stockmeyer, 1972; Stockmeyer, 1976].

3 Consensus Recognition

We now formally define the problem of determining whether a ranking is a Kemeny consensus.

Name: Kemeny Consensus Recognition

Given: An election (C, V) and a total order X .

Question: Is X a Kemeny consensus of the election?

Hudry [2013] observes that the Kemeny Consensus Recognition problem (there called Order Recognition) is in coNP, and conjectures it is coNP-complete.¹ Since it is easier to think about NP than about coNP, we will often look at the complement, i.e., determining whether X is not a consensus. As usual, the upper bound is easy to see: Note that X is not a Kemeny consensus if and only if there exists a total order whose distance to the election is less than that of X .

Also note that the Kemeny Consensus Recognition problem is not in NP unless $\text{NP} = \text{coNP}$, since the Kemeny score of an election (Kendall tau distance to a consensus) is greater than k if and only if there exists a total order that is a Kemeny consensus whose score is greater than k . So if Kemeny Consensus Recognition is in NP, then determining if the Kemeny score of an election is greater than k is in NP. This latter problem is coNP-complete, since it is in essence the complement of the problem Kemeny Score, which is NP-complete [Bartholdi *et al.*, 1989b].

The above does not imply that Kemeny Consensus Recognition is coNP-hard. It merely says that, assuming $\text{NP} \neq$

coNP, the problem is in coNP – NP. Under the assumption that $\text{NP} \neq \text{coNP}$, there are problems in coNP – NP that are not coNP-complete [Ladner, 1975]. A natural candidate of such a problem is graph nonisomorphism problem. Note that this problem has some “easiness” properties that are not shared by any natural coNP-complete problem, such as a zero-knowledge proof for the complement [Goldreich *et al.*, 1991] and a quasi-polynomial time algorithm [Babai, 2016].

We will now prove Hudry’s conjecture that Kemeny Consensus Recognition is coNP-complete (as Theorem 5).

Optimal solution recognition problems induced by optimization problems are very natural decision problems, but there are only a couple of results in the literature. Papadimitriou and Steiglitz [1978, Theorem 5] show that Minimum TSP Tour Recognition is coNP-complete. Armstrong and Jacobson [2003] study the global verification problem (which is the complement of optimal solution recognition) related to various NP optimization problems and show that the optimal solution recognition problems for Vertex Cover, MAX-SAT, and MAX- k -SAT ($k \geq 2$) are each coNP-complete.

Our proof of Theorem 5 will use the coNP-completeness of Minimum Vertex Cover Recognition.

Name: Minimum Vertex Cover Recognition

Given: A graph G and a set of vertices X .

Question: Is X a minimum vertex cover of G ?

Theorem 2 (Armstrong and Jacobson, 2003). *Minimum Vertex Cover Recognition is coNP-complete.*

The Kemeny Score problem was shown hard by a reduction from Feedback Arc Set (FAS) [Bartholdi *et al.*, 1989b]. In the full version [Fitzsimmons and Hemaspaandra, 2021], we show the following problem coNP-complete.

Name: Minimum FAS Recognition

Given: An irreflexive and antisymmetric directed graph G and a set of arcs X .

Question: Is X a minimum fas of G (a minimum set of arcs such that $G - X$ is acyclic)?

Theorem 3. *Minimum FAS Recognition is coNP-complete.*

Feedback arc sets and Kemeny consensus are very closely related [Bartholdi *et al.*, 1989b]. We use the following slightly unusual formulation of this relationship.

Lemma 4. *For G a directed graph, let $e(G)$ be the election with candidates $V(G)$ and for each arc $(a, b) \in A(G)$ one voter voting $a > b$ followed by all candidates in $V(G) - \{a, b\}$ in lexicographical order and one voter voting all candidates in $V(G) - \{a, b\}$ in reverse lexicographical order followed by $a > b$. This election is computable in polynomial time, and has G with all arc weights 2 as its induced weighted majority graph [McGarvey, 1953].*

For X a minimal fas of G (i.e., X is a fas of G and no strict subset of X is a fas), and $\hat{X}$ a total order consistent with $G - X$ (i.e., if $(a, b) \in A(G) - X$, then $a > b$ in $\hat{X}$), it holds that X is a minimum fas if and only if $\hat{X}$ is a Kemeny consensus of $e(G)$.

This gives us a reduction from Minimum FAS Recognition to Kemeny Consensus Recognition, which gives us the following theorem.

¹Hudry uses Turing reductions. We look at the standard notion of polynomial-time many-one reductions, which gives stronger results.

Theorem 5. *Kemeny Consensus Recognition is coNP-complete.*

Proof. Given G and X , if X is not a minimal fas (which can be determined in polynomial time), then output something that is not an instance of the problem. If X is a minimal fas, then output $e(G)$ (as defined in Lemma 4) and a total order consistent with $G - X$ (which can be computed in polynomial time, since $G - X$ is acyclic). $\square$

From the above, one might think that coNP-hardness for an optimal solution recognition problem follows from a straightforward modification of the reduction for the related NP-complete decision problem. But this is only the case when the witnesses of the two decision problems directly correspond to each other. This is usually not the case. See for example the proof of the analogous result for tournaments later in this paper (Theorem 15).

4 Manipulative-Actions-to-Consensus

In the previous section we showed that Kemeny Consensus Recognition is coNP-complete. Given the hardness of this problem, does it follow that manipulative actions with the goal to reach a specific consensus are hard? This is true if we look at decision problems such as “Given an election and a total order X , can we perform a manipulative action such that X is a consensus.” Such decision problems typically inherit the coNP-hardness (e.g., by having no manipulators). It is still interesting to look at these decision problems, since they may be complete for classes above coNP, which limits the tools we have to solve these problems. Standard approaches for solving problems in NP or coNP such as using SAT solvers are not appropriate for solving problems that are complete for higher levels of the polynomial hierarchy such as Σ_2^P .

We will also look at the problem of determining the manipulative action. It is possible that it is easy to determine the best action, even though it is hard to determine whether such an action leads to the desired outcome. In fact:

Observation 6. *Consider Kemeny-Manipulation-to-Consensus, in which we are given an election, a collection of manipulators, and a desired consensus X , and we ask if the manipulators can vote so that X is a Kemeny consensus of the resulting election. It is easy to see that a total order X can be made a consensus if and only if X is a consensus when all manipulators vote X (for details see the full version).*

And so the optimal action for the manipulators is straightforward, namely to vote X , and the complexity of the associated decision problem Kemeny-Manipulation-to-Consensus is the same as for the recognition problem, namely, coNP-complete.

Now we ask: What makes it easy to determine the manipulative action? Is it the election system (Kemeny)? Is it the manipulative action (manipulation)? Is it the manipulative goal (consensus)?

Note that the observation above has interesting repercussions for other manipulative actions and for other manipulative goals. For example, in bribery, we can assume that all bribed voters vote the same X , where X is a consensus after bribery. And if the goal of the manipulators is to make a

preferred candidate p a winner, we can assume that all manipulators vote the same X , where X is a consensus after manipulation. (Since if there is a manipulation such that p is a winner, then there is a manipulation with a consensus X that ranks p first. But then X is also a consensus when all manipulators vote X .)

Despite this simplicity of all manipulators/bribed voters voting the same, we will provide evidence in the next couple of sections that determining the optimal manipulation to obtain a Kemeny consensus is easy because of the combination of election system (Kemeny), manipulative action (manipulation), and manipulative goal (consensus).

5 Control-to-Consensus

Electoral control models whether the structure of an election can be modified to ensure a preferred outcome [Bartholdi *et al.*, 1992]. Control(-to-Winner) problems for Kemeny tend to be Σ_2^P -complete [Fitzsimmons *et al.*, 2019] (note that winner determination is already complete for parallel access to NP [Hemaspaandra *et al.*, 2005]). In this section we provide evidence that this is also the case for Control-to-Consensus. Note that this implies that, unless NP = coNP, the optimal control action to obtain a Kemeny consensus is not polynomial-time computable (in contrast to manipulation).

Σ_2^P lower bounds are often hard to prove, in part because there are fewer known Σ_2^P -complete problems (see Schaefer and Umans [2002] for a list) and also because one needs a closer correspondence between the two problems than for NP-hardness reductions.

We first show that optimal solution recognition for the Σ_2^P -complete problem Generalized Node Deletion (GND) [Rutenburg, 1994] is Π_2^P -complete.

Name: Minimum GND Recognition

Given: A graph G , integer ℓ , and set of vertices X .

Question: Is X a minimum set of vertices such that $G - X$ does not contain $K_{\ell+1}$ (a clique of size $\ell + 1$)?

Theorem 7. *Minimum GND Recognition is Π_2^P -complete.*

This is the first completeness result at the second level of the polynomial hierarchy for optimal solution recognition. For details, see the full version.

The natural deletion analogues of Minimum Vertex Cover (resp. FAS) Recognition where we are additionally given a delete limit k and ask if there exists a set of at most W vertices such that X is a minimum vertex cover (minimum fas, respectively) of $G - W$ are also Σ_2^P -complete (for details see the full version). Since there is a dearth of natural Σ_2^P -complete problems, these results are interesting in their own right.

We will now look at control by deleting candidates (CDC). We will show that the following problem is Σ_2^P -complete.

Name: Kemeny-CDC-to-Consensus

Given: An election (C, V) , delete limit k , and a total order X over C .

Question: Does there exist a set $D \subseteq C$ of at most k candidates such that X restricted to $C - D$ is a Kemeny consensus of $(C - D, V)$?

Though this problem is not the most natural, it does provide evidence that Kemeny-Control-to-Consensus problems are Σ_2^P -complete.

Theorem 8. *Kemeny-CDC-to-Consensus is Σ_2^P -complete.*

Note that in the definition of Kemeny-CDC-to-Consensus, it is important that X is a total order over C . If it were over $C - D$, we would be able to see which candidates are deleted from the problem instance (and the problem would be equivalent to Kemeny Consensus Recognition). However, this makes the problem different from the Σ_2^P -complete FAS problem, since total order X ranks all candidates. This means that the straightforward Σ_2^P analogue of the reduction from Minimum FAS Recognition to Kemeny Consensus Recognition from the proof of Theorem 5 does not work. In that reduction, the order was a total order consistent with the directed acyclic graph $G - X$, where X is a fas. However, before deletion, $G - X$ is not necessarily acyclic! To prove Σ_2^P -completeness of Kemeny-CDC-to-Consensus, we need different, less natural Σ_2^P -complete versions of Vertex Cover and Feedback Arc Set Recognition that look more like Kemeny-CDC-to-Consensus. In particular, we need to make sure that the solutions for Vertex Cover and Feedback Arc Set are (not necessarily optimal) solutions for the whole graph. Details can be found in the full version.

There are other types of control, most notably control by adding candidates and control by adding/deleting voters. As problems, these are more compelling. For example, the definition of control by deleting voters (CDV) to consensus is straightforward and natural.

Name: Kemeny-CDV-to-Consensus

Given: An election (C, V) , delete limit k , and a total order X .

Question: Does there exist a set $W \subseteq V$ of at most k voters such that X is a Kemeny consensus of $(C, V - W)$?

One might think that, in analogy to optimal action for manipulators being voting the consensus, the optimal action for CDV would be to simply delete voters furthest from the desired consensus (and for CAV to simply add voters closest to the desired consensus). However, the following example shows that this is not the case.

Example 9. *Consider an election with candidates $\{a, b, c\}$, five voters: three voting $a > b > c$, one voting $a > c > b$, and one voting $c > b > a$, delete limit 1, and desired consensus $a > c > b$.*

Note that $a > c > b$ is not a consensus. If we delete the voter furthest from the consensus (i.e., the voter voting $c > b > a$) then $a > c > b$ is not a consensus, but if we delete one of the $a > b > c$ voters then $a > c > b$ is a consensus.

This example with one of the $a > b > c$ voters and the $c > b > a$ voter as the unregistered voters and an add limit of 1 shows the analogous counterexample for Kemeny-CAV-to-Consensus.

We conjecture that all these control-to-consensus problems are Σ_2^P -complete. However, we cannot modify the approach above in a simple way, since one arc in a graph does not correspond to one voter in the corresponding election. This is also the reason that the complexity of “regular” Kemeny voter control(-to-winner) is still open [Fitzsimmons *et al.*, 2019].

6 Manipulation(-to-Winner)

Showing that manipulation is hard is hard! For example, it is not too hard to show that control for Borda is hard [Russell,

2007], but the complexity of (coalitional) manipulation for Borda was open for a long time and NP-completeness was shown only after discovering an appropriate NP-complete problem in scheduling [Davies *et al.*, 2014; Betzler *et al.*, 2011]. And proving the NP-completeness of manipulation for Copeland $^\alpha$ for $\alpha \neq 0.5$ involved construction of elaborate gadgets [Faliszewski *et al.*, 2008; Faliszewski *et al.*, 2010].

The reason that it is so hard to prove manipulation hard is that the manipulators do not follow any structure other than voting a total order. This means that basically all the structure needs to come from the nonmanipulators.

For Kemeny, we know from Section 4 that we can assume that all manipulators vote the same. So all we have to work with is one total order. Though we conjecture that Kemeny-Manipulation is Σ_2^P -complete, we have not succeeded in proving this. The closest we got is the following theorem, which is explained in more detail after the theorem statement. We note that this is the first Σ_2^P -complete manipulation result.

Theorem 10. *Slater-Manipulation where candidates have unary weights is Σ_2^P -complete, even for one manipulator.*

The Slater rule [Slater, 1961] can be viewed as a qualitative version of Kemeny. It is defined as follows. A ranking $>$ is a Slater consensus if the number of disagreements with the majority graph induced by the voters is minimal (note that for Slater we look at the induced majority graph while for Kemeny we look at the induced *weighted* majority graph). In our Slater proofs, we will often look at the Slater *score* of a ranking, which is the number of *agreements* with the majority graph, i.e., $\|C\|(\|C\| - 1)/2$ minus the number of disagreements. So, the higher the score, the better the ranking.

Candidates with weights for Kemeny are a natural notion [Kumar and Vassilvitskii, 2010]. For candidates with weights, the contribution of each candidate to the score is multiplied by its weight. For our result, we need only unary weights, which is a step in the direction of not needing weights.

The high-level reason that we obtain this result for Slater and not for Kemeny is that in Slater we can “freeze” certain arcs in the majority graph. For example, if we have three non-manipulators all voting $a > b$, and we have one manipulator, then the manipulator cannot change the contribution to the Slater score of the pair $\{a, b\}$. Note that this is not the case for Kemeny.

Candidates with weights also give more structure to the manipulator. For example, if we have two candidates a and b of weight 10, then the manipulator can rank $a > b$ or $b > a$. If we replace a by 10 little a ’s and 10 little b ’s, the manipulator can rank those in any messy order it wants.

Proof Sketch of Theorem 10. To show Σ_2^P -hardness, we will reduce from QSAT $_2$ [Stockmeyer, 1976; Wrathall, 1976]. Consider cnf formula $\phi = D_1 \wedge \dots \wedge D_{m-1}$ over variables $x_2, \dots, x_n$ and let $\phi' = (x_1 \vee D_1) \wedge \dots \wedge (x_1 \vee D_{m-1}) \wedge \neg x_1$. (Notice that ϕ' has m clauses over variables $x_1, \dots, x_n$.) Without loss of generality, assume that if ϕ is not satisfiable, then at most $m - 3$ clauses can be satisfied (this can be accomplished by doubling each clause). We will in polynomial time compute an election with one manipulator such that $\exists x_{n'+1} \dots x_n \neg (\exists x_2 \dots x_{n'} \phi(x_2, \dots, x_n))$ if and only if

the manipulator can vote such that the candidate $+_1$ becomes a winner.

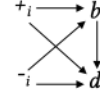
First note that if $\exists x_{n'+1} \dots x_n \neg(\exists x_2 \dots x_{n'} \phi(x_2, \dots, x_n))$, then $\exists x_{n'+1} \dots x_n$ such that any assignment with $x_1 = \text{true}$ satisfies $m - 1$ clauses of ϕ' and any assignment with $x_1 = \text{false}$ satisfies at most $m - 2$ clauses of ϕ' . If it is not the case that $\exists x_{n'+1} \dots x_n \neg(\exists x_2 \dots x_{n'} \phi(x_2, \dots, x_n))$, then any assignment with $x_1 = \text{true}$ satisfies $m - 1$ clauses of ϕ' and there is an assignment with $x_1 = \text{false}$ that satisfies m clauses of ϕ' .

Now apply the reduction from MAX-SAT to Slater score from Conitzer [2006] to ϕ' , with the following change. We replace each size M “super-candidate” (a group of M candidates that, for the purposes of Slater score, can be treated as one single candidate of weight M) by one candidate of weight M . This ensures that we only get Slater consensus of a specific form and no “rogue” consensus (this was not a problem in Conitzer [2006], since for the purposes of Slater scores it is enough that there exist a Slater consensus of the appropriate form; however, since we are interested in whether a specific candidate can be a winner or not, we need to preclude rogue consensus with a rogue winner). This computes a tournament² in which each variable x_i is represented by a subtournament T_i (which includes the vertices $+_i$ and $-_i$) and each clause by a candidate c_k . The relevant properties of the reduction are as follows.

- All Slater consensus rank $T_1 > \dots > T_n$.
- Slater consensus correspond to assignments satisfying a maximum number of clauses of ϕ' in the following way. For C_k a true clause, candidate c_k is ranked (in a specific way) among the candidates in a subtournament T_i whose ranking encodes an assignment to x_i that makes C_k true.
- If T_1 's ranking encodes $x_1 = \text{true}$, then candidate $+_1$ is ranked first. If T_1 's ranking encodes $x_1 = \text{false}$, then candidate $-_1$ is ranked first.
- $+_1$ is a Slater winner or $-_1$ is a Slater winner.
- There is an assignment that satisfies $\geq k$ clauses of ϕ' if and only if the Slater score is $\geq B + kM$ (here, B (the baseline score) and M are polynomial-time computable constants that are small enough to be given in unary).

We want to keep as much of this construction as possible. First we double every voter, so that the arc weights in the induced tournament are all 2. We have one manipulator. Note that one manipulator cannot change an arc of weight 2. We will now change the tournament a little, in such a way that the manipulator can “set” the values of the existential variables $(x_{n'+1}, \dots, x_n)$, but nothing else.

In the construction, we change how the existential variables are represented. Each such variable x_i will be represented by a graph consisting of four candidates $+_i, -_i, b_i, d_i$, each of weight M (recall that we allow unary weights for the candidates). These four candidates are connected by the following weight-2 arc:



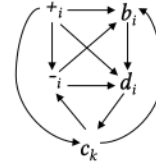
The only “undeclared” arc is between $+_i$ and $-_i$. This arc will be determined by the vote of the manipulator. $+_i > -_i$ will correspond to setting x_i to true and $-_i > +_i$ will correspond to setting x_i to false. Let T'_i be the subtournament after the manipulator vote.

For clause candidate c_k , we add the following arcs.

- If x_i occurs positively in C_k , add arcs $+_i \rightarrow c_k, c_k \rightarrow -_i, c_k \rightarrow b_i, d_i \rightarrow c_k$.
- If x_i occurs negatively in C_k , add arcs $-_i \rightarrow c_k, c_k \rightarrow +_i, c_k \rightarrow b_i, d_i \rightarrow c_k$.
- If x_i does not occur in C_k , add arcs $c_k \rightarrow +_i, c_k \rightarrow -_i, b_i \rightarrow c_k, d_i \rightarrow c_k$.

All other arcs are unchanged. In particular, all Slater consensus rank $T_1 > \dots > T_{n'} > T'_{n'+1} > \dots > T'_n$. Note that if we rank candidate c_k before or after T'_i , this contributes a baseline score of $2M$ to the Slater score. The only way a clause candidate c_k can gain points from T'_i over the baseline score of $2M$ is if c_k is ranked among the candidates in T'_i and the value of x_i encoded by the ranking of T'_i makes C_k true. In that case, we gain M extra points.

Example 11. For example, if x_i is true and x_i occurs positively in C_k , we obtain the subtournament below and we can order $+_i > c_k > -_i > b_i > d_i$ so that c_k gains $3M$ points from T'_i for the Slater score.



From this, we get the following, for a specific fixed assignment to $x_{n'+1}, \dots, x_n$ (and the manipulator voting accordingly).

- Slater consensus correspond to assignments satisfying a maximum number of clauses of ϕ' in the following way. For C_k a true clause, candidate c_k is ranked (in a specific way) among the candidates in a subtournament T_i or T'_i whose ranking encodes an assignment to x_i that makes C_k true.
- If T_1 's ranking encodes $x_1 = \text{true}$, then $+_1$ is ranked first. If it encodes $x_1 = \text{false}$, then $-_1$ is ranked first.
- $+_1$ is a Slater winner or $-_1$ is a Slater winner.
- There is an assignment that satisfies $\geq k$ clauses of ϕ' if and only if the Slater score is $\geq \hat{B} + kM$ (here, $\hat{B}$ is the baseline score of the new construction).

If $\exists x_{n'+1} \dots x_n \neg(\exists x_2 \dots x_{n'} \phi(x_2, \dots, x_n))$, then $\exists x_{n'+1} \dots x_n$ such that any assignment with $x_1 = \text{true}$ satisfies $m - 1$ clauses of ϕ' and any assignment with $x_1 = \text{false}$ satisfies at most $m - 2$ clauses of ϕ' . Let the manipulator vote

²For every pair of vertices a, b , $a \rightarrow b$ or $b \rightarrow a$, but not both.

according to the assignment to $x_{n'+1} \cdots x_n$. Then the Slater score of a total order starting with -1 is $< \widehat{B} + (m-1)M$ and the Slater score of a total order starting with $+1$ is $\geq \widehat{B} + (m-1)M$. It follows that $+1$ is a Slater winner.

For the converse, suppose the manipulator can vote such that $+1$ is a winner. Consider the assignment to $x_{n'+1} \cdots x_n$ induced by the manipulator. If $\phi(x_2, \dots, x_{n'})$ were satisfiable, then any assignment with $x_1 = \text{true}$ satisfies $m-1$ clauses of ϕ' and there is an assignment with $x_1 = \text{false}$ that satisfies m clauses of ϕ' . It follows that the Slater score $\geq \widehat{B} + mM$ and that the ranking of T_1 in any Slater consensus encodes that x_1 is false. This implies that -1 is always ranked first, which contradicts the assumption that $+1$ is not a winner. $\square$

Slater is an interesting system in itself (see, e.g., Hüllermeier and Fürnkranz [2004] for motivation from the preference learning literature). But here we are mostly interested in the closeness of Slater to Kemeny, and view Theorem 10 as supporting our conjecture that Kemeny-Manipulation is Σ_2^P -complete.

Many lower bound proofs for Kemeny transfer to Slater and vice versa by the following simple observation (this is implicit in any source comparing Kemeny and Slater and explicitly stated for tournaments where every arc has weight 1 in Bachmeier *et al.* [2019]).

Observation 12. *If all weights in the weighted majority graph are the same, then the Kemeny consensus and Slater consensus coincide.*

Looking back at the proofs of the results from the previous section, we immediately obtain the following corollaries.

Corollary 13. *Slater Consensus Recognition is coNP-complete.*

Corollary 14. *Slater-CDC-to-Consensus is Σ_2^P -complete.*

The definition of Slater from this section allows an even number of voters. Not all Slater definitions allow ties, i.e., Slater is sometimes defined only for the case where the majority graph is a tournament. And also Kemeny for tournaments is an interesting problem. The proofs from the previous section construct elections with an even number of voters and so do not give the analogous results about tournaments. It is much more difficult to prove hardness for tournaments. For example, feedback arc set is one of the original 21 NP-complete problems from Karp [1972], but the complexity of feedback arc set for tournaments was open for a long time. Alon [2006] showed NP-completeness by derandomizing the reduction from Ailon *et al.* [2008]. Conitzer [2006] gave a direct proof of the result. We will modify the lovely reduction from Conitzer [2006] to prove the following. For Slater this answers an open question from Hudry [2010]. For details see the full version.

Theorem 15. *Slater and Kemeny Consensus Recognition for tournaments is coNP-complete.*

7 Manipulation-to-Consensus

Recall from Observation 6 that for Kemeny-Manipulation-to-Consensus the optimal action for the manipulators is to vote

their desired consensus. In contrast we show that for Borda-Manipulation-to-Consensus it is hard to compute the optimal action for the manipulators. The Borda election system [de Borda, 1781] is an important rule that can be used to produce a consensus by ranking each candidate by their Borda score. For an m -candidate election, each voter contributes $m-i$ points to the candidate ranked i th in their vote. Note that in a Borda consensus candidates with the same score are tied.

We first show that for Borda it is not always the case that a manipulator should vote the desired consensus.

Example 16. *Let there be the following five nonmanipulative voters: Two voters voting $a > b > c > d$, two voters voting $b > a > c > d$, and one voter voting $b > c > a > d$. Let there be one manipulator with a preferred consensus of $a > b > c > d$.*

Before manipulation, the candidates have the following Borda scores: $\text{score}(a) = 11, \text{score}(b) = 13, \text{score}(c) = 6$, and $\text{score}(d) = 0$, and so the consensus is $b > a > c > d$.

If the manipulator votes their preferred consensus the resulting Borda scores are: $\text{score}(a) = 14, \text{score}(b) = 15, \text{score}(c) = 7$, and $\text{score}(d) = 0$, with the Borda consensus of $b > a > c > d$.

However, manipulation is possible when the manipulator instead votes $a > c > d > b$.

We now consider the complexity of Borda-Manipulation-to-Consensus. The proof from Davies *et al.* [2014], which shows that coalitional manipulation for Borda is NP-complete constructs an election such that manipulation is possible if and only if after manipulation the candidates $p, a_1, \dots, a_{q+1}$ are all tied with the highest Borda score and the remaining candidate a_{q+2} has a strictly lower score, i.e., the Borda consensus is $\{p, a_1, \dots, a_{q+1}\} > a_{q+2}$. It follows that:

Theorem 17. *Borda-Manipulation-to-Consensus is NP-complete.*

This immediately implies that the optimal action for the manipulators is not polynomial-time computable, unless $P = NP$.

8 Conclusion

We showed that even checking if a given ranking is a Kemeny consensus is coNP-complete. We also showed that, though determining whether a ranking is a Kemeny consensus is hard, the optimal action for the manipulators to reach a consensus is easy. We provided evidence that this simplicity is caused by the combination of election system (Kemeny), manipulative action (manipulation), and manipulative goal (consensus).

For future work, we are most interested in showing our conjecture that Kemeny-Manipulation(-to-Winner) is Σ_2^P -complete. In addition, the study of elections where candidates have weights is very natural and interesting.

Acknowledgments

This work was supported in part by NSF-DUE-1819546. Research done in part while Zack Fitzsimmons was on research leave at Rensselaer Polytechnic Institute. We thank the reviewers for their helpful feedback and suggestions.

References

- [Ailon *et al.*, 2008] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *JACM*, 55(5):Article 23, 2008.
- [Alon, 2006] N. Alon. Ranking tournaments. *SIDMA*, 20(1–2):137–142, 2006.
- [Armstrong and Jacobson, 2003] D. Armstrong and S. Jacobson. Studying the complexity of global verification for NP-hard discrete optimization problems. *J. Glob. Optim.*, 27:83–96, 2003.
- [Babai, 2016] L. Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proc. of STOC-16*, pages 684–697, June 2016.
- [Bachmeier *et al.*, 2019] G. Bachmeier, F. Brandt, C. Geist, P. Harrenstein, K. Kardel, D. Peters, and H. Seedig. k -Majority digraphs and the hardness of voting with a constant number of voters. *JCSS*, 105:130–157, 2019.
- [Bartholdi *et al.*, 1989a] J. Bartholdi, III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *SCW*, 6(3):227–241, 1989.
- [Bartholdi *et al.*, 1989b] J. Bartholdi, III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *SCW*, 6(2):157–165, 1989.
- [Bartholdi *et al.*, 1992] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Math. Comput. Model.*, 16(8/9):27–40, 1992.
- [Betzler *et al.*, 2011] N. Betzler, R. Niedermeier, and G. Woeginger. Unweighted coalitional manipulation under the Borda rule is NP-hard. In *Proc. of IJCAI-11*, pages 55–60, August 2011.
- [Conitzer, 2006] V. Conitzer. Computing Slater rankings using similarities among candidates. In *Proc. of AAAI-06*, pages 613–619, July 2006.
- [Davies *et al.*, 2014] J. Davies, G. Katsirelos, N. Narodytska, T. Walsh, and L. Xia. Complexity of and algorithms for the manipulation of Borda, Nanson’s and Baldwin’s voting rules. *AIJ*, 217:20–42, 2014.
- [de Borda, 1781] J.-C. de Borda. Mémoire sur les élections au scrutin. *Histoire de l’Académie Royale des Sciences*, pages 657–664, 1781.
- [Dwork *et al.*, 2001] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. of WWW-01*, pages 613–622, March 2001.
- [Faliszewski and Rothe, 2016] P. Faliszewski and J. Rothe. Control and bribery in voting. In *Handbook of Computational Social Choice*, pages 146–168. Cambridge University Press, 2016.
- [Faliszewski *et al.*, 2008] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *Proc. of AAMAS-08*, pages 983–990, May 2008.
- [Faliszewski *et al.*, 2010] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Manipulation of Copeland elections. In *Proc. of AAMAS-10*, pages 367–374, May 2010.
- [Fitzsimmons and Hemaspaandra, 2021] Z. Fitzsimmons and E. Hemaspaandra. Kemeny consensus complexity. Tech. Rep. arXiv:2105.08540 [cs.GT], arXiv.org, May 2021.
- [Fitzsimmons *et al.*, 2019] Z. Fitzsimmons, E. Hemaspaandra, A. Hoover, and D. Narváez. Very hard electoral control problems. In *Proc. of AAAI-19*, pages 1933–1940, January/February 2019.
- [Goldreich *et al.*, 1991] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *JACM*, 38(3):691–729, 1991.
- [Hemaspaandra *et al.*, 2005] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *TCS*, 349(3):382–391, 2005.
- [Hudry, 2010] O. Hudry. On the complexity of Slater’s problems. *EJOR*, 203(1):216–221, 2010.
- [Hudry, 2013] O. Hudry. Complexity of computing median linear orders and variants. *ENDM*, 42:57–64, 2013.
- [Hüllermeier and Fürnkranz, 2004] E. Hüllermeier and J. Fürnkranz. Comparison of ranking procedures in pairwise preference learning. In *Proc. of IPMU-04*, 2004.
- [Jackson *et al.*, 2008] B. Jackson, S. Schnable, and S. Aluru. Consensus genetic maps as media orders from inconsistent sources. *TCBB*, 5(2):161–171, 2008.
- [Karp, 1972] R. Karp. Reducibility among combinatorial problems. In *Proc. of Symposium on Complexity of Computer Computations*, pages 85–103, 1972.
- [Kemeny, 1959] J. Kemeny. Mathematics without numbers. *Daedalus*, 88:577–591, 1959.
- [Kumar and Vassilvitskii, 2010] R. Kumar and S. Vassilvitskii. Generalized distances between rankings. In *Proc. of WWW-10*, pages 571–580, April 2010.
- [Ladner, 1975] R. Ladner. On the structure of polynomial time reducibility. *JACM*, 22(1):155–171, 1975.
- [McGarvey, 1953] D. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.
- [Meyer and Stockmeyer, 1972] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proc. of FOCS-72*, pages 125–129, October 1972.
- [Papadimitriou and Steiglitz, 1978] C. Papadimitriou and K. Steiglitz. Some examples of difficult traveling salesman problems. *Oper. Res.*, 26(3):434–443, 1978.
- [Russell, 2007] N. Russell. Complexity of control of Borda count elections. Master’s thesis, Rochester Institute of Technology, 2007.
- [Rutenburg, 1994] V. Rutenburg. Propositional truth maintenance systems: Classification and complexity analysis. *AMAI*, 10(3):207–231, 1994.
- [Schaefer and Umans, 2002] M. Schaefer and C. Umans. Completeness in the polynomial-time hierarchy: Part I: A compendium. *SIGACT News*, 33(3):32–49, 2002.
- [Slater, 1961] P. Slater. Inconsistencies in a schedule of paired comparisons. *Biometrika*, 48(3/4):303–312, 1961.
- [Stockmeyer, 1976] L. Stockmeyer. The polynomial-time hierarchy. *TCS*, 3(1):1–22, 1976.
- [Wrathall, 1976] C. Wrathall. Complete sets and the polynomial-time hierarchy. *TCS*, 3(1):23–33, 1976.