Rank Aggregation Using Scoring Rules

Niclas Boehmer[1], Robert Bredereck[2], Dominik Peters[3]

[1] Algorithmics and Computational Complexity, Technische Universität Berlin
[2] Institut für Informatik, TU Clausthal
[3] CNRS, LAMSADE, Université Paris Dauphine–PSL
niclas.boehmer@tu-berlin.de, robert.bredereck@tu-clausthal.de, dominik.peters@lamsade.dauphine.fr

Abstract

To aggregate rankings into a social ranking, one can use scoring systems such as Plurality, Veto, and Borda. We distinguish three types of methods: ranking by score, ranking by repeatedly choosing a winner that we delete and rank at the top, and ranking by repeatedly choosing a loser that we delete and rank at the bottom. The latter method captures the frequently studied voting rules Single Transferable Vote (aka Instant Runoff Voting), Coombs, and Baldwin. In an experimental analysis, we show that the three types of methods produce different rankings in practice. We also provide evidence that sequentially selecting winners is most suitable to detect the "true" ranking of candidates. For different rules in our classes, we then study the (parameterized) computational complexity of deciding in which positions a given candidate can appear in the chosen ranking. As part of our analysis, we also consider the WINNER DETERMINATION problem for STV, Coombs, and Baldwin and determine their complexity when there are few voters or candidates.

1 Introduction

Rank aggregation, the task of aggregating several rankings into a single ranking, sits at the foundation of social choice as introduced by Arrow (1951). Besides preference aggregation, it has numerous important applications, for example in the context of meta-search engines (Dwork et al. 2001), of juries ranking competitors in sports tournaments (Truchon 1998), and multi-criteria decision analysis.

One of the best-known methods for aggregating rankings is Kemeny's (1959) method: A Kemeny ranking is a ranking that minimizes the average swap distance (Kendall-tau distance) to the input rankings. It is axiomatically attractive (Young and Levenglick 1978; Can and Storcken 2013; Bossert and Sprumont 2014) and has an interpretation as a maximum likelihood estimator (Young 1995) making it well-suited to epistemic social choice that assumes a ground truth.

However, Kemeny's method is hard to compute (Bartholdi, Tovey, and Trick 1989; Hemaspaandra, Spakowski, and Vogel 2005) which makes the method problematic to use, especially when there are many candidates to rank (for example, when ranking all applicants to a university). Even if computing the ranking is possible, it is coNP-hard to verify if a ranking is indeed a Kemeny ranking (Fitzsimmons and Hemaspaandra 2021). Thus, third parties cannot easily audit, interpret, or understand the outcome, making systems based on Kemeny's method potentially unaccountable. This limits its applicability in democratic contexts.

These two drawbacks motivate the search for computationally simpler and more transparent methods for aggregating rankings. There is a significant literature on polynomial-time approximation algorithms for Kemeny's method (Coppersmith, Fleischer, and Rudra 2006; Kenyon-Mathieu and Schudy 2007; Ailon, Charikar, and Newman 2008; van Zuylen and Williamson 2009), but these algorithms are typically not attractive beyond their approximation guarantee. In particular, they would typically not fare well in an axiomatic analysis, and are unlikely to be understood by and appealing to the general public (many are based on derandomization).

Instead, we turn to one of the fundamental tools of social choice: positional scoring rules. These rules transform voter rankings into scores for the candidates. For example, under the *Plurality* scoring rule, every voter gives 1 point to their top-ranked candidate. Under the *Veto* (or anti-plurality) scoring rule, voters give -1 point to their last-ranked candidate and zero points to all others. Under the *Borda* scoring rule, every voter gives m points to their top-ranked candidate, $m-1$ points to their second-ranked candidate, and so on, giving 1 point to their last-ranked candidate. We study three ways of using scoring rules to aggregate rankings:

- *Score*: We rank the candidates in order of their score, higher-scoring candidates being ranked higher.
- *Sequential-Winner*: We take the candidate c with the highest score and rank it top in the aggregate ranking. We then delete c from the input profile, re-calculate the scores, and put the new candidate with the highest score in the second position, and so on.
- *Sequential-Loser:* We take the candidate c with the lowest score and rank it last. We then delete c, re-calculate the scores, and put the new candidate with the lowest score in the second-to-last position, and so on.

Ranking by score is the obvious way of using scoring rules for rankings, and so it has been studied in the social choice literature (Smith 1973; Levenglick 1977). These rules are frequently used in practice. Examples include the European Song Contest, the "ARTU" aggregation of university rankings (using Borda), ranking countries by number of Olympic gold medals (plurality), and in a certain sense also participatory budgeting (k-approval, with a postprocessing step to turn the ranking into a knapsack). Sequential-Loser captures as special cases the previously studied rules Single Transferable Vote (also known as Instant Runoff Voting, among other names, which is used for political elections in Australia, Ireland, and some jurisdictions in the US), Coomb's method, and Baldwin's method. These are typically used as voting rules that elect a single candidate, but they can also be understood as rank aggregation methods. On the other hand, despite being quite natural, Sequential-Winner methods appear not to have been formally studied in the literature (to our knowledge). These rules mirror sequential decision making in real-world situations. For example, an academic department could vote on whom to make a job offer, and in case of rejections, repeatedly re-vote. Then the order in which offers go out would map to, for example, Sequential-Plurality-Winner. Another example are political parties who have to decide on a party list for a parliament election. In many cases (for example in Germany), these are voted on by party members, voting first for the first list place, then for the second, etc., using Plurality-like rules each time.

1.1 Our Contributions

Axiomatic Properties (Section 4) Based on the existing literature, we begin by describing some axiomatic properties of the methods in our three families. For example, we check which of the methods are Condorcet or majority consistent, and which are resistant to cloning. We also consider independence properties and state some characterization results.

Simulations (Section 5) To understand how and whether the three families of methods practically differ from each other, and how they relate to Kemeny's method, we perform extensive simulations based on synthetic data (sampled using the Mallows and Euclidean models). We find that, for Plurality and Borda, ranking by score and Sequential-Loser usually produce very similar results, whereas Sequential-Winner offers a new perspective (that is typically closer to Kemeny's method). Moreover, Sequential-Loser rules seem to be particularly well suited to identify the best candidates (justifying their usage as single-winner voting rules), while Sequential-Winner rules are best at avoiding low quality candidates.

Computational Complexity (Section 6) The rules in all three of our families are easy to compute in the sense that their description implies a straightforward algorithm for obtaining an output ranking. However, for the sequential rules there is a subtlety: During the execution of the rule, ties can occur. It matters how these are broken, because candidates could end up in significantly different positions. For high-stakes decisions and in democratic contexts, it would be important to know which output rankings are possible.

Thus, we study the computational problem of deciding whether a given candidate can end up in a given position. This and related problems have been studied in the literature under the name of *parallel universe tie-breaking*, including theoretical and experimental studies for some of the rules in our families (Conitzer, Rognlie, and Xia 2009; Brill and Fischer 2012; Mattei, Narodytska, and Walsh 2014; Freeman, Brill, and Conitzer 2015; Wang et al. 2019). We extend the results of that literature and find NP-hardness for all the sequential methods that we study. We show that the problem becomes tractable if the number of candidates is small. In contrast, for several methods we find that the problem remains hard even if the number of input rankings is small. Curiously, for few input rankings, methods based on Plurality, Borda, or Veto each induce a different parameterized complexity class.

Omitted proofs, additional results, and more details and experiments can be found in the full version on arXiv (Boehmer, Bredereck, and Peters 2022). The code of our experiments is available at github.com/n-boehmer/Rank-Aggregation.

2 Preliminaries

For $k \in \mathbb{N}$, write $[k] = \{1, \dots, k\}$.

Let $C = \{c_1, \dots, c_m\}$ be a set of m *candidates*. A *ranking* $\succ$ of C is a linear order (irreflexive, total, transitive) of C. We write $\mathcal{L}(C)$ for the set of all rankings of C.

A *(ranking) profile* $P = (\succ_1, \dots, \succ_n)$ is a list of rankings. We sometimes say that the rankings are *voters*.

For a subset $C' \subseteq C$ of candidates and ranking $\succ \in \mathcal{L}(C)$, we write $\succ|_{C'}$ for the ranking obtained by restricting $\succ$ to the set C'. For a profile P, we write $P|_{C'}$ for the profile obtained by restricting each of its rankings to C'.

A *social preference function*[1] f is a function that assigns to every ranking profile P a non-empty set $f(P) \subseteq \mathcal{L}(C)$ of rankings. Here, $f(P)$ may be a singleton but there can be more than one output ranking in case of ties. For a ranking $\succ$, we say that f *selects* $\succ$ *on* P if $\succ \in f(P)$.

For a ranking $\succ \in \mathcal{L}(C)$ and a candidate $c \in C$, let $\mathrm{pos}(\succ, c) = |\{d \in C : d \succ c\}| + 1$ be the *position* of c in $\succ$. For example, if $\mathrm{pos}(\succ, c) = 1$ then c is the most-preferred candidate in $\succ$. We write $\mathrm{cand}(\succ, r) \in C$ for the candidate ranked in position $r \in [m]$ in $\succ \in \mathcal{L}(C)$.

For a ranking $\succ \in \mathcal{L}(C)$, let $\mathrm{rev}(\succ)$ denote its reverse ranking, so $\mathrm{cand}(\succ, r) = \mathrm{cand}(\mathrm{rev}(\succ), m - r + 1)$ for each $r \in [m]$. For a profile $P = (\succ_1, \dots, \succ_n)$, we write $\mathrm{rev}(P) = (\mathrm{rev}(\succ_1), \dots, \mathrm{rev}(\succ_n))$.

For an integer $m \in \mathbb{N}$, a *scoring vector* $\mathbf{s}^{(m)} = (s_1, \dots, s_m) \in \mathbb{R}^m$ is a list of m numbers. A *scoring system* is a family of scoring vectors $(\mathbf{s}^{(m)})_{m \in \mathbb{N}}$ one for each possible number m of candidates. We assume that we have access to a polynomial-time algorithm that can compute $\mathbf{s}^{(m)}$ given m. For brevity, we sometimes write $\mathbf{s}$ for $(\mathbf{s}^{(m)})_{m \in \mathbb{N}}$. We will focus on three scoring systems:

- *Plurality* with $\mathbf{s}^{(m)} = (1, 0, \dots, 0)$ for each $m \in \mathbb{N}$,
- *Veto* with $\mathbf{s}^{(m)} = (0, \dots, 0, -1)$ for each $m \in \mathbb{N}$,
- *Borda* with $\mathbf{s}^{(m)} = (m, m - 1, \dots, 1)$ for each $m \in \mathbb{N}$.

Given a profile P over m candidates, the $\mathbf{s}$-*score* of candidate c is $\mathrm{score}_{\mathbf{s}}(P, c) = \sum_{i \in [n]} \mathbf{s}^{(m)}_{\mathrm{pos}(\succ_i, c)}$. We say that a candidate is an $\mathbf{s}$-*winner* if it has maximum $\mathbf{s}$-score, and an $\mathbf{s}$-*loser* if it has minimum $\mathbf{s}$-score. For a scoring system $\mathbf{s}$ we denote

[1]This terminology is due to Young and Levenglick (1978). The term *social welfare function* from Arrow (1951) usually refers to resolute functions that may only output a single ranking.

by $\mathbf{s}^*$ the scoring system where we reverse each scoring vector and multiply all its entries by -1, i.e., for each $m \in \mathbb{N}$ and $i \in [m]$, we have $(\mathbf{s}^*)_i^{(m)} = -\mathbf{s}_{m-i+1}^{(m)}$. Note that $(\mathbf{s}^*)^* = \mathbf{s}$ for every $\mathbf{s}$, that Plurality$^* =$ Veto, that Veto$^* =$ Plurality, and that Borda* is the same as Borda, up to a shift.

For two rankings $\succ_1$ and $\succ_2$, their *swap distance* (or *Kendall-tau distance*) $\kappa(\succ_1, \succ_2)$ is the number of pairs of candidates on whose ordering the two rankings disagree, i.e., $\kappa(\succ_1, \succ_2) = |\{(c, d) \in C \times C : c \succ_1 d \text{ and } d \succ_2 c\}|$. Note that the maximum swap distance between two rankings over m candidates is $\binom{m}{2}$. Given a profile P, *Kemeny's rule* selects those rankings which minimize the average swap distance to the rankings in P, i.e., $\arg\min_{\succ \in \mathcal{L}(C)} \sum_{i \in N} \kappa(\succ, \succ_i)$. We refer to the selected rankings as *Kemeny rankings*.

3 Scoring-Based Rank Aggregation

We now formally define the three families of scoring-based social preference functions that we study.

Definition 3.1 ($\mathbf{s}$-Score)**.** Let $\mathbf{s}$ be a scoring system. For the social preference function $\mathbf{s}$-Score on profile P, we have $\succ \in$ $\mathbf{s}$-Score(P) if and only if for all $c, d \in C$ with $\mathrm{score}_{\mathbf{s}}(P, c) > \mathrm{score}_{\mathbf{s}}(P.d)$, we have $c \succ d$.

Definition 3.2 (Sequential-$\mathbf{s}$-Winner; Seq.-$\mathbf{s}$-Winner)**.** Let $\mathbf{s}$ be a scoring system. The social preference function Seq.-$\mathbf{s}$-Winner is defined recursively as follows: For a profile P, we have $\succ \in$ Seq.-$\mathbf{s}$-Winner(P) if and only if

- the top choice $c = \mathrm{cand}(\succ, 1)$ is an $\mathbf{s}$-winner in P,
- if $|C| > 1$, then $\succ|_{C\setminus\{c\}} \in$ Seq.-$\mathbf{s}$-Winner$(P|_{C\setminus\{c\}})$.

Definition 3.3 (Sequential-$\mathbf{s}$-Loser; Seq.-$\mathbf{s}$-Loser)**.** Let $\mathbf{s}$ be a scoring system. The social preference function Seq.-$\mathbf{s}$-Loser is defined recursively as follows: For a profile P, we have $\succ \in$ Seq.-$\mathbf{s}$-Loser(P) if and only if

- the bottom choice $c = \mathrm{cand}(\succ, |C|)$ is an $\mathbf{s}$-loser in P,
- if $|C| > 1$, then $\succ|_{C\setminus\{c\}} \in$ Seq.-$\mathbf{s}$-Loser$(P|_{C\setminus\{c\}})$.

Example 3.4. *Let P be the following ranking profile:*

$$3 \times a \succ b \succ c, \quad 2 \times b \succ c \succ a, \quad 2 \times c \succ b \succ a.$$

Then for the three methods based on Plurality, we have:

- *Plurality-Score$(P) = \{a \succ b \succ c,\ a \succ c \succ b\}$,*
- *Seq.-Plurality-Winner$(P) = \{a \succ b \succ c\}$, and*
- *Seq.-Plurality-Loser$(P) = \{b \succ a \succ c,\ c \succ a \succ b\}$.*

We sometimes view Seq.-$\mathbf{s}$-Winner (or Seq.-$\mathbf{s}$-Loser) rules as round-based voting rules, where in each round an $\mathbf{s}$-winner (or an $\mathbf{s}$-loser) is deleted from the profile and added in the highest (or lowest) position of the ranking that has not yet been filled. If there are multiple $\mathbf{s}$-winners (or $\mathbf{s}$-losers) in one round, each selection gives rise to different output rankings. Seq.-Plurality-Loser is also known as *STV*, Seq.-Veto-Loser as *Coombs*, and Seq.-Borda-Loser as *Baldwin*.

Sequential-Winner and Sequential-Loser rules are formally related: A candidate is an $\mathbf{s}$-winner in a profile P if and only if it is an $\mathbf{s}^*$-loser in the reverse profile $\mathrm{rev}(P)$. Thus, we have:

Lemma 3.5. *Let $\mathbf{s}$ be a scoring system. Then for each ranking profile P and for every ranking $\succ \in \mathcal{L}(C)$, we have:*

$$\succ \in \textit{Sequential-}\mathbf{s}\textit{-Winner}(P)$$
$$\iff \mathrm{rev}(\succ) \in \textit{Sequential-}\mathbf{s}^*\textit{-Loser}(\mathrm{rev}(P)).$$

For example, this lemma establishes a close connection between Seq.-Veto-Winner and Seq.-Plurality-Loser, as a ranking $\succ$ is selected under Seq.-Veto-Winner on profile P if and only if $\mathrm{rev}(\succ)$ is selected under Seq.-Plurality-Loser on profile $\mathrm{rev}(P)$. This equivalence will prove useful in our axiomatic analysis and in our complexity results.

4 Axiomatic Properties

In this section, we will briefly and informally discuss some axiomatic properties and characterizations of the methods in our three families. A more formal treatment appears in the full version. See Table 1 for an overview.

A desirable property of a ranking aggregation rule is that if one candidate is deleted from the profile, then the relative rankings of the other candidates does not change (*independence of irrelevant alternatives*, IIA). Arrow's (1951) impossibility theorem shows that this property cannot be satisfied by unanimous non-dictatorial rules. Young (1988) proves that Kemeny's method satisfies a weaker version that he calls *local IIA*: removing the candidate that appears in the first or last position in the Kemeny ranking does not change the ranking of the other candidates. Splitting this property into its two parts, we can easily see from their definitions that Seq.-$\mathbf{s}$-Winner satisfies independence at the top, and Seq.-$\mathbf{s}$-Loser satisfies independence at the bottom.

Another influential axiom is known as *consistency* or *reinforcement*. A rule f satisfies reinforcement if whenever some ranking $\succ$ is chosen in two profiles, i.e., $\succ \in f(P) \cap f(P')$, then it is also chosen if we combine the profiles into one, and in fact $f(P + P') = f(P) \cap f(P')$. All the methods in this paper satisfy reinforcement. Notably, Young (1988) shows that Kemeny is the only anonymous, neutral, and unanimous rule satisfying reinforcement and local IIA. Focusing on Seq.-$\mathbf{s}$-Loser, Freeman, Brill, and Conitzer (2014) define *reinforcement at the bottom* to mean that if the same candidate c is placed in the last position in the selected ranking in two profiles, then c is also placed in the last position in the selected ranking in the combined profile. They show that independence at the bottom and reinforcement at the bottom characterize Seq.-$\mathbf{s}$-Loser rules (under mild additional assumptions). Using Lemma 3.5, a simple adaptation of their proof shows that Seq.-$\mathbf{s}$-Winner rules can be similarly characterized by independence at the top and reinforcement at the top. ($\mathbf{s}$-Score methods do not satisfy similar independence assumptions; they have been characterized by Levenglick (1977) and Smith (1973).)

Refining their characterization of Seq.-$\mathbf{s}$-Loser rules, Freeman, Brill, and Conitzer (2014) characterize Seq.-Plurality-Loser (aka STV) as the only Seq.-$\mathbf{s}$-Loser rule satisfying independence of clones (Tideman 1987), Seq.-Veto-Loser (aka Coombs) as the only one that, in case a strict majority of voters have the same ranking, copies that ranking as the output ranking, and Seq.-Borda-Loser (aka Baldwin) as the only one always placing a Condorcet winner in the first position. Using Lemma 3.5, we can similarly characterize Seq.-Plurality-Winner as the only method in its class that copies a majority ranking.

		Score			Sequential-Winner			Sequential-Loser		
	Kemeny	Plurality	Veto	Borda	Plurality	Veto	Borda	Plurality	Veto	Borda
Independence at the top	✓	×	×	×	✓	✓	✓	×	×	×
Independence at the bottom	✓	×	×	×	×	×	×	✓	✓	✓
Reinforcement	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reinforcement at the top	×	✓	✓	✓	✓	✓	✓	×	×	×
Reinforcement at the bottom	×	✓	✓	✓	×	×	×	✓	✓	✓
Condorcet winner at top	✓	×	×	×	×	×	×	×	×	✓
Copy majority	✓	×	×	×	✓	×	×	×	✓	×
Independence of clones	×	×	×	×	×	×	×	✓	×	×

Table 1: An overview of the axiomatic properties of our studied rules. See the full version for definitions. These results are either easy to see, or follow directly from results of Freeman, Brill, and Conitzer (2014).

5 Simulations

We analyze our three families of scoring-based ranking rules for Plurality and Borda on synthetically generated profiles.[2]

5.1 Setup

To deal with ties in the computation of our rules, each time we sample a ranking profile over candidates C, we also sample a ranking $\succ_{\text{tie}} \in \mathcal{L}(C)$ uniformly at random and break ties according to $\succ_{\text{tie}}$ for all rules. To quantify the difference between two rankings $\succ_1, \succ_2 \in \mathcal{L}(C)$, we use their normalized swap distance $\kappa(\succ_1, \succ_2)/\binom{m}{2}$, i.e., their swap distance divided by maximum possible swap distance.

(Normalized) Mallows We conduct simulations on profiles generated using the Mallows model (Mallows 1957) (as observed by Boehmer et al. (2021) real-world profiles often seem to be close to some Mallows profile). This model is parameterized by a dispersion parameter $\phi \in [0,1]$ and a central ranking $\succ^* \in \mathcal{L}(C)$. Then, a profile is assembled by sampling rankings i.i.d. so that the probability of sampling a ranking $\succ \in \mathcal{L}(C)$ is proportional to $\phi^{\kappa(\succ,\succ^*)}$. We use the normalization of the Mallows model proposed by Boehmer et al. (2021), which is parameterized by a normalized dispersion parameter $\text{norm-}\phi \in [0,1]$. This parameter is then converted to a dispersion parameter ϕ such that the expected swap distance between a sampled vote and the central vote is $\text{norm-}\phi \cdot (m(m-1)/4)$. Then $\text{norm-}\phi = 0$ results in profiles only containing the central vote, and $\text{norm-}\phi = 1$ leads to profiles where all rankings are sampled with the same probability, so that on average rankings disagree with the central ranking $\succ^*$ on half of the pairwise comparisons. Choosing $\text{norm-}\phi = 0.5$ leads to profiles where rankings on average disagree with $\succ^*$ on a quarter of the pairwise comparisons.

5.2 Comparison of Our Ranking Methods

We analyze the average normalized swap distance between the rankings selected by our three families of scoring-based ranking methods on profiles containing 100 rankings over 10 candidates. For this, we sampled 10 000 profiles for each $\text{norm-}\phi \in \{0, 0.1, \ldots, 0.9, 1\}$ and depict the results in Figure 1(a). Let us first focus on Plurality: We find that the rankings produced by Seq.-Plurality-Loser and Plurality-Score are quite similar, whereas the ranking produced by Seq.-Plurality-Winner is substantially different. This observation is particularly strong for $\text{norm-}\phi \leq 0.3$: In such profiles, all the rankings are similar to each other. Accordingly, many candidates initially have a Plurality score of zero, and thus there are many ties in the execution of Plurality-Score and Seq.-Plurality-Loser (for the latter, ties occur in more than half of the rounds). Thus, the rankings computed by the two rules fundamentally depend on the (shared) random tie-breaking order $\succ_{\text{tie}}$. In contrast, for Seq.-Plurality-Winner, for $\text{norm-}\phi \leq 0.3$, no ties in its execution appear. Thus, Seq.-Plurality-Winner is able to meaningfully distinguish the weaker candidates on these profiles.

Turning to $\text{norm-}\phi \geq 0.3$ (where more candidates have non-zero Plurality score and thus the tie-breaking is no longer as important), Seq.-Plurality-Loser and Plurality-Score are still clearly more similar to each other than to Seq.-Plurality-Winner; this indicates that Seq.-Winner rules add a new perspective to existing scoring-based ranking rules.

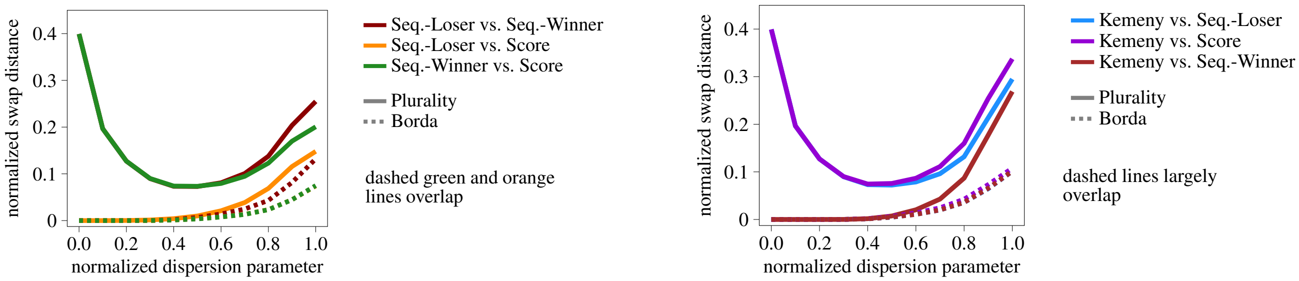
When using Borda scores, the rankings returned by the three methods are quite similar. This is intuitive given that Borda scores capture the general strength of candidates in a profile much better than Plurality scores. Thus, the Borda score of a candidate also changes less drastically in case a candidate is deleted. Increasing $\text{norm-}\phi$, the selected rankings become more different from each other (as profiles get more chaotic, leading to more similar Borda scores of candidates). Interestingly, for large values of $\text{norm-}\phi$, Borda-Score has the same (small) distance to the other two rules, while Seq.-Borda-Winner and -Loser are more different.

5.3 Comparison to Kemeny Ranking

To assess which method produces the "most accurate" rankings, we compare them to Kemeny's method. For 10 000 profiles for each $\text{norm-}\phi \in \{0, 0.1, \ldots, 0.9, 1\}$, in Figure 1(b), we show the average normalized swap distance of the Kemeny ranking to the rankings selected by our rules.

For Plurality, Seq.-Plurality-Winner is closest to the Kemeny ranking for every value of $\text{norm-}\phi$, while Seq.-Plurality-Loser and Plurality-Score have a much larger

[2] In the full version, we describe the results of further experiments. For instance, we analyze in which parts of the computed ranking the considered methods agree or disagree most, the frequency and position of ties, and the influence of the number of alternatives and voters on our results.We repeat our experiments on profiles sampled from Euclidean models and obtain similar results.



(a) Pairs of scoring-based methods

(b) Scoring-based methods vs. Kemeny ranking

Figure 1: Pairwise average normalized swap distance between rankings produced by different methods for Plurality (solid) and Borda (dashed) on Mallows profiles with 10 candidates and 100 voters.

distance to the Kemeny ranking. For norm-$\phi \leq 0.3$, Seq.-Plurality-Loser and Plurality-Score are notably far away from the Kemeny ranking. As before, this is because, for these two rules, large parts of the ranking are simply determined by the random tie-breaking order in such profiles. Seq.-Plurality-Winner is not affected by this, and it is very close to Kemeny for norm-$\phi \leq 0.5$. For norm-$\phi \geq 0.7$, their distance from the Kemeny ranking becomes more similar for our three methods. This is intuitive, recalling that for norm-$\phi = 1$, profiles are "chaotic", with many different rankings having comparable quality.

For Borda, all three methods have a similar small distance to the Kemeny ranking. This distance increases steadily from 0 for norm-$\phi = 0$ to around 0.1 for norm-$\phi = 1$.

6 Complexity

We study various computational problems related to Sequential-Winner and Sequential-Loser rules. By breaking ties arbitrarily, it is easy to compute *some* ranking that is selected by such a rule. However, in some (high-stakes) applications, it might not be sufficient to simply output some selected ranking. For instance, some candidate could claim that there also exist other rankings selected by the same rule where that candidate is ranked higher. To check such claims, and get some understanding of the rankings that can be selected in the presence of ties, we need an algorithm that for a given candidate d and position k, decides whether d is ranked in position k in some ranking selected by the rule. Accordingly, we introduce the following computational problem:

POSITION-k DETERMINATION for social preference function f	
Given:	A ranking profile P over candidate set C, a designated candidate $d \in C$, and an integer $k \in [\lvert C \rvert]$.
Question:	Is there a ranking $\succ$ selected by f on P where d is in position k, i.e., $\succ \in f(P)$ with $\mathrm{pos}(\succ, d) = k$?

Where possible, we will design (parameterized) algorithms that solve this problem (Dorn and Schlotter 2017). We also prove hardness results, which will apply even to restricted versions of this problem that are most relevant in practice. Specifically, we would expect candidates to mainly be interested if they can be ranked highly. Thus, we introduce the TOP-k DETERMINATION problem, where we ask whether a given candidate can be ranked in one of the first k positions.[3] Lastly, the special case of both problems with $k = 1$ is of particular importance: The WINNER DETERMINATION problem asks whether the designated candidate can be ranked in the first position.

For the three Sequential-Loser rules, it is known that their WINNER DETERMINATION problem is NP-complete. For STV, this was stated by Conitzer, Rognlie, and Xia (2009), and for Baldwin and Coombs, this was proven by Mattei, Narodytska, and Walsh (2014). We will see that the corresponding TOP-k DETERMINATION problems for the Sequential-Winner rules are also NP-complete. Thus, since almost all of our problems turn out to be NP-hard, we take a more fine-grained view. In particular, we will study the influence of the number n of voters and the number m of candidates on the complexity of our problems. This analysis is not only of theoretical interest but also practically relevant, as in many applications one of the two parameters is considerably smaller than the other (e.g., in political elections m is typically much smaller than n, while in applications such as meta-search engines or ranking applicants, n is often much smaller than m). Tables 2 and 3 provide overviews of our results.

6.1 Parameter Number of Candidates

We start by considering the parameter m, the number of candidates. It is easy to see that POSITION-k DETERMINATION for all Sequential-Winner and Sequential-Loser rules is fixed-parameter tractable with respect to m (by iterating over all $m!$ possible output rankings). However, it is possible to improve the dependence on the parameter in the running time.

Theorem 6.1. *For every scoring system* s*,* POSITION-k DETERMINATION *can be solved in time*

- $\mathcal{O}(2^m \cdot nm^2)$ *and* $\mathcal{O}(m^k \cdot nm^2)$ *for Sequential-*s*-Winner,*
- $\mathcal{O}(2^m \cdot nm^2)$ *and* $\mathcal{O}(m^{m-k} \cdot nm^2)$ *for Sequential-*s*-Loser.*

Proof (algorithm). We present an algorithm for Seq.-s-Winner (the results for Seq.-s-Loser follow by applying Lemma 3.5). We use dynamic programming. Call a subset $C' \subseteq C$ an *elimination set* if there is a selected

[3] If we have an algorithm for POSITION-k DETERMINATION, we can solve the TOP-k DETERMINATION problem by using the algorithm for positions $i = 1, \ldots, k$. (This is a Turing reduction.)

		n	m
Sequential-Plurality-Loser (STV)	NP-c. (Thm. 6.2)	FPT (Obs. 6.4)	FPT (Thm. 6.1)
Sequential-Veto-Loser (Coombs)	NP-c. (Thm. 6.5)	W[1]-h. (Thm. 6.6), XP (Thm. 6.7)	FPT (Thm. 6.1)
Sequential-Borda-Loser (Baldwin)	NP-c. (Thm. 6.8)	NP-c. for $n = 8$ (Thm. 6.8)	FPT (Thm. 6.1)

Table 2: Our results for Sequential-Loser rules. All hardness results hold for WINNER DETERMINATION; all algorithmic results also apply to POSITION-k DETERMINATION. The unparameterized NP-hardness results in the first column were already stated or proven by Conitzer, Rognlie, and Xia (2009) and Mattei, Narodytska, and Walsh (2014).

ranking where the candidates from C' are ranked in the first $|C'|$ positions. We introduce a table T with entry $T[C']$ for each subset $C' \subseteq C$. $T[C']$ will be true iff C' is an elimination set. We initialize the table by setting $T[\emptyset]$ to true. Now we compute T for each subset $C' \subseteq C$ in increasing order of the size of the subset using the following recurrence: We set $T[C']$ to true if there is a candidate $c \in C'$ such that $T[C' \setminus \{c\}]$ is true and c is an s-winner in $P|_{C \setminus (C' \setminus \{c\})}$.

After filling the table, we return "true" if and only if there is a subset $C' \subseteq C \setminus \{d\}$ with $|C'| = k-1$ such that $T[C']$ is true and d is an s-winner in $P|_{C \setminus C'}$. By filling the complete table we get a running time in $\mathcal{O}(2^m \cdot nm^2)$. However, it is sufficient to only fill the table for all subsets of size at most $k-1$, resulting in a running time in $\mathcal{O}(m^k \cdot nm^2)$. □

6.2 Sequential Loser

We study Seq.-Plurality/Veto/Borda-Loser (aka STV, Coombs, and Baldwin). The WINNER DETERMINATION problem is NP-hard for all three rules. Table 2 shows an overview of our results. In particular, we get a clear separation of the rules for the number n of voters:

- Seq.-Plurality-Loser admits a simple FPT algorithm,
- Seq.-Veto-Loser is W[1]-hard but in XP, and
- Seq.-Borda-Loser is NP-hard for 8 voters.

Plurality Conitzer, Rognlie, and Xia (2009) stated that WINNER DETERMINATION for Seq.-Plurality-Loser (aka STV) is NP-hard. This result has been frequently cited and used. The proof was omitted in the conference paper, and to our knowledge no proof has ever appeared in published work. To aid future research, we include a simple reduction here.

Theorem 6.2. WINNER DETERMINATION *for Sequential-Plurality-Loser (aka STV) is NP-hard.*

Proof. We reduce from the NP-hard variant of 3-SAT where each clause contains at most three literals and each literal appears exactly twice (Berman, Karpinski, and Scott 2003). Let φ be a formula fulfilling these restrictions with clause set $F = \{c_1, \ldots, c_m\}$ and variable set $X = \{x_1, \ldots, x_n\}$. Let $L = X \cup \overline{X}$ be the set of literals. We construct a ranking profile with candidate set $C = \{d, w\} \cup F \cup L$, where d is our designated candidate, and the following voters:

100 voters	$d \succ \ldots$	
99 voters	$w \succ d \succ \ldots$	
98 voters	$c_j \succ w \succ d \succ \ldots$	$\forall j \in [m]$
60 voters	$\ell \succ \overline{\ell} \succ w \succ d \succ \ldots$	$\forall \ell \in L$
2 voters	$\ell \succ c_j \succ w \succ d \succ \ldots$	$\forall j \in [m], \ell \in c_j$

For this ranking profile, in every execution of Seq.-Plurality-Loser the first n eliminated candidates must be a subset $L' \subseteq L$ of literals such that for every variable we select either its positive literal or its negative literal (but not both). In other words, L' must satisfy $\ell \in L' \leftrightarrow \overline{\ell} \notin L'$. To see this, note that all literal candidates initially have a Plurality score of 64, which is the lowest Plurality score in the profile, and that all other candidates have a higher Plurality score. Thus, in the first round an arbitrary literal ℓ of some variable x is eliminated. This increases the Plurality score of the opposite literal $\overline{\ell}$ to 124. In the second round, we have to eliminate again an arbitrary literal (however, this time a literal corresponding to a variable different from x). We repeat this process for n rounds until for each variable exactly one of the corresponding literals has been eliminated. We claim that an execution of Seq.-Plurality-Loser eliminates d last if and only if the assignment that sets all literals from L' to *true* satisfies φ.

Suppose φ is satisfied by some variable assignment α, and consider an execution of Seq.-Plurality-Loser that begins by eliminating the n literals set to true in α. After this, the scores of the remaining candidates are:

(i) d has 100 points and w has 99 points,
(ii) c_j for $j \in [m]$ has between 100 and 104 points (as at least one of the literals in c_j has been eliminated), and
(iii) each literal $\ell \in L$ set to false by α has 124 points.

In the next round, w is eliminated, giving 99 points to d. In the next m rounds, each clause candidate c_j is eliminated, giving its points to d. Then, remaining literals are eliminated, also each giving their points to d. Thus, d is the last remaining candidate and ranked in first position in the selected ranking.

Let $L' \subseteq L$ be the set of literals eliminated in the first n rounds in some execution of Seq.-Plurality-Loser (recall $\ell \in L' \leftrightarrow \overline{\ell} \notin L'$). Suppose that the assignment α setting all literals from L' to true does not satisfy φ. After L' has been eliminated, the scores of the remaining candidates are:

(i) d has 100 points and w has 99 points,
(ii) c_j for $j \in [m]$: if α satisfies c_j, it has between 100 and 104 points, otherwise it has 98 points, and
(iii) each literal $\ell \in L$ set to false by α has 124 points.

In the next round, one of the unsatisfied clauses is eliminated, redistributing its 98 points to w, bringing the score of w to 197. Note that while w remains uneliminated, d cannot gain additional points (because all voters prefer w to d, except the 100 voters who have d ranked top). Thus, d will be eliminated before w, and thus d cannot be eliminated last. □

		n	k	$n+k$	m
Sequential-Plurality-Winner	NP-c.	W[1]-h., XP	W[1]-h., XP	FPT	FPT
Sequential-Veto-Winner	NP-c.	FPT	W[2]-h., XP	FPT	FPT
Sequential-Borda-Winner	NP-c.	NP-h. for $n = 8$	W[1]-h., XP	?	FPT

Table 3: Our results for Sequential-Winner rules. All hardness results hold for the TOP-k DETERMINATION problem; all algorithmic results also apply to the general POSITION-k DETERMINATION problem.

Motivated by this hardness result, we turn to parameterized complexity. By Theorem 6.1, the problem is solvable in $\mathcal{O}(2^m \cdot nm^2)$ time. We show that unless the Exponential Time Hypothesis (ETH)[4] is false, we cannot hope to substantially improve the exponential part of this running time.

Theorem 6.3. *If the ETH is true, then* WINNER DETERMINATION *for Sequential-Plurality-Loser (aka STV) cannot be solved in* $2^{o(m)} \cdot \mathrm{poly}(n, m)$ *time.*

Turning to the number n of voters, observe that initially only at most n candidates have a non-zero Plurality score. All other candidates (which are not ranked first in any ranking) will be eliminated immediately, without thereby changing the Plurality scores of other candidates. After these eliminations, we are left with at most n candidates. This makes it easy to see that POSITION-k DETERMINATION is fixed-parameter tractable with respect to n (by using Theorem 6.1).

Observation 6.4. POSITION-k DETERMINATION *for Sequential-Plurality-Loser (aka STV) is solvable in* $\mathcal{O}(2^n \cdot nm^2)$ *time.*

Veto For Seq.-Veto-Loser (aka Coombs), Mattei, Narodytska, and Walsh (2014) showed that WINNER DETERMINATION is NP-hard. We give another NP-hardness proof that also implies an ETH-based lower bound for parameter m.

Theorem 6.5. WINNER DETERMINATION *for Sequential-Veto-Loser (aka Coombs) is NP-complete. If the ETH is true, then the problem cannot be solved in* $2^{o(m)} \cdot \mathrm{poly}(n, m)$ *time.*

For the parameter n, we show that the problem is W[1]-hard, based on an involved reduction from MULTICOLORED INDEPENDENT SET. This suggests that Seq.-Veto-Loser behaves quite differently from Seq.-Plurality-Loser.

Theorem 6.6. WINNER DETERMINATION *for Sequential-Veto-Loser (aka Coombs) is W[1]-hard with respect to the number n of voters.*

On the positive side, WINNER DETERMINATION and even POSITION-k DETERMINATION are solvable in polynomial time if n is constant. The reason is that for Seq.-Veto-Loser, the "status" of an execution is fully captured by the *bottom list* containing the bottom-ranked candidate of each voter, i.e., for a ranking profile $(\succ_1, \cdots \succ_n)$ over m candidates the bottom list is $(\mathrm{cand}(\succ_1, m), \ldots, \mathrm{cand}(\succ_n, m))$. If we know the current bottom list, we can deduce which candidates have been eliminated thus far (these are exactly the candidates that appear behind the currently bottom candidate of a voter in its original vote). As there are only m^n possibilities for the bottom list, dynamic programming over all possible bottom lists yields an XP algorithm for POSITION-k DETERMINATION.

Theorem 6.7. POSITION-k DETERMINATION *for Sequential-Veto-Loser is in XP with respect to the number n of voters.*

Borda We conclude by studying Seq.-Borda-Loser (aka Baldwin). Mattei, Narodytska, and Walsh (2014) proved that WINNER DETERMINATION for this rule is NP-hard, adapting an earlier reduction about hardness of manipulation due to Davies et al. (2014). In fact, by giving a construction based on weighted majority graphs and using tools from Bachmeier et al. (2019), we prove that this NP-hardness persists even for only $n = 8$ voters. This result suggests that the Borda scoring system leads to the hardest computational problems.

Theorem 6.8. *Let $n \geq 8$ be a fixed even integer. Then* WINNER DETERMINATION *for Sequential-Borda-Loser (aka Baldwin), restricted to instances with exactly n voters, is NP-complete. In addition, if the ETH is true, then the problem cannot be solved in* $2^{o(m)} \cdot \mathrm{poly}(m)$ *time.*

6.3 Sequential Winner

We briefly summarize our results for Seq.-Plurality/Veto/Borda-Winner (see Table 3); for details see the full version. WINNER DETERMINATION is trivial for these rules, so we focus on TOP-k DETERMINATION. For all rules, TOP-k DETERMINATION is NP-hard and W[1]-hard with respect to k. For parameter n, the picture is more diverse: For Borda, it is again NP-hard for $n = 8$, while Plurality and Veto switch their role (we have an FPT algorithm for Veto and W[1]-hardness for Plurality). Due to Lemma 3.5, this switch is unsurprising, but the W[1]-hardness requires a separate (though similar) proof as Theorem 6.6.

7 Future Directions

There are many directions for future work. For example, do the hard problems we have identified become tractable if preferences are structured, for example single-peaked? (For Coombs, see (Grofman and Feld 2004, Prop. 2).) Instead of parallel universes, one could break ties immediately (e.g., by some fixed order), and focus on finding the fastest algorithms for computing the output ranking. (Computing STV in this model is known to be P-complete (Csar et al. 2017).) The Borda-Score rule is known to give a 5-approximation of Kemeny's method (Coppersmith, Fleischer, and Rudra 2006); do any other of the rules from our families provide an approximation? Finally, one could try to extend our results to other scoring vectors, and potentially prove dichotomy theorems.

[4] The ETH states that 3-SAT with n variables cannot be solved in $2^{o(n)} \cdot \mathrm{poly}(n)$ time (Impagliazzo, Paturi, and Zane 2001).

Acknowledgments

NB was supported by the DFG project ComSoc-MPMS (NI 369/22).

References

Ailon, N.; Charikar, M.; and Newman, A. 2008. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM*, 55(5): 1–27.

Arrow, K. J. 1951. *Social Choice and Individual Values*. New Haven: Cowles Foundation, 1st edition.

Bachmeier, G.; Brandt, F.; Geist, C.; Harrenstein, P.; Kardel, K.; Peters, D.; and Seedig, H. G. 2019. k-majority digraphs and the hardness of voting with a constant number of voters. *Journal of Computer and System Sciences*, 105: 130–157.

Bartholdi, J.; Tovey, C. A.; and Trick, M. A. 1989. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2): 157–165.

Berman, P.; Karpinski, M.; and Scott, A. D. 2003. Approximation hardness of short symmetric instances of MAX-3SAT. *Electronic Colloquium on Computational Complexity*.

Boehmer, N.; Bredereck, R.; Faliszewski, P.; Niedermeier, R.; and Szufa, S. 2021. Putting a compass on the map of elections. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, 59–65.

Boehmer, N.; Bredereck, R.; and Peters, D. 2022. Rank aggregation using scoring rules. *arXiv:2209.08856*.

Bossert, W.; and Sprumont, Y. 2014. Strategy-proof preference aggregation: Possibilities and characterizations. *Games and Economic Behavior*, 85: 109–126.

Brill, M.; and Fischer, F. 2012. The price of neutrality for the ranked pairs method. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 1299–1305.

Can, B.; and Storcken, T. 2013. Update monotone preference rules. *Mathematical Social Sciences*, 65(2): 136–149.

Conitzer, V.; Rognlie, M.; and Xia, L. 2009. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 109–115.

Coppersmith, D.; Fleischer, L.; and Rudra, A. 2006. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 776–782.

Csar, T.; Lackner, M.; Pichler, R.; and Sallinger, E. 2017. Winner determination in huge elections with MapReduce. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 451–458.

Davies, J.; Katsirelos, G.; Narodytska, N.; Walsh, T.; and Xia, L. 2014. Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules. *Artificial Intelligence*, 217: 20–42.

Dorn, B.; and Schlotter, I. 2017. Having a hard time? Explore parameterized complexity. In Endriss, U., ed., *Trends in Computational Social Choice*, chapter 11, 209–230. AI Access.

Dwork, C.; Kumar, R.; Naor, M.; and Sivakumar, D. 2001. Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on the World Wide Web (WWW)*, 613–622.

Fitzsimmons, Z.; and Hemaspaandra, E. 2021. Kemeny consensus complexity. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, 196–202.

Freeman, R.; Brill, M.; and Conitzer, V. 2014. On the axiomatic characterization of runoff voting rules. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 675–681.

Freeman, R.; Brill, M.; and Conitzer, V. 2015. General tiebreaking schemes for computational social choice. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 1401–1409.

Grofman, B.; and Feld, S. L. 2004. If you like the alternative vote (a.k.a. the instant runoff), then you ought to know about the Coombs rule. *Electoral Studies*, 23(4): 641–659.

Hemaspaandra, E.; Spakowski, H.; and Vogel, J. 2005. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3): 382–391.

Impagliazzo, R.; Paturi, R.; and Zane, F. 2001. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4): 512–530.

Kemeny, J. G. 1959. Mathematics without numbers. *Daedalus*, 88: 577–591.

Kenyon-Mathieu, C.; and Schudy, W. 2007. How to rank with few errors. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, 95–103.

Levenglick, A. B. 1977. *Characterizations of Social Decision Functions*. Ph.D. thesis, City University of New York.

Mallows, C. L. 1957. Non-null ranking models. I. *Biometrika*, 44(1/2): 114–130.

Mattei, N.; Narodytska, N.; and Walsh, T. 2014. How hard is it to control an election by breaking ties? In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, 1067–1068.

Smith, J. H. 1973. Aggregation of preferences with variable electorate. *Econometrica*, 41(6): 1027–1041.

Tideman, T. N. 1987. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3): 185–206.

Truchon, M. 1998. Figure skating and the theory of social choice. *Cahier 98-16 du Centre de Recherche en Économie et Finance Appliquées, Québec*.

van Zuylen, A.; and Williamson, D. P. 2009. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research*, 34(3): 594–620.

Wang, J.; Sikdar, S.; Shepherd, T.; Zhao, Z.; Jiang, C.; and Xia, L. 2019. Practical algorithms for multi-stage voting rules with parallel universes tiebreaking. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 2189–2196.

Young, H. P. 1988. Condorcet's theory of voting. *The American Political Science Review*, 82(4): 1231–1244.

Young, H. P. 1995. Optimal voting rules. *Journal of Economic Perspectives*, 9(1): 51–64.

Young, H. P.; and Levenglick, A. 1978. A consistent extension of Condorcet's election principle. *SIAM Journal on Applied Mathematics*, 35(2): 285–300.