How Similarity Helps to Efficiently Compute Kemeny Rankings[*]

Nadja Betzler
Institut für Informatik
Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2
D-07743 Jena, Germany
betzler@minet.uni-jena.de

Michael R. Fellows
PC Research Unit
Office of DVC (Research)
University of Newcastle
Callaghan, NSW 2308, Australia
michael.fellows@newcastle.edu.au

Jiong Guo
Institut für Informatik
Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2
D-07743 Jena, Germany
guo@minet.uni-jena.de

Rolf Niedermeier
Institut für Informatik
Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2
D-07743 Jena, Germany
niedermeier@minet.uni-jena.de

Frances A. Rosamond
PC Research Unit
Office of DVC (Research)
University of Newcastle
Callaghan, NSW 2308, Australia
frances.rosamond@newcastle.edu.au

ABSTRACT

The computation of Kemeny rankings is central to many applications in the context of rank aggregation. Unfortunately, the problem is NP-hard. We show that the Kemeny score (and a corresponding Kemeny ranking) of an election can be computed efficiently whenever the *average* pairwise distance between two input votes is not too large. In other words, Kemeny Score is fixed-parameter tractable with respect to the parameter "average pairwise Kendall-Tau distance d_a". We describe a fixed-parameter algorithm with running time $16^{\lceil d_a \rceil} \cdot \text{poly}$. Moreover, we extend our studies to the parameters "maximum range" and "average range" of positions a candidate takes in the input votes. Whereas Kemeny Score remains fixed-parameter tractable with respect to the parameter "maximum range", it becomes NP-complete in case of an average range value of two. This excludes fixed-parameter tractability with respect to the parameter "average range" unless P=NP.

Categories and Subject Descriptors

F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms and Problems*; G.2.1 [**Mathematics of Computing**]: Discrete Mathematics—*Combinatorics*; I.2.8 [**Computing Methodologies**]: Artifical Intelligence—*Problem Solving, Control Methods, and Search*; J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms

Algorithms

Keywords

Rank aggregation, NP-hard problem, exact algorithm, fixed-parameter tractability, structural parameterization

1. INTRODUCTION

Aggregating inconsistent information has many applications ranging from voting scenarios to meta search engines and fighting spam [1, 8, 11, 14]. In some sense, one deals with *consensus problems* where one wants to find a solution to various "input demands" such that these demands are met as well as possible. Naturally, contradicting demands cannot be fulfilled at the same time. Hence, the consensus solution has to provide a balance between opposing requirements. The concept of *Kemeny consensus* (or Kemeny ranking) is among the most important conflict resolution proposals in this context. In this paper, extending and improving previous results [3], we study new algorithmic approaches based on parameterized complexity analysis [13, 17, 21] for efficiently computing optimal Kemeny consensus solutions in practically relevant special cases. To this end, we employ the "similarity" between votes by measuring their *average* pairwise distance.

Kemeny's voting scheme can be described as follows. An *election* (V, C) consists of a set V of n votes and a set C of m candidates. A *vote* is a preference list of the candidates, that is, a permutation on C. For instance, in the case of three candidates a, b, c, the order $c > b > a$ would mean that candidate c is the best-liked and candidate a is the least-liked for this voter. A "Kemeny consensus" is a preference list that is "closest" to the preference lists of the voters. For

[*] Most of the results of this paper have been presented at COMSOC'08 under the title "Computing Kemeny Rankings, Parameterized by the Average KT-Distance".

Cite as: How Similarity Helps to Efficiently Compute Kemeny Rankings, Nadja Betzler, Michael R. Fellows, Jiong Guo, Rolf Niedermeier, Frances A. Rosamond, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 657–664

each pair of votes v, w, the so-called *Kendall-Tau distance* (*KT-distance* for short) between v and w, also known as the inversion distance between two permutations, is defined as

$$\text{KT-dist}(v, w) = \sum_{\{c,d\} \subseteq C} d_{v,w}(c, d),$$

where the sum is taken over all unordered pairs $\{c, d\}$ of candidates, and $d_{v,w}(c, d)$ is 0 if v and w rank c and d in the same order, and 1 otherwise. Using divide-and-conquer, the KT-distance can be computed in $O(m \cdot \log m)$ time [20]. The *score* of a preference list l with respect to an election (V, C) is defined as $\sum_{v \in V} \text{KT-dist}(l, v)$. A preference list l with the minimum score is called a *Kemeny consensus* of (V, C) and its score $\sum_{v \in V} \text{KT-dist}(l, v)$ is the *Kemeny score* of (V, C), denoted as K-score(V, C). The underlying decision problem is as follows:

> Kemeny Score
> *Input:* An election (V, C) and an integer $k > 0$.
> *Question:* Is K-score$(V, C) \leq k$?

Known results. Bartholdi et al. [2] showed that Kemeny Score is NP-complete, and it remains so even when restricted to instances with only four votes [14, 15]. Given the computational hardness of Kemeny Score on the one side and its practical relevance on the other side, polynomial-time approximation algorithms have been studied. The Kemeny score can be approximated to a factor of 8/5 by a deterministic algorithm [23] and to a factor of 11/7 by a randomized algorithm [1]. Recently, a polynomial-time approximation scheme (PTAS) has been developed [19]. However, its running time is completely impractical. Conitzer, Davenport, and Kalagnanam [11, 8] performed computational studies for the efficient exact computation of a Kemeny consensus, using heuristic approaches such as greedy and branch-and-bound. Their experimental results encourage the search for practically relevant, efficiently solvable special cases. These experimental investigations focus on computing strong admissible bounds for speeding up search-based heuristic algorithms. In contrast, our focus is on exact algorithms with provable asymptotic running time bounds for the developed algorithms. Hemaspaandra et al. [18] provided further, exact classifications of the classical computational complexity of Kemeny elections. More specifically, whereas Kemeny Score is NP-complete, they provided $\mathbf{P}_{\|}^{\mathbf{NP}}$-completeness results for other, more general versions of the problem. Very recently, a parameterized complexity study based on various problem parameterizations has been initiated [3]. There, fixed-parameter tractability results for the parameters "Kemeny score", "number of candidates" and "maximum KT-distance between two input votes" are reported.

Finally, it is interesting to note that Conitzer [7] uses a (different) notion of similarity (which is, furthermore, imposed on candidates rather than voters) to efficiently compute the closely related Slater rankings. Using the concept of similar candidates, he identifies efficiently solvable special cases, also yielding a powerful preprocessing technique for computing Slater rankings.

New results. Our main result is that Kemeny Score can be solved in $16^{\lceil d_a \rceil} \cdot \text{poly}(n, m)$ time, where d_a denotes the average KT-distance between the pairs of input votes. This means a significant improvement over the previous algorithm for the maximum KT-distance $d_{\max}$ between pairs of input votes, which has running time $(3d_{\max} + 1)! \cdot \text{poly}(n, m)$ [3]. Clearly, $d_a \leq d_{\max}$. In addition, using similar ideas, we can show that Kemeny Score can be solved in $32^{r_{\max}} \cdot \text{poly}(n, m)$ time, where $r_{\max}$ denotes the maximum range of candidate positions of an election (see Section 2 for a formal definition). In contrast, these two fixed-parameter tractability results are complemented by an NP-completeness result for the case of an average range of candidate positions of only two, thus destroying hopes for fixed-parameter tractability with respect to this parameterization.

2. PRELIMINARIES

Let the *position* of a candidate c in a vote v, denoted by $v(c)$, be the number of candidates that are better than c in v. That is, the leftmost (and best) candidate in v has position 0 and the rightmost has position $m - 1$. For an election (V, C) and a candidate $c \in C$, the *average position* $p_a(c)$ of c is defined as

$$p_a(c) := \frac{1}{n} \cdot \sum_{v \in V} v(c).$$

For an election (V, C), the average KT-distance d_a is defined as[1]

$$d_a := \frac{1}{n(n-1)} \cdot \sum_{u,v \in V, u \neq v} \text{KT-dist}(u, v).$$

Note that an equivalent definition is given by

$$d_a := \frac{1}{n(n-1)} \cdot \sum_{a,b \in C} \#v(a > b) \cdot \#v(b > a),$$

where for two candidates a and b the number of votes in which a is ranked better than b is denoted by $\#v(a > b)$. The latter definition is useful if the input is provided by the outcomes of the pairwise elections of the candidates including the margins of victory. Furthermore, we define

$$d := \lceil d_a \rceil.$$

Further, for an election (V, C) and for a candidate $c \in C$, the *range* $r(c)$ of c is defined as

$$r(c) := \max_{v,w \in V} \{|v(c) - w(c)|\} + 1.$$

The maximum range $r_{\max}$ of an election is given by $r_{\max} := \max_{c \in C} r(c)$ and the average range r_a is defined as

$$r_a := \frac{1}{m} \sum_{c \in C} r(c).$$

Finally, we briefly introduce the relevant notions of parameterized complexity theory [13, 17, 21]. Parameterized algorithmics aims at a multivariate complexity analysis of problems. This is done by studying relevant problem parameters and their influence on the computational complexity of problems. The hope lies in accepting the seemingly inevitable combinatorial explosion for NP-hard problems, but confining it to the parameter. Thus, the decisive question is whether a given parameterized problem is *fixed-parameter*

[1] To simplify the presentation, the following definition counts the pair (u, v) *as well as* the pair (v, u), thus having to divide by $n(n - 1)$ to obtain the correct average distance value.

$$\begin{array}{|lcl|}\hline v_1 & : & a > b > c > d > e > f > \dots \\ & & \vdots \\ v_i & : & a > b > c > d > e > f > \dots \\ v_{i+1} & : & b > a > d > c > f > e > \dots \\ & & \vdots \\ v_{2i} & : & b > a > d > c > f > e > \dots \\ \hline \end{array}$$

Figure 1: Small maximum range but large average KT-distance.

tractable (FPT) with respect to the parameter. In other words, for an input instance I together with the parameter k, we ask for the existence of a solving algorithm with running time $f(k) \cdot \text{poly}(|I|)$ for some computable function f.

3. ON PARAMETERIZATIONS OF KEMENY SCORE

This section discusses the "art" of finding different, practically relevant parameterizations of Kemeny Score. Our paper focusses on structural parameterizations, that is, structural properties of input instances that may be exploited to develop efficient solving algorithms for Kemeny Score. To this end, here we investigate the realistic scenario (which, to some extent, is also motivated by previous experimental results [11, 8]) that the given preference lists of the voters show some form of similarity. More specifically, we consider the parameters "average KT-distance" between the input votes, "maximum range of candidate positions", and "average range of candidate positions". Clearly, the maximum value is always an upper bound for the average value. The parameter "average KT-distance" reflects the situation that in an ideal world all votes would be the same, and differences occur to some (limited) form of noise which makes the actual votes different from each other (see [12, 10, 9]). With average KT-distance as parameter we can affirmatively answer the question whether a consensus list that is closest to the input votes can efficiently be found. By way of contrast, the parameterization by position range rather reflects the situation that whereas voters can be more or less decided concerning groups of candidates (e.g., political parties), they may be quite undecided and, thus, unpredictable, concerning the ranking within these groups. If these groups are small this can also imply small range values, thus making the quest for a fixed-parameter algorithm in terms of range parameterization attractive.

It is not hard to see, however, that the parameterizations by "average KT-distance" and by "range of position" can significantly differ. As described in the following, there are input instances of Kemeny Score that have a small range value and a large average KT-distance, and vice versa. This justifies separate investigations for both parameterizations; these are performed in Sections 4 and 5, respectively. We end this section with some concrete examples that exhibit the announced differences between our notions of vote similarity, that is, our parameters under investigation. First, we provide an example where one can observe a small maximum candidate range whereas one has large average KT-distance, see Figure 1. The election in Figure 1 consists of $n = 2i$ votes such that there are two groups of i identical votes. The votes of the second group are obtained from the first group by swapping neighboring pairs of candidates. Clearly, the maximum range of candidates is 2. However, for m candidates the average KT-distance d_a is

$$d_a = \frac{2 \cdot (n/2)^2 \cdot (m/2)}{n(n-1)} > m/4$$

and, thus, d_a is unbounded for an unbounded number of candidates.

$$\begin{array}{|lcllllllllllllll|}\hline v_1 & : & a & > & b & > & c & > & d & > & e & > & f & > & \dots \\ v_2 & : & b & > & c & > & d & > & e & > & f & > & \dots & > & a \\ v_1' & : & a & > & b & > & c & > & d & > & e & > & f & > & \dots \\ & & \vdots & & & & & & & & & & & & \\ \hline \end{array}$$

Figure 2: Small average KT-distance but large maximum range.

Second, we present an example where the average KT-distance is small but the maximum range of candidates is large, see Figure 2. In the election of Figure 2 all votes are equal except that candidate a is at the last position in the second vote, but on the first position in all other votes. Thus, the maximum range equals the range of candidate a which equals the number of candidates, whereas by adding more copies of the first vote the average KT-distance can be made smaller than one.

Finally, we have a somewhat more complicated example displaying a case where one observes small average KT-distance but large average range of candidates.[2] To this end, we make use of the following construction based on an election with m candidates. Let V_m be a set of m votes such that every candidate is in one of the votes at the first and in one of the votes at the last position; the remaining positions can be filled arbitrarily. Then, for some $N > m^3$, add N further votes V_N in which all candidates have the same arbitrary order. Then, the average KT-distance of the constructed election is

$$d_a = D(V_m) + D(V_N) + D(V_N, V_m),$$

where $D(V_m)$ $(D(V_N))$ is the average KT-distance within the votes of V_m (V_N) and $D(V_N, V_m)$ is the average KT-distance between pairs of votes with one vote from V_N and the other vote from V_m. Since m^2 is an upper bound for the pairwise (and average) KT-distance between any two votes, it holds that $D(V_m) \leq m^2$, $D(V_N) = 1$, and $D(V_N, V_m) \leq m^2$. Further, we have $m \cdot (m-1)$ ordered pairs of votes within V_m, $N \cdot m$ pairs between V_N and V_m, and $N \cdot (N-1)$ pairs within V_N. Since $N > m^3$ it follows that

$$d_a \leq \frac{m(m-1) \cdot m^2 + Nm \cdot m^2 + N(N-1) \cdot 1}{N(N-1)} \leq 3.$$

In contrast, the range of every candidate is m, thus the average range is m.

4. PARAMETER AVERAGE KT-DISTANCE

[2] Clearly, this example also exhibits the situation of a large maximum candidate range with a small average KT-distance. We chose nevertheless to present the example from Figure 2 because of its simplicity.

In this section, we further extend the range of parameterizations studied so far (see [3]) by giving a fixed-parameter algorithm with respect to the parameter "average KT-distance". We start with showing how the average KT-distance can be used to upper-bound the range of positions that a candidate can take in any optimal Kemeny consensus. Based on this crucial observation, we then state the algorithm.

4.1 A Crucial Observation

Our fixed-parameter tractability result with respect to the average KT-distance of the votes is based on the following lemma.

Lemma 1. *Let d_a be the average KT-distance of an election (V, C) and $d = \lceil d_a \rceil$. Then, in every optimal Kemeny consensus l, for every candidate $c \in C$ with respect to its average position $p_a(c)$ we have $p_a(c) - d < l(c) < p_a(c) + d$.*

Proof. The proof is by contradiction and consists of two claims: First, we show that we can find a vote with Kemeny score less than $d \cdot n$, that is, the Kemeny score of the instance is less than $d \cdot n$. Second, we show that in every Kemeny consensus every candidate is in the claimed range. More specifically, we prove that every consensus in which the position of a candidate is not in a "range d of its average position" has a Kemeny score greater than $d \cdot n$, a contradiction to the first claim.

Claim 1: K-score$(V, C) < d \cdot n$.

Proof of Claim 1: To prove Claim 1, we show that there is a vote $v \in V$ with $\sum_{w \in V}$ KT-dist$(v, w) < d \cdot n$, implying this upper bound for an optimal Kemeny consensus as well. By definition,

$$d_a = \frac{1}{n(n-1)} \cdot \sum_{v,w \in V, v \neq w} \text{KT-dist}(v, w) \qquad (1)$$

$$\Rightarrow \exists v \in V \text{ with } d_a \geq \frac{1}{n(n-1)} \cdot n \cdot \sum_{w \in V, v \neq w} \text{KT-dist}(v, w) \qquad (2)$$

$$= \frac{1}{n-1} \cdot \sum_{w \in V, v \neq w} \text{KT-dist}(v, w) \qquad (3)$$

$$\Rightarrow \exists v \in V \text{ with } d_a \cdot n > \sum_{w \in V, v \neq w} \text{KT-dist}(v, w). \qquad (4)$$

Since we have $d = \lceil d_a \rceil$, Claim 1 follows directly from Inequality (4).

The next claim shows the given bound on the range of possible candidates positions.

Claim 2: In every optimal Kemeny consensus l, every candidate $c \in C$ fulfills $p_a(c) - d < l(c) < p_a(c) + d$.

Proof of Claim 2: We start by showing that, for every candidate $c \in C$, we have

$$\text{K-score}(V, C) \geq \sum_{v \in V} |l(c) - v(c)|. \qquad (5)$$

Note that, for every candidate $c \in C$, for two votes v, w we must have KT-dist$(v, w) \geq |v(c) - w(c)|$. Without loss of generality, assume that $v(c) > w(c)$. Then, there must be at least $v(c) - w(c)$ candidates that have a smaller position than c in v and that have a greater position than c in w. Further, each of these candidates increases the value of KT-dist(v, w) by one. Based on this, Inequality (5) directly follows as, by definition, K-score$(V, C) = \sum_{v \in V}$ KT-dist(v, l).

To simplify the proof of Claim 2, in the following, we shift the positions in l such that $l(c) = 0$. Accordingly, we shift the positions in all votes in V, that is, for every $v \in V$ and every $a \in C$, we decrease $v(a)$ by the original value of $l(c)$. Clearly, shifting all positions does not affect the relative differences of positions between two candidates. Then, let the set of votes in which c has a nonnegative position be V^+ and let V^- denote the remaining set of votes, that is, $V^- := V \backslash V^+$.

Now, we show that if candidate c is placed outside of the given range in an optimal Kemeny consensus l, then K-score$(V, C) > d \cdot n$. The proof is by contradiction. We distinguish two cases:

Case 1: $l(c) \geq p_a(c) + d$.
As $l(c) = 0$, in this case $p_a(c)$ becomes negative. Then,

$$0 \geq p_a(c) + d \Leftrightarrow -p_a(c) \geq d.$$

It follows that $|p_a(c)| \geq d$. The following shows that Claim 2 holds for this case.

$$\sum_{v \in V} |l(c) - v(c)| = \sum_{v \in V} |v(c)| \qquad (6)$$

$$= \sum_{v \in V^+} |v(c)| + \sum_{v \in V^-} |v(c)|. \qquad (7)$$

Next, replace the term $\sum_{v \in V^-} |v(c)|$ in (7) by an equivalent term that depends on $|p_a(c)|$ and $\sum_{v \in V^+} |v(c)|$. For this, use the following, derived from the definition of $p_a(c)$:

$$n \cdot p_a(c) = \sum_{v \in V^+} |v(c)| - \sum_{v \in V^-} |v(c)|$$
$$\Leftrightarrow \sum_{v \in V^-} |v(c)| = n \cdot (-p_a(c)) + \sum_{v \in V^+} |v(c)|$$
$$= n \cdot |p_a(c)| + \sum_{v \in V^+} |v(c)|.$$

The replacement results in

$$\sum_{v \in V} |l(c) - v(c)| = 2 \cdot \sum_{v \in V^+} |v(c)| + n \cdot |p_a(c)|$$
$$\geq n \cdot |p_a(c)| \geq n \cdot d.$$

This says that K-score$(V, C) \geq n \cdot d$, a contradiction to Claim 1.

Case 2: $l(c) \leq p_a(c) - d$.
Since $l(c) = 0$, the condition is equivalent to $0 \leq p_a(c) - d \Leftrightarrow d \leq p_a(c)$, and we have that $p_a(c)$ is nonnegative. Now, we show that Claim 2 also holds for this case.

$$\sum_{v \in V} |l(c) - v(c)| = \sum_{v \in V} |v(c)| = \sum_{v \in V^+} |v(c)| + \sum_{v \in V^-} |v(c)|$$
$$\geq \sum_{v \in V^+} v(c) + \sum_{v \in V^-} v(c) = p_a(c) \cdot n \geq d \cdot n.$$

Thus, also in this case, K-score$(V, C) \geq n \cdot d$, a contradiction to Claim 1. □

Based on Lemma 1, for every position we can define the set of candidates that can take this position in an optimal Kemeny consensus. The subsequent definition will be useful for the formulation of the algorithm.

Definition 1. *Let (V, C) be an election. For every integer $i \in \{0, \dots, m-1\}$, let P_i denote the set of candidates that can assume the position i in an optimal Kemeny consensus, that is, $P_i := \{c \in C \mid p_a(c) - d < i < p_a(c) + d\}$.*

Using Lemma 1, we can easily show the following.

Lemma 2. *For every position i, $|P_i| \leq 4d$.*

Proof. The proof is by contradiction. Assume that there is a position i with $|P_i| > 4d$. Due to Lemma 1, for every candidate $c \in P_i$ the positions which c may assume in an optimal Kemeny consensus can differ by at most $2d-1$. This is true because, otherwise, candidate c could not be in the given range around its average position. Then, in a Kemeny consensus, each of the at least $4d+1$ candidates must hold a position that differs at most by $2d-1$ from position i. As there are only $4d-1$ such positions ($2d-1$ on the left and $2d-1$ on the right of i), one obtains a contradiction. □

4.2 Basic Idea of the Algorithm

In Subsection 4.4, we will present a dynamic programming algorithm for Kemeny Score. It exploits the fact that every candidate can only appear in a fixed range of positions in an optimal Kemeny consensus.[3] The algorithm "generates" a Kemeny consensus from the left to the right. It tries out all possibilities for ordering the candidates locally and then combines these local solutions to yield an optimal Kemeny consensus.

More specifically, according to Lemma 2, the number of candidates that can take a position i in an optimal Kemeny consensus for any $0 \leq i \leq m-1$ is at most $4d$. Thus, for position i, we can test all possible candidates. Having chosen a candidate for position i, the remaining candidates that could also assume i must either be left or right of i in a Kemeny consensus. Thus, we test all possible two-partitionings of this subset of candidates and compute a "partial" Kemeny score for every possibility. For the computation of the partial Kemeny scores at position i we make use of the partial solutions computed for the position $i-1$.

4.3 Definitions for the Algorithm

To state the dynamic programming algorithm, we need some further definitions. For $i \in \{0, \dots, m-1\}$, let $I(i)$ denote the set of candidates that could be "inserted" at position i *for the first time*, that is,

$$I(i) := \{c \in C \mid c \in P_i \text{ and } c \notin P_{i-1}\}.$$

Let $F(i)$ denote the set of candidates that must be "forgotten" at latest at position i, that is,

$$F(i) := \{c \in C \mid c \notin P_i \text{ and } c \in P_{i-1}\}.$$

For our algorithm, it is essential to subdivide the overall Kemeny score into *partial Kemeny scores* (pK). More precisely, for a candidate c and a subset R of candidates with $c \notin R$, we set

$$\mathrm{pK}(c, R) := \sum_{c' \in R} \sum_{v \in V} d_v^R(c, c'),$$

where for $c \notin R$ and $c' \in R$ we have $d_v^R(c, c') := 0$ if in v we have $c > c'$, and $d_v^R(c, c') := 1$, otherwise. Intuitively, the partial Kemeny score denotes the score that is "induced" by candidate c and the candidate subset R if the candidates of R have greater positions than c in an optimal Kemeny consensus.[4] Then, for a Kemeny consensus $l := c_0 > c_1 > \dots > c_{m-1}$, the overall Kemeny score can be expressed by partial Kemeny scores as follows.

$$\text{K-score}(V, C) = \sum_{i=0}^{m-2} \sum_{j=i+1}^{m-1} \sum_{v \in V} d_{v,l}(c_i, c_j) \tag{8}$$

$$= \sum_{i=0}^{m-2} \sum_{c' \in R} \sum_{v \in V} d_v^R(c_i, c') \text{ for } R := \{c_j \mid i < j < m\} \tag{9}$$

$$= \sum_{i=0}^{m-2} \mathrm{pK}(c_i, \{c_j \mid i < j < m\}). \tag{10}$$

Next, consider the corresponding three-dimensional dynamic programming table T. Roughly speaking, define an entry for every position i, every candidate c that can assume i, and every candidate subset $C' \subseteq P_i \backslash \{c\}$. The entry stores the "minimum partial Kemeny score" over all possible orders of the candidates of C' under the condition that c takes position i and all candidates of C' take positions smaller than i. To define the dynamic programming table formally, we need some further notation.

Let $\Pi(C')$ denote the set of all possible orders of the candidates in C', where $C' \subseteq C$. Further, consider a Kemeny consensus in which every candidate of C' has a position smaller than every candidate in $C \backslash C'$. Then, the *minimum partial Kemeny score restricted to* C' is defined as

$$\min_{(d_1 > d_2 > \dots > d_x) \in \Pi(C')} \left\{ \sum_{s=1}^{x} \mathrm{pK}(d_s, \{d_j \mid s < j < m\} \cup (C \backslash C')) \right\}$$

with $x := |C'|$. That is, it denotes the minimum partial Kemeny score over all orders of C'. We define an entry of the dynamic programming table T for a position i, a candidate $c \in P_i$, and a candidate subset $P'_i \subseteq P_i$ with $c \notin P'_i$. For this, we define $L := \bigcup_{j \leq i} F(j) \cup P'_i$. Then, an entry $T(i, c, P'_i)$ denotes the minimum partial Kemeny score restricted to the candidates in $L \cup \{c\}$ under the assumptions that c is at position i in a Kemeny consensus, all candidates of L have positions smaller than i, and all other candidates have positions greater than i. That is, for $|L| = i-1$, define

$$T(i, c, P'_i) := \min_{(d_1 > \dots > d_{i-1}) \in \Pi(L)} \sum_{s=0}^{i-1} \mathrm{pK}(d_s, C \backslash \{d_j \mid j \leq s\}) + \mathrm{pK}(c, C \backslash (L \cup \{c\})).$$

4.4 Dynamic Programming Algorithm

[3] In contrast, the previous dynamic programming algorithms [3] for the parameters "maximum range of candidate positions" and "maximum KT-distance" rely on decomposing the input whereas here we rather have a decomposition of the score into partial scores. Further, here we obtain a much better running time by using a more involved dynamic programming approach.

[4] By convention and somewhat counterintuitively, we say that a candidate c has a greater position than a candidate c' in a vote if $c' > c$.

Input: An election (V, C) and, for every $0 \leq i < m$, the set P_i of candidates that can assume position i in an optimal Kemeny consensus.
Output: The Kemeny score of (V, C).

```
Initialization:
01 for  $i = 0, \dots, m-1$
02   for all  $c \in P_i$
03     for all $P'_i \subseteq P_i \setminus \{c\}$
04       $T(i, c, P'_i) := +\infty$
05 for all $c \in P_0$
06   $T(0, c, \emptyset) := \mathrm{pK}(c, C \setminus \{c\})$

Update:
07 for  $i = 1, \dots, m-1$
08   for all $c \in P_i$
09     for all $P'_i \subseteq P_i \setminus \{c\}$
10       if  $|P'_i \cup \bigcup_{j \leq i} F(j)| = i - 1$
         and $T(i-1, c', (P'_i \cup F(i)) \setminus \{c'\})$ is defined then
11   $T(i, c, P'_i) = \min_{c' \in P'_i \cup F(i)} T(i-1, c', (P'_i \cup F(i)) \setminus \{c'\})$
         $+ \mathrm{pK}(c, (P_i \cup \bigcup_{i<j<m} I(j)) \setminus (P'_i \cup \{c\}))$

Output:
12 K-score $= \min_{c \in P_{m-1}} T(m-1, c, P_{m-1} \setminus \{c\})$
```

Figure 3: Dynamic programming algorithm for Kemeny Score

The algorithm is displayed in Figure 3. It is easy to modify the algorithm such that it outputs an optimal Kemeny consensus: for every entry $T(i, c, P'_i)$, one additionally has to store a candidate c' that minimizes $T(i-1, c', (P'_i \cup F(i)) \setminus \{c'\})$ in line *11*. Then, starting with a minimum entry for position $m-1$, one reconstructs an optimal Kemeny consensus by iteratively adding the "predecessor" candidate. The asymptotic running time remains unchanged. Moreover, in several applications, it is useful to compute not just *one* optimal Kemeny consensus but to enumerate all of them. At the expense of an increased running time, which clearly depends on the number of possible optimal consensus rankings, our algorithm can be extended to provide such an enumeration by storing all possible predecessor candidates.

Lemma 3. *The algorithm in Figure 3 correctly computes* Kemeny Score.

Proof. For the correctness, we have to show two points: First, all table entries are well-defined, that is, for an entry $T(i, c, P'_i)$ concerning position i there must be exactly $i-1$ candidates that have positions smaller than i. This condition is assured by line *10* of the algorithm.[5]

Second, we must ensure that our algorithm finds an optimal solution. Due to Equality (10), we know that the Kemeny score can be decomposed into partial Kemeny scores. Thus, it remains to show that the algorithm considers a decomposition that leads to an optimal solution. For every position, the algorithm tries all candidates in P_i. According to Lemma 1, one of these candidates must be the "correct" candidate c for this position. Further, for c we can observe that the algorithm tries a sufficient number of possibilities to partition all remaining candidates $C \setminus \{c\}$ such that they have either smaller or greater positions than i. More precisely, every candidate from $C \setminus \{c\}$ must be in exactly one of the following three subsets:

1. The set F of candidates that have already been forgotten, that is, $F := \bigcup_{0 \leq j \leq i} F(j)$.
2. The set of candidates that can assume position i, that is, $P_i \setminus \{c\}$.
3. The set I of candidates that are not inserted yet, that is, $I := \bigcup_{i<j<m} I(j)$.

Due to Lemma 1 and the definition of $F(j)$, we know that a candidate from F cannot take a position greater than $i-1$ in an optimal Kemeny consensus. Thus, it is sufficient to explore only those partitions in which the candidates from F have positions smaller than i. Analogously, one can argue that for all candidates in I, it is sufficient to consider partitions in which they have positions greater than i. Thus, it remains to try all possibilities for partitioning the candidates from P_i. This is done in line *09* of the algorithm. Thus, the algorithm returns an optimal Kemeny score. □

Theorem 1. Kemeny Score *can be solved in $O(16^d \cdot (d^2 \cdot m + d \cdot m^2 \log m \cdot n) + n^2 \cdot m \log m)$ time with average KT-distance d_a and $d = \lceil d_a \rceil$. The size of the dynamic programming table is $O(16^d \cdot d \cdot m)$.*

Proof. The dynamic programming procedure requires the set of candidates P_i for $0 \leq i < m$ as input. To determine P_i for all $0 \leq i < m$, one needs the average positions of all candidates and the average KT-distance d_a of (V, C). To determine d_a, compute the pairwise distances of all pairs of votes. As there are $O(n^2)$ pairs and the pairwise KT-distance can be computed in $O(m \log m)$ time [20], this takes $O(n^2 \cdot m \log m)$ time. The average positions of all candidates can be computed in $O(n \cdot m)$ time by iterating once over every vote and adding the position of every candidate to a counter variable for this candidate. Thus, the input for the dynamic programming algorithm can be computed in $O(n^2 \cdot m \log m)$ time.

Concerning the dynamic programming algorithm itself, due to Lemma 2, for $0 \leq i < m$, the size of P_i is upper-bounded by $4d$. Then, for the initialization as well as for the update, the algorithm iterates over m positions, $4d$ candidates, and 2^{4d} subsets of candidates. Whereas the initialization in the innermost instruction (line *04*) can be done in constant time, in every innermost instruction of the update phase (line *11*) one has to look for a minimum entry and one has to compute a pK-score. To find the minimum, one has to consider all candidates from $P'_i \cup F(i)$. As $P'_i \cup F(i)$ is a subset of P_{i-1}, it can contain at most $4d$ candidates. Further, the required pK-score can be computed in $O(n \cdot m \log m)$ time. Thus, for the dynamic programming we arrive at the running time of $O(m \cdot 4d \cdot 2^{4d} \cdot (4d + n \cdot m \log m)) = O(16^d \cdot (d^2 \cdot m + d \cdot m^2 \log m \cdot n))$.

Concerning the size of the dynamic programming table, there are m positions and any position can be assumed by

[5] It can still happen that a candidate takes a position outside of the required range around its average position. Since such an entry cannot lead to an optimal solution according to Lemma 1, this does not affect the correctness of the algorithm. To improve the running time it would be convenient to "cut away" such possibilities. We leave considerations in this direction to future work.

at most $4d$ candidates. The number of considered subsets is bounded from above by 2^{4d}. Hence, the size of the table T is $O(16^d \cdot d \cdot m)$. □

Finally, let us discuss the differences between the dynamic programming algorithm used for the "maximum pairwise KT-distance" in [3] and the algorithm presented in this work. In [3], the dynamic programming table stored all possible orders of the candidates of a given subset of candidates. In this work, we eliminate the need to store all orders by using the decomposition of the Kemeny score into partial Kemeny scores. This allows us to restrict the considerations for a position to a candidate and its order relative to all other candidates.

5. SMALL CANDIDATE RANGE

In this section, we consider two further parameterizations, namely "maximum range" and "average range" of candidates. As exhibited in Section 3, the range parameters in general are "orthogonal" to the distance parameterizations dealt with in Section 4. Whereas for the parameter "maximum range" we can obtain fixed-parameter tractability by using the dynamic programming algorithm given in Figure 3, the Kemeny Score problem becomes NP-complete already in case of an average range of two.

5.1 Parameter Maximum Range

In the following, we show how to bound the number of candidates that can assume a position in an optimal Kemeny consensus by a function of the maximum range. This enables the application of the algorithm from Figure 3.

Lemma 4. *Let $r_{\max}$ be the maximum range of an election (V, C). Then, for every candidate its relative order in an optimal consensus with respect to all but at most $3r_{\max}$ candidates can be computed in $O(n \cdot m^2)$ time.*

Proof. We use an observation that follows directly from the Extended Condorcet criterion [22]: If for two candidates $b, c \in C$ we have $v(b) > v(c)$ for all $v \in V$, then in every Kemeny consensus l it holds that $l(b) > l(c)$. Thus, it follows that for $b, c \in C$ with $\max_{v \in V} v(b) < \min_{v \in V} v(c)$, in an optimal Kemeny consensus l we have $l(b) < l(c)$. That is, for two candidates with "non-overlapping range" their relative order in an optimal Kemeny consensus can be determined using this observation. Clearly, all these candidate pairs can be computed in $O(n \cdot m^2)$ time.

Next, we show that for every candidate c there are at most $3r_{\max}$ candidates whose range overlaps with the range of c. The proof is by contradiction. Let the range of c go from position i to j, with $i < j$. Further, assume that there is a subset of candidates $S \subseteq C$ with $|S| \geq 3r_{\max} + 1$ such that for every candidate $s \in S$ there is a vote $v \in V$ with $i \leq v(s) \leq j$. Now, consider an arbitrary input vote $v \in V$. Since there are at most $3r_{\max}$ positions p with $i - r_{\max} \leq p \leq j + r_{\max}$ for one candidate $s \in S$ it must hold that $v(s) < i - r_{\max}$ or $v(s) > j + r_{\max}$. Thus, the range of s is greater than $r_{\max}$, a contradiction. Hence, there can be at most $3r_{\max}$ candidates that have a position in the range of c in a vote $v \in V$. As described above, for all other candidates we can compute the relative order in $O(n \cdot m^2)$ time. Hence, the lemma follows. □

As a direct consequence of Lemma 4, we conclude that every candidate can assume one of at most $3r_{\max}$ consecutive positions in an optimal Kemeny consensus. Recall that for a position i the set of candidates that can assume i in an optimal consensus is denoted by P_i (see Definition 1). Then, using the same argument as in Lemma 2, one obtains the following.

Lemma 5. *For every position i, $|P_i| \leq 6r_{\max}$.*

In complete analogy to Theorem 1, one arrives at the following.

Theorem 2. Kemeny Score *can be solved in $O(32^{r_{\max}} \cdot (r_{\max}^2 \cdot m + r_{\max} \cdot m^2 \log m \cdot n) + n^2 \cdot m \log m)$ time with maximum range $r_{\max}$. The size of the dynamic programming table is $O(32^{r_{\max}} \cdot r_{\max} \cdot m)$.*

5.2 Parameter Average Range

Theorem 3. Kemeny Score *is NP-complete for elections with average range two.*

Proof. The proof uses a reduction from an arbitrary instance $((V, C), k)$ of Kemeny Score to a Kemeny Score-instance $((V', C'), k)$ with average range less than two. The construction of the election (V', C') is given in the following. To this end, let $a_i, 1 \leq i \leq |C|^2$, be new candidates not occurring in C.

- $C' := C \uplus \{a_i \mid 1 \leq i \leq |C|^2\}$.
- For every vote $v = c_1 > c_2 > \cdots > c_m$ in V, put the vote $v' := c_1 > c_2 > \cdots > c_m > a_1 > a_2 > \cdots > a_{m^2}$ into V'.

It follows from the extended Condorcet criterion [22] that if a pair of candidates has the same order in all votes, it must have this order in a Kemeny consensus as well. Thus, in a Kemeny consensus it holds that $a_i > a_j$ for $i > j$ and, therefore, adding the candidates from $C' \backslash C$ does not increase the Kemeny score. Hence, an optimal Kemeny consensus of size k for (V', C') can be transformed into an optimal Kemeny consensus of size k for (V, C) by deleting the candidates of $C' \backslash C$. The average range of (V', C') is bounded as follows:

$$\begin{aligned} r_a &= \frac{1}{m + m^2} \cdot \sum_{c \in C'} r(c) \\ &= \frac{1}{m + m^2} \cdot \left(\sum_{c \in C} r(c) + \sum_{c \in C' \backslash C} r(c) \right) \\ &\leq \frac{1}{m + m^2} \cdot (m^2 + m^2) < 2. \end{aligned}$$

Clearly, the reduction can be easily modified to work for every constant value of at least two by choosing a C' of appropriate size. □

6. CONCLUSION

Compared to earlier work [3], we significantly improved the running time for the natural parameterization "maximum KT-distance" for the Kemeny Score problem. There have been some experimental studies [11, 8] that hinted that the Kemeny problem is easier when the votes are close to a consensus and, thus, tend to have a small average distance. Our results for the average distance parameterization can also be regarded as a theoretical explanation with provable guarantees for this behavior. Moreover, we provided fixed-parameter tractability in terms of the parameter "maximum range of positions", whereas this is excluded for the parameter "average range of positions" unless P=NP. These results are of particular interest because we indicated in Section 3 that the parameters "position range" and "pairwise distance" are independent of each other.

As challenges for future work, we envisage the following:

- Extend our findings to the Kemeny Score problem with input votes that may have ties or that may be incomplete (also see [3]).
- Improve the running time as well as the memory consumption (which is exponential in the parameter)—we believe that still significant improvements are possible.
- Implement the algorithms, perhaps including heuristic improvements of the running times, and perform experimental studies.
- Investigate typical values of the average KT-distance and the maximum candidate range, either under some distributional assumption or for real-world data.

Finally, we want to advocate parameterized algorithmics [13, 17, 21] as a very helpful tool for better understanding and exploiting the numerous natural parameters occuring in voting szenarios with associated NP-hard combinatorial problems. Only few investigations in this direction have been performed so far, see, for instance [4, 5, 6, 16].

7. ACKNOWLEDGEMENTS

We are grateful to an anonymous referee of *COMSOC 2008* for constructive feedback. This work was supported by the DFG, research project DARE, GU 1023/1, Emmy Noether research group PIAF, NI 369/4, and project PALG, NI 369/8 (Nadja Betzler and Jiong Guo). Michael R. Fellows and Frances A. Rosamond were supported by the Australian Research Council. This work was done while Michael Fellows stayed in Jena as a recipient of the Humboldt Research Award of the Alexander von Humboldt foundation, Bonn, Germany.

8. ADDITIONAL AUTHORS

9. REFERENCES

[1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM*, 55(5), 2008. Article 23 (October 2008).

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

[3] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. Fixed-parameter algorithms for Kemeny scores. In *Proc. of 4th AAIM*, volume 5034 of *LNCS*, pages 60–71. Springer, 2008.

[4] N. Betzler, J. Guo, and R. Niedermeier. Parameterized computational complexity of Dodgson and Young elections. In *Proc. of 11th SWAT*, volume 5124 of *LNCS*, pages 402–413. Springer, 2008.

[5] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. In *Proc. of 2nd COCOA '08*, volume 5165 of *LNCS*, pages 43–53. Springer, 2008.

[6] R. Christian, M. R. Fellows, F. A. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.

[7] V. Conitzer. Computing Slater rankings using similarities among candidates. In *Proc. 21st AAAI*, pages 613–619. AAAI Press, 2006.

[8] V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing Kemeny rankings. In *Proc. 21st AAAI*, pages 620–626. AAAI Press, 2006.

[9] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximun likelihood estimation. In *Proc. of 2nd COMSOC*, pages 181–192, 2008.

[10] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *Proc. of 21st UAI*, pages 145–152. AUAI Press, 2005.

[11] A. Davenport and J. Kalagnanam. A computational study of the Kemeny rule for preference aggregation. In *Proc. 19th AAAI*, pages 697–702. AAAI Press, 2004.

[12] M. J. A. N. de Caritat (Marquis de Condorcet). Essai sur l'application de l'analyse à la probabilité des décisions redues à la pluralité des voix. Paris: L'Imprimerie Royal, 1785.

[13] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[14] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proc. of 10th WWW*, pages 613–622, 2001.

[15] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited, 2001. Manuscript.

[16] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland voting broadly resist bribery and control. In *Proc. of 22nd AAAI*, pages 724–730. AAAI Press, 2007.

[17] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[18] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349:382–391, 2005.

[19] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proc. 39th STOC*, pages 95–103. ACM, 2007.

[20] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison Wesley, 2006.

[21] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[22] M. Truchon. An extension of the Condorcet criterion and Kemeny orders. Technical report, cahier 98-15 du Centre de Recherche en Économie et Finance Appliquées, Université Laval, Québec, Candada, 1998.

[23] A. van Zuylen and D. P. Williamson. Deterministic algorithms for rank aggregation and other ranking and clustering problems. In *Proc. 5th WAOA*, volume 4927 of *LNCS*, pages 260–273. Springer, 2007.