Partial Kernelization for Rank Aggregation: Theory and Experiments

Nadja Betzler*, Robert Bredereck*, and Rolf Niedermeier

Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Germany
{nadja.betzler, robert.bredereck, rolf.niedermeier}@uni-jena.de

Abstract. RANK AGGREGATION is important in many areas ranging from web search over databases to bioinformatics. The underlying decision problem KEMENY SCORE is NP-complete even in case of four input rankings to be aggregated into a "median ranking". We study efficient polynomial-time data reduction rules that allow us to find *optimal* median rankings. On the theoretical side, we improve a result for a "partial problem kernel" from quadratic to linear size. On the practical side, we provide encouraging experimental results with data based on web search and sport competitions, e.g., computing optimal median rankings for real-world instances with more than 100 candidates within milliseconds.

1 Introduction

We investigate the effectiveness of data reduction for computing *optimal* solutions of the NP-hard RANK AGGREGATION problem. Kemeny's corresponding 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 or a *ranking* is a total order of all candidates. For instance, in case of three candidates a, b, c, the order $c > b > a$ means that candidate c is the best-liked one and candidate a is the least-liked one. For each pair of votes v, w, the *Kendall-Tau distance* (also known as the number of inversions) between v and w is defined as

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

where $d_{v,w}(c, d)$ is set to 0 if v and w rank c and d in the same order, and is set to 1, otherwise. The *score* of a ranking l with respect to an election (V, C) is defined as $\sum_{v \in V} \text{KT-dist}(l, v)$. A ranking l with a minimum score is called a *Kemeny ranking* of (V, C) and its score is the *Kemeny score* of (V, C). The central problem considered in this work is as follows:

RANK AGGREGATION: Given an election (V, C), find a Kemeny ranking of (V, C).

Its decision variant KEMENY SCORE asks whether there is a Kemeny ranking of (V, C) with score at most some additionally given positive integer k. The RANK AGGREGATION problem has numerous applications, ranging from building meta-search engines for the web or spam detection [7] over databases [8] to the construction of genetic maps

* Supported by the DFG, research project PAWS, NI 369/10.

Published in the Proceedings of the 5th International Symposium on Parameterized and Exact Computation (IPEC)*, Chennai, India, December 2010, Volume 6478 in Lecture Notes in Computer Science, pp. 26–37.* © *Springer*

in bioinformatics [9]. Kemeny rankings are also desirable in classical voting scenarios such as the determination of a president (see, for example, `www.votefair.org`) or the selection of the best qualified candidates for job openings. The wide range of applications is due to the fulfillment of many desirable properties from the social choice point of view [18], including the *Condorcet property*: if there is a candidate (*Condorcet winner*) who is better than every other candidate in more than half of the votes, then this candidate is also ranked first in every Kemeny ranking.

Previous work. First computational complexity studies of Kemeny Score go back to Bartholdi et al. [2], showing its NP-hardness. Dwork et al. [7] showed that the problem remains NP-hard even in the case of four votes. Moreover, they identified its usefulness in aggregating web search results and provided several approximation and heuristic algorithms. Recent papers showed constant-factor approximability [1,17] and an (impractical) PTAS [11]. Schalekamp and van Zuylen [15] provided a thorough experimental study of approximation and heuristic algorithms. Due to the importance of computing optimal solutions, there have been some experimental studies in this direction [5,6]: An integer linear program and a branch-and-bound approach were applied to random instances generated under a noise model (motivated by the interpretation of Kemeny rankings as maximum likelihood estimators [5]). From a parameterized complexity perspective, the following is known. First fixed-parameter tractability results have been shown with respect to the single parameters number of candidates, Kemeny score, maximum range of candidate positions, and average KT-distance d_a [3]. The *average KT-distance*

$$d_a := \sum_{v,w \in V, v \neq w} \text{KT-dist}(v,w)\big)/(n(n-1))$$

will also play a central role in this work. Moreover, Kemeny Score remains NP-hard when the average range of candidate positions is two [3], excluding hope for fixed-parameter tractability with respect to this parameterization. Simjour [16] further introduced the parameter “Kemeny score divided by the number of votes” (also showing fixed-parameter tractability) and improved the running times for the fixed-parameter algorithms corresponding to the parameterizations by average KT-distance and Kemeny score. Recently, Karpinski and Schudy [10] devised subexponential-time fixed-parameter algorithms for the parameters Kemeny score, d_a, and Kemeny score divided by the number of votes. Mahajan et al. [12] studied above guarantee parameterization with respect to the Kemeny score. Introducing the new concept of partial kernelization, it has been shown that with respect to the average KT-distance d_a one can compute in polynomial time an equivalent instance where the number of candidates is at most $162d_a^2+9d_a$ [4]. This equivalent instance is called *partial kernel*[1] with respect to the parameter d_a because it only bounds the number of candidates but not the number of votes instead of bounding the total instance size (as one has in classical problem kernels).

Our contributions. On the theoretical side, we improve the previous partial kernel [4] from $162d_a^2 + 9d_a$ candidates to $11d_a$ candidates. Herein, the central point is to exploit “stronger majorities”, going from “$>_{2/3}$-majorities” as used before [4] to “$\geq_{3/4}$-majorities”. In this line, we also prove that the consideration of “$\geq_{3/4}$-majorities” is

[1] A formal definition of partial kernels appears in [4].

optimal in the sense that "$\geq_s$-majorities" with $s < 3/4$ do not suffice. Moreover, our "linear partial kernel" result with respect to d_a is based on results for related parameterizations based on "dirty pairs according to $\geq_s$-majorities" which might be considered as parameterizations of independent interest.

On the practical side, we provide strong empirical evidence for the usefulness of data reduction rules associated with the above mentioned kernelization. An essential property of our data reduction rules is that they can break instances into several subinstances to be handled independently, that is, the relative order between the candidates in two different subinstances in a Kemeny ranking is already determined. This also means that for hard instances which we could not completely solve we were still able to compute "partial rankings" of the top and bottom ranked candidates. Finally, we employ some of the known fixed-parameter algorithms and integer linear programming to solve sufficiently small parts of the instances remaining after data reduction. Our experiments showed that the data reduction rules allow for drastically decreased running times and are crucial to solve larger instances. For example, using data reduction rules decreased the running time for a test instance from winter sports with 33 candidates from 103 seconds to few milliseconds. Furthermore, several instances with more than 100 candidates, which could not be solved without data reduction, were solved in a few seconds.

2 Majority-based data reduction rules

We start with some definitions and sketch some relevant previous results [4]. Then we show how to extend the previous results to obtain a linear partial kernel for the average KT-distance by providing a new data reduction rule. We also show the "limitations" of our new reduction rule. Finally, we provide two more reduction rules of practical relevance.

Definitions and previous results. The data reduction framework from previous work [4] introduces a "dirtiness concept" and shows that one can delete some "non-dirty candidates" by a data reduction rule leading to a partial kernel with respect to the average KT-distance. The "dirtiness" of a pair of candidates is measured by the amount of agreement of the votes for this pair. To this end, we introduce the following notation. For an election (V, C), two candidates $c, c' \in C$, and a rational number $s \in]0.5, 1]$, we write

$$c \geq_s c'$$

if at least $\lceil s \cdot |V| \rceil$ of the votes prefer c to c'. A candidate pair $\{c, c'\}$ is *dirty according to the $\geq_s$-majority* if neither $c \geq_s c'$ nor $c' \geq_s c$. All remaining pairs are *non-dirty according to the $\geq_s$-majority*. This directly leads to the parameter n_d^s denoting the number of dirty pairs according to the $\geq_s$-majority. Previous work only considered $>_{2/3}$-majorities[2] and provided a reduction rule such that the number of candidates in a reduced instance is at most quadratic in $\min\{n_d^{>2/3}, d_a\}$. In this work, we provide a

[2] To simplify matters, we write "$>_{2/3}$" instead of "$\geq_s$ with $s > 2/3$", and if the value of s is clear from the context, then we speak of "dirty pairs" and omit "according to the $\geq_s$-majority".

Table 1. Partial kernelization and polynomial-time solvability. The term dirty refers to the $\geq_s$-majority for the respective values of s. The number of dirty pairs is n_d^s. A linear partial kernel w.r.t. the average KT-distance follows directly from the linear partial kernel w.r.t. n_d^s (Theorem 1).

value of s	**partial kernel result**	**special case: no dirty pairs**
$2/3 < s < 3/4$	quadratic partial kernel w.r.t. n_d^s [4, Theorem 2]	polynomial-time solvable [4, Proposition 3]
$3/4 \leq s \leq 1$	linear partial kernel w.r.t. n_d^s (Theorem 1)	

linear partial kernel with respect to n_d^s according to the $\geq_s$-majority for $s \geq 3/4$ and show that this leads to a linear partial kernel with respect to d_a.

We say that c and c' are *ordered according to the $\geq_s$-majority* in a preference list l if $c \geq_s c'$ and $c > c'$ in l. If all candidate pairs are non-dirty with respect to the $\geq_s$-majority for an $s > 2/3$, then there exists a *$\geq_s$-majority order*, that is, a preference list in which all candidate pairs are ordered according to the $\geq_s$-majority [4]. Furthermore, the corresponding $>_{2/3}$-majority order can be found in polynomial time and is a Kemeny ranking [4]. Candidates appearing only in non-dirty pairs according to a $\geq_s$-majority are called *non-dirty candidates* and all remaining candidates are *dirty candidates* according to the $\geq_s$-majority. Note that with this definition a non-dirty pair can also be formed by two dirty candidates. See Table 1 for an overview of partial kernelization and polynomial-time solvability results.

We end with some notation needed to state our data reduction rules. For a candidate subset $C' \subseteq C$, a ranking fulfills the condition $C' > C \setminus C'$ if every candidate from C' is preferred to every candidate from $C \setminus C'$. A subinstance of (V, C) *induced* by a candidate subset $C' \subseteq C$ is given by (V', C') where every vote in V' one-to-one corresponds to a vote in V keeping the relative order of the candidates from C'.

2.1 New results exploiting $\geq_{3/4}$-majorities

We improve the partial kernel upper bound [4] for the parameter d_a from quadratic to linear, presenting a new data reduction rule. The crucial idea for the new reduction rule is to consider $\geq_{3/4}$-majorities instead of $>_{2/3}$-majorities. We further show that the new reduction rule is tight in the sense that it does not work for $>_{2/3}$-majorities.

Reduction Rule. The following lemma allows us to formulate a data reduction rule that deletes all non-dirty candidates and additionally may break the remaining set of dirty candidates into several subsets to be handled independently from each other.

Lemma 1. *Let $a \in C$ be a non-dirty candidate with respect to the $\geq_{3/4}$-majority and $b \in C \setminus \{a\}$. If $a \geq_{3/4} b$, then in every Kemeny ranking one must have "$a > b$"; if $b \geq_{3/4} a$, then in every Kemeny ranking one must have "$b > a$".*

Proof. Let the *partial score* of a candidate subset C' be the Kemeny score of the subinstance induced by C'. We consider the case $a \geq_{3/4} b$; the case $b \geq_{3/4} a$ follows in complete analogy. The proof is by contradiction. Assume that there is a Kemeny ranking l with "$\ldots > b > D > a > \ldots$" for some $D \subseteq C \setminus \{a, b\}$. Since a is non-dirty, for every candidate $d \in D$, it must either hold that $a \geq_{3/4} d$ or $d \geq_{3/4} a$. Let $D_1 := \{d \in D \mid a \geq_{3/4} d\}$ and $D_2 := \{d \in D \mid d \geq_{3/4} a\}$. Consider the ranking l' obtained from l through replacing

$$b > D > a \quad \text{by} \quad D_2 > a > b > D_1,$$

where the positions of all other candidates remain unchanged and the candidates within D_1 and D_2 have the same relative order as within D.

We show that the score of l is greater than the score of l' contradicting that l is a Kemeny ranking. The only pairs of candidates that have different orders in l and l' and thus can contribute with different partial scores to the scores of l and l' are $\{a, d\}$ and $\{d_2, d\}$ for all $d \in D_1 \cup \{b\}$ and all $d_2 \in D_2$. Consider any $d \in D_1 \cup \{b\}$ and $d_2 \in D_2$. Since $|\{v \in V \mid d_2 \geq_{3/4} a\}| \geq 3/4 \cdot |V|$ and $|\{v \in V \mid a \geq_{3/4} d\}| \geq 3/4 \cdot |V|$, the intersection of these two sets must contain at least $|V|/2$ elements, that is, there must be at least $|V|/2$ votes with "$\ldots > d_2 > \ldots > a > \ldots > d > \ldots$". Thus, the partial score of $\{d_2, d\}$ in l is at least as high as its partial score in l'. The partial score of every pair $\{a, d\}$ with $d \in D_1 \cup \{b\}$ in l' is strictly less than the partial score in l. Since $|D_1 \cup \{b\}| \geq 1$, the score of l' is smaller than the score of l and thus l cannot be a Kemeny ranking, a contradiction. □

As a direct consequence of Lemma 1 we can partition the candidates of an election (V, C) as follows. Let $N := \{n_1, \ldots, n_t\}$ denote the set of non-dirty candidates with respect to the $\geq_{3/4}$-majority such that $n_i \geq_{3/4} n_{i+1}$ for $1 \leq i \leq t-1$. Then, $D_0 := \{d \in C \setminus N \mid d \geq_{3/4} n_1\}$, $D_i := \{d \in C \setminus N \mid n_i \geq_{3/4} d \text{ and } d \geq_{3/4} n_{i+1}\}$ for $1 \leq i \leq t-1$, and $D_t := \{d \in C \setminus N \mid n_t \geq_{3/4} d\}$.

3/4-Majority Rule. *Let (V, C) be an election and N and $D_0, \ldots, D_t$ be the sets of non-dirty and dirty candidates as specified above. Replace the original instance by the $t+1$ subinstances induced by D_i for $i \in \{0, \ldots, t\}$.*

The soundness of the 3/4-Majority Rule follows directly from Lemma 1 and it is straightforward to verify its running time $O(nm^2)$. An instance reduced by the 3/4-Majority Rule contains only dirty candidates with respect to the original instance. Making use of a simple relation between the number of dirty candidates and the average KT-distance as also used previously [4], one can state the following.

Theorem 1. *Let d_a be the average KT-distance and n_d^s be the number of dirty pairs according to the $\geq_s$-majority for an $s \geq 3/4$. Then,* KEMENY SCORE *admits a partial kernel with less than* $\min\{11d_a, 2n_d^s\}$ *candidates which can be computed in* $O(nm^2)$ *time.*

Proof. It is easy to adapt the 3/4-Majority Rule to work for the decision problem: An instance is reduced by deleting all candidates from N, reordering every vote such that $D_0 > D_1 > \cdots > D_t$ where within D_i, $0 \leq i \leq t$, the order of the candidates remains unchanged, and decreasing the Kemeny score appropriately. It is not hard to verify that the 3/4-Majority Rule can be executed in $O(m^2 n)$ time by using a table storing the outcomes for all pairs of candidates. After applying the adapted reduction rule, only dirty candidates remain. Let their number be i. Since every dirty candidate

Table 2. Properties "induced" by $\geq_s$-majorities for different values of s.

value of s	properties
$1/2 \leq s \leq 2/3$	a $\geq_s$-majority order does not necessarily exist (Example 1)
$2/3 < s < 3/4$	a $\geq_s$-majority order exists (follows from [4, Theorem 4]) **but** a non-dirty candidate and a dirty candidate do not have to be ordered according to the $\geq_s$-majority in a Kemeny ranking (Theorem 2)
$3/4 \leq s \leq 1$	a $\geq_s$-majority order exists (follows from [4, Theorem 4]) **and** in every Kemeny ranking every non-dirty candidate is ordered according to the $\geq_s$-majority with respect to all other candidates (Lemma 1)

must be involved in at least one candidate pair that is not ordered according to the $\geq_{3/4}$-majority, there must be at least $i/2$ candidate pairs that contribute with at least $|V|/4 \cdot 3|V|/4$ to the average KT-distance of the original input instance. By definition of the average KT-distance, it follows that

$$d_a \geq \frac{1}{|V|(|V|-1)} \cdot \frac{i}{2} \cdot \frac{|V|}{4} \cdot \frac{3|V|}{4} > \frac{3}{32} \cdot i \Rightarrow 11 \cdot d_a > i.$$

□

Tightness results. We investigate to which $\geq_s$-majorities the results obtained for $\geq_{3/4}$-majorities extend. An overview of properties for a Kemeny ranking for different values of s is provided in Table 2.

For the $>_{2/3}$-majority, instances without dirty candidates are polynomial-time solvable [4]. More precisely, the $>_{2/3}$-majority order is a Kemeny ranking. A simple example shows that, for any $s \leq 2/3$, a $\geq_s$-majority order does not always exist (**Example 1**): Consider the election consisting of the three candidates a, b, and c and the three votes "$a > b > c$", "$b > c > a$", and "$c > a > b$". Here, $a \geq_{2/3} b$, $b \geq_{2/3} c$, and $c \geq_{2/3} a$. No linear order fulfills all three relations.

The existence of a data reduction rule analogously to the $3/4$-Majority Rule for $\geq_s$-majorities for $s < 3/4$ would be desirable since such a rule might be more effective: There are instances for which a candidate is dirty according to the $\geq_{3/4}$-majority but non-dirty according to a $\geq_s$-majority with $s < 3/4$. Hence, for many instances, the number of dirty pairs according to the $\geq_{3/4}$-majority assumes higher values than it does according to smaller values of s. In the following, we discuss why an analogous s-Majority Rule with $s < 3/4$ cannot exist. The decisive point of the $3/4$-Majority Rule is that, in a Kemeny ranking, every non-dirty candidate must be ordered according to the $\geq_{3/4}$-majority with respect to *every* other candidate. The following theorem shows that this is not true for $\geq_s$-majorities with $s < 3/4$. We omit the corresponding intricate construction.

Theorem 2. *Consider a $\geq_s$-majority for any rational $s \in]2/3, 3/4[$. For a non-dirty candidate x and a dirty candidate y, $x \geq_s y$ does not imply $x > y$ in a Kemeny ranking.*

2.2 Exploiting the Condorcet property

We present a well-known data reduction rule of practical relevance and show that it reduces an instance at least as much as the $3/4$-Majority Rule. The reduction rule is based on the following easy-to-verify observation.

Observation 1 *Let $C' \subseteq C$ be a candidate subset with $c' \geq_{1/2} c$ for every $c' \in C'$ and every $c \in C \setminus C'$. Then there must be a Kemeny ranking fulfilling $C' > C \setminus C'$.*

To turn Observation 1 into a reduction rule, we need a polynomial-time algorithm to identify appropriate "winning subsets" of candidates. We use the following simple strategy, called *winning subset routine*: For every candidate c, compute a minimal winning subset M_c by iteratively adding every candidate c' with $c' >_{1/2} c''$, $c'' \in M_c$, to M_c. After this, we choose a smallest winning subset.

Condorcet-Set Rule. *If the winning subset routine returns a subset C' with $C' \neq C$, then replace the original instance by the two subinstances induced by C' and $C \setminus C'$.*

It is easy to see that the Condorcet-Set Rule can be carried out in $O(nm^3)$ time. The following proposition shows that the Condorcet-Set Rule is at least as powerful as the $3/4$-Majority Rule, implying that the Condorcet-Set Rule provides a partial kernel with less than $11d_a$ candidates.

Proposition 1. *An instance where the Condorcet-Set Rule has been exhaustively applied cannot be further reduced by the $3/4$-Majority Rule.*

Proposition 1 shows that the $3/4$-Majority Rule cannot lead to a "stronger" reduction of an instance than the Condorcet-Set Rule does. However, since the Condorcet-Set Rule has a higher running time, that is $O(nm^3)$ compared to $O(nm^2)$, applying the $3/4$-Majority Rule before the Condorcet-Set Rule may lead to an improved running time in practice. This is also true for the consideration of the following "special case" of the Condorcet-Set Rule also running in $O(nm^2)$ time.

Condorcet Rule. *If there is a candidate $c \in C$ with $c \geq_{1/2} c'$ for every $c' \in C \setminus \{c\}$, then delete c.*

Our experiments will show that combining the Condorcet-Set Rule with the other rules significantly speeds up the practical running times for many instances. On the contrary, it is easy to see that other than the Condorcet-Set Rule the Condorcet Rule alone does not yield a partial kernel.

3 Experimental results

To solve sufficiently small remaining parts of the instances left after the application of our data reduction rules, we implemented three exact algorithms. First, an extended version of the search tree algorithm showing fixed-parameter tractability with respect to the Kemeny score [3]. Second, a dynamic programming algorithm running in $O(2^m \cdot nm^2)$ time for m candidates and n votes [3,14]. Third, the integer linear program [5, Linear Program 3] which was the fastest exact algorithm in previous experimental studies [5,15]. We use the freely available ILP-solver GLPK[3] to solve the ILP.

[3] http://www.gnu.org/software/glpk/

Table 3. We describe the results for different combinations of the data reduction rules for the three instances corresponding to the search terms "blues", "gardening", and "classical guitar". The first column encodes the combination of reduction rules used: the first digit is "1" if the Condorcet-Set Rule is applied, the second if the Condorcet Rule is applied and the last digit is "1" if the $3/4$-Majority Rule is applied. We give the running times of data reduction in seconds and the profiles describing the result of the data reduction process. The profiles are to read as follows. Every "1" stands for a position for which a candidate was determined in a Kemeny ranking and higher numbers for groups of candidates whose "internal" order could not be determined by the data reduction rules. Sequences of i ones are abbreviated by 1^i. For example, for the search term "gardening" and the combination 111, we know the order of the best 54 candidates, we know the set of candidates that must assume positions 55-64 without knowledge of their relative orders, and so on.

	blues		gardening		classical guitar	
	time	profile	time	profile	time	profile
001	0.03	$1^2 > 5 > 1 > 101 > 1 > 2$	0.01	$1 > 2 > 1 > 102$	0.03	$1 > 114$
010	0.10	$1^{74} > 9 > 1^{29}$	0.05	$1^{54} > 43 > 1^9$	0.06	$1^6 > 92 > 1^{17}$
011	0.10	$1^{74} > 9 > 1^{29}$	0.05	$1^{54} > 43 > 1^9$	0.07	$1^6 > 92 > 1^{17}$
100	0.84	$1^{74} > 9 > 1^{29}$	0.95	$1^{54} > 20 > 1^3 > 9 > 1^{10} > 4 > 1^6$	1.89	$1^6 > 7 > 1^{50} > 35 > 1^{17}$
101	0.10	$1^{74} > 9 > 1^{29}$	1.03	$1^{54} > 20 > 1^3 > 9 > 1^{10} > 4 > 1^6$	2.03	$1^6 > 7 > 1^{50} > 35 > 1^{17}$
110	0.10	$1^{74} > 9 > 1^{29}$	0.10	$1^{54} > 20 > 1^3 > 9 > 1^{10} > 4 > 1^6$	0.19	$1^6 > 7 > 1^{50} > 35 > 1^{17}$
111	0.10	$1^{74} > 9 > 1^{29}$	0.11	$1^{54} > 20 > 1^3 > 9 > 1^{10} > 4 > 1^6$	0.18	$1^6 > 7 > 1^{50} > 35 > 1^{17}$

Our algorithms are implemented in C++ using several libraries of the boost package. Our implementation consists of about 4000 lines of code. All experiments were carried out on a PC with 3 GHz and 4 GB RAM (CPU: Intel Core2Quad Q9550) running under Ubuntu 9.10 (64 bit) Linux. Source code and test data are available under the GPL Version 3 license under http://theinf1.informatik.uni-jena.de/kconsens/.

In the following, first we describe our results for two different types of web search data, and second we investigate instances obtained from sport competitions.

3.1 Search result rankings

A prominent application of RANK AGGREGATION is the aggregation of search result rankings obtained from different web search engines. We queried the same 37 search terms as Dwork et al. [7] and Schalekamp and van Zuylen [15] to generate rankings. We used the search engines Google, Lycos, MSN Live Search, and Yahoo! to generate rankings of 1000 candidates. We consider two search results as identical if their URL is identical up to some canonical form (cutting after the top-level domain). Results not appearing in all rankings are ignored. Ignoring the term "zen budism" with only 18 candidates, this results in 36 instances having between 55 and 163 candidates. We start with a systematic investigation of the performance of the individual reduction rules followed by describing our results for the web instances.

We systematically applied all combinations of reduction rules, always sticking to the following rule ordering: If applied, the Condorcet-Set Rule is applied last and the $3/4$-Majority Rule is applied first. After a successful application of the Condorcet-Set Rule, we "jump" back to the other rules (if "activated"). Examples for the combinations are given in Table 3. This led to the following observations.

Table 4. Web data instances with more than 100 candidates. The first column denotes the search term, the second the number of candidates, the third the running time of data reduction in seconds, and the last column the "profiles" remaining after data reduction.

search term	# cand.	time		structure of reduced instance	
affirmative action	127	0.21	1^{27}	$> 41 >$	1^{59}
alcoholism	115	0.10	1^{115}		
architecture	122	0.16	1^{36}	$> 12 > 1^{30} > 17 >$	1^{27}
blues	112	0.10	1^{74}	$> 9 >$	1^{29}
cheese	142	0.20	1^{94}	$> 6 >$	1^{42}
classical guitar	115	0.19	1^{6}	$> 7 > 1^{50} > 35 >$	1^{17}
Death+Valley	110	0.11	1^{15}	$> 7 > 1^{30} > 8 >$	1^{50}
field hockey	102	0.17	1^{37}	$> 26 > 1^{20} > 4 >$	1^{15}
gardening	106	0.10	1^{54}	$> 20 > 1 > 1 > 9 > 1^{8} > 4 >$	1^{9}
HIV	115	0.13	1^{62}	$> 5 > 1^{7} > 20 >$	1^{21}
lyme disease	153	3.08	1^{25}	$> 97 >$	1^{31}
mutual funds	128	2.08	1^{9}	$> 45 > 1^{9} > 5 > 1 > 49 >$	1^{10}
rock climbing	102	0.07	1^{102}		
Shakespeare	163	0.26	1^{100}	$> 10 > 1^{25} > 6 >$	1^{22}
telecommuting	131	1.60	1^{9}	$> 109 >$	1^{13}

First, surprisingly, the Condorcet Rule alone led to a stronger reduction than the $3/4$-Majority Rule in most of the instances whereas the $3/4$-Majority Rule never led to a stronger reduction than the Condorcet Rule. Second, for several instances the Condorcet-Set Rule led to a stronger reduction than the other two rules, for example, for gardening and classical guitar (see Table 3). It led to a stronger reduction for 14 out of the 36 instances and restricted to the 15 instances with more than 100 candidates (given in Table 4), it led to a stronger reduction for eight of them. The running times for the Condorcet-Set Rule in combination with the other rules are given in the left part of Fig. 1. Applying the Condorcet Rule before the Condorcet-Set Rule led to a significant speed-up. Additionally applying the $3/4$-Majority Rule changes the running time only marginally. Note that jumping back to the "faster" rules after applying the Condorcet-Set Rule is crucial to obtain the given running times. In the following, by "our reduction rules", we refer to all three rules applied in the order: $3/4$-Majority Rule, Condorcet Rule, and Condorcet-Set Rule.

For all instances with more than 100 candidates, the results of our reduction rules are displayed in Table 4: the data reduction rules are not only able to reduce candidates at the top and the last positions but also partition some instances into several smaller subinstances. Out of the 36 instances, 22 were solved directly by the reduction rules and one of the other algorithms in less than five minutes. Herein, the reduction rules always contributed with less than four seconds to the running time. For all other instances we still could compute the "top" and the "flop" candidates of an optimal ranking. For example, for the search term "telecommuting" there remains a subinstance with 109 candidates but we know the best nine candidates (and their order). The effectiveness in terms of top candidates of our reduction rules combined with the dynamic programming

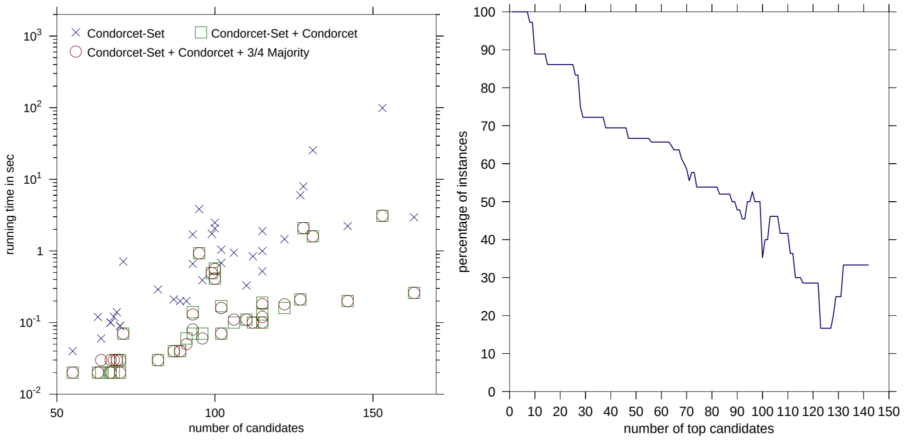

Fig. 1. Left: Running times of different combinations of reduction rules. To improve readability, we omitted the data points for the Condorcet-Set Rule combined with the $3/4$-Majority Rule which was usually worse and in no case outperformed the best running times for the other combinations. Right: Percentage of the web search instances for which the x top candidates could be determined by data reduction and dynamic programming within five minutes. For a given number x of top positions, we only considered instances with at least x candidates.

algorithm is illustrated in Fig. 1. For example, we were able to compute the top seven candidates for all instances and the top 40 candidates for 70 percent of the instances.

3.2 Impact rankings

We generated rankings that measure the "impact in the web" of different search terms. For a search engine, a list of search terms is ranked according to the number of the hits of each single term. We used Ask, Google, MSN Live Search, and Yahoo! to generate rankings for all capitals, all nations, and the 103 richest people of the world.[4] Our biggest instance is built from a list of 1349 mathematicians.[5]

As for the capitals, in less than a second, our algorithms (reduction rules and any of the other algorithms for solving subinstances up to 11 candidates) computed the following "profile" of a Kemeny ranking: $1^{45} > 34 > 1^{90} > 43 > 1^{26}$ (see Table 3 for a description of the profile concept). The final Kemeny ranking starts as follows: London $>$ Paris $>$ Madrid $>$ Singapore $>$ Berlin $> \cdots$. For aggregating the nation rankings, our algorithms were less successful. However, we could still compute the top 6 and the flop 12 candidates. Surprisingly, the best represented nation in the web seems to be Indonesia, followed by France, the United States, Canada, and Australia. The instance consisting of the 103 richest persons could be solved exactly in milliseconds by the

[4] http://en.wikipedia.org/wiki/List_of{capitals_by_countries, richest_people}

[5] http://aleph0.clarku.edu/∼djoyce/mathhist/chronology.html

data reduction rules. In contrast, for the mathematicians we could only compute the top 31 and flop 31 candidates but could not deal with a subinstance of 1287 candidates between. For the mathematicians instance, the search strategy for minimal subsets for the Condorcet-Set Rule as given in Section 2 led to a running time of more than a day. Hence, we used a cutoff of 20 candidates for the size of the minimal subsets. This decreased the running time to less than one hour.

3.3 Winter sport competitions

For ski jumping and cross skiing, we considered the world cup rankings from the seasons 2005/2006 to 2008/2009,[6] ignoring candidates not appearing in all four rankings. Without data reduction, the ski jumping instance, consisting of 33 candidates, was solved by the ILP-solver GLPK in 103 seconds whereas the search tree and dynamic programming algorithms did not find a solution within five minutes. In contrast, the instance was solved in milliseconds by only applying the reduction rules. The cross skiing instance, consisting of 69 candidates, could not be solved without data reduction within five minutes by any of our algorithms but was reduced in 0.04 seconds such that one component with 12 and one component with 15 candidates were left while all other positions could be determined by the reduction rules. The remaining parts could be solved, for example by the dynamic programming algorithm, within 0.12 and 0.011 seconds.

4 Conclusion

Our experiments showed that the described data reduction rules allow for the computation of optimal Kemeny rankings for real-world instances of non-trivial sizes within seconds. We could solve large instances with more than 50 candidates by means of data reduction while the corresponding instances could not be directly solved by the ILP, the previously fastest exact algorithm [5], or the two other implemented fixed-parameter algorithms. A key feature of the data reduction rules is to break instances into smaller, independent instances. A crucial observation in the experiments with the different data reduction rules regards certain cascading effects, that is, jumping back to the faster-to-execute rules after a successful application of the Condorcet-Set Rule significantly improves the running time. This shows that the order of applying data reduction rules is important. We could not observe a specific behavior of our data reduction rules for the different types of data under consideration. However, a further extension of the data sets and experiments in this direction seem interesting.

On the theoretical side, we improved the previous partial kernel [4] with respect to the parameter average KT-distance from quadratic to linear size. It is open whether there is a linear partial kernel with respect to the $\geq_s$-majority for any $s < 3/4$. A natural step in answering this question seems to investigate whether for two *non-dirty* candidates a, b, there must be a Kemeny ranking with $a > b$ if $a \geq_s b$. An important extension of RANK AGGREGATION is to consider “constraint rankings”, that is, the problem input additionally contains a prespecified order of some candidate pairs in the

[6] Obtained from `http://www.sportschau.de/sp/wintersport/`

consensus list [17]. Here, our data reduction rules cannot be applied anymore. New reduction rules to be developed for this scenario could also be used in "combination" with the search tree algorithm [3] in an "interleaving mode" [13].

Acknowledgements. We are grateful to the anonymous referees of *IPEC-2010* and of the *Third International Workshop on Computational Social Choice (COMSOC-2010)* for constructive feedback helping to improve this work.

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). → Page 2.
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. → Page 2.
3. N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. Fixed-parameter algorithms for Kemeny rankings. *Theoretical Computer Science*, 410(45):4554–4570, 2009. → Pages 2, 7, and 12.
4. N. Betzler, J. Guo, C. Komusiewicz, and R. Niedermeier. Average parameterization and partial kernelization for computing medians. *Journal of Computer and System Sciences*, 2010. doi:10.1016/j.jcss.2010.07.005. → Pages 2, 3, 4, 5, 6, and 11.
5. V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing Kemeny rankings. In *Proc. of 21st AAAI*, pages 620–626. AAAI Press, 2006. → Pages 2, 7, and 11.
6. A. Davenport and J. Kalagnanam. A computational study of the Kemeny rule for preference aggregation. In *Proc. of 19th AAAI*, pages 697–702. AAAI Press, 2004. → Page 2.
7. C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proc. of 10th WWW*, pages 613–622, 2001. → Pages 1, 2, and 8.
8. R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proc. of 22nd ACM SIGMOD*, pages 301–312. ACM, 2003. → Page 1.
9. B. N. Jackson, P. S. Schnable, and S. Aluru. Consensus genetic maps as median orders from inconsistent sources. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):161–171, 2008. → Page 2.
10. M. Karpinski and W. Schudy. Approximation schemes for the betweenness problem in tournaments and related ranking problems. In *Proc. of 21st ISAAC*, 2010. To appear. → Page 2.
11. C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proc. of 39th STOC*, pages 95–103. ACM, 2007. → Page 2.
12. M. Mahajan, V. Raman, and S. Sikdar. Parameterizing above or below guaranteed values. *Journal of Computer and System Sciences*, 75:137–153, 2009. → Page 2.
13. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. → Page 12.
14. V. Raman and S. Saurabh. Improved fixed parameter tractable algorithms for two "edge" problems: MAXCUT and MAXDAG. *Information Processing Letters*, 104(2):65–72, 2007. → Page 7.
15. F. Schalekamp and A. van Zuylen. Rank aggregation: Together we're strong. In *Proc. of 11th ALENEX*, pages 38–51. SIAM, 2009. → Pages 2, 7, and 8.
16. N. Simjour. Improved parameterized algorithms for the Kemeny aggregation problem. In *Proc. of 4th IWPEC*, volume 5917 of *LNCS*, pages 312–323. Springer, 2009. → Page 2.
17. A. van Zuylen and D. P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research*, 34:594–620, 2009. → Pages 2 and 12.
18. H. Young and A. Levenglick. A consistent extension of Condorcet's election principle. *SIAM Journal on Applied Mathematics*, 35(2):285–300, 1978. → Page 2.