To appear in Journal of Computer and System Sciences, 2010

Average Parameterization and Partial Kernelization for Computing Medians[☆]

Nadja Betzler[a,1], Jiong Guo[b,2], Christian Komusiewicz[*,a,3], Rolf Niedermeier[a]

[a] *Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany*
[b] *Universität des Saarlandes, Campus E 1.4, D-66123 Saarbrücken, Germany*

Abstract

We propose an effective polynomial-time preprocessing strategy for intractable median problems. Developing a new methodological framework, we show that if the input objects of generally intractable problems exhibit a sufficiently high degree of similarity between each other *on average*, then there are efficient exact solving algorithms. In other words, we show that the median problems Swap Median Permutation, Consensus Clustering, Kemeny Score, and Kemeny Tie Score all are fixed-parameter tractable with respect to the parameter "average distance between input objects". To this end, we develop the novel concept of "partial kernelization" and, furthermore, identify polynomial-time solvable special cases for the considered problems.

Key words: polynomial-time preprocessing, data reduction, fixed-parameter tractability, rank aggregation, consensus clustering

1. Introduction

In median problems one is given a set of objects and the task is to find a "consensus object" that minimizes the sum of distances to the given input objects. Our new approach to solve generally intractable (mostly NP-hard) median problems considers an average measure for the similarity between the input objects by summing over all pairwise object distances divided by the number

[☆] A preliminary version of this work appears in the *Proceedings of the 9th Latin American Theoretical Informatics Symposium 2010*, Springer, LNCS, to appear.

[*] Corresponding author. Phone: +49-3641-946325. Fax: +49-3641-946322
Email addresses: nadja.betzler@uni-jena.de (Nadja Betzler), jguo@mmci.uni-saarland.de (Jiong Guo), c.komus@uni-jena.de (Christian Komusiewicz), rolf.niedermeier@uni-jena.de (Rolf Niedermeier)

[1] Supported by the DFG, project PAWS, NI 369/10.
[2] Supported by the DFG Excellence Cluster "Multimodal Computing and Interaction".
[3] Supported by a PhD fellowship of the Carl-Zeiss-Stiftung and the DFG, project PABI, NI 369/7.

of these pairs. Based on this, we develop an algorithmic framework for showing that if the input objects are sufficiently "similar on average", then there are provably effective data reduction rules. In terms of parameterized algorithmics [12, 14, 24], this means that we show that the four median problems we study are fixed-parameter tractable with respect to the parameter "average distance between input objects". To the best of our knowledge, this parameter has only been studied for the Kemeny Score problem [6, 27] by using exponential-time dynamic programming and search tree methods. This work complements these results by polynomial-time preprocessing through data reduction.

Marx [22] studies average parameterization for the Consensus Patterns problem. He also shows fixed-parameter tractability; however in his case the parameter relates to the solution quality whereas our parameter can be easily computed without knowing a solution because it directly relates to the input structure.

Let us briefly discuss the naturalness of average parameterization for two prominent median problems tackled in this paper. First, consider the NP-hard Consensus Clustering problem (see, e.g., [23, 2, 8]). Roughly speaking, the goal here is to find a median partition for a given set of partitions all over the same base set; this is motivated by the often occurring task to reconcile clustering information [5, 15, 23]. It is plausible that this reconciliation is only meaningful when the given input partitions have a sufficiently high degree of average similarity, because otherwise the median partition found may be meaningless since it tries to fit the demands of strongly opposing clustering proposals. Our algorithms are tailored for being efficient when there is "enough" consensus in the input. If this is not fulfilled, a standard way of coping with too heterogeneous input partitions is to cluster the partitions and then to use Consensus Clustering in each "cluster of partitions", where high average similarity is to be expected [15].

As a second prominent NP-hard problem, we study the computation of Kemeny rankings (also known as rank aggregation) arising in the area of voting (see, e.g., [1, 2, 11, 13, 17]). As Conitzer and Sandholm [10] pointed out, one potential view of voting is that there exists a "correct" outcome (ranking), and each voter's vote corresponds to a noisy perception of this correct outcome (see [9, 11] for practical studies in this direction). Studying an average parameterization with respect to the pairwise distance between input votes naturally reflects this view on voting. We develop efficient algorithms for computing Kemeny rankings in case of a reasonably small average distance between votes, again developing an effective preprocessing technique.

Within our framework, two points deserve particular attention. First, the identification of *polynomial-time solvable special cases* of the underlying problems. Second, a novel concept of kernelization based on polynomial-time data reduction that does not yield problem kernels in the classical sense of parameterized algorithmics but only "*partial problem kernels*". Roughly speaking, in (at least) "two-dimensional" problems as we study here (for instance, one dimension being the size of the base set and the other being the number of input subsets over this base set), this means that at least one dimension can be re-

duced such that its size only depends on the parameter value. This somewhat "weaker" concept of kernelization promises to be of wider practical use.

On the way to proving our results with respect to the parameter "average distance", we introduce another measurement of dissimilarity—the "number of dirty elements"—which can be considered as an alternative parameterization. We also show fixed-parameter tractability with respect to this parameterization. As we will see, both parameterizations are closely related. In comparison, the "average distance" seems to be the more intuitive and easier to understand parameter whereas the "dirty element" parameterization seems to yield stronger results.

Our work is organized as follows. In the next section, we present our algorithmic framework, using the Swap Median Permutation problem [25] as running example for showing fixed-parameter tractability with respect to the average swap distance between the input permutations. Our concrete main results refer to problems in the areas of data clustering and rank aggregation. More precisely, we study the NP-hard problems Consensus Clustering and to compute a Kemeny consensus in voting with and without ties. More details about the studied problems and the corresponding literature are provided in the respective sections.

We conclude with briefly describing the essential concepts of parameterized complexity [12, 14, 24] as used in this work. A problem with input instance I and parameter k is *fixed-parameter tractable* if it can be solved by an exact algorithm with running time $f(k) \cdot \text{poly}(|I|)$ for some computable function f only depending on the parameter k. Moreover, a problem with instance (I, k) is called *kernelizable* [7, 16] if there is a polynomial-time algorithm that computes an equivalent instance (I', k') where the size of I' is a function of k and $k' \leq k$. The new instance (I', k') is reduced in size and called *problem kernel*.

2. Framework and Swap Median Permutation

In this work, we are concerned with consensus problems. Roughly speaking, the common feature of all these problems is that one is given a number of combinatorial objects (such as permutations, partitions etc.) over a base set U and wants to find a *median* object over U that minimizes the sum of "distances" to all input objects.

The general outline of our framework reads as follows.

Step 1. Identify a polynomial-time solvable special case. This is done by defining a "dirtiness" concept for elements from the base set U and proving that an instance of the underlying consensus problem can easily be solved when the input objects do not induce any dirty elements.
Step 2. Show that the number of dirty elements from U is bounded from above by a polynomial only depending on the average distance between the given combinatorial objects.
Step 3. Develop polynomial-time data reduction rules which shrink the number of non-dirty elements from U, generating an equivalent problem instance of

smaller size. Then show that the number of non-dirty elements in the reduced instance can be bounded from above by a polynomial only depending on the number of dirty elements and, thus, also the average distance.
Step 4. Make use of the fact that the desired median combinatorial object can be found in a running time only depending on the number of elements in U, and not depending on the number of combinatorial objects.

When applicable, this framework yields fixed-parameter tractability with respect to both parameters "average distance" and "number of dirty pairs". In general, fixed-parameter tractability would also follow for non-polynomial functions in Steps 2 and 3, but all our results provide polynomial bounds. A special feature of our framework is that in Step 3 we perform a *partial kernelization*, a concept that should be of general interest. Herein, the term "partial" refers to the fact that only the size of the base set is reduced, but not the number of input objects.

To illustrate our framework for efficiently solving "similar-on-average" median problems, we use the Swap Median Permutation problem (SMP for short) as a running example.[4] Herein, the combinatorial objects are permutations over the set $\{e_1, \ldots, e_m\}$; the distance between two permutations is the *swap distance* defined as follows: A *swap* operation interchanges two elements of a permutation. Thus, swapping e_i and e_j in the identity permutation

$$e_1 \cdots e_{i-1}\, e_i\, e_{i+1} \cdots e_{j-1}\, e_j\, e_{j+1} \cdots e_m$$

leads to

$$e_1 \cdots e_{i-1}\, e_j\, e_{i+1} \cdots e_{j-1}\, e_i\, e_{j+1} \cdots e_m.$$

The minimum number of swaps needed to transform a permutation π_1 into a permutation π_2 (or vice versa) is called the *swap distance* between π_1 and π_2, denoted by $\mathrm{dist}(\pi_1, \pi_2)$. Concerning notation, we follow Popov [25]. The formal problem definition of SMP reads as follows:

Input: A set of permutations $\{\pi_1, \pi_2, \ldots, \pi_n\}$ over $\{e_1, e_2, \ldots, e_m\}$.
Output: A median permutation π with minimum distance $\sum_{i=1}^{n} \mathrm{dist}(\pi, \pi_i)$.

The *average swap distance* d for an input instance of SMP is defined as

$$d := \big(\sum_{i \neq j} \mathrm{dist}(\pi_i, \pi_j)\big)/(n \cdot (n-1)).$$

The computation of the swap distance between two permutations can be carried out in $O(nm)$ time [3] by exploiting the tight relation between swap distances and *permutation cycles*. Given two permutations π_1 and π_2 of a set U, a permutation cycle of π_1 with respect to π_2 is a subset of π_1 whose elements, compared to π_2, trade positions in a circular fashion. In particular, an element e having the same position in both π_1 and π_2 builds a cycle by itself. For example,

[4] We remark that the question of the NP-hardness of SMP seems unsettled, cf. [25].

with respect to permutation $e_1e_2e_3e_4e_5e_6$, permutation $e_3e_5e_1e_4e_6e_2$ has three permutation cycles (e_1, e_3), (e_4), and (e_2, e_5, e_6). With respect to π_2, the cycle representation of π_1 as a product of disjoint permutation cycles is unique (up to the ordering of the cycles) and can be computed in $O(m^2)$ time [3]. The central observation behind the swap distance computation made by Amir et al. [3] is as follows: The swap distance between π_1 and π_2 is $m - c(\pi_1)$, where $c(\pi_1)$ is the number of permutation cycles in π_1 with respect to π_2.

First, according to Step 1, we need to define "dirty" elements. A *dominating position* of an element e is a position such that e occurs at this position in more than $n/2$ input permutations. An element is called *dirty* if it has no dominating position; otherwise, it is called *non-dirty*. Lemma 1 not only leads to the polynomial-time solvability of the special case but also is crucial for the correctness of a data reduction rule used in Step 3. In the following, we use $\pi[i]$ to denote the element at position i of a permutation π.

Lemma 1. *Every median permutation places the non-dirty elements according to their dominating positions.*

Proof. Let π be a median permutation where a non-dirty element e does not take its dominating position i, say e has position j in π with $i \neq j$. Now consider the permutation π' obtained from π by swapping e and $\pi[i]$. We show that π' has smaller distance to the input permutations than π. Let $r_\pi(\pi_l)$ be the cycle representation of an input permutation π_l with respect to π and let $r_{\pi'}(\pi_l)$ be the one with respect to π'. Since i is the dominating position of e, we have more than $n/2$ input permutations π_l with $e = \pi_l[i]$. Then, compared to $r_\pi(\pi_l)$, we create in $r_{\pi'}(\pi_l)$ a new permutation cycle consisting only of e for each π_l by swapping e to position i. Moreover, in $r_\pi(\pi_l)$ of each of these permutations π_l, e and $\pi_l[i]$ are in the same permutation cycle. Thus, we increase the number of permutation cycles by at least one in $r_{\pi'}(\pi_l)$ for each π_l. For each of the remaining less than $n/2$ input permutations π_l, we have $\text{dist}(\pi', \pi_l) \leq \text{dist}(\pi, \pi_l) + 1$, because $\text{dist}(\pi, \pi') = 1$. Altogether, π' has a distance to the input permutations smaller than the one π has. □

Lemma 2. *SMP without dirty elements can be solved in $O(nm)$ time.*

Proof. Due to Lemma 1 and the observation that the dominating positions of the elements can be easily computed in $O(nm)$ time, the claim directly follows. □

Next, according to Step 2, we have to bound the number of dirty elements.

Lemma 3. *Given an SMP-instance with average swap distance d, there are less than $4d$ dirty elements.*

Proof. For each dirty element e, let $\{i_1, i_2, \ldots, i_l\}$, $l \leq n$, be the set of positions where e occurs in the input permutations. For $1 \leq j \leq l$, let $\text{occ}(i_j)$ denote the number of input permutations π with $\pi[i_j] = e$. Thus, $\Sigma_{j=1}^{l}\text{occ}(i_j) = n$ and, since e is dirty, $\text{occ}(i_j) \leq n/2$ for $1 \leq j \leq l$. Overall, there are $\Sigma_{j=1}^{l}(\text{occ}(i_j) \cdot$

$(n - \mathrm{occ}(i_j)))/2$ pairs of input permutations such that, for each of these pairs π and π', $\mathrm{pos}_\pi(e) \neq \mathrm{pos}_{\pi'}(e)$ with $\mathrm{pos}_\pi(e)$ denoting the position of e in π. This sum is always greater than $n^2/4$. Moreover, for each of these pairs π and π', e is contained in a size-at-least-two permutation cycle of π with respect to π'. Since every permutation cycle with size k needs exactly $k-1$ swap operations to sort the elements in it [3] and one swap operation can sort at most two elements, we need altogether more than $(x/2) \cdot (n^2/4)$ swap operations to sort the dirty elements for all pairs of input permutations, where x denotes the number of dirty elements. Dividing this number of operations by $n \cdot (n-1)/2$ (note that in our definition of average distance we count every pair twice, and hence divide by $n \cdot (n-1)$ instead) yields a lower bound on the average swap distance, which is then more than $x/4$, showing the claim. □

According to Step 3, the number of non-dirty elements needs to be bounded. To this end, we present the following data reduction rule.

Reduction Rule. *In each of the input permutations, swap all non-dirty elements to their dominating positions. Remove all non-dirty elements.*

Lemma 4. *The data reduction rule above yields an equivalent SMP-instance with at most* $4d$ *elements, and it can be executed in* $O(nm)$ *time.*

Proof. According to Lemma 1, each non-dirty element should take its dominating position. Thus, we can already count the number of swap operations needed to sort them in the input permutations, that is, to swap them to their dominating positions. Since each swap operation can sort only one element and the dirty elements cannot occupy the dominating positions of non-dirty elements in any median permutation, swapping non-dirty elements does not affect the dirty elements. Thus, the reduction rule is correct. The bound on the size of the reduced instance derives from Lemma 3. The $O(nm)$ running time can be achieved by iterating over all elements. For each non-dirty element e, swapping e to its dominating position in one input permutation needs constant time. □

Finally, according to Step 4, it remains to observe that for the median permutation we clearly have $O((\lceil 4d \rceil)!)$ possibilities. Hence, simply testing all of them and taking a best one, we obtain the following proposition.

Proposition 1. Swap Median Permutation *is fixed-parameter tractable with respect to the parameter average swap distance as well as with respect to the number of dirty elements.*

3. Consensus Clustering

Our second application of the framework deals with the NP-hard Consensus Clustering problem. It arises in attempts to reconcile clustering information. The goal is to find a *median partition* for a given set of partitions, which all are over the same base set. The problem is defined as follows.

Input: A set $\mathcal{C} = \{C_1, \ldots, C_n\}$ of partitions over a base set S.
Output: A partition C of S with minimum distance $\sum_{C_i \in \mathcal{C}} \text{dist}(C, C_i)$.

Consensus Clustering was introduced in the area of clustering of gene expression data [23]. Its NP-hardness was shown by Křivánek and Morávek [21] and later also by Wakabayashi [28]. Bonizzoni et al. [8] showed that Consensus Clustering is APX-hard even if the input consists of only three partitions, whereas the maximization version has a polynomial-time approximation scheme (PTAS). For the minimization version of Consensus Clustering, the best approximation factor achievable in polynomial time is 4/3 [2]. Various heuristics for Consensus Clustering have been experimentally evaluated [5, 15].

Following Goder and Filkov [15], we call two elements $a, b \in S$ *co-clustered* with respect to a partition C if a and b occur together in a subset of C and *anti-clustered* if a and b occur in different subsets of C. Given a set $\mathcal{C}$ of partitions, we denote with $\text{co}(a, b)$ the number of partitions in $\mathcal{C}$ in which a and b are co-clustered and with $\text{anti}(a, b)$ the number of partitions in $\mathcal{C}$ in which a and b are anti-clustered. Define the *distance* $\text{dist}(C_i, C_j)$ between two input partitions C_i and C_j as the number of unordered pairs $\{a, b\}$ of elements from the base set S such that a and b are co-clustered in one of C_i and C_j and anti-clustered in the other. Our parameter d denoting the *average distance* of a given Consensus Clustering instance is then defined as

$$d := \Big(\sum_{C_i, C_j \in \mathcal{C}} \text{dist}(C_i, C_j) \Big) / \big(n \cdot (n-1) \big).$$

Our overall goal is to show that Consensus Clustering is fixed-parameter tractable with respect to the average distance d. To this end, we follow the framework presented in Section 2. Recall that Step 1 was to identify a polynomial-time solvable special case using a dirtiness concept.

Definition 1. A pair of elements $a, b \in S$ is called a *dirty pair* $a\#b$ of a set $\mathcal{C}$ of n partitions if $\text{co}(a, b) \geq n/3$ and $\text{anti}(a, b) \geq n/3$. Moreover, the predicate (ab) is true iff $\text{co}(a, b) > 2n/3$, and the predicate $a \leftrightarrow b$ is true iff $\text{anti}(a, b) > 2n/3$.

To show that an input instance of Consensus Clustering *without* dirty pairs is polynomial-time solvable, we need the following.

Lemma 5. *Let $\{a, b, c\}$ be a set of elements where a and c do not form a dirty pair. Then, $(ab) \wedge (bc) \Rightarrow (ac)$ and $a \leftrightarrow b \wedge (bc) \Rightarrow a \leftrightarrow c$.*

Proof. Since a and c do not form a dirty pair, by definition, c can only be co-clustered with a in either less than one third of the partitions or more than two thirds of the partitions. However, since (ab) and (bc), this implies that c has to be co-clustered with a in more than one third of all partitions, thus implying (ac). The argumentation for $a \leftrightarrow b$ and (bc) implying $a \leftrightarrow c$ works in an analogous manner. □

Proposition 2. Consensus Clustering *without dirty pairs is solvable in polynomial time.*

Proof. Let C be an optimal solution, that is, C is a partition of S with minimum distance to the input partitions. It suffices to show that in C the following two statements are true.

1. If (ab), then a and b are co-clustered in C.
2. If $a \leftrightarrow b$, then a and b are anti-clustered in C.

Clearly, since there are no dirty pairs, any pair $a, b \in S$ must fulfill either (ab) or $a \leftrightarrow b$. Hence, the two statements directly specify for each element from S in which subset in C it will end up.

To prove the first statement, suppose that there is an optimal solution C not fulfilling it. Then, there must exist two subsets S_i and S_j in C with $a \in S_i$ and $b \in S_j$. One can further partition both S_i and S_j into each time two subsets. More specifically, let $S_i^1 := \{x \in S_i : (ax)\}$ and $S_i^2 := S_i \setminus S_i^1$. The sets S_j^1 and S_j^2 are defined analogously with respect to b. In this way, by replacing S_i and S_j with $S_i^1 \cup S_j^1$, S_i^2, and S_j^2, one obtains a modified partition C'. Consider any $x \in S_i^1$ and any $y \in S_i^2$. Then, $x \leftrightarrow y$ follows from (ax), $a \leftrightarrow y$, and Lemma 5. The same is true with respect to S_j^1 and S_j^2. Moreover, if $x \in S_i^1$ and $y \in S_j^2$, this means that (ax) and $b \leftrightarrow y$, implying by Lemma 5 and using (ab) that $x \leftrightarrow y$. It remains to consider $x \in S_i^1$ and $y \in S_j^1$. Then, again the application of Lemma 5 yields (xy). Thus, C' is a better partition than C is because in C' now (ab) holds for all elements $a, b \in S_i^1 \cup S_j^1$ (without causing any increased cost elsewhere). This contradicts the optimality of C, proving the first statement. The second statement is proved analogously. □

As required by Step 2 of the framework in Section 2, the next lemma estimates the number of dirty pairs with the help of the average distance d.

Lemma 6. *An input instance of* Consensus Clustering *with average distance d contains less than $9d/4$ dirty pairs.*

Proof. We claim that every dirty pair $a\#b$ contributes more than $4n^2/9$ to the overall distance $\sum_{C_i, C_j \in \mathcal{C}} \mathrm{dist}(C_i, C_j)$. Given that, the statement of Lemma 6 follows by observing that $\sum_{C_i, C_j \in \mathcal{C}} \mathrm{dist}(C_i, C_j) = d \cdot n \cdot (n-1)$. Hence, it remains to prove the claim.

To prove the claim, first recall that for every dirty pair $\mathrm{co}(a, b) \geq n/3$ and $\mathrm{anti}(a, b) \geq n/3$. Clearly, $\mathrm{co}(a, b) + \mathrm{anti}(a, b) = n$. To show that a dirty pair $a\#b$ contributes more than $4n^2/9$ to the overall distance, note that any pair makes the contribution $\mathrm{co}(a, b) \cdot (n - \mathrm{co}(a, b)) + \mathrm{anti}(a, b) \cdot (n - \mathrm{anti}(a, b)) = 2 \cdot \mathrm{co}(a, b) \cdot \mathrm{anti}(a, b)$. It is easy to see that under the given constraints then the minimum contribution is greater than $2 \cdot (n/3) \cdot (2n/3) = 4n^2/9$. □

Step 3 of our framework now calls for a polynomial-time data reduction that reduces the number of elements that do not appear in any dirty pair. We call these elements *non-dirty elements* and all other elements *dirty elements.*

Roughly speaking, the aim of our reduction rule is to find subsets of S that contain many non-dirty elements that are all co-clustered in more than $2n/3$ input partitions. If these subsets are too large, then we can reduce the instance. In order to find such subsets, we describe a partition of S that is based on its non-dirty elements. In the following, let S_1 denote the non-dirty elements of S, and S_2 the dirty elements. First, we describe a partition $P_1 = \{S_1^1, \ldots, S_1^l\}$ of S_1 into equivalence classes according to the non-dirty pairs in S_1. Then, we show that these equivalence classes also induce a partition of S_2.

For each equivalence class $S_1^i \in P_1$, we demand

- $\forall a \in S_1^i \, \forall b \in S_1^i : (ab)$ and
- $\forall a \in S_1^i \, \forall b \in S \setminus S_1^i : a \leftrightarrow b$.

Observe that, by Lemma 5, the partition P_1 of S_1 that fulfills these requirements is well-defined, since the predicate (ab) describes a transitive relation over S_1. Using P_1, we define the subsets S_2^i of S_2 as follows:

$$S_2^i := \{a \in S_2 \mid \exists b \in S_1^i : (ab)\}.$$

Informally, each S_2^i is the set of elements $a \in S_2$ that are often co-clustered with at least one element $b \in S_1^i$. We also define one additional set S_2^0 that contains all elements $a \in S_2$ such that there is no $b \in S_1$ for which (ab) holds.

Finally, we obtain a set of subsets $P = \{S^0, S^1, \ldots, S^l\}$ of S by setting $S^i = S_1^i \cup S_2^i$ for $1 \leq i \leq l$ and $S^0 = S_2^0$. We call this set of subsets *non-dirty-based*. The following lemma shows that P is indeed a partition of S, and also provides some further structural properties of P.

Lemma 7. *Let $P = \{S^0, S^1, \ldots, S^l\}$ be a non-dirty-based set of subsets of S constructed as described above. Then, P is a partition of S, and for each $S^i \in P$ it holds that*

- $\forall a \in S^i \, \forall b \in S : (ab) \Rightarrow b \in S^i$ *and*
- $\forall a, b \in S^i, i \geq 1 : (ab) \vee a\#b$.

Proof. First, we show that P is a partition. By Lemma 5, it is easy to verify that the claim holds for the partition P_1 of S_1. By definition, $\bigcup_{i=0}^{l} S_2^i = S_2$. We now show that for each $a \in S_2$ there is exactly one set S_2^i that contains a, and, thus, that P is a partition of S. By definition, S^0 does not overlap with any other set S^i, $i \geq 1$. Now, suppose that there are two sets S_2^i, $i \geq 1$, and S_2^j, $j \geq 1$, $j \neq i$, that contain a. Then there are two elements $b \in S_1^i$ and $c \in S_1^j$ such that (ab) and (ac) holds. Since P_1 is a partition of S_1, we have $c \notin S_1^i$ and thus also $b \leftrightarrow c$. But then it follows from Lemma 5 that $b \leftrightarrow a$ holds (since we have $b \leftrightarrow c$ and (ca)). This clearly contradicts (ab). We have thus shown that P is a partition of S.

We now show that for each $S^i \in P$ it holds that $\forall a \in S^i \forall b \in S : (ab) \Rightarrow b \in S^i$. Suppose that there is a pair of elements $a \in S^i$ and $b \in S^j$, $j \neq i$, for which (ab) holds. By definition, this can be only the case if $a \in S_2$ and $b \in S_2$.

Without loss of generality, assume that $i \geq 1$. This means that there is some element $c \in S_1^i$ with (ac). However, by Lemma 5, then also (cb) must hold. This contradicts $b \notin S^i$.

Finally, we show that for each $S^i \in P$, $i \geq 1$, it holds that $\forall a, b \in S^i : (ab) \vee a\#b$. Suppose that there is some S^i containing two elements a and b for which $a \leftrightarrow b$ holds. By definition of S_1^i, one of a and b must be from S_2^i, say $a \in S_2^i$, and there must be some $c \in S_1^i$ such that (ac) holds. By Lemma 5, we have $c \leftrightarrow b$. This means, however, that, also by Lemma 5, we have $b \leftrightarrow d$ for all $d \in S_1^i$. This contradicts $b \in S^i$. □

Informally, Lemma 7 says that inside any $S^i \in P$ we have only pairs that are *co-clustered* in more than $2n/3$ input partitions or dirty pairs; between two subsets $S^i \in P$ and $S^j \in P$ we have only dirty pairs or pairs that are *anti-clustered* in more than $2n/3$ input partitions. Clearly, the elements in S_1^i then are co-clustered in more than $2n/3$ partitions with *all* elements in S^i and are anti-clustered in more than $2n/3$ partitions with *all* elements in $S \setminus S^i$. This means that an S^i with too many elements in S_1^i is forced to become a set of an optimal partition. With the subsequent data reduction rule, we remove these sets from the input.

We introduce the following notation for subsets of S. For some set $E \subseteq S$, we denote with $\mathrm{dp}(E)$ the dirty pairs among the elements of E, that is, for a dirty pair $a\#b$ we have $a\#b \in \mathrm{dp}(E)$ if $a \in E$ and $b \in E$. Analogously, for two sets $E \subseteq S$ and $F \subseteq S$, we define $\mathrm{dp}(E, F)$ as the set of dirty pairs between E and F, that is, for a dirty pair $a\#b$ we have $a\#b \in \mathrm{dp}(E, F)$ if $a \in E$ and $b \in F$ or vice versa.

Reduction Rule. *Let P be a non-dirty-based partition of S. If there is some $S^i \in P$ such that*

$$|S_1^i| > |\,\mathrm{dp}(S^i)| + |\,\mathrm{dp}(S^i, S \setminus S^i)|,$$

then output S^i as one of the sets of the solution and remove the elements of S^i from all input partitions.

Lemma 8. *The data reduction rule above is correct.*

Proof. Let S^i be as described in the reduction rule. We show that every optimal partition C contains one set C^j such that $S^i = C^j$. In the following, we call the subsets of S in a partition C of S *clusters*. For our proof, we only consider clusters C^j that contain at least one element of S^i. In what follows, we partition each such C^j into four subsets. Figure 1 shows these sets and their relation to S^i.

- $C_1^j := \{a \in C^j \cap S^i \mid \forall b \in C^j \setminus S^i : a \leftrightarrow b\}$ contains those elements from S^i that do not appear in dirty pairs with elements from $C^j \setminus S^i$.
- $C_2^j := \{a \in C^j \cap S^i \mid \exists b \in C^j \setminus S^i : a\#b\}$ contains the (dirty) elements from S^i that form a dirty pair with some element from $C^j \setminus S^i$.
- $C_3^j := C^j \cap \{a \in S \setminus S^i \mid \exists b \in S^i \cap C^j : a\#b\}$ contains the elements of $C^j \setminus S^i$ that form a dirty pair with some element from $S^i \cap C^j$.

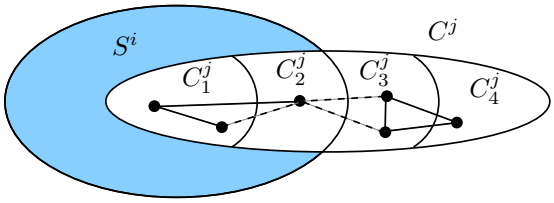

Figure 1: The subsets of a cluster C^j with respect to the set S^i as defined in the proof of Lemma 8. Solid edges are between elements a and b for which (ab) holds; dashed edges are between elements that form a dirty pair; elements a and b for which $a \leftrightarrow b$ holds have no edge between them.

- $C_4^j := C^j \setminus (C_1^j \cup C_2^j \cup C_3^j)$ contains all other elements.

We prove our claim in three steps. First, we show that $|C_1^j| \geq |C_2^j|$ implies $C^j = (C_1^j \cup C_2^j)$. Then, we show that there is exactly one C^j with $C^j = (C_1^j \cup C_2^j)$. Finally, we show that in an optimal partition, there is no C^j with $|C_1^j| < |C_2^j|$. The first two claims show that there is exactly one cluster C^j with $C^j \subseteq S^i$. The third claim shows that there can be no other clusters that have nonempty intersection with S^i. Altogether, this means that in an optimal clustering there is exactly one cluster C_j with $C_j \cap S^i \neq \emptyset$, which proves the correctness of the reduction rule.

Now, we show that in an optimal partition C, there is no C^j such that $|C_1^j| \geq |C_2^j|$ and $C^j \neq (C_1^j \cup C_2^j)$, since for any partition C that contains such a cluster C^j there is an alternative partition C' that has lower cost and is constructed as follows: replace the cluster C^j by two new clusters $C_1^j \cup C_2^j$ and $C_3^j \cup C_4^j$. We now show that C' has lower cost than C. Let $d(C)$ denote the cost of the partition C, and let $d(C')$ be the cost of the partition C'. Clearly, the costs of C and C' differ only in the costs for the pairs that contain one element from $C_1^j \cup C_2^j$ and one from $C_3^j \cup C_4^j$. For each pair of elements $a \in C_1^j \cup C_2^j$ and $b \in C_3^j \cup C_4^j$, C' saves a cost of $\text{anti}(a, b)$ compared to C, since these two elements now appear in different clusters. However, this means that C' has an additional cost of $\text{co}(a, b)$ for each such pair. Note that, by definition, the following holds:

- $(ab) \Rightarrow (\text{co}(a, b) - \text{anti}(a, b) > n/3)$,
- $a \leftrightarrow b \Rightarrow (\text{anti}(a, b) - \text{co}(a, b) > n/3)$, and
- $a\#b \Rightarrow (|\,\text{anti}(a, b) - \text{co}(a, b)| \leq n/3)$.

Overall, the cost difference between C and C' is then

$$\begin{aligned} d(C) - d(C') &= \sum_{a \in C_1^j \cup C_2^j} \sum_{b \in C_3^j \cup C_4^j} \text{anti}(a,b) - \text{co}(a,b) \\ &\overset{(*)}{>} \sum_{a \in C_1^j} \sum_{b \in C_3^j} \frac{n}{3} - \sum_{a \in C_2^j} \sum_{b \in C_3^j} \frac{n}{3} \\ &\overset{(**)}{\geq} 0. \end{aligned}$$

Inequality $(*)$ follows from the following four facts:

1. $\forall a \in C_1^j \cup C_2^j \forall b \in C_4^j : a \leftrightarrow b$,
2. $\forall a \in C_1^j \forall b \in C_3^j : a \leftrightarrow b$,
3. $\forall a \in C_2^j \forall b \in C_3^j : a \leftrightarrow b \vee a \# b$, and
4. $C_2^j \cup C_3^j \cup C_4^j \neq \emptyset$.

Inequality $(**)$ follows from the fact that $|C_1^j| \geq |C_2^j|$. Thus, we have shown that in an optimal partition there can be no clusters C^j with $|C_1^j| \geq |C_2^j|$ and $C^j \neq (C_1^j \cup C_2^j)$. Hence, we can have clusters C^j of two types, those with $C^j = (C_1^j \cup C_2^j)$ and those with $|C_1^j| < |C_2^j|$.

Next, we show that in an optimal solution, there is exactly one cluster with $C^j = (C_1^j \cup C_2^j)$. Let C_{iso} be the set of clusters C^j with $C^j = (C_1^j \cup C_2^j)$. Let C be a partition that creates more than one cluster in C_{iso}. We show that there is an alternative partition C' that merges two clusters of C_{iso} to a new cluster and that has lower cost than C. First, there must be two clusters $C^j \in C_{\text{iso}}$ and $C^l \in C_{\text{iso}}$ such that $|(C^j \cup C^l) \cap S_1| > \text{dp}(C^j \cup C^l)$, because, otherwise, the union of all clusters in C_{iso} has more dirty pairs than non-dirty elements. However, this is also the case for all other clusters C^h, since for these clusters we have $|C_1^h| < |C_2^h|$, which means that then S^i has more dirty pairs than non-dirty elements, contradicting the precondition of the reduction rule. Our alternative partition C' merges C^j and C^l into a new cluster $C^j \cup C^l$. Otherwise, it does not differ from C. The costs of C and C' differ only with respect to pairs that contain one element $a \in C^j$ and one element $b \in C^l$. For each pair, putting the elements in the same cluster instead of two different clusters saves $\text{co}(a,b)$ and costs $\text{anti}(a,b)$. The cost difference between C and C' is thus

$$\begin{aligned} d(C) - d(C') &= \sum_{a \in C^j} \sum_{b \in C^l} \text{co}(a,b) - \text{anti}(a,b) \\ &\overset{(*)}{\geq} \sum_{a \in (C^j \cup C^l) \cap S_1} \frac{n}{3} - |\,\text{dp}(C^j, C^l)| \cdot \frac{n}{3} \\ &\overset{(**)}{>} 0. \end{aligned}$$

Inequality $(*)$ follows from the two facts

1. $\forall a \in C^j \forall b \in C^l : (ab) \vee a \# b$ and

2. $\forall a \in (C^j \cup C^l) \cap S_1 \forall b \in C^j \cup C^l : (ab)$.

Inequality $(**)$ follows from the fact that $|(C^j \cup C^l) \cap S_1| > \mathrm{dp}(C^j \cup C^l)$. Hence, partition C is clearly not optimal. We have thus shown that in an optimal partition there is at most one cluster C^j with $C^j = (C_1^j \cup C_2^j)$, and possibly some other clusters C^l with $|C_1^l| < |C_2^l|$. Furthermore, by the precondition of the reduction rule, this means that there must be *exactly* one cluster C^j with $C^j = (C_1^j \cup C_2^j)$ in an optimal partition C.

We complete the proof of the correctness of the reduction rule by showing that in an optimal partition there is no cluster C^l with $|C_1^l| < |C_2^l|$. Let C^j be the cluster with $C^j = (C_1^j \cup C_2^j)$. We show that an optimal partition C never contains a cluster C^l with $|C_1^l| < |C_2^l|$, since then we can obtain a better partition C' by removing $C_1^l \cup C_2^l$ from C^l and merging $C_1^l \cup C_2^l$ and C^j into a new cluster $C^j \cup C_1^l \cup C_2^l$. First, observe that, by the precondition of the reduction rule, we have $|(C^j \cup C^l) \cap S_1| > \mathrm{dp}(C^j \cup C^l) + \mathrm{dp}(C^j \cup C^l, S \setminus (C^j \cup C^l))$. Otherwise, we would have $|S^i \cap S_1| < \mathrm{dp}(S^i) + \mathrm{dp}(S^i, S \setminus S^i)$, since already $C^j \cup C_1^l \cup C_2^l$ has less non-dirty elements than dirty pairs, and for each other cluster C^h from D, there are more dirty pairs than non-dirty elements (since $|C_1^h| < |C_2^h|$). We now compare the cost of C with the cost of C'. First, the costs have changed for pairs with $a \in C^j$ and $b \in C_1^l \cup C_2^l$, where in C' we have—compared to C—an additional cost of $\mathrm{anti}(a,b)$ and save a cost of $\mathrm{co}(a,b)$, since a and b are now in the same clusters. Second, the costs have changed for pairs $a \in C_1^l \cup C_2^l$ and $b \in C_3^l \cup C_4^l$, where in C' we have—compared to C—an additional cost of $\mathrm{co}(a,b)$ and save a cost of $\mathrm{anti}(a,b)$. Overall, the cost difference is

$$
\begin{aligned}
d(C) - d(C') = & \sum_{a \in C^j} \sum_{b \in C_1^l \cup C_2^l} \mathrm{co}(a,b) - \mathrm{anti}(a,b) \\
& + \sum_{a \in C_1^l \cup C_2^l} \sum_{b \in C_3^l \cup C_4^l} \mathrm{anti}(a,b) - \mathrm{co}(a,b) \\
\overset{(*)}{>} & -|\,\mathrm{dp}(C^j, C^l \cap S^i)| \cdot \frac{n}{3} + \sum_{a \in C^j \cap S_1} \sum_{b \in C_1^l \cup C_2^l} \frac{n}{3} \\
& - |\,\mathrm{dp}(C_2^j, C_3^j)| \cdot \frac{n}{3} + \sum_{a \in C_1^l} \sum_{b \in C_3^l \cup C_4^l} \frac{n}{3} \\
\overset{(**)}{\geq} & \frac{n}{3} \cdot (|C^j \cap S_1| \cdot |C_1^l \cup C_2^l| + |C_1^l| \cdot |C_3^l \cup C_4^l|) \\
& - \frac{n}{3} \cdot (|\,\mathrm{dp}(C^j, C^l \cap S^i)| + |\,\mathrm{dp}(C_2^l, C_3^l)|) \\
\overset{(***)}{>} & 0.
\end{aligned}
$$

Inequality $(*)$ follows from the following facts:

1. $\forall a \in C^j \forall b \in C_1^l \cup C_2^l : (ab) \vee a \# b$,
2. $\forall a \in C_1^l \forall b \in C_3^l \cup C_4^l : a \leftrightarrow b$, and
3. $\forall a \in C_2^l \forall b \in C_3^l \cup C_4^l : a \leftrightarrow b \vee a \# b$.

Inequality $(**)$ is straightforward, and inequality $(***)$ follows from the fact that $|(C^j \cup C_1^l) \cap S_1| > \mathrm{dp}(C^j \cup C_1^l \cup C_2^l) + \mathrm{dp}(C^j \cup C_1^l \cup C_2^l, S \setminus (C^j \cup C_1^l \cup C_2^l))$. An optimal partition thus does not contain a cluster C^l with $|C_1^l| < |C_2^l|$. Therefore, an optimal partition contains exactly one cluster C^j that contains all the elements from S_i and no other elements, proving the correctness of the reduction rule. □

In the following theorem, we combine Steps 3 and 4 of our framework: we show that exhaustively applying the reduction rule yields an equivalent instance whose number of elements is less than $9d$, and that this implies the fixed-parameter tractability of Consensus Clustering.

Theorem 1.
1. *Each instance of* Consensus Clustering *can be reduced in polynomial time to an equivalent instance with less than* $9d$ *elements in the base set. A resulting reduced instance contains only dirty elements.*
2. Consensus Clustering *is fixed-parameter tractable with respect to the average distance* d *between the input partitions as well as with respect to the number of dirty elements.*

Proof. Clearly, the reduction rule can be performed exhaustively in polynomial time. Therefore, consider an instance I that is reduced with respect to the reduction rule. With S_1 we denote the non-dirty elements of I, and with S_2 we denote the elements of S that appear in dirty pairs. By Lemma 6, the number of dirty pairs in I is less than $9d/4$. Hence, the size of the set S_2 containing the elements appearing in dirty pairs is less than $9d/2$. It remains to bound the number of non-dirty elements. For this, consider the non-dirty based partition P of S. Since the reduction rule cannot be applied, the number of non-dirty elements of each set $S^i \in P$ is bounded by the number of dirty pairs that contain at least one element from S^i. The overall size of the set $|S_1|$ containing the non-dirty elements can thus be bounded by

$$|S_1| \leq \sum_{S^i \in P} (\mathrm{dp}(S^i) + \mathrm{dp}(S^i, V \setminus S^i)) < 9d/2.$$

The second inequality stems from the fact that we have at most $9d/4$ dirty pairs and that the dirty pairs between different sets $S^i, S^j \in P$ have to be counted twice. Hence, a reduced instance contains at most $|S_1| + |S_2| < 2 \cdot (9d/2) < 9d$ elements. We can solve Consensus Clustering by trying all possible partitions (whose number is clearly a function of d), computing their costs in polynomial time, and then outputting the best partition. □

4. Kemeny Rankings

In the third application of our framework, we investigate the problem of finding a "consensus ranking", that is, a so-called Kemeny ranking [18]. We first consider the NP-hard Kemeny Score problem and, second, the somewhat harder to attack generalization Kemeny Tie Score.

4.1. Kemeny Score

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" with respect to the so-called *Kendall-Tau distance* to the preference lists of the voters. For each pair of votes v, w, the Kendall-Tau distance (*KT-distance* for short) between v and w, also known as the inversion distance between two permutations, is defined as

$$\mathrm{dist}(v, w) = \sum_{\{a,b\} \subseteq C} d_{v,w}(a, b),$$

where the sum is taken over all unordered pairs $\{a, b\}$ of candidates, and $d_{v,w}(a, b)$ is 0 if v and w rank a and b 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 $\sum_{v \in V} \mathrm{dist}(l, v)$. A preference list l with the minimum score is called a *Kemeny consensus* of (V, C) and its score $\sum_{v \in V} \mathrm{dist}(l, v)$ is the *Kemeny score* of (V, C). The Kemeny Score problem is defined as follows:

Input: An election (V, C).
Output: A Kemeny consensus l with minimum score $\sum_{v \in V} \mathrm{dist}(l, v)$.

To show our results, it will be useful to decompose the Kemeny score of a preference list into "partial scores". More precisely, for a preference list l and a candidate pair $\{a, b\}$, the *partial score* of l with respect to $\{a, b\}$ is

$$s_l(\{a, b\}) := \sum_{v \in V} d_{v,l}(a, b).$$

The partial score of l with respect to a subset P of candidate pairs is $s_l(P) := \sum_{p \in P} s_l(p)$.

Bartholdi et al. [4] showed that the decision version of Kemeny Score is NP-complete, and it remains so even when restricted to instances with only four votes [13]. The Kemeny score can be approximated to a factor of 8/5 by a deterministic algorithm [29] and to a factor of 11/7 by a randomized algorithm [2]. A polynomial-time approximation scheme (PTAS) for Kemeny Score is provided by Kenyon-Mathieu and Schudy [19]. However, its running time is impractical. Conitzer et al. [11] performed computational studies for the efficient exact computation of a Kemeny consensus, using heuristic approaches such as greedy and branch-and-bound. Schalekamp and van Zuylen [26] experimentally evaluated the quality of different approximation algorithms and heuristics. Hemaspaandra et al. [17] provided further exact classifications of the 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.

For an election (V, C), the average KT-distance d, the average parameterization for Kemeny Score, is defined as

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

The Kemeny Score problem is known to be fixed-parameter tractable with respect to the parameter d [6, 27]. The currently fastest algorithm is a branching algorithm running in $5.823^d \cdot \text{poly}(n, m)$ time [27]. We extend these results by showing that the approach presented in Section 2 can be applied to Kemeny Score.

To identify a polynomial-time solvable special case as described in Step 1 of our framework, it is crucial to develop a concept of dirtiness.[5] For Kemeny Score this is realized as follows. Let (V, C) denote an election. An unordered pair of candidates $\{a, b\} \subseteq C$ with neither $a > b$ nor $a < b$ in more than 2/3 of the votes is called a *dirty pair* and a and b are called *dirty candidates*. All other pairs of candidates are called *non-dirty pairs*, and candidates that appear only in non-dirty pairs are called *non-dirty candidates*. Note that with this definition a non-dirty pair can also be formed by two dirty candidates. Let D denote the set of dirty candidates and n_d denote the number of dirty pairs in (V, C). For two candidates a, b, we write $a >_{2/3} b$ if $a > b$ in more than 2/3 of the votes. We say that a and b are *ordered according to the* 2/3*-majority* in a preference list l if $a >_{2/3} b$ and $a > b$ in l.

Proposition 3. Kemeny Score *without dirty pairs is solvable in polynomial time.*

Proof. For an input instance (V, C) of Kemeny Score without dirty pairs, we show that the preference list "induced" by the 2/3-majority of the candidate pairs is optimal.

First, we show by contradiction that there is a preference list $l_{2/3}$ where for all candidate pairs $\{a, b\}$ with $a, b \in C$ and $a >_{2/3} b$, one has $a > b$. Assume that such a preference list does not exist. Then, there must be three candidates $a, b, c \in C$ that violate transitivity, that is, $a >_{2/3} b$, $b >_{2/3} c$, and $c >_{2/3} a$. Since $a >_{2/3} b$ and $b >_{2/3} c$, there must be at least $n/3$ votes with $a > b > c$. Since a and c do not form a dirty pair, it follows that $a >_{2/3} c$, a contradiction.

Second, we show by contradiction that $l_{2/3}$ is optimal. Assume that there is a Kemeny consensus l with a non-empty set P of candidate pairs that are not ordered according to the 2/3-majority; that is, $P := \{\{c, c'\} : c > c' \text{ in } l \text{ and } c' >_{2/3} c\}$. All candidate pairs that are not in P are ordered equally in l and $l_{2/3}$. Thus, the partial score with respect to them is the same for l and $l_{2/3}$. For every candidate pair $\{c, c'\} \in P$, the partial score $s_l(\{c, c'\})$ is more than $2n/3$ and the

[5] In earlier work on Kemeny Score [6] the term "dirty" is used in a different way to obtain fixed-parameter tractability results with respect to other parameters. In contrast to our framework, the previous results for the parameterization by the average KT-distance [6, 27] do not classify the candidates into different groups.

partial score $s_{l_{2/3}}(\{c, c'\})$ is less than $n/3$. Thus, the score of $l_{2/3}$ is smaller than the score of l, a contradiction to the optimality of l. □

Following Step 2 of our framework, the next lemma shows how the number of dirty pairs and, thus, also the number of dirty candidates, is bounded from above by a function linear in the average KT-distance d.

Lemma 9. *Given an instance of* Kemeny Score *with average KT-distance d, there are less than $9d/2$ dirty pairs.*

Proof. For an election (V, C) with average KT-distance d, let i denote the number of dirty pairs. Every dirty pair $\{a, b\} \subseteq C$ contributes more than $n/3 \cdot 2n/3$ to the overall sum of KT-distances. Recall that

$$d = \left(\sum_{v,w \in V} \text{dist}(v, w) \right) / (n(n-1)) = \left(\sum_{\{c,d\} \subseteq C} \sum_{v,w \in V} d_{v,w}(c, d) \right) / (n(n-1)).$$

Thus,

$$d > \frac{1}{n(n-1)} \cdot i \cdot \frac{n}{3} \cdot \frac{2n}{3} > \frac{2}{9} \cdot i \Leftrightarrow \frac{9}{2} \cdot d > i.$$

□

The following three lemmas establish the basis for a polynomial-time data reduction rule as required in Step 3 of our framework. The basic idea is to consider the order that is induced by the 2/3-majorities of the non-dirty pairs and then to show that a dirty candidate can only "influence" the order of candidates that are not "too far away" from it in this order. Then, it is safe to remove non-dirty candidates that cannot be influenced by any dirty candidate.

Lemma 10. *For an election containing n_d dirty pairs, in every Kemeny consensus at most n_d non-dirty pairs are not ordered according to their 2/3-majorities.*

Proof. For an election (V, C) with n_d dirty pairs, let l be a Kemeny consensus with $P := \{\{c, c'\} : c > c' \text{ in } l \text{ and } c' >_{2/3} c\}$ and $|P| > n_d$. Then, we show that l cannot be optimal.

Let $l_{2/3}$ denote a preference list with $c > c'$ for all pairs with $c >_{2/3} c'$ and the remaining dirty pairs are ordered arbitrarily. This can be done without violating transitivity. More precisely, due to Proposition 3, all non-dirty candidates can be ordered according to the 2/3-majority. Analogously, one can show that every dirty candidate can be ordered according to the 2/3-majority with respect to all non-dirty candidates and that two dirty candidates that form a non-dirty pair do not violate transitivity if ordered according to the 2/3-majority of this pair. Since the remaining dirty pairs can be ordered arbitrarily, they can be ordered without violating transitivity as well.

We show that $\text{score}(l) > \text{score}(l_{2/3})$. Let C_P denote the set of all pairs of candidates of C, that is, $C_P := \{\{c, c'\} : c, c' \in C, c \neq c'\}$, and D_P denote the

set of all dirty pairs in (V, C). Then, score(l) and score$(l_{2/3})$ can be decomposed into partial scores depending on candidate pairs of P, D_P, and $C_P \backslash (D_P \cup P)$:

$$\text{score}(l) = s_l(P) + s_l(D_P) + s_l(C_P \backslash (D_P \cup P))$$

Now, consider score(l) − score$(l_{2/3})$. Since all pairs $p \in C_P \backslash (D_P \cup P)$ are ordered according to the 2/3-majority in l and in $l_{2/3}$, the partial scores for them are equal. The partial score for every non-dirty pair is more than $2n/3$ if it is not ordered according to the 2/3-majority, and less than $n/3$ otherwise. Together with the fact that for a dirty pair the difference of the partial scores of the two possible orders is at most $n/3$, one has

$$s_l(D_P) - s_{l_{2/3}}(D_P) \geq -|D_P| \cdot n/3,$$

and

$$s_l(P) - s_{l_{2/3}}(P) > |P| \cdot n/3.$$

Since $|P| > |D_P|$, it follows that score(l)−score$(l_{2/3}) > n/3 > 0$ and, thus, l cannot be optimal. □

In the following, we show that the bound on the number of "incorrectly" ordered non-dirty pairs from Lemma 10 can be used to fix the relative order of two candidates forming a non-dirty pair. For this, it will be useful to have a concept of distance of candidates with respect to the order induced by the 2/3-majority. For an election (V, C) and a non-dirty pair $\{c, c'\}$, define $\text{dist}(c, c') := |\{b \in C : b \text{ is non-dirty and } c >_{2/3} b >_{2/3} c'\}|$ if $c >_{2/3} c'$ and $\text{dist}(c, c') := |\{b \in C : b \text{ is non-dirty and } c' >_{2/3} b >_{2/3} c\}|$ if $c' >_{2/3} c$.

Lemma 11. *Let (V, C) be an election and let $\{c, c'\}$ be a non-dirty pair. If $\text{dist}(c, c') \geq n_d$, then in every Kemeny consensus $c > c'$ iff $c >_{2/3} c'$.*

Proof. Let l be a preference list such that there is a non-dirty pair $\{c, c'\}$ with $c > c'$ in l, $c' >_{2/3} c$, and $\text{dist}(c, c') \geq n_d$. We show that l cannot be a Kemeny consensus. Since $\text{dist}(c, c') \geq n_d$, there are at least n_d non-dirty candidates e with $c' >_{2/3} e >_{2/3} c$. Since $c > c'$ in l, these candidates e cannot be ordered according to the 2/3-majority with respect to c or c' in l. Hence, there are at least n_d pairs formed by the candidates e and c or c' in l, which, together with the pair $\{c, c'\}$, give more than n_d non-dirty pairs that are not ordered according to the 2/3-majority. This contradicts Lemma 10 and l cannot be optimal. □

Finally, the next lemma enables us to fix the position in a Kemeny consensus for a non-dirty candidate that has a large distance to all dirty candidates.

Lemma 12. *If for a non-dirty candidate c it holds that $\text{dist}(c, c_d) > 2n_d$ for all dirty candidates $c_d \in D$, then in every Kemeny consensus c is ordered according to the 2/3-majority with respect to all candidates from C.*

Proof. Assume that there is a non-dirty candidate c with $\text{dist}(c, c_d) > 2n_d$ for all $c_d \in D$ and that there is a preference list l with $e > c$ for a candidate e with $c >_{2/3} e$. Then, we show that l cannot be optimal.

Since $\text{dist}(c, c_d) > 2n_d$ for all dirty candidates $c_d \in D$, it follows from Lemma 11 that all dirty candidates must be ordered according to the 2/3-majority with respect to c. Thus, e must be a non-dirty candidate. Due to Lemma 11, $\text{dist}(e, c) < n_d$. Since for all $c_d \in D$ one has $\text{dist}(c, c_d) > 2n_d$, it follows from $\text{dist}(e, c) < n_d$ that $\text{dist}(e, c_d) > n_d$ for all $c_d \in D$ as well. Thus, in a Kemeny consensus, e must be ordered according to the 2/3-majority with respect to all dirty candidates due to Lemma 11. For a candidate $c_d \in D$ one has $c >_{2/3} c_d$ iff $e >_{2/3} c_d$ since for all $c_d \in D$ one has $\text{dist}(c, c_d) > 2n_d$ and $\text{dist}(e, c) < n_d$. Hence, there is no dirty candidate $c_d \in D$ with $e > c_d > c$ in l, that is, all candidates $f_i, i = 1, \ldots, s$, with $e > f_i > \cdots > f_s > c$ in l must be non-dirty. Then, analogously to the proof of Proposition 3, one can show that ordering $c, e, f_1, \ldots, f_s$ according to the 2/3-majority gives a consensus with score less than the score of l. Thus, l cannot be optimal. □

The correctness of the following data reduction rule follows directly from Lemma 12. It is not hard to verify that it can be carried out in $O(n \cdot m^2)$ time.

Reduction Rule. *For an election with n_d dirty pairs, let c be a non-dirty candidate with $\text{dist}(c, c_d) > 2n_d$ for all $c_d \in D$. Let $C_l := \{c' \in C : c' >_{2/3} c\}$ and $C_r := \{c' \in C : c >_{2/3} c'\}$. Delete c and reorder every vote such that $C_l > C_r$ and the order of the candidates within C_l and C_r remains unchanged.*

In the following, we show that after exhaustively applying the reduction rule, the number of non-dirty candidates is bounded by a quadratic function of d.

Theorem 2. *Each instance of* Kemeny score *with average KT-distance d can be reduced in polynomial time to an equivalent instance with at most $9d + 162 \cdot d^2$ candidates and with at most $x_d + 32x_d^2$ candidates with x_d denoting the number of dirty candidates.*

Proof. Consider an instance of Kemeny Score with average KT-distance d. According to Lemma 9, there are at most $9d/2$ dirty pairs and, thus, at most $9d$ dirty candidates. For every non-dirty candidate who is not deleted after exhaustively applying the reduction rule, there must be a dirty candidate c_d with $\text{dist}(c, c_d) \leq 2n_d \leq 9d$. Thus, for every dirty candidate there can be at most $2 \cdot 9d = 18d$ non-dirty candidates that are not deleted. Then, in total, there can be at most $9d \cdot 18d \leq 162d^2$ non-dirty candidates left. Thus, a reduced instance can consist of at most $9d + 162d^2$ candidates. □

4.2. Kemeny Tie Score

A practically relevant extension of Kemeny Score is Kemeny Tie Score [1, 17]. Here, one additionally allows the voters to classify sets of equally liked candidates, that is, a preference list is no longer defined as a permutation of the candidates, but for two (or more) candidates a, b one can have $a = b$. The

term $d_{v,w}(a,b)$ that denotes the contribution of the candidate pair $\{a,b\}$ to the KT-distance between two votes v and w is modified as follows [17]. One has $d_{v,w}(a,b) = 2$ if $a > b$ in v and $b > a$ in w, $d_{v,w}(a,b) = 0$ if a and b are ordered in the same way in v and w, and $d_{v,w}(a,b) = 1$, otherwise. In the literature there are different demands for the consensus itself. For example, Hemaspaandra et al. [17] allow that the consensus list can contain ties as well whereas Ailon [1] requires the consensus list to be a "full ranking", that is, a permutation of the candidates. We consider here the more general setting of Hemaspaandra et al. [17]. Note that Kemeny Tie Score does not only generalize Kemeny Score but also includes other interesting special cases such as p-ratings and top-m lists [1].

Regarding the complexity of Kemeny Tie Score, clearly all hardness results for Kemeny Score carry over. Regarding algorithmic results, this is only true for some of them. In particular, previous approaches [6, 27] only provide fixed-parameter tractability with respect to the average KT-distance for Kemeny Score. In contrast, the question of fixed-parameter tractability of Kemeny Tie Score with respect to the average KT-distance has been open so far. Here, we answer this question positively by showing that the new method for partial kernelization introduced in Section 2 also applies to Kemeny Tie Score.

To apply Step 1 of our framework, we extend the definition of dirtiness as given for Kemeny Score. For an instance with ties, we say $a =_{2/3} b$ if $a = b$ in more than $2n/3$ votes. Then, a pair of candidates a, b is dirty if neither $a >_{2/3} b$ nor $a =_{2/3} b$ nor $a <_{2/3} b$. We use $a \geq_{2/3} b$ to denote $(a >_{2/3} b) \vee (a =_{2/3} b)$.

Proposition 4. Kemeny Tie Score *without dirty pairs is solvable in polynomial time.*

Proof. The basic idea of the proof is the same as for Proposition 3: Show by contradiction that all pairs of candidates must be ordered according to their 2/3-majorities. For ties, this leads to an extensive case distinction for all possible orders of two candidates a and b.

Case I: $a >_{2/3} b$. **a)** Assume that $a = b$ in a preference list l. Then, we show that l cannot be a Kemeny consensus. Let T denote the set of candidates that are tied with a and b in l. If T is empty, then replacing $a = b$ by $a > b$ obviously gives a consensus with lower score. In the following, we describe how to order the candidates of T such that we have $a > b$ and the partial score with respect to candidate pairs inside $T \cup \{a,b\}$ is smaller than the partial score of l with respect to the same set of pairs.

Similarly to the proof of Proposition 3, we can show that there is no candidate $c \in C$ with $c \geq_{2/3} a$ and $c \leq_{2/3} b$. Hence, the candidates of T can be partitioned into three groups:

- $T_l := \{c \in T : c \geq_{2/3} a \text{ and } c >_{2/3} b\}$,
- $T_m := \{c \in T : a >_{2/3} c \text{ and } c >_{2/3} b\}$, and
- $T_r := \{c \in T : a >_{2/3} c \text{ and } b \geq_{2/3} c\}$.

For a subset $C' \subseteq C$ and a candidate $a \in C \backslash C'$, we write $a = C'$ if a is tied with all candidates in C'. We show that the partial score of $T_l = a > T_m > b = T_r$ (new) is smaller than the partial score of $a = b = T$ (old).

The considered partial scores can be decomposed such that they depend on the relative order between subsets of candidates. More precisely, for two subsets $C', C'' \subseteq C$, $C' \times C''$ denotes all pairs $\{c', c''\}$ of candidates with $c' \in C'$ and $c'' \in C''$. Then, the partial scores depend on

$$s(\{a\} \times T_l) + s(\{a\} \times (\{b\} \cup T_m \cup T_r)) + s(\{b\} \times T_r) + s(\{b\} \times (T_l \cup T_m)) \\ + s(T_l \times T_m) + s(T_l \times T_r) + s(T_m \times T_r)$$

and the relative order between the candidates within the T_m, T_l and T_r. Since in both considered orders the candidates within these subsets are tied, there is no different partial score for the corresponding pairs.

Now, we compare the "old" with the "new" partial score showing that the new partial score is smaller. All candidates of T_l are tied with a in the new and in the old consensus. All candidates of $\{b\} \cup T_m \cup T_r$ are tied with a in the old consensus and are beaten by a in the new consensus. Since for all these candidates we have that a is better in more than two third of the input votes, the score of the new consensus is smaller with respect to them.

All candidates of T_r are tied with b in the new and in the old consensus. All candidates of $T_l \cup T_m$ are tied with b in the old consensus and better than b in the new consensus. Since every such candidate is better than b in more than two third of the input votes, the score of the new consensus is smaller with respect to them.

It remains to consider the order between the candidates of the different subsets:

1. T_l, T_m: For every $t_l \in T_l$ and for every $t_m \in T_m$, we have that t_l is better than or equal to a in more than two thirds of the votes whereas a is strictly better than t_m in more than two thirds of the votes. It follows that t_l must be better than t_m in at least one third of the votes and, thus, in more than two thirds of the votes.
2. T_l, T_r: (analogous to 1.)
3. T_m, T_r: For every $t_r \in T_r$ and for every $t_m \in T_m$, we have that b is better than or equal to t_r in more than two thirds of the votes whereas t_l is strictly better than b in more than two thirds of the votes. It follows that t_l is better than t_r in more than two third of the votes and the new score is smaller.

Case Ib) $a >_{2/3} b$ and there is a consensus with $a < b$: This case can be excluded in analogy to Case Ia) with T containing all candidates c with $b \geq c \geq a$ in the consensus.

Case II) $a =_{2/3} b$: We show that a consensus with $a > b$ cannot have minimum score. Partition the set of candidates $C \backslash \{a, b\}$ into the following three subsets:

- $C_l := \{c \in C \backslash \{a,b\} : c >_{2/3} a \text{ and } c >_{2/3} b\}$,
- $C_m := \{c \in C \backslash \{a,b\} : c =_{2/3} a \text{ and } c =_{2/3} b\}$, and
- $C_r := \{c \in C \backslash \{a,b\} : c <_{2/3} a \text{ and } c <_{2/3} b\}$.

The remaining possibilities are "$d \geq_{2/3} a$ and $d <_{2/3} b$", "$d \leq_{2/3} b$ and $d >_{2/3} a$", or "$a >_{2/3} d$ and $d >_{2/3} b$", for $d \in C \setminus \{a,b\}$. All of them can be excluded by simple counting arguments. Let S denote the set of candidates between a and b in the consensus, that is, $S := \{s \in C : a \geq s \geq b\}$ in the consensus. We partition S into the three subsets, $S_l := C_l \cap S$, $S_m := C_m \cap S$, and $S_r := C_r \cap S$.

Now, we show that the following order gives a consensus with smaller score:

$$S_l > a = S_m = b > S_r.$$

Similarly to Case Ia) the partial score can be decomposed into $s(\{a\} \times \{b\}) + s(\{a,b\} \times S_r) + s(\{a,b\} \times S_m) + s(\{a,b\} \times S_l) + s(S_l \times S_m) + s(S_l \times S_r) + s(S_m \times S_r)$. Then, a simple calculation shows that the new score is smaller than the old one. □

For Kemeny Tie Score we can bound the number of candidates by a function only depending on the average KT-distance by proving lemmas analogous to Lemmas 9-12. In what follows, we only describe the differences.

The "tie-variant" of Lemma 10 says that there are at most $5 \cdot n_d$ (instead of n_d) non-dirty pairs that are not ordered according to their 2/3-majorities in a Kemeny consensus. The reason is that within the otherwise analogous proof we now use $s_l(D_P) - s_{l_{2/3}}(D_p) \geq -|D_p| \cdot 5/3 \cdot n$. The factor $5/3 \cdot n$ is due to the fact that the difference of the partial scores for two possible orders of a dirty pair is only bounded by $5/3 \cdot n$.

Next, it is crucial to adapt the distance function between two candidates appropriately. More precisely, for two candidates a, b with $a \geq_{2/3} b$, one defines

$$\text{dist}(a,b) := |\{c \in C : a \geq_{2/3} c \geq_{2/3} b \text{ and } c \text{ is non-dirty}\}|.$$

Then, Lemmas 11 and 12 can be directly transferred to the case with ties simply replacing n_d by $5n_d$ (due to the variant of Lemma 10). This results in the following reduction rule:

Reduction Rule. *Let c be a non-dirty candidate with $\text{dist}(c, c_d) > 10n_d$ for all $c_d \in D$. Let $C_l := \{c' \in C : c' >_{2/3} c\}$, $C_m := \{c' \in C : c' =_{2/3} c\}$, and $C_r := \{c' \in C : c >_{2/3} c'\}$. Delete c and reorder every vote such that $C_l > C_m > C_r$ and the order of the candidates within C_l, C_m, and C_r remains unchanged.*

Regarding the last step of our framework, it is known that Kemeny Tie Score is fixed-parameter tractable with respect to the number of candidates. More specifically, the reduced instances can be solved by a dynamic programming

algorithm with running time $2^m \cdot \text{poly}(n, m)$ [6]. Altogether, this leads to the following theorem.[6]

Theorem 3. 1. *Each instance of* Kemeny Tie Score *with average KT-distance d can be reduced in polynomial time to an equivalent instance with at most $O(d^2)$ candidates and at most $O(x_d^2)$ candidates for an instance with x_d dirty candidates.*

2. Kemeny Tie Score *is fixed-parameter tractable with respect to the average KT-distance d as well as with respect to the number of dirty candidates.*

5. Conclusion

In this work, we focussed on the efficient computation of "small" partial kernels. The corresponding fixed-parameter tractability results were derived in a straightforward way. It remains a task for future research to improve these brute-force algorithms (operating on the kernelized instances) by more sophisticated approaches. More specifically, in case of Kemeny Score one may employ fixed-parameter algorithms with respect to different parameterizations [6, 27] whereas in case of Consensus Clustering no non-trivial exact algorithm seems to be available so far.

In applications one can easily determine the average distance and the number of dirty elements of the considered median problem and then decide whether the developed fixed-parameter algorithm should replace the otherwise used algorithm. Other related parameterizations which can not be easily computed "in advance" refer to distance measures from the input objects to the solution. Among these, our results directly extend to the parameter "maximum distance of the input objects from the solution" since this parameter is an upper bound for the average distance. In contrast, the "average distance of the input objects from the solution" is clearly a lower bound for the "average distance between the input objects". Hence, it is an interesting open question to investigate the parameterized complexity with respect to this parameter.

Based on ongoing experimental evaluations, we are confident that the developed efficient data reduction rules and the ensuing fixed-parameter tractability results will prove practical usefulness in real-world applications.

References

[1] N. Ailon. Aggregation of partial rankings, p-ratings, and top-m lists. *Algorithmica*, 2008. Available electronically. 2, 19, 20

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM*, 55(5), 2008. Article 23 (October 2008), 27 pages. 2, 7, 15

[6] Recall that in contrast to Kemeny Score without ties, for Kemeny Tie Score the fixed-parameter tractability with respect to the average KT-distance has been open so far.

[3] A. Amir, Y. Aumann, G. Benson, A. Levy, O. Lipsky, E. Porat, S. Skiena, and U. Vishne. Pattern matching with address errors: rearrangement distances. *Journal of Computer and System Sciences*, 75(6):359–370, 2009. 4, 5, 6

[4] 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. 15

[5] M. Bertolacci and A. Wirth. Are approximation algorithms for consensus clustering worthwhile? In *Proc. 7th SDM*, pages 437–442. SIAM, 2007. 2, 7

[6] 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. 2, 16, 20, 23

[7] H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *Proc. 4th IWPEC*, volume 5917 of *Lecture Notes in Computer Science*, pages 17–37. Springer, 2009. 3

[8] P. Bonizzoni, G. D. Vedova, R. Dondi, and T. Jiang. On the approximation of correlation clustering and consensus clustering. *Journal of Computer and System Sciences*, 74(5):671–696, 2008. 2, 7

[9] V. Conitzer. Computing Slater rankings using similarities among candidates. In *Proc. 21st AAAI*, pages 613–619. AAAI Press, 2006. 2

[10] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *Proc. 21st UAI*, pages 145–152. AUAI Press, 2005. 2

[11] V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing Kemeny rankings. In *Proc. 21st AAAI*, pages 620–626. AAAI Press, 2006. 2, 15

[12] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999. 2, 3

[13] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proc. 10th WWW*, pages 613–622, 2001. 2, 15

[14] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006. 2, 3

[15] A. Goder and V. Filkov. Consensus clustering algorithms: Comparison and refinement. In *Proc. 10th ALENEX*, pages 109–117. SIAM, 2008. 2, 7

[16] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007. 3

[17] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349:382–391, 2005. 2, 15, 19, 20

[18] J. Kemeny. Mathematics without numbers. *Daedalus*, 88:571–591, 1959. 14

[19] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proc. 39th STOC*, pages 95–103. ACM, 2007. 15

[20] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison Wesley, 2006. 15

[21] M. Křivánek and J. Morávek. NP-hard problems in hierarchical-tree clustering. *Acta Informatica*, 23(3):311–323, 1986. 7

[22] D. Marx. Closest substring problems with small distances. *SIAM Journal on Computing*, 38(4):1382–1410, 2008. 2

[23] S. Monti, P. Tamayo, J. P. Mesirov, and T. R. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1–2):91–118, 2003. 2, 7

[24] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. 2, 3

[25] V. Y. Popov. Multiple genome rearrangement by swaps and by element duplications. *Theoretical Computer Science*, 385(1-3):115–126, 2007. 3, 4

[26] F. Schalekamp and A. van Zuylen. Rank aggregation: together we're strong. In *Proc. 11th ALENEX*, pages 38–51. SIAM, 2009. 15

[27] N. Simjour. Improved parameterized algorithms for the Kemeny aggregation problem. In *Proc. 4th IWPEC*, volume 5917 of *LNCS*, pages 312–323. Springer, 2009. 2, 16, 20, 23

[28] Y. Wakabayashi. The complexity of computing medians of relations. *Resenhas*, 3(3):323–350, 1998. 7

[29] A. van Zuylen and D. P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research*, 34:594–620, 2009. 15