Proc. 9th LATIN, 2010

Average Parameterization and Partial Kernelization for Computing Medians

Nadja Betzler[1]*, Jiong Guo[2]**, Christian Komusiewicz[1]***, and Rolf Niedermeier[1]

[1] Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany
{nadja.betzler,c.komus,rolf.niedermeier}@uni-jena.de
[2] Universität des Saarlandes, Campus E 1.4, D-66123 Saarbrücken, Germany
jguo@mmci.uni-saarland.de

Abstract. We propose an effective polynomial-time preprocessing strategy for intractable median problems. Developing a new methodological framework, we show that if the input instances 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 new concept of "partial kernelization" and identify interesting polynomial-time solvable special cases for the considered problems.

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 in general 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 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 [10, 12, 18], 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 [5, 20] by using exponential-time dynamic

* Supported by the DFG, research project PAWS, NI 369/10.
** Supported by the Excellence Cluster on Multimodal Computing and Interaction (MMCI). Main work was done while the author was with Friedrich-Schiller-Universität Jena.
*** Supported by a PhD fellowship of the Carl-Zeiss-Stiftung.

programming and search tree methods. This work complements these results by polynomial-time preprocessing through data reduction. Marx [16] 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 parameters relate to the input structure.

Let us briefly discuss the naturalness of average parameterization for two prominent median problems tackled in this paper. First, we show the fixed-parameter tractability of the NP-hard Consensus Clustering problem (see, e.g., [2, 6, 17]). 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 [4, 13, 17]. 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.[1] Second, we also deal with the computation of Kemeny rankings (also known as rank aggregation), an NP-hard problem from the area of voting (see, e.g., [1, 2, 9, 11, 14]). As Conitzer and Sandholm [8] 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 [7, 9] for practical studies in this direction). Studying an average parameterization with respect to the pairwise distance between input votes reflects the view on voting proposed by Conitzer and Sandholm [8]. 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 interesting *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 reduced such that its size only depends on the parameter value. This somewhat “weaker” concept of kernelization promises to be of wider practical use.

Due to the lack of space, most proofs are deferred to a full version of the paper.

[1] Indeed, 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 [13].

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, given a number of combinatorial objects (such as permutations, partitions etc.) over a base set U, 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 upper-bounded by a (typically polynomial) function only depending on the average distance between the given combinatorial objects.
Step 3. Show that the number of non-dirty elements from U can be upper-bounded by a (typically polynomial) function only depending on the number of dirty elements (and, thus, also the average distance). This is achieved by developing polynomial-time data reduction rules which shrink the number of non-dirty elements (and thus U), generating an equivalent problem instance of smaller size.
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 the parameter average distance. Note that a special feature of our framework is that in Step 3 we actually perform some sort of *partial kernelization*[2], a concept that may be of general interest. 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.[3] Herein, the combinatorial objects are permutations over the set $\{e_1, \ldots, e_m\}$, and the distance between two permutations is the *swap distance* defined as follows: A *swap* operation interchanges two elements of a permutation. For instance, 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 $d(\pi_1, \pi_2)$. Concerning notation, we follow the recent paper of Popov [19]. 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 $\sum_{i=1}^{n} d(\pi, \pi_i)$.

[2] The term "partial" refers to the fact that only the size of the base set is reduced, but not the number of input objects.
[3] We remark that the question of the NP-hardness of SMP seems unsettled, cf. [19].

Now, the *average swap distance* d for an input instance of SMP is defined as $d := \left(\sum_{i \neq j} d(\pi_i, \pi_j)\right)/(n \cdot (n-1))$. We present a first application of our framework using SMP as the concrete running example. After that, in the next two sections, we will provide our main results.

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 of permutations. 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, 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). 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 shows the polynomial-time solvability of the special case but also the correctness of a data reduction rule used in Step 3.

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

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

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.*

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. Record the number of the employed swap operations, which needs to be added to the distance of the median permutation of the resulting instance.

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

Theorem 1. Swap Median Permutation *is fixed-parameter tractable with respect to the parameter average swap distance.*

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. Due to its practical relevance, Consensus Clustering has been intensively studied in terms of the usefulness of various heuristics and accompanying experiments [4, 13]. 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 $\sum_{C_i \in \mathcal{C}} d(C, C_i)$.

Consensus Clustering finds applications for example in the field of bioinformatics. Bonizzoni et al. [6] showed that Consensus Clustering is APX-hard even if the input consists of only three partitions. So far, the best approximation factor achievable in polynomial time is 4/3 [2].

Following Goder and Filkov [13], 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 $\mathrm{co}(a, b)$ the number of partitions in $\mathcal{C}$ in which a and b are co-clustered and with $\mathrm{anti}(a, b)$ we denote the number of partitions in $\mathcal{C}$ in which a and b are anti-clustered. Define the *distance* $d(C_i, C_j)$ 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. Thus, our parameter d denoting the *average distance* of a given Consensus Clustering instance is defined as $d := \left(\sum_{C_i, C_j \in \mathcal{C}} d(C_i, C_j)\right) / \left(n \cdot (n-1)\right)$.

Our overall goal is to show that Consensus Clustering is fixed-parameter tractable with respect to the average parameter d. To this end, we follow the approach 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 $\mathrm{co}(a, b) \geq n/3$ and $\mathrm{anti}(a, b) \geq n/3$. Moreover, the predicate (ab) is true iff $\mathrm{co}(a, b) > 2n/3$, and the predicate $a \leftrightarrow b$ is true iff $\mathrm{anti}(a, b) > 2n/3$.*

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 $a \leftrightarrow b \wedge (bc) \Rightarrow a \leftrightarrow c$.*

Theorem 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 the claim. 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 $S_i^1 := \{x \in S_i \mid (ax)\}$ 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 upper-bounds 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.*

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*. To this end, we analyze the structure of an input instance. The idea is to find subsets of S that contain many non-dirty elements that are all pairwisely co-clustered in more than $2n/3$ input partitions. First, we partition the input base set S into two subsets S_1, which contains the non-dirty elements, and S_2, which contains the elements that appear in dirty pairs. In the following, we describe a partition of S_1 into equivalence classes according to the non-dirty pairs in S_1. Moreover, these equivalence classes also induce a partition of S_2. First we describe the partition $P_1 = \{S_1^1, \ldots, S_1^l\}$ of S_1. 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 subsets S_2^i of S_2. Each S_2^i is defined as the set of elements $a \in S_2$ that have at least one element $b \in S_1^i$ such that (ab) holds. 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 $S^0 = S_2^0$. The following lemma shows that P is indeed a partition of S, and also gives some further structural property of P.

Lemma 7. *Let $P = \{S^0, S^1, \ldots, S^l\}$ be a 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*
- $\forall a, b \in S^i, 1 \leq i \leq l : (ab) \vee a\#b$.

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^i_1 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^i_1 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 partition of S according to Lemma 7. If there is some $S^i \in P$ such that $|S^i_1| > |\mathrm{dp}(S^i)| + |\mathrm{dp}(S^i, S \setminus S^i)|$, 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 above reduction rule is correct.*

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 3. Consensus Clustering *is fixed-parameter tractable with respect to the average distance d between the input partitions. Each instance of* Consensus Clustering *can be reduced in polynomial time to an equivalent instance with less than $9d$ elements in the base set.*

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 [11]. We first consider the NP-hard Kemeny Score problem and, second, the somewhat harder to attack generalization Kemeny Tie Score.

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 $\text{dist}(v, w) = \sum_{\{a,b\} \subseteq C} d_{v,w}(a, b)$, 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. The *score* of a preference list l with respect to an election (V, C) is defined as $\sum_{v \in V} \text{dist}(l, v)$. A preference list l with the minimum score is called a *Kemeny consensus* of (V, C) and its score $\sum_{v \in V} \text{dist}(l, v)$ 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 $\sum_{v \in V} \text{dist}(l, v)$.

Kemeny Score is NP-complete even when restricted to instances with only four votes [11]. The Kemeny score can be approximated to a factor of 8/5 by a deterministic algorithm [21] and to a factor of 11/7 by a randomized algorithm [2]. Recently, a polynomial-time approximation scheme (PTAS) for Kemeny Score has been developed [15]. However, its running time is impractical.

For an election (V, C), the average KT-distance d, the parameter of this work, is defined as $d := \left(\sum_{u,v \in V, u \neq v} \text{dist}(u, v)\right)/(n(n-1))$. The Kemeny Score problem is known to be in FPT with respect to the parameter d [5, 20]. There is a branching algorithm for Kemeny Score which runs in $(5.823)^d \cdot \text{poly}(n, m)$ time [20]. 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. 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. Further, 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. To show the following 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 $s_l(P) := \sum_{p \in P} s_l(p)$.

Theorem 4. 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, $P := \{\{c, c'\} \mid c > c' \text{ in } l \text{ and } c' >_{2/3} c\}$. 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 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 upper-bounded 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 at most $9d/2$ dirty pairs.*

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 can be influenced by no 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.*

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 (V, C) and a non-dirty pair $\{c, c'\}$, define $\operatorname{dist}(c, c') := |\{b \in C : b \text{ is non-dirty and } c >_{2/3} b >_{2/3} c'\}|$ if $c >_{2/3} c'$ and $\operatorname{dist}(c, c') := |\{b \in C : b \text{ is non-dirty and } c' >_{2/3} b >_{2/3} c\}|$ if $c' >_{2/3} c$.

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

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 $\mathrm{dist}(c, c_d) > 2n_d$ 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.*

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. Let c be a non-dirty candidate. If for all $c_d \in D$ one has $\mathrm{dist}(c, c_d) > 2n_d$, then delete c and record the partial score with respect to all candidate pairs that contain c and are ordered according to the 2/3-majority. This score will be added to the Kemeny score of the resulting instance.

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

Theorem 5. *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.*

Kemeny Tie Score. A practically relevant extension of Kemeny Score is Kemeny Tie Score [1, 14]. 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 $d_{vw}(a, b)$ that denotes the contribution of the candidate pair $\{a, b\}$ to the KT-distance between two votes v and w is modified as follows [14]. 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. Note that in the literature there are different demands for the consensus itself. For example, Hemaspaandra et al. [14] 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 used in [14]. Further, note that Kemeny Tie Score not only generalizes Kemeny Score but also includes other interesting special cases like p-ratings and top-m lists [1].

Previous approaches [5, 20] 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 can answer this question positively by showing that the new method for partial kernelization introduced in Section 2 also applies to Kemeny Tie Score. This indicates that the new method can be more powerful than the dynamic programming approach in [5].

For an instance with ties, we say $a =_{2/3} b$ if $a = b$ in more than $2n/3$ votes. Then, the concept of dirtiness is adapted such that a pair of candidates a, b is dirty if neither $a >_{2/3} b$ nor $a =_{2/3} b$ nor $a <_{2/3} b$. Further, we use $a \geq_{2/3} b$ to denote $(a >_{2/3} b) \vee (a =_{2/3} b)$.

Theorem 6. Kemeny Tie Score *without dirty pairs is solvable in polynomial time.*

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

Moreover, due to the definition of the KT-distance for the case with ties, the bound on the number of non-dirty candidates is twice as high as in the case without ties.

Theorem 7. Kemeny Tie Score *is fixed-parameter tractable with respect to the average KT-distance d. 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.*

5 Conclusion

In applications one can easily determine the average distance parameter 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 computed "in advance" refer to distance measures from the input rankings to the solution. Among these, our results directly extend to the parameter "maximum distance of the input rankings from the solution" since this parameter is an upper bound for the average distance. In contrast, the "average distance of the input rankings from the solution" is clearly a lower bound for the "average distance between the input rankings". Hence, it is an interesting open question to investigate the parameterized complexity with respect to this parameter.

References

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

[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.

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

[5] 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.

[6] 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.

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

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

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

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

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

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

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

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

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

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

[17] 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.

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

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

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

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