Acta Informatica
https://doi.org/10.1007/s00236-024-00463-x

ORIGINAL ARTICLE

Parameterized aspects of distinct Kemeny rank aggregation

Koustav De[1] · Harshil Mittal[2] · Palash Dey[1] · Neeldhara Misra[2]

Received: 8 February 2024 / Accepted: 6 August 2024

Abstract

The Kemeny method is one of the popular tools for rank aggregation. However, computing an optimal Kemeny ranking is NP-hard. Consequently, the computational task of finding a Kemeny ranking has been studied under the lens of parameterized complexity with respect to many parameters. We study the parameterized complexity of the problem of computing all distinct Kemeny rankings. We consider the target Kemeny score, number of candidates, average distance of input rankings, maximum range of any candidate, and unanimity width as our parameters. For all these parameters, we already have FPT algorithms. We find that any desirable number of Kemeny rankings can also be found without substantial increase in running time. We also present FPT approximation algorithms for Kemeny rank aggregation with respect to these parameters.

1 Introduction

Aggregating individual ranking over a set of alternatives into one societal ranking is a fundamental problem in social choice theory in particular and artificial intelligence in general. Immediate examples of such applications include aggregating the output of various search engines [1], recommender systems [2], etc. The Kemeny rank aggregation method is often the method of choice in such applications due to its many desirable properties like Condorcet consistency that is electing the Condorcet winner (if it exists), etc. A Condorcet winner is a candidate who defeats every other candidate in pairwise election. The Kemeny method outputs a ranking R with minimum sum of dissatisfaction of individual voters known as *Kemeny score* of R; the dissatisfaction of a voter with ranking Q with respect to R is quantified as the number of pairs of candidates that Q and R order differently [3]. This quantity is also

✉ Koustav De
koustavde7@kgpian.iitkgp.ac.in

✉ Harshil Mittal
mittal_harshil@iitgn.ac.in

Palash Dey
palash.dey@cse.iitkgp.ac.in

Neeldhara Misra
neeldhara.m@iitgn.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal, 721302, India

[2] Department of Computer Science and Engineering, Indian Institute of Technology Gandhinagar, Gandhinagar, Gujarat, 382424, India

Published online: 15 August 2024

called the Kendall-Tau distance between Q and R. A ranking with minimum Kemeny score is called the Kemeny ranking.

The computational question of finding optimal Kemeny rankings is intractable in very restricted settings (for instance, even with a constant number of voters). Therefore, it has been well-studied from both approximation and parameterized perspectives. A problem is said to be *fixed-parameter tractable* or FPT with respect to a parameter k if it admits an algorithm whose running time can be described as $f(k) \cdot n^{O(1)}$ where the input size is n, implying that the algorithm is efficient for instances where the parameter is "small" [4]. For the Kemeny rank aggregation problem, the following parameters (among others) have enjoyed attention in the literature:

- *Range.* The range of a candidate in a profile is the difference between its positions in the votes which rank him/her the lowest and the highest [5]. The maximum and average range of a profile is defined as, respectively, the maximum and average ranges of individual candidates. Profiles which are "homogeneous", i.e. where most candidates are viewed somewhat similarly by the voters, are likely to have low values for range, while a single polarizing candidate can skew the max range parameter considerably.
- *KT-distance.* The average (respectively, maximum) KT distance is the average (respectively, maximum) of the Kendall-Tau distances between all pairs of votes [5]. Recall that the KT distance between a pair of rankings is the number of pairs that are ordered *differently* by the two rankings under consideration.

A pair of candidates are said to be *unanimous* with respect to a voting profile if all votes rank them in the same relative order. Consider the following "unanimity graph" associated with a profile P and defined as follows: every candidate is represented by a vertex, and there is an edge between a pair of candidates if and only if they are unanimous with respect to the profile. We use G_P to denote this graph. Note that the structure of the *complement* of this graph, denoted $\overline{G_P}$, carries information about candidates about whom the voters are not unanimous in their opinion. In particular, for every pair of candidates a and b that have an edge between them in the complement of the unanimity graph, there is at least one voter who prefers a over b and at least one who prefers b over a. Thus every edge signals a lack of consensus, and one could think of the number of edges in this graph as a measure of the distance of the profile from an "automatic consensus", which is one that can be derived from the information about unanimous pairs alone. Motivated by this view, we consider also the following structural parameter:

Unanimity width. For an input voting profile P, we define a graph G_P, called "unanimity graph" or the comparability graph, on the set of candidates where we have an edge from i to j if and only if every ranking in P puts i before j; we denote its complement by $\overline{G_P}$ and call it co-comparability graph of P. We call the pathwidth of $\overline{G_P}$ the unanimity width of P [6] (refer to Sect. 2 for the formal definition of pathwidth).

Our contribution concerns enumerating optimal Kemeny rankings. In recent times, there is considerable research interest in finding a set of diverse optimal or near-optimal solutions of an optimization problem. The two natural parameters, e.g. the number r of distinct solutions in the set and the minimum required distance s (scatteredness parameter) between any two solutions in the set quantifies what the term "diverse" means for a set of solutions. Indeed, it is often difficult to encode all aspects of a complex system into a neat computational problem. In such scenarios, having a diverse set of optimal solutions for a problem Γ allows the user to pick a solution which meets other aspects which are not captured in Γ. In the context of rank aggregation, such other external constraints may include gender fairness, demographic balance, etc. For the Kemeny rank aggregation method, Arrighi et al. [7] present a parameterized algorithm to output a set of diverse Kemeny rankings with respect to unanimity width as the parameter.

However, note that external requirements are often independent of the constraints in the optimization problem, and consequently they may not be correlated with diversity based on distance parameters. In particular, for useful externalities like gender fairness or geographic balance—these features of the candidates may not have any relation with their position in the voters' rankings, and therefore, diversity *between* solutions may not imply diversity *within* any of the solutions. This becomes particularly stark when most near-optimal rankings do not meet the external requirements. Indeed, there is a substantial literature [8, 9] that considers the problem of accounting for these requirements explicitly, and studies trade-offs between optimality of solutions and the degree to which demands of diversity can be met.

In this paper, we shift our focus from finding diverse solutions to finding as many *distinct* solutions as possible. Enumerating solutions is a fundamental goal for any optimization problem. The literature on counting optimal Kemeny rankings is arguably limited considering that even finding one is hard in very restricted settings, and that instances could have exponentially many rankings — which would be too expensive to enumerate. Indeed, consider a profile that consists of two votes over m candidates, where one vote ranks the candidates in lexicographic order and the other ranks the candidates in reverse lexicographic order. For this instance, every ranking is an optimal ranking. However, note that real world preferences often have additional structure: for example, profiles with an odd number of votes that are single-peaked [10] or single-crossing [10] have unique optimal solutions. To address scenarios where the number of optimal solutions is large, we allow the user to specify the number r of optimal solutions that she wants the algorithm to output. In our problem called DISTINCT OPT KEMENY RANKING AGGREGATION, the input is a set of rankings over a set of candidates and an integer r, and we need to output max$\{r$, number of optimal solutions$\}$ Kemeny rankings.

1.1 Our contributions

Algorithms for Distinct Kemeny Rank Aggregation The first parameter that we consider is the optimal Kemeny score k, also called the *standard parameter*. Many applications of rank aggregation, for example, faculty hiring, etc. exhibit correlation among the individual rankings — everyone in the committee may tend to prefer some candidate with strong academic background than some other candidate with weak track record. In such applications, the optimal Kemeny score k, average Kendall-Tau distance d (a.k.a. Bubble sort distance) among input rankings, maximum range of the positions of any candidate r_{max}, and unanimity width w will be small, and an FPT algorithm becomes useful. We show that there is an algorithm for DISTINCT OPT KEMENY RANKING AGGREGATION running in time $\mathcal{O}^*\left(2^k\right)$ [Theorem 1]. We next consider the number of candidates, m as the parameter and present an algorithm running in time $\mathcal{O}^*\left(2^m r^{\mathcal{O}(1)}\right)$ [Theorem 2] where r is the required number of solutions. For d and r_{max}, we present algorithms with running time $\mathcal{O}^*(16^d \cdot r)$ and $\mathcal{O}^*\left(32^{r_{max}} \cdot r\right)$ [Theorem 3 and 4] respectively. Our last parameter is the unanimity width w which is the pathwidth of the co-comparability graph of the unanimity order and we present an algorithm running in time $\mathcal{O}^*\left(2^{\mathcal{O}(w)} \cdot r\right)$ [Theorem 5].

Some instances may have a few optimal solutions, but have many close-to-optimal solutions. To address such cases, we study the DISTINCT APPROXIMATE KEMENY RANKING AGGREGATION problem where the user gives a real number $\lambda \geq 1$ as input and looks for max$\{r$, number of optimal solutions$\}$ rankings with Kemeny score at most λ times the optimal Kemeny score. For this problem, we design algorithms with running time $\mathcal{O}^*\left(2^{\lambda k}\right)$ [Corollary 1], $\mathcal{O}^*\left(2^m r^{\mathcal{O}(1)}\right)$ [Corollary 2] and $\mathcal{O}^*\left(16^{\lambda d} \cdot r\right)$ [Corollary 6].

We observe that the running time of all our algorithms are comparable with the respective parameterized algorithms for the problem of finding one Kemeny ranking. We note that this phenomenon is in sharp contrast with the diverse version of Kemeny rank aggregation where we have an FPT algorithm only for unanimity width as the parameter. To begin with, the algorithm, as presented in [6], cannot be used to find only optimal solutions; it can find only approximately optimal solutions. However, one can set the parameters of the algorithm in [6] to find all λ approximate rankings in time $\mathcal{O}^*((w!(\lambda-1)\mathrm{OPT})^{\mathcal{O}(m!)})$ where OPT is the optimal Kemeny score of the input rankings.

1.2 Related work

Kemeny rule [3] shows us its most significant and popular mechanism for ranking aggregation. However, Bartholdi et al. [11] have established that KEMENY SCORE, Kemeny ranking and Kemeny winner are NP-hard. The Kemeny ranking aggregation approach is NP-hard, even if we have only four input rankings to aggregate [1]. Fixed-parameter algorithms for Kemeny voting rule have been proved to be an effective and important area for research by Betzler et al. [5] considering structural parameterizations such as "number of candidates", "solution size i.e. Kemeny Score", "average pairwise distance", "maximum range", "average range" of candidates in an election. A multi-parametric algorithm for DIVERSE KEMENY RANK AGGREGATION over partially ordered votes has been studied in [6]. Arrighi et al. [7] developed a FPT algorithm parameterized by unanimity width for the Kemeny rank aggregation method to output a set of diverse Kemeny rankings. A small error in the construction proof from [1] has been rectified by Biedl et al. [12] and they have established the approximation factor of $2-2/k$, improving from the previous approximation factor of 2.

Further classification in more exact manner of the classical computational complexity of Kemeny elections has been provided by Hemaspaandra et al. [13]. With respect to the practical relevance of the computational hardness of the KEMENY SCORE, polynomial-time approximation algorithms have been developed where a factor of $8/5$ is seen in [14] and a factor of $11/7$ is proved in [15]. Kenyon-Mathieu and Schudy [16] proposed a polynomial-time approximation scheme (PTAS) for finding a Kemeny ranking. However, their algorithm is not very useful in practice. There are quite a few works which develop practical heuristics for this problem [17–19].

Polynomial time algorithms producing good solutions for rank aggregation rule is a consequence of thorough computational studies [16, 21]. Cornaz et al. [10] have established polynomial time computability of the single-peaked and single-crossing widths and have proposed new fixed-parameter tractability results for the computation of an optimal ranking according to the Kemeny rule by following the results of Guo te al. [21]. In social choice theory [11, 22], the ideas related to diverse sets of solutions have found tremendous applicability. The study in [23] introduced the (j,k)-Kemeny rule which is a generalization of Kemeny's voting rule that aggregates ballots containing weak orders with j indifference classes into a weak order with k indifference classes. In social choice theory, different values of j and k yield various rules of the interest of the community turning up as special cases. The minimum Kendall-Tau distance between pairs of solutions has a nice analogy with min Hamming distance over all pairs of solutions as shown in [24, 25].

2 Preliminaries

For an integer ℓ, we denote the set $\{1,\ldots,\ell\}$ by $[\ell]$ and $[\ell]_0 = [\ell] \cup \{0\}$. For two integers a, b, we denote the set $\{i \in \mathbb{N} : a \le i \le b\}$ by $[a, b]$. Given two integer tuples $(x_1,\ldots,x_\ell), (y_1,\ldots,y_\ell) \in \mathbb{N}^\ell$, we say $(x_1,\ldots,x_\ell) >_{\text{lex}} (y_1,\ldots,y_\ell)$ if there exists an integer $i \in [\ell]$ such that we have (i) $x_j = y_j$ for every $j \in [i-1]$, and (ii) $x_i > y_i$.

Let $\mathcal{C}$ be a set of candidates and $\Pi = \{\pi_1,\ldots,\pi_n\}$ a multi-set of n rankings (complete orders) on $\mathcal{C}$. For a ranking π and a candidate c let us define $\text{pos}_\pi(c)$ to be $|\left\{c' \in \mathcal{C} : c' \succ_\pi c\right\}|$. We precisely define the *range* $r(c)$ in a set of rankings Π to be $\max_{\pi_i,\pi_j\in\Pi}\left\{|\text{pos}_{\pi_i}(c) - \text{pos}_{\pi_j}(c)|\right\} + 1$. We denote the set of all complete orders over $\mathcal{C}$ by $\mathcal{L}(\mathcal{C})$. The Kemeny score of a ranking $Q \in \mathcal{L}(\mathcal{C})$ with respect to Π is

$$\text{Kemeny}_\Pi(Q) = \sum_{i=1}^{n} \text{d}_{\text{KT}}(Q, \pi_i)$$

where $\text{d}_{\text{KT}}(\cdot,\cdot)$ is the Kendall-Tau distance – the number of pairs of candidates whom the linear orders order differently – between two linear orders, and $N_\Pi(x \succ y)$ is the number of linear orders in Π where x is preferred over y. A Kemeny ranking of Π is a ranking Q which has the minimum $\text{Kemeny}_\Pi(Q)$; the score $\text{Kemeny}_\Pi(Q)$ is called the optimal Kemeny score of Π.

We now define our problems formally. For a set of rankings Π, we denote the set of (optimal) Kemeny rankings and rankings with Kemeny score at most some integer k for Π respectively by $K(\Pi)$ and $K(\Pi, k)$, and the minimum Kemeny score by $k_{\text{OPT}}(\Pi)$.

Definition 1 (Distinct OPT Kemeny Ranking Aggregation). Given a set of rankings (complete orders) Π over a set of candidates $\mathcal{C}$ and integer r, compute $\ell = \min\{r, |K(\Pi)|\}$ distinct Kemeny rankings $\pi_1,\ldots,\pi_\ell$. We denote an arbitrary instance of it by $(\mathcal{C}, \Pi, r)$.

For a set of rankings Π over a set of candidates $\mathcal{C}$, we say that a complete order π respects unanimity order if we have $x \succ_\pi y$ whenever $x \succ y$ for all $\succ \in \Pi$.

Definition 2 (Distinct approximate Kemeny Ranking Aggregation). Given a set of ranking (complete order) Π over a set of candidates $\mathcal{C}$, an approximation factor $\lambda \ge 1$, and integer r, compute $\ell = \min\{r, |K(\Pi, \lambda \cdot k_{\text{OPT}}(\Pi))|\}$ distinct rankings $\pi_1,\ldots,\pi_\ell$ such that each ranking $\pi_i, i \in [\ell]$ respects unanimity order with respect to Π and the Kemeny score of each ranking $\pi_i, i \in [\ell]$ is at most $\lambda \cdot k_{\text{OPT}}(\Pi)$. We denote an arbitrary instance of it by $(\mathcal{C}, \Pi, \lambda, r)$.

Definition 3 (Distinct Kemeny Ranking Aggregation). Given a list of partial votes Π over a set of candidates $\mathcal{C}$, and integers k and r, compute $\ell = \min\{r, |K(\Pi, k)|\}$ distinct rankings $\pi_1,\ldots,\pi_\ell$ such that the Kemeny score for each ranking π_i is at most k and each $\pi_i, i \in [\ell]$ respects unanimity order. We denote an arbitrary instance of it by $(\mathcal{C}, \Pi, k, r)$.

We use $\mathcal{O}^*(\cdot)$ to hide polynomial factors. That is, we denote $\mathcal{O}(f(k)\text{poly}(n))$ as $\mathcal{O}^*(f(k))$ where n is the input size.

We define a path decomposition of a graph $G = (V, E)$ by a tuple $\mathcal{P} = (\mathcal{B}_i)_{i\in[t]}$ where each bag $\mathcal{B}_i \subseteq V$, t is the number of bags in $\mathcal{P}$ and $\mathcal{P}$ satisfies the following additional constraints: (1) $\bigcup_{i\in[t]} \mathcal{B}_i = V$, (2) $\exists i \in [t]$ such that $u, v \in \mathcal{B}_i$ for each $(u, v) \in E$ and (3) $\mathcal{B}_i \cap \mathcal{B}_k \subseteq \mathcal{B}_j$ for each $i, j, k \in [t]$ satisfying $i < j < k$. The width of $\mathcal{P}$ denoted by $w(\mathcal{P})$ is defined as $\max_{i\in[t]} |\mathcal{B}_i| - 1$. The *pathwidth* of G is denoted by $pw(G)$ which is defined as the minimum width of a path decomposition of G.

3 Algorithms for Distinct Kemeny Ranking Aggregation

We start with an easy Turing reduction from Distinct OPT Kemeny Ranking Aggregation to Distinct Kemeny Ranking Aggregation.

Observation 1 *Suppose there exists an algorithm for* Distinct Kemeny Ranking Aggregation *running in time $\mathcal{O}(f(m,n))$ where m is the number of candidates and n is the number of input votes. Then there exists an algorithm for* Distinct OPT Kemeny Ranking Aggregation *running in time $\mathcal{O}(f(m,n)\log(mn))$.*

Proof We note that the optimal Kemeny score belongs to the set $\{0, 1, \ldots, n\binom{m}{2}\}$. To solve Distinct OPT Kemeny Ranking Aggregation, we perform a binary search in the range from 0 to $n\binom{m}{2}$ to find the smallest k such that the algorithm for Distinct Kemeny Ranking Aggregation returns at least one ranking. □

We now present a bounded search based FPT algorithm for Distinct Kemeny Ranking Aggregation parameterized by the optimal Kemeny score. Hence, we also have an FPT algorithm for Distinct OPT Kemeny Ranking Aggregation parameterized by the optimal Kemeny score.

Theorem 1 *Let k be the Kemeny score of a Kemeny ranking. There is an* FPT *algorithm for* Distinct Kemeny Ranking Aggregation *parameterized by k which runs in time $\mathcal{O}^*\left(2^k\right)$. Hence, we have an* FPT *algorithm for* Distinct OPT Kemeny Ranking Aggregation *parameterized by k_{OPT} which runs in time $\mathcal{O}^*\left(2^{k_{OPT}}\right)$.*

Proof Due to observation 1, it is enough to present an algorithm for Distinct Kemeny Ranking Aggregation. We design an algorithm for a more general problem Distinct Kemeny Ranking Aggregation′ where every output ranking needs to respect the relative order of some set of pair of candidates given as input. If the set of pairs of candidates is empty, then the new problem is the same as Distinct Kemeny Ranking Aggregation.

Let $(\mathcal{C}, \Pi, k, r)$ be an arbitrary instance of Distinct Kemeny Ranking Aggregation. We define $\mathcal{X} = \{a \succ b : a, b \in \mathcal{C},$ every ranking in Π prefers a over $b\}$ to be the unanimity order of Π. We find a solution of Distinct Kemeny Ranking Aggregation′ instance $(\mathcal{C}, \Pi, k, r, \mathcal{X})$. We now design a bounded search based algorithm. We maintain a set $\mathcal{S}$ of solutions, which is initialized to the empty set. If every pair of candidates belong to $\mathcal{X}$ and $k \geq 0$, then we put the ranking induced by $\mathcal{X}$ in $\mathcal{S}$. If $k < 0$, then we discard this branch. Otherwise, we pick a pair (a, b) of candidates not present in $\mathcal{X}$, solve $(\mathcal{C}, \Pi, k - |\{\pi \in \Pi : b \succ a \text{ in } \pi\}|, r,$ transitive closure of $\mathcal{X} \cup \{a \succ b\})$ and $(\mathcal{C}, \Pi, k - |\{\pi \in \Pi : a \succ b \text{ in } \pi\}|, r,$ transitive closure of $\mathcal{X} \cup \{b \succ a\})$ recursively, and put solutions found in $\mathcal{S}$. We note that, since (a, b) is not a unanimous order of Π, the target Kemeny score k decreases by at least one on both the branches of the search tree. Hence, the height of the search tree is at most k. Thus, the number of leaves and nodes in the search tree are at most respectively 2^k and $2 \cdot 2^k$. After the search terminates, we output $\min\{r, |\mathcal{S}|\}$ rankings from $\mathcal{S}$. If $\mathcal{S}$ remains empty set, report that there is no ranking whose Kemeny score is at most k. The computation at each node of the search tree (except the recursive calls) clearly takes a polynomial amount of time. Hence, the runtime of our algorithm is $\mathcal{O}^*\left(2^k\right)$. The correctness of our algorithm follows from the observation that every ranking R whose Kemeny score is at most k, appears in a leaf node of the search tree of our algorithm. □

Running the algorithm in Theorem 1 with target Kemeny score λk where k is the optimal Kemeny score gives us the following result.

Corollary 1 *There is an algorithm for* Distinct approximate Kemeny Ranking Aggregation *running in time* $\mathcal{O}^*\left(2^{\lambda k}\right)$ *parameterized by both* λ *and* k.

We now consider the number of candidates as parameter and present a dynamic programming based FPT algorithm for Distinct Kemeny Ranking Aggregation.

Theorem 2 *There is an algorithm for* Distinct Kemeny Ranking Aggregation *which runs in time* $\mathcal{O}^*\left(2^m r^{\mathcal{O}(1)}\right)$. *In particular,* Distinct Kemeny Ranking Aggregation *and* Distinct OPT Kemeny Ranking Aggregation *are* FPT *parameterized by the number of candidates since the number* r *of output rankings can be at most* $m!$.

Proof Let $(\mathcal{C}, \Pi, k, r)$ be an arbitrary instance of Distinct Kemeny Ranking Aggregation. We maintain a dynamic programming table $\mathcal{T}$ indexed by the set of all possible non-empty subsets of $\mathcal{C}$. For a subset $\mathcal{S} \subseteq \mathcal{C}, \mathcal{S} \neq \emptyset$, the table entry $\mathcal{T}[\mathcal{S}]$ stores at most $\min\{r, |\mathcal{S}|!\}$ distinct rankings on $\mathcal{S}$ which have the least Kemeny score when the votes are restricted to $\mathcal{S}$. We initialize table $\mathcal{T}$ for the trivial cases like $\mathcal{T}[\mathcal{S}] = ()$ when $|\mathcal{S}| = 0$, $\mathcal{T}[\mathcal{S}] =$ (the element from $\mathcal{S}$) when $|\mathcal{S}| = 1$ and $\mathcal{T}[\mathcal{S}] = (x \succ y)$ when $\mathcal{S} = \{x, y\}$ and $x \succ y$ has the least Kemeny score when Π is restricted to $\{x, y\}$ or $\mathcal{T}[\mathcal{S}] = (x \succ y,\ y \succ x)$ when $\mathcal{S} = \{x, y\}$ and both $x \succ y$ and $y \succ x$ have the least Kemeny score when Π is restricted to $\{x, y\}$. To update the table entry $\mathcal{T}[\mathcal{S}]$ for $|\mathcal{S}| \geq 3$, we include to that entry $\min\{r, |\mathcal{S}|!\}$ rankings that have the least Kemeny score (when the votes are restricted to $\mathcal{S}$) among all rankings of the form $c > \pi$, where c is a candidate in $\mathcal{S}$ and π is a ranking stored in $\mathcal{T}[\mathcal{S} \setminus \{c\}]$. Updating each table entry takes at most $\mathcal{O}^\star(r^{\mathcal{O}(1)})$ time. As there are $2^m - 1$ table entries, the running time of our algorithm is at most $\mathcal{O}^\star\left(2^m r^{\mathcal{O}(1)}\right)$.

We now present the proof of correctness of our algorithm. Suppose we have $\mathcal{S} = \{c_1, ..., c_\ell\}$ and $c_1 \succ ... \succ c_\ell$ be a ranking in $\mathcal{T}[\mathcal{S}]$. Then $c_1 \succ ... \succ c_\ell$ is a Kemeny ranking if the votes in Π are restricted to $\mathcal{S}$. But then $c_2 \succ ... \succ c_\ell$ is a Kemeny ranking if votes are restricted to $\mathcal{S} \setminus \{c_1\}$. If not, then suppose $c_2' \succ ... \succ c_\ell'$ be a ranking with Kemeny score less than $c_2 \succ ... \succ c_\ell$. Then the Kemeny score of $c_1 \succ c_2' \succ ... \succ c_\ell'$ is less than the Kemeny score of $c_1 \succ c_2 \succ ... \succ c_\ell$ contradicting our assumption that $c_1 \succ ... \succ c_\ell$ is a Kemeny ranking when votes are restricted to $\mathcal{S}$. Hence, the update procedure of our dynamic programming algorithm is correct. □

Corollary 2 follows from the algorithm presented in the proof of Theorem 2.

Corollary 2 Distinct approximate Kemeny Ranking Aggregation *is FPT parameterized by the number of candidates* m.

Proof Consider an instance $(\mathcal{C}, \Pi, \lambda, r)$ of Distinct approximate Kemeny Ranking Aggregation. We run the algorithm proposed at Theorem 2 on the instances $(\mathcal{C}, \Pi, 0, 1), (\mathcal{C}, \Pi, 1, 1), \ldots$ of Distinct Kemeny Ranking Aggregation. We stop once we encounter an instance, say $(\mathcal{C}, \Pi, k^*, 1)$, for which the algorithm in Theorem 2 returns a Kemeny ranking with Kemeny score at most k^*. Note that k^* is the optimum Kemeny score for the election profile $(\mathcal{C}, \Pi)$. Next, we run the algorithm of Theorem 2 on the instance $(\mathcal{C}, \Pi, \lambda \cdot k^*, r)$ of Distinct Kemeny Ranking Aggregation to get the desired output. As $k^* \leq \binom{m}{2} \cdot |\Pi|$, the overall running time of the algorithm is at most $\mathcal{O}^*\left(2^m r^{\mathcal{O}(1)}\right)$. So, as $r \leq m!$, it follows that Distinct approximate Kemeny Ranking Aggregation is FPT parameterized by the number of candidates m. □

Our next parameter is the "average pairwise distance (Kendall-Tau distance)" d of the input rankings. We present a dynamic programming based FPT algorithm parameterized by d.

Theorem 3 *Let d be the average KT-distance of an election* $(\Pi, \mathcal{C})$. *There is an* FPT *for* Distinct OPT Kemeny Ranking Aggregation *parameterized by d which runs in time* $\mathcal{O}^\star\left(16^d \cdot r\right)$.

Proof Let $|\mathcal{C}| = m, \ |\Pi| = n$ and average position of a candidate $c \in \mathcal{C}$ in Π is defined as $p_{avg}(c) := \frac{1}{n} \cdot \sum_{v \in \Pi} v(c)$ where $v(c) := |\left\{c' \in \mathcal{C} : c' \succ c \text{ in } v \in \Pi\right\}|$. Formally for an election $(\Pi, \mathcal{C})$, $d := \frac{\sum_{v \in \Pi} \sum_{w \in \Pi} \mathrm{d}_{\mathrm{KT}}(v,w)}{n \cdot (n-1)}$. Betzler et al. [5] proved that Kemeny ranking can be found in time $O*(16^{\lceil d \rceil})$ and their proof holds for just d as well. Following the proof of both Lemma 6 and Lemma 7 from Betzler et al. [5], we have a set of candidates say $P_i := \left\{c \in \mathcal{C} \mid p_{avg}(c) - d < i < p_{avg}(c) + d\right\}$ for each position $i \in [m-1]_0$ in an optimal Kemeny Consensus and we know that $|P_i| \leqslant 4d \ \ \forall i \in [m-1]_0$. Our FPT dynamic programming algorithm is an extension of the algorithm presented in Fig. 4. of section 6.4 of [5].

Let the subset of candidates that are forgotten at latest at position i, be denoted by $F(i) := P_{i-1} \setminus P_i$ and the subset of candidates that are introduced for the first time at position i be denoted by $I(i) := P_i \setminus P_{i-1}$. We maintain a three dimensional dynamic programming table $\mathcal{T}$ indexed by $\forall i \in [m-1]_0$, $\forall\, c \in P_i$ and $\forall P_i' \subseteq P_i \setminus \{c\}$ of size at most $\mathcal{O}\left(16^d \cdot d \cdot m\right)$. We define the partial Kemeny Score $\text{pK-score}(c, \mathcal{R}) := \sum_{c' \in \mathcal{R}} \sum_{v \in \Pi} d_v^{\mathcal{R}}(c, c')$ where $d_v^{\mathcal{R}}(c, c') := 0$ if $c \succ_v c'$ and $d_v^{\mathcal{R}}(c, c') := 1$ otherwise and $\mathcal{R} \subseteq \mathcal{C}$. At each table entry $\mathcal{T}(i, c, P_i')$, we store a sequence of at most r number of partial Kemeny Scores sorted in non-decreasing order by considering and iterating over the entries in $\mathcal{T}(i-1, c', (P_i' \cup F(i)) \setminus \left\{c'\right\})$ $\forall c' \in P_i' \cup F(i)$ and we store the tuple

$$\left(\mathcal{T}(i-1, c', (P_i' \cup F(i)) \setminus \left\{c'\right\}) + \text{pK-score}(c, (P_i \cup \bigcup_{i<j<m} I(j)) \setminus (P_i' \cup \{c\}))\right)_{c' \in P_i' \cup F(i)}$$

in that table entry unlike storing only the minimum partial Kemeny Score at each table entry. K-score of an election is the Kemeny Score of an optimal Kemeny ranking. $\text{K-score}(\Pi, \mathcal{C}) = \sum_{i=0}^{m-2} \text{pK-score}(c_i, \mathcal{R}_i)$, where $\mathcal{R}_i$ is the set of candidates that follow c_i in the Kemeny ranking.

At each entry of the table candidate c takes position i and all of P_i' take position smaller than i. The initialization step is same as the algorithm presented in Fig. 4. of section 6.4 of [5] but the difference lies in the update step of that algorithm. Though we are storing Kemeny score in each table entry, we can enumerate Kemeny ranking(s) from them within asymptotic bound of our current run time by iteratively ordering the candidate(s) for which we get minimum partial Kemeny Score in a particular table entry. We output first r number of optimal Kemeny rankings whose K-scores are stored in the entry $T(m-1, c, P_{m-1} \setminus \{c\})$. Correctness of Lemma 8 of [5] ensures the correctness of our algorithm for generating at most r number of optimal Kemeny Rankings.

Updating each table entry takes time at most $r \cdot (4d + nm \log m)$ time. Hence, the overall runtime is bounded above by $\mathcal{O}^\star\left(16^d \cdot r\right)$. □

We next consider the “maximum range" r_{max} of candidate positions in the input rankings, as our parameter. We again present a dynamic programming based FPT algorithm parameterized by r_{max}.

Theorem 4 *Let r_{max} be the maximum candidate position range of an election* $(\Pi, \mathcal{C})$. *There exists an* FPT *dynamic programming algorithm for* Distinct OPT Kemeny Ranking Aggregation *parameterized by r_{max} which runs in time* $\mathcal{O}^*\left(32^{r_{max}} \cdot r\right)$.

Proof Following the proof of both Lemma 9 and Lemma 10 from [5], we have here certainly the size of the set of candidates P_i for every position i, $|P_i| \leqslant 6r_{max}$. We maintain a dynamic programming table $\mathcal{T}$ of size $\mathcal{O}\left(32^{r_{max}} \cdot r_{max} \cdot m\right)$ indexed by $\forall i \in [m-1]_0$, $\forall\, c \in P_i$ and $\forall P_i' \subseteq P_i \setminus \{c\}$. The proof of Theorem 4 follows immediately from a complete analogy to the proof of Theorem 3. □

Our final parameter is the unanimity width of the input rankings. We present a dynamic programming based FPT algorithm.

Theorem 5 Distinct OPT Kemeny Rank Aggregation *admits an* FPT *algorithm in the combined parameter unanimity width w and number of rankings r, which runs in time $\mathcal{O}^*\left(2^{\mathcal{O}(w)} \cdot r\right)$.*

Proof The problem of finding a Kemeny consensus is known to admit an FPT algorithm in the parameter w (Section 3, [7]). We adapt this algorithm to prove Theorem 5. Consider an instance $(\mathcal{C}, \Pi, r)$ of Distinct OPT Kemeny Ranking Aggregation. Let m denote the number of candidates in $\mathcal{C}$, and let n denote the number of voters in Π. For any candidates $a, b \in \mathcal{C}$, let $cost(a, b)$ denote the number of voters in Π who prefer b over a. Note that for any linear ordering π of candidates, $\text{Kemeny}_{\Pi}(\pi) = \sum_{a,b\in\mathcal{C}:a\succ b \text{ in } \pi} cost(a, b)$. Let ρ denote the unanimity order of Π. Let G_ρ denote the comparability graph of ρ. Using Lemma 3 of [6], let's construct a nice ρ-consistent path decomposition, say $\mathcal{P} = (B_1, \ldots, B_{2m})$, of $\overline{G_\rho}$ of width $w' \leq 5w+4$ in time $\mathcal{O}\left(2^{\mathcal{O}(w)} \cdot m\right)$. A path decomposition $\mathcal{P}' = (B_1, \ldots, B_z)$, is nice if $B_1 = B_z = \emptyset$ and either $B_{i+1} = B_i \cup \{v\}$; $v \notin B_i$ or $B_{i+1} = B_i \setminus \{w\}$; $w \in B_i$ for each $i \in [z-1]$. $\mathcal{P}$ is ρ-consistent if $\forall\,(x, y) \in \rho$, $\max\left(\{i \in [2m] | y \in B_i\}\right) \not< \min\left(\{i \in [2m] | x \in B_i\}\right)$. For each $1 \leq i \leq 2m$,

- Let $forg(i)$ denote the set of candidates that have been forgotten up to i^{th} bag. That is, $forg(i) = \left(B_1 \cup \ldots \cup B_{i-1}\right) \setminus B_i$.
- For each candidate $v \in B_i$, let $\mathcal{A}(i, v)$ denote the cost incurred by the virtue of placing all candidates of $forg(i)$ before v. That is, $\mathcal{A}(i, v) = \sum_{u\in forg(i)} cost(u, v)$.
- For each candidate $v \in B_i$ and each $T \subseteq B_i \setminus \{v\}$, let $\mathcal{B}(i, v, T)$ denote the cost incurred by the virtue of placing all candidates of T before v. That is, $\mathcal{B}(i, v, T) = \sum_{u\in T} cost(u, v)$.
- For each $T \subseteq B_i$, let $C(i, T)$ be a set that consists of first $min\left(r, |forg(i) \uplus T|!\right)$ orderings, along with their Kemeny scores, if all linear extensions of ρ on $forg(i) \uplus T$ were to be sorted in ascending order of their Kemeny scores. That is, $C(i, T)$ consists of the tuples $(\pi_1, k_1), (\pi_2, k_2), \ldots$, where $\pi_1, \pi_2, \ldots$ are the first $min\left(r, |forg(i) \uplus T|!\right)$ orderings in the sorted order, and $k_1, k_2, \ldots$ are their respective Kemeny scores.

Recall that every Kemeny consensus extends ρ (Lemma 1, [5]). So, if all linear extensions of ρ on $\mathcal{C}$ were to be sorted in ascending order of their Kemeny scores, then all Kemeny consensuses would appear in the beginning. Thus, Π has r distinct (optimal) Kemeny rankings if and only if $C(2m, \phi)$ contains r orderings of the same Kemeny score. Let's use DP to find all $\mathcal{A}(\cdot, \cdot)$'s, $\mathcal{B}(\cdot, \cdot, \cdot)$'s and $C(\cdot, \cdot)$'s as follows:

- First, let's compute and store $\mathcal{A}(i, \cdot)$'s in a table for $i = 1, \ldots, 2m$ (in that order) in time $\mathcal{O}\left(w' \cdot m \cdot \log(m \cdot n)\right)$ as follows: We set $\mathcal{A}(1, u) = 0$, where u denotes the candidate introduced by B_1. Now, consider $i \geq 2$ and a candidate $v \in B_i$. Let's describe how to find $\mathcal{A}(i, v)$.

  Introduce node: Suppose that B_i introduces a candidate, say x. Note that $forg(i) = forg(i-1)$. So, if $v \neq x$, we set $\mathcal{A}(i, v) = \mathcal{A}(i-1, v)$. Now, suppose that $v = x$. Let's show that $cost(u, x) = 0$ for all $u \in forg(i)$. Consider $u \in forg(i)$. In $\mathcal{P}$, u is

forgotten before x is introduced. So, $\{u, x\} \notin E(\overline{G_\rho})$. That is, u and x are comparable in ρ. Also, due to ρ-consistency of $\mathcal{P}$, we have $(x, u) \notin \rho$. Therefore, $(u, x) \in \rho$. That is, all voters in Π prefer u over x. So, $cost(u, x) = 0$. Thus, we set $\mathcal{A}(i, x) = 0$.
Forget node: Suppose that B_i forgets a candidate, say x. Note that $forg(i) = forg(i-1) \uplus \{x\}$. So, we set $\mathcal{A}(i, v) = \mathcal{A}(i-1, v) + cost(x, v)$.

- Next, let's compute and store all $\mathcal{B}(\cdot, \cdot, \cdot)$'s in a table in time $\mathcal{O}\big(w' \cdot 2^{w'} \cdot m \cdot \log(m \cdot n)\big)$ as follows: Consider $1 \leq i \leq 2m$ and $v \in B_i$. We have $\mathcal{B}(i, v, \phi) = 0$. Let's set $\mathcal{B}(i, v, T)$ for non-empty subsets $T \subseteq B_i \setminus \{v\}$ (in ascending order of their sizes) as $\mathcal{B}(i, v, T \setminus \{u\}) + cost(u, v)$, where u is an arbitrary candidate in T.
- Next, let's compute and store $C(i, \cdot)$'s in a table in time $\mathcal{O}\big(w' \cdot 2^{w'} \cdot m^2 \cdot r \cdot \log(m \cdot n \cdot r)\big)$ for $i = 1, \ldots, 2m$ (in that order) as follows: We set $C(1, \phi) = \{(, 0)\}$ and $C(1, \{u\}) = \{(u, 0)\}$, where u denotes the candidate introduced by B_1. Now, consider $i \geq 2$. Let's describe how to find $C(i, \cdot)$'s.
Introduce node: Suppose that B_i introduces a candidate, say x. For each $T \subseteq B_i$ that does not contain x, we set $C(i, T) = C(i-1, T)$. Now, let's find $C(i, T)$ for all subsets $T \subseteq B_i$ that contain x (in ascending order of their sizes) as follows: First, let's consider $T = \{x\}$. Recall that $(u, x) \in \rho$ for all $u \in forg(i)$. So, x is the last candidate in all linear extensions of ρ on $forg(i) \uplus \{x\}$. Also, in any such ordering, the pairs of the form (u, x), where $u \in forg(i)$, contribute 0 to Kemeny score. Thus, we put the tuples $\big(\pi_1 > x, s_1\big), \big(\pi_2 > x, s_2\big), \ldots$ in $C(i, \{x\})$, where $(\pi_1, s_1), (\pi_2, s_2), \ldots$ denote the tuples of $C(i-1, \phi)$, and $\pi_1 > x, \pi_2 > x, \ldots$ denote the orderings obtained by appending x to $\pi_1, \pi_2, \ldots$ respectively. Now, let's consider a subset $T \subseteq B_i$ of size ≥ 2 that contains x. Let's describe how to find $C(i, T)$. Let $\Delta(i, T)$ denote the set of all candidates $c \in T$ such that c is not unanimously preferred over any other candidate of $forg(i) \uplus T$. That is, there is no other candidate $u \in forg(i) \uplus T$ such that $(c, u) \in \rho$. Recall that x appears after all candidates of $forg(i)$ in any linear extension of ρ on $forg(i) \uplus T$. So, it is clear that in any such ordering, the last candidate (say y) belongs to $\Delta(i, T)$ ensuring $\Delta(i, T)$ to be non-empty always. Moreover,

  - The pairs of the form (u, y), where $u \in forg(i)$, together contribute $\mathcal{A}(i, y)$ to Kemeny score.
  - The pairs of the form (u, y), where $u \in T \setminus \{y\}$, together contribute $\mathcal{B}(i, y, T \setminus \{y\})$ to Kemeny score.

  So, to find $C(i, T)$, let's proceed as follows: We compute $\Delta(i, T)$. For each possible choice $y \in \Delta(i, T)$ of the last candidate, let's form a set, say $\Gamma(y)$, that consists of the following tuples:

  - $\Big(\pi_1^y > y, s_1^y + \mathcal{A}(i, y) + \mathcal{B}\big(i, y, T \setminus \{y\}\big)\Big)$
  - $\Big(\pi_2^y > y, s_2^y + \mathcal{A}(i, y) + \mathcal{B}\big(i, y, T \setminus \{y\}\big)\Big)$ and so on

  where $(\pi_1^y, s_1^y), (\pi_2^y, s_2^y), \ldots$ denote the tuples of $C(i, T \setminus \{y\})$, and $\pi_1^y > y, \pi_2^y > y, \ldots$ denote the orderings obtained by appending y to $\pi_1^y, \pi_2^y, \ldots$ respectively. Finally, let's sort all tuples of $\biguplus_{y \in \Delta(i,T)} \Gamma(y)$ in ascending order of their Kemeny scores, and put the first $min\big(r, |forg(i) \uplus T|!\big)$ of them in $C(i, T)$.
Forget node: Suppose that B_i forgets a candidate, say x. For each $T \subseteq B_i$, as $forg(i) \uplus T = forg(i-1) \uplus \big(T \uplus \{x\}\big)$, we set $C(i, T) = C(i-1, T \uplus \{x\})$.

This concludes the proof of Theorem 5. □

Corollary 3 Distinct approximate Kemeny Ranking Aggregation *is FPT in the combined parameter unanimity width w and number of rankings r.*

Proof Consider an instance Distinct approximate Kemeny Ranking Aggregation. As in the algorithm described in the proof of Theorem 5, we find all $\mathcal{A}(\cdot,\cdot)$'s, $\mathcal{B}(\cdot,\cdot,\cdot)$'s and $C(\cdot,\cdot)$'s. Note that Π has r distinct approximately optimal Kemeny rankings if and only if $C(2m,\phi)$ contains r orderings, and the Kemeny score of the r^{th} ordering is at most λ times the Kemeny score of the first ordering. The overall running time of the algorithm is at most $\mathcal{O}^*\left(2^{\mathcal{O}(w)}\cdot r\right)$. This proves Corollary 3. □

Our last result is an FPT algorithm for Distinct approximate Kemeny Ranking Aggregation parameterized by the average Kendall-Tau distance d and the approximation parameter λ. Here we aim to relate the position of a candidate c in a λ-approximate ranking π, i.e. a ranking whose Kemeny Score denoted by K-score (π) has value at most $\lambda\cdot k_{OPT}$ where k_{OPT} denotes the optimal Kemeny Score, to its average position in the set of votes Π denoted by $p_{avg}(c)$.

Lemma 1 $p_{avg}(c)-\lambda\cdot d<\pi(c)<p_{avg}(c)+\lambda\cdot d$ *where $\pi(c)$ denotes position of c in π and d is average KT-distance.*

Proof There can be two cases for a vote $v\in\Pi$.

Case 1 $v(c)\leq\pi(c)$

In Case 1 there are $\pi(c)-1$ candidates that appear before c in π. Note that at most $v(c)-1$ of them can appear before c in v. Hence, at least $\pi(c)-v(c)$ of them must appear after c in v. Thus, $\mathrm{d_{KT}}(v,\pi)\geq\pi(c)-v(c)$.

Case 2 $v(c)>\pi(c)$

Here in Case 2, we come up with $\mathrm{d_{KT}}(v,\pi)\geq v(c)-\pi(c)$ arguing similarly to Case 1.

$$\begin{aligned}\text{K-score}(\pi)&=\sum_{v\in\Pi}\mathrm{d_{KT}}(v,\pi)\\&=\sum_{v\in\Pi:v(c)\leq\pi(c)}\mathrm{d_{KT}}(v,\pi)+\sum_{v\in\Pi:v(c)>\pi(c)}\mathrm{d_{KT}}(v,\pi)\\&\geq\sum_{\substack{v\in\Pi:\\v(c)\leq\pi(c)}}(\pi(c)-v(c))+\sum_{\substack{v\in\Pi:\\v(c)>\pi(c)}}(v(c)-\pi(c))\quad\left[\text{using Case1and Case 2}\right]\end{aligned}\tag{1}$$

Note that

$$\begin{aligned}&\sum_{v\in\Pi:v(c)\leq\pi(c)}(\pi(c)-v(c))+\sum_{v\in\Pi:v(c)>\pi(c)}(v(c)-\pi(c))\\&=\sum_{v\in\Pi}v(c)-2\sum_{\substack{v\in\Pi:\\v(c)\leq\pi(c)}}v(c)+\pi(c)\cdot(2\cdot|\{v\in\Pi:v(c)\leq\pi(c)\}|-n)\\&=n\cdot p_{avg}(c)-n\pi(c)-2\sum_{\substack{v\in\Pi:\\v(c)\leq\pi(c)}}v(c)+\pi(c)\cdot(2\cdot|\{v\in\Pi:v(c)\leq\pi(c)\}|)\\&\geq n\left(p_{avg}(c)-\pi(c)\right)\end{aligned}\tag{2}$$

Similarly,

$$\begin{aligned}
&\sum_{v\in\Pi:v(c)\le\pi(c)}(\pi(c)-v(c))+\sum_{v\in\Pi:v(c)>\pi(c)}(v(c)-\pi(c))\\
&=-\sum_{v\in\Pi}v(c)+2\sum_{\substack{v\in\Pi:\\ v(c)>\pi(c)}}v(c)+\pi(c)\cdot(-2\cdot|\{v\in\Pi:v(c)>\pi(c)\}|+n)\\
&=-n\cdot p_{avg}(c)+n\pi(c)+2\sum_{\substack{v\in\Pi:\\ v(c)>\pi(c)}}v(c)-\pi(c)\cdot(2\cdot|\{v\in\Pi:v(c)>\pi(c)\}|)\\
&\ge -n\left(p_{avg}(c)-\pi(c)\right) \qquad (3)
\end{aligned}$$

Now let's show that

$$\text{K-score}\,(\pi)<\lambda\cdot n\cdot d \qquad (4)$$

We have

$$\begin{aligned}
&d=\frac{\sum_{v\in\Pi}\sum_{w\in\Pi}\mathrm{d_{KT}}\,(v,w)}{n\cdot(n-1)}\ge\frac{n\cdot\sum_{w\in\Pi,w\ne v^\star}\mathrm{d_{KT}}\,(v^\star,w)}{n\cdot(n-1)}>\frac{\sum_{w\in\Pi,w\ne v^\star}\mathrm{d_{KT}}\,(v^\star,w)}{n}\\
&\left[\exists v^\star\in\Pi\text{ for which }\sum_{w\in\Pi,w\ne v^\star}\mathrm{d_{KT}}\left(v^\star,w\right)\text{ is minimum}\right]\\
&\implies \text{K-score}\left(v^\star\right)<n\cdot d\\
&\text{So, }k_{OPT}\le\text{K-score}\left(v^\star\right)<n\cdot d \qquad (5)\\
&\text{K-score}\,(\pi)\le\lambda\cdot k_{OPT}<\lambda\cdot n\cdot d\ \left[\text{Using Eq. (5)and proving Eq.(4)}\right] \qquad (6)\\
&\text{Now }\lambda\cdot n\cdot d>\text{K-score}\,(\pi)\ge n\cdot\left(p_{avg}(c)-\pi(c)\right)\left[\text{Using Eq.(1), (2)\&(4)}\right]\\
&\implies p_{avg}(c)-\lambda\cdot d<\pi(c) \qquad (7)\\
&\text{Again }\lambda\cdot n\cdot d>-n\cdot\left(p_{avg}(c)-\pi(c)\right)\left[\text{Using Eqs.(1), (3)\&(4)}\right]\\
&\implies \pi(c)<p_{avg}(c)+\lambda\cdot d \qquad (8)\\
&\text{Hence }p_{avg}\,(c)-\lambda\cdot d<\pi\,(c)<p_{avg}\,(c)+\lambda\cdot d\left[\text{Using Eqs.(7)\&(8)}\right] \qquad (9)
\end{aligned}$$

Equation (9) concludes the proof of Lemma 1. □

The following Lemma 2 depends on the Lemma 7 from [5].

Lemma 2 $|P_i|\le 4\lambda d-1\ \ \forall i\in[m-1]_0$

Proof We prove this lemma by contradiction. For this, we assume that for a position i, we have $|P_i|>4\lambda d$. Every candidate $c\in P_i$ has at most $2\lambda d-1$ different positions around its average position in a λ-approximately optimal Kemeny consensus π based on the proof of Lemma 1. In Lemma 1 we have established that $p_{avg}\,(c)-\lambda\cdot d<\pi(c)<p_{avg}\,(c)+\lambda\cdot d$. Hence, only those candidates in π have i as their common position for which

$$|i-\pi(c)|\le 2\lambda d-1 \qquad (10)$$

$$\Rightarrow i-(2\lambda d-1)\le\pi(c)\le i+(2\lambda d-1) \qquad (11)$$

Since, our assumption is $|P_i|\ge 4\lambda d$, therefore, each of these $4\lambda d$ candidates must hold a position which differs at most by $2\lambda d-1$ around position i. But from Eq. (11), we know that only those candidates in λ-approximately optimal Kemeny consensus π qualify for P_i

whose $\pi(c)$ lies in the range $2\lambda d - 1$ left and $2\lambda d - 1$ right around position i. Therefore, we have $4\lambda d - 1$ such positions. Hence, we approach towards contradiction. □

We now use the dynamic programming algorithm of Theorem 3 to claim the following Theorem 6. Its proof of correctness follows from Lemma 2.

Theorem 6 *There exists an FPT dynamic programming algorithm for* DISTINCT APPROXIMATE KEMENY RANKING AGGREGATION *parameterized by both* λ *and* d *which runs in time* $\mathcal{O}^*(16^{\lambda d} \cdot r)$.

4 Concluding remarks and future work

We consider the problem of finding distinct rankings that have a good Kemeny score in either exact or approximate terms, and propose algorithms that are tractable for various natural parameterizations of the problem. We show that many optimal or close to optimal solutions can be computed without significant increase in the running time compared with the algorithms to output a single solution, which is in sharp contrast with the diverse version of the problem.

We propose three main themes for future work. The first would be to extend these studies to other voting rules, and possibly identify meta theorems that apply to classes of voting rules. The second would be to understand if the structural parameters that we studied are correlated with some natural distance notion on the solution space: in other words, for a given distance notion, do all similar-looking instances have similar parameter values? Finally, we would also like to establish algorithmic lower bounds for the question of finding a set of diverse solutions that match the best known algorithms in the current literature.

References

1. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Shen, V.Y., Saito, N., Lyu, M.R., Zurko, M.E. (eds.) Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1–5, 2001, pp. 613–622. ACM, New York, NY, United States (2001). https://doi.org/10.1145/371920.372165
2. Pennock, D.M., Horvitz, E., Giles, C.L.: Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In: Kautz, H.A., Porter, B.W. (eds.) Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30–August 3, 2000, Austin, Texas, USA, pp. 729–734. AAAI Press / The MIT Press, Austin, Texas, USA (2000). http://www.aaai.org/Library/AAAI/2000/aaai00-112.php
3. Kemeny, J.G.: Mathematics without numbers. Daedalus **88**(4), 577–591 (1959)
4. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Cham, Switzerland (2015). https://doi.org/10.1007/978-3-319-21275-3
5. Betzler, N., Fellows, M.R., Guo, J., Niedermeier, R., Rosamond, F.A.: Fixed-parameter algorithms for kemeny rankings. Theor. Comput. Sci. **410**(45), 4554–4570 (2009). https://doi.org/10.1016/j.tcs.2009.08.033
6. Arrighi, E., Fernau, H., Lokshtanov, D., Oliveira Oliveira, M., Wolf, P.: Diversity in kemeny rank aggregation: A parameterized approach. In: Zhou, Z. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19–27 August 2021, pp. 10–16. ijcai.org, Montreal, Canada (2021). https://doi.org/10.24963/ijcai.2021/2 . Main Track
7. Arrighi, E., Fernau, H., Oliveira Oliveira, M., Wolf, P.: Width Notions for Ordering-Related Problems. In: Saxena, N., Simon, S. (eds.) 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020). Leibniz International Proceedings in Informatics

(LIPIcs), vol. 182, pp. 9–1918. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2020). https://doi.org/10.4230/LIPIcs.FSTTCS.2020.9

8. Slivkins, A., Radlinski, F., Gollapudi, S.: Learning optimally diverse rankings over large document collections. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. ICML'10, pp. 983–990. Omnipress, Madison, WI, USA (2010)
9. Agrawal, R.: The continuum-armed bandit problem. SIAM J. Control Optim. **33**(6), 1926–1951 (1995). https://doi.org/10.1137/S0363012992237273
10. Cornaz, D., Galand, L., Spanjaard, O.: Kemeny elections with bounded single-peaked or single-crossing width. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. IJCAI '13, pp. 76–82. AAAI Press, Beijing, China (2013)
11. Bartholdi, J., Tovey, C.A., Trick, M.A.: Voting schemes for which it can be difficult to tell who won the election. Soc. Choice Welf. **6**(2), 157–165 (1989)
12. Biedl, T., Brandenburg, F.J., Deng, X.: On the complexity of crossings in permutations. Discrete Math. **309**(7), 1813–1823 (2009)
13. Hemaspaandra, E., Spakowski, H., Vogel, J.: The complexity of kemeny elections. Theor. Comput. Sci. **349**(3), 382–391 (2005)
14. Van Zuylen, A., Williamson, D.P.: Deterministic pivoting algorithms for constrained ranking and clustering problems. Math. Op. Res. **34**(3), 594–620 (2009)
15. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: ranking and clustering. J. ACM (JACM) **55**(5), 1–27 (2008)
16. Kenyon-Mathieu, C., Schudy, W.: How to rank with few errors. In: Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing, pp. 95–103 (2007)
17. Schalekamp, F., Zuylen, A.v.: Rank aggregation: Together we're strong. In: 2009 Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments (ALENEX), pp. 38–51 (2009). SIAM
18. Davenport, A., Kalagnanam, J.: A computational study of the Kemeny rule for preference aggregation. AAAI **4**, 697–702 (2004)
19. Conitzer, V., Davenport, A., Kalagnanam, J.: Improved bounds for computing Kemeny rankings. AAAI **6**, 620–626 (2006)
20. Zuylen, A., Williamson, D.P.: Deterministic algorithms for rank aggregation and other ranking and clustering problems. In: Kaklamanis, C., Skutella, M. (eds.) Approximation and Online Algorithms, 5th International Workshop, WAOA 2007, Eilat, Israel, October 11–12, 2007. Revised Papers. Lecture Notes in Computer Science,**4927**, pp. 260–273. Springer, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77918-6_21
21. Guo, J., Hüffner, F., Niedermeier, R.: A structural view on parameterizing problems: Distance from triviality. In: Downey, R., Fellows, M., Dehne, F. (eds.) Parameterized and Exact Computation, pp. 162–173. Springer, Berlin, Heidelberg (2004)
22. Klamler, C.: The Dodgson ranking and its relation to Kemeny's method and Slater's rule. Soc. Choice Welf. **23**(1), 91–102 (2004)
23. Zwicker, W.S.: Cycles and intractability in a large class of aggregation rules. J. Artif. Intell. Res. **61**, 407–431 (2018)
24. Hebrard, E., Hnich, B., O'Sullivan, B., Walsh, T.: Finding diverse and similar solutions in constraint programming. AAAI **5**, 372–377 (2005)
25. Hebrard, E., O'Sullivan, B., Walsh, T.: Distance constraints in constraint satisfaction. IJCAI **2007**, 106–111 (2007)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.