Omega 119 (2023) 102893

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Beyond kemeny rank aggregation: A parameterizable-penalty framework for robust ranking aggregation with ties[☆]

Sina Akbari[a,*], Adolfo R. Escobedo[a]

School of Computing and Augmented Intelligence, Arizona State University, PO. Box 878809, Tempe, AZ 85287-8809, United States

ARTICLE INFO

Article history:
Received 20 September 2022
Accepted 28 April 2023
Available online 9 May 2023

Keywords:
Decision making/process
Rank aggregation
Kemeny aggregation
Kendall-tau distance
Condorcet criterion
Computational social choice

ABSTRACT

Rank Aggregation has ubiquitous applications in operations research, artificial intelligence, computational social choice, and various other fields. Interest in this problem has increased due in part to the need to consolidate lists of rankings and scores output by different decision-making processes and algorithms. Although most attention has focused on the variant of this problem induced by the Kemeny-Snell distance, other robust rank aggregation problems have been proposed. This work delves into the rank aggregation problem under the generalized Kendall-tau distance —a parameterizable-penalty distance measure for comparing rankings with ties— which contains Kemeny aggregation as a special case. First, it derives exact and heuristic solution methods. Second, it introduces a social choice property (GXCC) that encloses existing variations of the Condorcet criterion as special cases, thereby expanding this seminal social choice concept beyond Kemeny aggregation for the first time. GXCC offers both computational and theoretical advantages. In particular, GXCC may help to divide the original problem into smaller subproblems, while still ensuring that solving them independently yields the optimal solution to the original problem. Experiments on two benchmark datasets conducted herein show that the featured exact and heuristic solution methods can benefit from GXCC. Finally, this work derives new theoretical insights into the effects of the generalized Kendall-tau distance penalty parameter on the optimal ranking and on the proposed social choice property.

1. Introduction

Rank aggregation is an important problem in operations research and artificial intelligence, and it has been studied in various other fields including crowdsourcing [1,2], bioinformatics [3], and computational social choice [4]. Its wide array of applications include meta-search engines [5], journals ranking [6,7], information retrieval [8], ride pooling [9], supplier selection [10], and network inference [11,12]. Generally speaking, rank aggregation can be utilized whenever a set of judges (human or non-human) express their preferences over a set of items, and it is necessary to find a ranking that best represents these preferences collectively. It has been advocated as a systematic approach to guide decision-making processes, especially in multi-criteria decision-making (MCDM) [13–16]. MCDM methods evaluate alternatives based on predefined criteria and subsequently sort or rank them based on the evaluations [16]. Prominent examples include AHP (analytic hierarchy process) [17], ANP (analytic network process) [18], and ELECTRE (ELimination and Choice Expressing REality) [19]. Since different MCDM methods produce conflicting rankings, finding an overall consensus ranking that resolves these disagreements is of paramount importance [16]. In the context of information retrieval, analogous concerns fall under the umbrella of data fusion, where the goal is to derive a collective ranking of different information retrieval systems [20]; rank aggregation methods have been effective in this context as well [21]. Furthermore, rank aggregation has also gained attention over the past few years as a robust mechanism for consolidating heterogeneous ordered lists output by different machine learning techniques [20]. It has been used for this purpose, for example, in meta-search engines and spam detection [22,23], feature selection [24–26], natural language processing [27,28], recommendation systems [29], data query [30], and label ranking [31,32].

Rank aggregation methodologies can be categorized into *distance-based* and *ad hoc* methods [33], the latter of which is further divided into elimination and non-elimination methods. A prominent elimination ad hoc method is Ranked Choice Voting [4,34]. The popular *score-based* methods fall into the non-

☆ Area - Decision Analysis and Preference-Driven Analytics. This manuscript was processed by Associate Editor Salvatore Corrente.

* Corresponding author.
E-mail addresses: sina.akbari@asu.edu (S. Akbari), adres@asu.edu (A.R. Escobedo).

https://doi.org/10.1016/j.omega.2023.102893
0305-0483/

elimination ad hoc category; these methods rank items by assigning scores to the items according to some function based on their positions in each of the input rankings. Examples of score-based methods are Borda rule [35] and Copeland rule [36].

This work focuses on distance-based methods to take advantage of their rigorous axiomatic foundations and associated socio-theoretic properties, which translate into higher robustness to outliers and manipulations [4]. Note, however, that their aggregation problems tend to be more computationally demanding and are often NP-hard [4]. The reason is that techniques in this category return a ranking among all possible solutions from a large combinatorial space with the least distance to all input rankings according to a specified distance measure. There are various distance measures between rankings including Kemeny-Snell [37], Kendall-tau [38], and Spearman's footrule [39]. We refer the reader to Diaconis [40] and Fagin et al. [41] for descriptions of various such distance measures between strict rankings (rankings without ties) and non-strict rankings (rankings that *may* contain ties), respectively.

Dealing with non-strict rankings in real-world applications is the rule rather than the exception [42,43]. Their prominence has increased in recent years due to their enhanced flexibility in representing preference data. In particular, it may not possible for humans to express their preferences strictly over more than a very small number of items, or a subset of items may be considered indistinguishable to a specific MCDM or machine learning algorithm (e.g, it may award the same score to multiple items). What is more, forcing the judges to express their preferences in a strict manner may not reflect their true opinion. Therefore, developing rank aggregation frameworks capable of handling this type of ranking data is crucial. However, it is worth noting that allowing this flexibility in expressing the judge's preferences comes at a higher computational cost, as there are $n!$ possible strict ranking solutions as opposed to approximately $0.5n!(1.4)^{n+1} >> n!$ possible non-strict solutions, where n is the number of items [44]. To better appreciate the difference in magnitudes, when $n = 5$, there are 120 strict rankings and approximately 452 non-strict rankings; when $n = 50$, there are 3.04×10^{64} strict rankings and approximately 4.31×10^{71} non-strict rankings.

Kemeny aggregation (KEM-AGG) is perhaps the most widely studied variant of the distance-based rank aggregation problem. Its popularity is largely due to the fact that the Kemeny-Snell distance function underlying this problem uniquely satisfies a key set of axioms, namely anonymity, commutativity, extension, non-negativity, scaling, and the triangular inequality [37]. In addition, the optimal solution to KEM-AGG satisfies various desired social choice properties including the Condorcet criterion, consistency, and neutrality [45]. However, KEM-AGG is NP-hard when there are four or more input rankings [23,46]. For this reason, the dominant focus on solution methods for this problem has been on approximation and heuristic algorithms (see e.g., [23,47–51]) and relatively less attention has been devoted to exact methods. In particular, KEM-AGG for strict rankings has been formulated with binary programming in Conitzer et al. [52] and Cook [33] and for non-strict rankings in Yoo and Escobedo [53]. Other exact methods include the specialized branch and bound algorithm of Emond and Mason [43] and the iterative exact algorithms of Azzini and Munda [54] and Rico et al. [55] for strict and non-strict rankings, respectively. Exact methods are capable of solving instances mostly with tens and no more than a few hundred items reliably. For example, the largest strict ranking instance solved exactly in Emond and Mason [43], Conitzer et al. [52], Betzler et al. [56], had 15, 40, and 200 items, respectively, and the largest non-strict ranking instance solved exactly in Yoo and Escobedo [53] had 210 items.

Recent works have explored how the solution to KEM-AGG can be accelerated by leveraging certain social choice properties, which are provably satisfied by the optimal solution(s) to this problem. Based on these properties, certain instances can be partitioned into a set of subproblems, such that solving them independently to optimality and concatenating the solutions is guaranteed to produce an optimal solution to the larger original problem. A prevalent partitioning scheme is based on the seminal Condorcet criterion and its variants, including the Extended Condorcet criterion (XCC) [57] for strict rankings and the Non-strict Extended Condorcet criterion (NXCC) [53] for non-strict rankings. These techniques have been used to accelerate exact formulations of KEM-AGG [53,56,58] and lower bounding techniques [59]. Yoo and Escobedo [53] reported that NXCC accelerated their exact binary programming formulation by at least 25% and up to 96% on tested instances from the Preflib database [60]. As shown later in this paper, they can accelerate other exact methods and improve the performance and run time of heuristics as well. It is worth mentioning that Betzler et al. [56] introduced another partitioning technique based on the *3/4-Majority Rule*; however, the authors proved that XCC partitioning is always at least as good as partitioning using the 3/4-Majority Rule. Additionally, Milosz and Hamel [61] introduced a related approach that finds the relative ordering of certain item-pairs in the optimal solution(s). While it was shown to be more effective than XCC in providing the partial structure of the optimal solution(s) to KEM-AGG, the associated algorithm has a complexity of $O(n^3)$, where n is the number of items—whereas XCC has a complexity of $O(n^2)$—and it is only applicable for strict rankings.

The Kendall-tau distance [62] is another widely used distance between strict rankings. It is equivalent to the Kemeny-Snell distance whenever the input rankings are strict, but unlike the latter, it is not capable of handling non-strict rankings, i.e., it is not a distance measure in the presence of ties as it violates the triangular inequality [63]. Fagin et al. [63] proposed the generalized Kendall-tau distance (a parameterizable-penalty distance measure, among various metrics for comparing non-strict rankings), which includes the Kemeny-Snell distance as a special case. Accordingly, KEM-AGG represents only one variant of the Parameterizable-penalty Rank Aggregation (RANK-AGG(p)) framework, which is introduced herein to capitalize on the robust and flexible framework for handling ties induced by the generalized Kendall-tau distance. After formally defining RANK-AGG(p), this paper presents several exact and heuristic solution methods, and it generalizes the Condorcet criterion and its variants to expedite its solution. It is important to mention that another special case of RANK-AGG(p) has been studied in the literature, specifically by Brancotte et al. [64] and Andrieu et al. [65], who present an exact formulation and partitioning scheme, respectively. However, the general form of this rank aggregation problem has received little to no attention. In summary, this paper makes the following contributions:

- Define Parameterizable-penalty Rank Aggregation, which includes the Kemeny aggregation as a special case.
- Introduce an exact formulation, a constraint relaxation, and a heuristic algorithm for Parameterizable-penalty Rank Aggregation.
- Present a new social choice property (GXCC) that generalizes the Condorcet criterion and its variants beyond Kemeny aggregation for the first time
- Derive a partitioning method with respect to GXCC for expediting both exact and inexact approaches.
- Derive theoretical insights regarding the effect of the generalized Kendall-tau distance penalty parameter on the resulting partitions and solutions of the problem.

The rest of this paper is organized as follows. Section 2 introduces the notation used throughout the paper and establishes some preliminaries. Section 3 introduces various exact and heuristic methods. Section 4 generalizes the Condorcet criterion and its variants. Section 5 studies the effect of the generalized Kendall-tau distance penalty parameter on the optimal solution. Section 6 presents the computational results. Finally, Section 7 concludes the paper and discusses future directions of research.

2. Notation and preliminaries

Rankings can be divided into strict and non-strict. Strict rankings do not contain ties, while non-strict rankings allow for the possibility of ties. It is important to note that all strict rankings are included in the set of non-strict rankings, meaning that the latter *may* have ties. Rankings can also be further classified as complete or incomplete. In complete rankings, all items are ranked, while in incomplete rankings, some items may be unranked. Reasons for this include practicality, feasibility, and judiciousness [66]. This study focuses on complete non-strict rankings, but the methods proposed can be applied to strict rankings due to the relationship between the two classes of rankings.

Let $\mathcal{X} = \{1, 2, \ldots, n\}$ be the set of items, $\mathcal{L} = \{1, 2, \ldots, m\}$ be the set of indices of input rankings over $\mathcal{X}$, and $\boldsymbol{\Pi} \subset \mathcal{X}^n$ be the set of all possible complete ranking vectors over $\mathcal{X}$. Additionally, let $\boldsymbol{\pi}^l$ be the input ranking $l \in \mathcal{L}$, and π^l_i be the rank of item i in $\boldsymbol{\pi}^l$. Furthermore, let $\boldsymbol{\Lambda} = \{(i, j) \,|\, i, j \in \mathcal{X}, \;\; j > i\}$ be the set of distinct pairs of items. This paper focuses on complete rankings, where all items are explicitly ranked in the input and output rankings. The input rankings and the consensus ranking(s) can be strict or non-strict.

The preference relationship $i \succ_{\boldsymbol{\pi}} j$ indicates that item i is preferred over item j in $\boldsymbol{\pi}$, i.e., $\pi_i < \pi_j$, and $i \approx_{\boldsymbol{\pi}} j$ indicates that i and j are tied in $\boldsymbol{\pi}$, i.e., $\pi_i = \pi_j$. As a convention, let a *full rank reversal* denote the case where two rankings $\boldsymbol{\pi}^1, \boldsymbol{\pi}^2$ fully disagree over the relative orderings of items i and j (one of them ranks i ahead of j, and the other has the reverse opinion); additionally, let a *partial rank reversal* denote the case where i and j are tied in one ranking, but not in the other.

Definition 1. Let $s_{ij} = |l \in \mathcal{L} : i \succ_{\boldsymbol{\pi}^l} j|$ and $t_{ij} = |l \in \mathcal{L} : i \approx_{\boldsymbol{\pi}^l} j|$ be the number of input rankings in which item i is preferred over item j, and the number of input rankings in which i and j are tied, respectively.

Definition 2. (Yoo and Escobedo [53]) Item i is said to be pairwise preferred by a decisive majority over item j if $s_{ij} > s_{ji} + t_{ij}$.

For the rest of the paper, we use the term *pairwise preferred* instead of pairwise preferred by a decisive majority, for succinctness.

Definition 3. The Kemeny-Snell distance between two complete rankings $\boldsymbol{\pi}^1, \boldsymbol{\pi}^2$, denoted by $d_{KS}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2)$, is given by

$$d_{KS}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = \frac{1}{2} \sum_{i,j \in \mathcal{X}} \left| \text{sign}(\pi^1_i - \pi^1_j) - \text{sign}(\pi^2_i - \pi^2_j) \right|.$$

The function *sign*(v) returns 1 if $v > 0$, -1 if $v < 0$, and 0 otherwise. In the case of strict rankings, d_{KS} counts the number of full rank reversals. In the case of non-strict rankings, every full rank reversal has twice the weight of every partial rank reversal.

Definition 4. The consensus ranking obtained from KEM-AGG can be mathematically stated as

$$\boldsymbol{\pi}^*_{KS} = \underset{\boldsymbol{\pi} \in \boldsymbol{\Pi}}{\text{argmin}} \sum_{l \in \mathcal{L}} d_{KS}(\boldsymbol{\pi}, \boldsymbol{\pi}^l). \quad (1)$$

There is an equivalent rank aggregation problem to problem (1). Specifically, Emond and Mason [43] showed that whenever the rankings are complete, the aggregate ranking obtained by minimizing the cumulative Kemeny-Snell distance between the aggregate ranking and the input rankings is equivalent to the aggregate ranking obtained by maximizing the extended Kendall's tau correlation coefficient (τ_x) between the aggregate ranking and the input rankings. In mathematical terms, we have that

$$\underset{\boldsymbol{\pi} \in \boldsymbol{\Pi}}{\text{argmin}} \sum_{l \in \mathcal{L}} d_{KS}(\boldsymbol{\pi}, \boldsymbol{\pi}^l) = \underset{\boldsymbol{\pi} \in \boldsymbol{\Pi}}{\text{argmax}} \sum_{l \in \mathcal{L}} \tau_x(\boldsymbol{\pi}, \boldsymbol{\pi}^l). \quad (2)$$

Despite its name, this correlation measure differs from the distance that is the subject of this work, which is defined as follows.

Definition 5. The Kendall-tau distance between two complete strict rankings $\boldsymbol{\pi}^1, \boldsymbol{\pi}^2$, denoted by $d_{KT}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2)$, is given by

$$d_{KT}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = \sum_{(i,j) \in \boldsymbol{\Lambda}} K_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2), \quad (3)$$

where $K_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2)$ is set to 1 if the relative orderings of i and j are different in $\boldsymbol{\pi}^1$ and $\boldsymbol{\pi}^2$, and 0 otherwise. In other words

$$K_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = \begin{cases} 1 & \text{if} \quad (i \succ_{\boldsymbol{\pi}^1} j \;\wedge\; j \succ_{\boldsymbol{\pi}^2} i) \quad \vee \quad (j \succ_{\boldsymbol{\pi}^1} i \;\wedge\; i \succ_{\boldsymbol{\pi}^2} j) \\ 0 & \text{otherwise.} \end{cases}$$

It is straightforward to verify that whenever the input rankings are strict, d_{KT} and d_{KS} are equivalent; however, unlike d_{KS}, d_{KT} is not capable of handling ties [63]. Fagin et al. [63] proposed a generalization of the Kendall-tau distance for non-strict rankings using bucket orders, otherwise known as weak orders. A bucket order $\boldsymbol{B}$ is a transitive, total, and reflexive binary relation $\succ$ in which buckets $B_1, \ldots, B_t$ form a partition of $\mathcal{X}$ such that $i \succ j$ if and only if $i \in B_k$ and $j \in B_{k'}$, with $k < k'$. Members of the same bucket are considered as being tied. The position of bucket B_k is defined as $pos(B_k) = (\sum_{k' < k} |B_{k'}|) + (|B_k| + 1)/2$, and it indicates the average location within bucket B_k. A bucket order becomes a linear order when the cardinality of all buckets equals one. A non-strict ranking $\boldsymbol{\pi}$ can be mapped to a bucket order by letting $\pi_i = pos(\bar{B})$, where $\bar{B}$ is the bucket containing item i [63].

Next, we restate the definition of the generalized Kendall-tau distance introduced by Fagin et al. [63]. Given a fixed penalty parameter $0 \le p \le 1$ and two rankings $\boldsymbol{\pi}^1$ and $\boldsymbol{\pi}^2$, let $K^{(p)}_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2)$ be the contribution to the distance function, for each pair $(i, j) \in \boldsymbol{\Lambda}$. There are three cases with respect to the relative orderings of items i and j in $\boldsymbol{\pi}^1$ and $\boldsymbol{\pi}^2$:

Cases 1. There is a strict ordering between i and j in $\boldsymbol{\pi}^1$ and $\boldsymbol{\pi}^2$. If i and j are in the same order in both rankings, set $K^{(p)}_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = 0$; otherwise, set $K^{(p)}_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = 1$.

Cases 2. Both rankings tie i and j. In this case, set $K^{(p)}_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = 0$.

Cases 3. One of the rankings ties i and j, but not the other. In this case, set $K^{(p)}_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = p$.

Piecing together the above three cases, $K^{(p)}_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2)$ can be succinctly written as

$$K^{(p)}_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = \begin{cases} 1 & \text{if} \quad (i \succ_{\boldsymbol{\pi}^1} j \;\wedge\; j \succ_{\boldsymbol{\pi}^2} i) \quad \vee \quad (j \succ_{\boldsymbol{\pi}^1} i \;\wedge\; i \succ_{\boldsymbol{\pi}^2} j) \\ p & \text{if} \quad (i \approx_{\boldsymbol{\pi}^1} j \;\wedge (i \succ_{\boldsymbol{\pi}^2} j \;\vee\; j \succ_{\boldsymbol{\pi}^2} i)) \\ & \quad \vee (i \approx_{\boldsymbol{\pi}^2} j \;\wedge (i \succ_{\boldsymbol{\pi}^1} j \;\vee\; j \succ_{\boldsymbol{\pi}^1} i)) \\ 0 & \text{otherwise.} \end{cases}$$

Considering all distinct item-pairs, the *Kendall-tau distance with penalty parameter p*, denoted as $K^{(p)}$, can be abbreviated as

$$K^{(p)}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = \sum_{(i,j) \in \boldsymbol{\Lambda}} K^{(p)}_{ij}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2). \quad (4)$$

Note that Case 1 corresponds to a full rank reversal, and Case 3 corresponds to a partial rank reversal. Additionally, Eq. (4) induces the Kemeny-Snell distance (scaled by 1/2) as a special case, namely for $p = 1/2$. The $K^{(p)}$ distance is a metric for $1/2 \leq p \leq 1$, a near metric for $0 < p < 1/2$, and not a metric for $p = 0$ [63]. The ensuing example helps illustrate the use of this distance.

Example 1. Define two non-strict rankings of four item $\boldsymbol{\pi}^1 = (1, 2, 3, 3)$ and $\boldsymbol{\pi}^2 = (2, 1, 1, 1)$; the bucket orders corresponding to these two rankings are $\boldsymbol{B}^1 = \{\{1\}, \{2\}, \{3, 4\}\}$ and $\boldsymbol{B}^2 = \{\{2, 3, 4\}, \{1\}\}$, respectively. The example highlights all three cases of the distance: $K^{(p)}_{12}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = 1$ (Case 1), $K^{(p)}_{34}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = 0$ (Case 2), and $K^{(p)}_{23}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = p$ (Case 3). Considering all distinct item-pairs, we obtain $K^{(p)}(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2) = 3 + 2p$.

There are other variants of the Kendall-tau and Kemeny-Snell distances worth discussing, namely those of [67–73]. Each of these distances cannot be considered as a special case of the generalized Kendall-tau distance defined in Fagin et al. [63] and vice versa. In particular, Lee and Philip [68], Kumar and Vassilvitskii [69], and Durand and Pascual [73] all propose slightly different generalizations of Kendall tau distance for comparing *strict* rankings where item-pairs/position-pairs are weighted. Chee et al. [70] generalized the Kendall-tau distance from another angle by focusing on swapping adjacent intervals instead of adjacent items. Fagin et al. [67] generalized the Kendall-tau distance for comparing top-k lists, defined as rankings wherein out of n total items, only a small number of them, k, are explicitly ordered. The items in a top-k list are assumed to be pairwise preferred over all absent items. In a somewhat related but distinct direction, Gilbert et al. [72] proposed a set-wise generalization of the Kemeny-Snell distance where instead of counting pairwise disagreements, the measure counts the number of k-wise disagreements, i.e., the number of disagreement in a subset of top-choice alternatives of cardinality at most k. The reviewed works solely focus on strict rankings, therefore, the generalized Kendall-tau distance, in its general form, cannot be considered one of their special cases. Furthermore, the generalized Kendall-tau distance does not place weights on item-pairs/position-pairs and only focuses on complete rankings (not top-k lists), therefore, it does not include any of the reviewed distances, in their general form, as a special case.

3. Parameterizable-penalty rank aggregation problem

Setting aside the rather unacceptable process of breaking ties randomly, there are three prevalent treatments for handling partial rank reversals: 1) Assuming full agreement [74]; 2) Assuming full disagreement [64,65]; and 3) Reflecting a level of agreement halfway between the two extremes [37]. To elaborate, assume that every full rank reversal has unit weight. Then, each partial rank reversal has a weight of 0, 1, and 0.5 under treatments 1–3, respectively. The entire agreement-disagreement spectrum is covered by the $K^{(p)}$ distance (via the penalty parameter p), which has been utilized for comparing non-strict rankings in numerous applications such as multiagent system evaluation [75], CP-nets [76], and social network analysis [77]. Despite the high flexibility of this distance measure in handling partial rank reversals, its associated rank aggregation problem has received little to no attention in the literature. The general form of the distance has not been studied in the context of ranking aggregation; however, Brancotte et al. [64] and Andrieu et al. [65] have used it for this purpose for the special case induced by setting $p = 1$ (i.e., treatment 2).

This section is organized as follows. Section 3.1 formally defines RANK-AGG(p), proves it is NP-hard and proposes an exact formulation. Section 3.2 devises a constraint relaxation method for solving the formulation more efficiently. Finally, Section 3.3 presents a novel heuristic algorithm.

3.1. Definition and formulation

RANK-AGG(p) seeks a ranking $\boldsymbol{\pi}^*$—either strict or non-strict—with the lowest cumulative $K^{(p)}$ distance to all the input rankings.

Definition 6. The optimal ranking obtained from RANK-AGG(p) can be mathematically stated as

$$\boldsymbol{\pi}^* = \text{argmin}_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \sum_{l \in \mathcal{L}} K^{(p)}(\boldsymbol{\pi}, \boldsymbol{\pi}^l) = \text{argmin}_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \sum_{l \in \mathcal{L}} \sum_{(i,j) \in \boldsymbol{\Lambda}} K^{(p)}_{ij}(\boldsymbol{\pi}, \boldsymbol{\pi}^l). \tag{5}$$

Theorem 1. *RANK-AGG(p) is NP-hard for* $m \geq 4$.

Proof. KEM-AGG was shown to be NP-hard for $m \geq 4$ by an encoding of the feedback arc set problem [23,46]. Since KEM-AGG is only a special case of RANK-AGG(p), the latter inherits the computational complexity of the former. □

It is pertinent to add that $m = 2$ has a trivial solution as both of the input lists are optimal solutions. The computational complexity of the feedback arc set problem and KEM-AGG for $m = 3$ is an open problem [23], to the best of our knowledge.

Before proceeding, it is worth adding that due to the relationship depicted in Eq. (2), RANK-AGG(p) also includes the rank aggregation technique associated with the extended Kendall's tau correlation coefficient, which maximizes the value of that measure between the aggregate ranking and the input rankings, as a special case.

To introduce an exact formulation for RANK-AGG(p), the cumulative $K^{(p)}$ distance between a given ranking $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ and all the input rankings is re-expressed equivalently as $\sum_{(i,j)\in\boldsymbol{\Lambda}} K^{(p)}_{ij}(\boldsymbol{\pi})$, where

$$K^{(p)}_{ij}(\boldsymbol{\pi}) = \begin{cases} s_{ji} + pt_{ij} & \text{if } i \succ_{\boldsymbol{\pi}} j, \\ s_{ij} + pt_{ij} & \text{if } j \succ_{\boldsymbol{\pi}} i, \\ p(s_{ij} + s_{ji}) & \text{if } i \approx_{\boldsymbol{\pi}} j. \end{cases} \tag{6}$$

Eq. (6) states that, whenever item i is ranked ahead of item j in $\boldsymbol{\pi}$, the imposed $K^{(p)}$ distance between $\boldsymbol{\pi}$ and all the input rankings for this item-pair equals the number of input rankings where j is ranked ahead of i, plus p-times the number of input rankings where i and j are tied. Whenever the pair is tied, the imposed $K^{(p)}$ distance is p-times the number of input rankings where there is strict ordering between i and j.

Brancotte et al. [64] proposed a mixed-integer linear programming formulation for solving Problem (5) for the special case induced by fixing $p = 1$. Herein, we revise their objective function to reflect any possible value of p as follows:

$$\min \quad \sum_{i \in \mathcal{X}} \sum_{j \in \mathcal{X}} \Big[(s_{ji} + p\, t_{ij}) x_{i \succ j} + (s_{ij} + p\, t_{ij}) x_{j \succ i} + \; p(s_{ij} + s_{ji}) x_{i \approx j} \Big] \tag{7a}$$

$$\text{s.t.} \quad x_{i \succ j} + x_{j \succ i} + x_{i \approx j} = 1 \qquad \forall (i, j) \in \boldsymbol{\Lambda} \tag{7b}$$

$$x_{i \succ j} - x_{k \succ j} - x_{i \succ k} \geq -1 \qquad \forall i, j, k \in \mathcal{X}; i \neq j \neq k \tag{7c}$$

$$2x_{i \succ j} + 2x_{j \succ i} + 2x_{j \succ k} + 2x_{k \succ j} - x_{i \succ k} - x_{k \succ i} \geq 0 \quad \forall i, j, k \in \mathcal{X}; k > i, j \neq k \tag{7d}$$

$$x_{i \succ j}, x_{i \approx j} \in \mathbb{B} \qquad \forall i, j \in \mathcal{X}. \tag{7e}$$

Decision variable $x_{i \succ j}$ is equal to 1 if item i is ranked ahead of item j and 0 otherwise, and decision variable $x_{i \approx j}$ is equal to 1 if i and j are tied, and 0 otherwise. The objective function (7a) minimizes the cumulative $K^{(p)}$ distance to all the input rankings using

Eq. (6). Constraint (7b) enforces that, for every distinct item-pair (i, j), either i is ranked ahead of j, j is ranked of i, or i and j are tied. Constraint (7c) enforces preference-transitivity by preventing preference-cycles [78,79]; for example, if i is ranked ahead of j, and j is ranked ahead of k, then i must be ranked of k as well. Constraint (7d) enforces that if i and j are tied, and j and k are tied, then i and k must be tied as well (see Yoo and Escobedo [53] for additional types of preference-cycles involving non-strict rankings avoid through these constraints). Constraint (7e) specifies the domain of the variables.

Yoo and Escobedo [53] report that their formulation for KEM-AGG with non-strict rankings, denoted as GKBP, outperformed the variant of Formulation (7) induced by fixing $p = 1/2$. Because GKBP takes advantage of the specific relationship between the Kemeny-Snell distance and the extended Kendall's correlation coefficient [43,80], it cannot be directly applied to model RANK-AGG(p). Nevertheless, inspired by its computational performance and the fact that its constraints are equivalent to the *axiomatic facet defining inequalities* of the weak order polytope [53,78], we propose an alternative formulation for solving Problem (5) that uses the same set of constraints. The proposed formulation is a non-linear binary programming model and is given by:

$$\min \sum_{i\in\mathcal{X}}\sum_{j\in\mathcal{X}}(s_{ji} + pt_{ij})y_{ij} + \sum_{(i,j)\in\boldsymbol{\Lambda}} \big(p(s_{ij} + s_{ji}) - s_{ij} - s_{ji} - 2pt_{ij}\big)y_{ij}y_{ji} \tag{8a}$$

$$\text{s.t.} \quad y_{ij} + y_{ji} \geq 1 \qquad \forall (i, j) \in \boldsymbol{\Lambda} \tag{8b}$$

$$y_{ij} - y_{kj} - y_{ik} \geq -1 \qquad \forall i, j, k \in \mathcal{X}; \quad i \neq j \neq k \tag{8c}$$

$$y_{ij} \in \mathbb{B} \qquad \forall i, j \in \mathcal{X}; \quad i \neq j. \tag{8d}$$

Here, the decision variable y_{ij} is equal to 1 if item i is ranked ahead of or tied with item j, and 0 otherwise. Item i is ranked ahead of item j if $y_{ij} = 1, y_{ji} = 0$ (giving $y_{ij}y_{ji} = 0$), and items i and j are tied whenever $y_{ij} = y_{ji} = 1$ (giving $y_{ij}y_{ji} = 1$). The objective function (8a) minimizes the cumulative $K^{(p)}$ distance to all the input rankings. Constraint (8b) enforces that i and j cannot be simultaneously dispreferred over each other. Constraint (8c) imposes preference-transitivity, and Constraint (8d) specifies the domain of variables. Let $\overline{\boldsymbol{\pi}}$ be an arbitrary non-strict ranking induced by a feasible solution to Formulation (8); the rank of item i in $\overline{\boldsymbol{\pi}}$ is obtained as $\overline{\boldsymbol{\pi}}_i := n - \sum_{j\in\mathcal{X}:i\neq j} y_{ij}$.

The objective function (8a) can be linearized using a technique proposed by Glover and Woolsey [81]. For each distinct item-pair (i, j), the binary product $y_{ij}y_{ji}$ in the objective function is replaced by the auxiliary continuous variable z_{ij}, with the addition of four constraints: $z_{ij} \leq y_{ij}$, $z_{ij} \leq y_{ji}$, $z_{ij} \geq y_{ij} + y_{ji} - 1$, $z_{ij} \geq 0$. Since the objective coefficient of $y_{ij}y_{ji}$, i.e., $\big(p(s_{ij} + s_{ji}) - s_{ij} - s_{ji} - 2p\,t_{ij}\big)$, is always less than or equal to zero, constraint $z_{ij} \geq y_{ij} + y_{ji} - 1$ is actually redundant; that is, whenever $y_{ij} = y_{ji} = 1$, the objective function has the incentive to set z_{ij} to its maximum value of 1 and there is no need for this constraint. The full mixed-integer linear program for Problem (5) is as follows:

$$\min \sum_{i\in\mathcal{X}}\sum_{j\in\mathcal{X}}(s_{ji} + pt_{ij})y_{ij} + \sum_{(i,j)\in\boldsymbol{\Lambda}} \big(p(s_{ij} + s_{ji}) - s_{ij} - s_{ji} - 2p\,t_{ij}\big)z_{ij} \tag{9a}$$

$$\text{s.t.} \quad y_{ij} + y_{ji} \geq 1 \qquad \forall (i, j) \in \boldsymbol{\Lambda} \tag{9b}$$

$$y_{ij} - y_{kj} - y_{ik} \geq -1 \qquad \forall i, j, k \in \mathcal{X}; \quad i \neq j \neq k \tag{9c}$$

$$z_{ij} \leq y_{ij} \qquad \forall (i, j) \in \boldsymbol{\Lambda} \tag{9d}$$

$$z_{ij} \leq y_{ji} \qquad \forall (i, j) \in \boldsymbol{\Lambda} \tag{9e}$$

$$z_{ij} \geq 0 \qquad \forall (i, j) \in \boldsymbol{\Lambda} \tag{9f}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i, j \in \mathcal{X}; \quad i \neq j. \tag{9g}$$

It is possible to derive a lower bound on Problem (5) using the pairwise comparison information provided in Eq. (6).

Proposition 1. *A lower bound on Problem (5) can be obtained as:*

$$LB = \sum_{(i,j)\in\boldsymbol{\Lambda}} \min\,\big(s_{ji} + p\,t_{ij},\ s_{ij} + p\,t_{ij},\ p(s_{ij} + s_{ji})\big). \tag{10}$$

Proof. For every distinct item-pair, Eq. (10) selects the smallest contribution among all three possible preference relationships between the items. □

Proposition 1 effectively generalizes the lower bound for KEM-AGG with strict rankings introduced in Davenport and Kalagnanam [82] and with non-strict rankings introduced in Akbari and Escobedo [59]. This lower bound can be boosted by detecting preference-cycles in the input rankings, as the solution obtained by selecting the smallest contribution for each distinct item-pair may not be transitive [52,61]. Yet another lower bound can be obtained by solving the linear programming relaxation of Formulations (7) or (9).

3.2. Constraint relaxation method

Formulation (9) has $O(n^3)$ preference-transitivity constraints (i.e., Constraints (9c)) which makes solving it to optimality very difficult and practically impossible for large values of n. However, only a very small fraction of these constraints are typically necessary to solve rank aggregation problem instances to optimality [83]. We use this insight to develop a constraint relaxation (CR) method [84,85] to solve instances that are practically unsolvable with off-the-shelf methods. The pseudocode of CR is presented in Algorithm 1. It begins by dropping all preference-transitivity constraints from Formulation (9)—this is denoted as the *Relaxed Formulation*. At each iteration of CR, the Relaxed Formulation is solved and the solution is inspected to determine whether there are unsatisfied preference-transitivity constraints, which are added to the model. This process is repeated until the solution does not violate any preference-transitivity constraints. CR is guaranteed to obtain an optimal solution, as all preference-transitivity constraints (which are finite) are added to the Relaxed Formulation in the worst-case scenario.

3.3. The least imposed cost heuristic (LICH)

In this section, we develop a greedy iterative algorithm, denoted as the Least Imposed Cost Heuristic (LICH), for solving RANK-AGG(p). Placing item i at any position of a bucket order contributes a certain amount to the objective function (9a); denote this imposed cost as $\upsilon(i)$. The algorithm works by iteratively adding an item among a small number of positions in a working bucket order, namely the available item with the lowest associated υ-value.

LICH's pseudocode is presented in Algorithm 2 and is summarized as follows. In the first iteration, one item needs to be selected to initialize the working bucket order. Placing item i in the first

Algorithm 1: Constraint Relaxation (CR) Method .

Input : p, $\left[s_{ij}\right] \in \mathbb{Z}^{n.n}$, $\left[t_{ij}\right] \in \mathbb{Z}^{n.n}$
Output: Optimal solution to Formulation (9)

$t := 0$;
$\Xi := \{(i,j,k) \mid i,j,k \in \mathcal{X};\ i \neq j \neq k\}$; `// set of all item-triplets`
$\Xi' := \emptyset$; `// set of item-triplets whose preference-transitivity constraints are included in the Relaxed Formulation (see the next line)`
Build the Relaxed Formulation:

$$\min \sum_{i\in\mathcal{X}}\sum_{j\in\mathcal{X}}(s_{ji}+p\,t_{ij})y_{ij} + \sum_{(i,j)\in\boldsymbol{\Lambda}}\left(p(s_{ij}+s_{ji})-s_{ij}-s_{ji}-2p\,t_{ij}\right)z_{ij}$$

$$\text{s.t.}\quad (9b), (9d)-(9g)$$

$$y_{ij}-y_{kj}-y_{ik} \geq -1 \qquad \forall (i,j,k) \in \Xi'$$

Preference_Transitivity_Violation = True;
while *Preference_Transitivity_Violation is True* **do**
  Preference_Transitivity_Violation = False;
  Solve the Relaxed Formulation and obtain solution y_{ij}, where $i, j \in \mathcal{X},\ i \neq j$;
  for $(i,j,k) \in \Xi \setminus \Xi'$ **do**
    if $y_{ij}-y_{kj}-y_{ik} \ngeq -1$ **then**
      Preference_Transitivity_Violation = True;
      $\Xi' \leftarrow \Xi' \cup \{(i,j,k)\}$;

Return $\boldsymbol{\pi}^* = [n - \sum_{j\in\mathcal{X}:i\neq j} y_{ij}$ *for* i *in* $\mathcal{X}]$

Algorithm 2: The Least Imposed Cost Heuristic (LICH).

Input : $p, \mathcal{X}$, $[s_{ij}] \in \mathbb{R}^{n\times n}$, $[t_{ij}] \in \mathbb{R}^{n\times n}$
Output: Solution non-strict ranking

$i' := \arg\min_{i\in\mathcal{X}} \sum_{j\in\mathcal{X}} s_{ji} + p\,t_{ij}$;
$\boldsymbol{B} := \{\{i'\}\}$; `// initialize the working bucket order`
$\mathcal{X}^r := \mathcal{X}\setminus\{i'\}$; `// set of remaining items`
for $t = 1, \ldots, n-1$ **do**
  Let B_w be the last bucket of the working bucket order $\boldsymbol{B}$;
  for $i \in \mathcal{X}^r$ **do**
    $\upsilon(i) = \min\left(\sum_{j\in B_w}(s_{ji}+p\,t_{ij}),\ \sum_{j\in B_w}(s_{ij}+p\,t_{ij}),\ \sum_{j\in B_w}p(s_{ij}+s_{ji})\right)$;
  $i' = \arg\min_{i\in\mathcal{X}^r}\upsilon(i)$; `// find the item with the lowest imposed cost`
  $\mathcal{X}^r \leftarrow \mathcal{X}^r\setminus\{i'\}$; `// remove` i' `from the set of remaining items`
  `// The next block of code adds` i' `to the working bucket order in a way that it induces the lowest imposed cost`
  if $\upsilon(i') = \sum_{j\in B_w} p(s_{i'j}+s_{ji'})$ **then**
    $\boldsymbol{B} \leftarrow \{B_1, \ldots, B_{w-1}, B_w \cup \{i'\}\}$;
  if $\upsilon(i') = \sum_{j\in B_w}(s_{i'j}+pt_{i'j})$ **then**
    $\boldsymbol{B} \leftarrow \{B_1, \ldots, B_{w-1}, B_w, \{i'\}\}$;
  if $\upsilon(i') = \sum_{j\in B_w}(s_{ji'}+pt_{i'j})$ **then**
    $\boldsymbol{B} \leftarrow \{B_1, \ldots, B_{w-1}, \{i'\}, B_w\}$;

Merge adjacent buckets of $\boldsymbol{B}$ if doing so improves the value of objective function (9a);
Obtain $\boldsymbol{\pi}$ from $\boldsymbol{B}$ (as explained in Section 2);
Return $\boldsymbol{\pi}$;

place, assuming that it is ranked ahead of all other items, imposes the following cost:

$$\upsilon(i) = \sum_{j\in\mathcal{X}\setminus\{i\}} s_{ji} + p\,t_{ij}.$$

A working bucket order $\boldsymbol{B}$ is initialized by placing the item with the lowest imposed cost in the first bucket.

In the subsequent iterations, the remaining items are compared with only the items in the last bucket of the working bucket order, for the sake of efficiency. At each iteration and for each remaining item i, three different imposed costs are calculated based on where i is added to the working bucket order: 1) in the last bucket, 2) a new bucket placed right after the last bucket, and 3) a new bucket placed right before the last bucket. For each item, consider the minimum of the three calculated imposed costs. Formally, let B_w be the last bucket of the working bucket order $\boldsymbol{B}$ and $\mathcal{X}^r$ be the set of remaining items to be placed in the working bucket order; then, calculate

$$\upsilon(i) = \min\left(\sum_{j\in B_w} p(s_{ij}+s_{ji}),\ \sum_{j\in B_w}(s_{ij}+p\,t_{ij}),\ \sum_{j\in B_w}(s_{ji}+p\,t_{ij})\right) \quad \forall i \in \mathcal{X}^r.$$

The item with the lowest imposed cost overall is added to the working bucket order in the appropriate manner (according to the aforementioned three cases). As a post-processing subroutine, adjacent buckets are merged if doing so decreases the value of objective function (9a).

Theorem 2. *Algorithm 2 has a time complexity of* $O(n^3)$.

Proof. The worst time complexity of Algorithm 2 occurs when the working bucket order has only one bucket; in this case, the last bucket of the working bucket order is always of maximum size. In this case, the number of distinct item-pairs for which we need to calculate the imposed cost is given by

$$\begin{aligned} &n(n-1) + (n-1)(1) + (n-2)(2) + \cdots + (1)(n-1) \\ &= n(n-1) + \sum_{i=1}^{n-1}(n-i)i \\ &= n(n-1) + \frac{1}{6}(n-1)n(n+1). \end{aligned}$$

The imposed costs of each item-pair can be obtained in constant time. Therefore, the complexity of the full algorithm is $O(n^3)$. □

Note that the worst time complexity of Algorithm 2 occurs when at least $n-1$ items are tied in the optimal ranking, and its time complexity reduces to $O(n^2)$ in the case of strict rankings, as all buckets are singletons in the latter case.

4. Generalizing the condorcet criterion and its variants

Marquis de Condorcet [86] proposed one of the most eminent social choice properties in voting theory, which has come to be known as the Condorcet criterion (CC). This property declares that an election candidate (i.e., item) that is pairwise preferred over all other candidates must be declared as the top-ranked candidate in the outcome of the election (i.e., the optimal ranking); such a candidate is denoted as the *Condorcet winner*. CC can be formally stated as [87]

$$\text{if } \exists\, i \in \mathcal{X}:\ s_{ij} > s_{ji} \quad \forall j \in \mathcal{X}\setminus\{i\} \implies i \succ_{\boldsymbol{\pi}} j \quad \forall j \in \mathcal{X}\setminus\{i\},$$

where $\boldsymbol{\pi}$ is the consensus ranking(s). In an analogous fashion, the *Condorcet loser* is a candidate who is pairwise dispreferred over all other candidates. A voting rule is said to be *Condorcet consistent* if it always selects the Condorcet winner as the top-ranked item in its consensus ranking $\boldsymbol{\pi}$, when one exists [4]. Apart from KEM-AGG, there are other Condorcet consistent rank aggregation methods such as Dodgson's rule [88], maximin rule [89], and the ranked pairs rule [90]. It is worth mentioning that Smith [91] proposed an item-set version of the Condorcet criterion that has come to be known as the *Smith set* in the literature. The winning (losing) Smith set is defined as the smallest nonempty set of items that are pairwise preferred (dispreferred) over every item outside of the set. The Smith set may help decision-makers to exclude irrelevant items from consideration.

Truchon [57] proposed the Extended Condorcet criterion (XCC), which generalizes CC to guarantee an ordering of item-subsets in the consensus ranking(s). XCC states that if $\mathcal{X}$ can be arranged into a partition $\mathbf{X}=\{X_1, X_2, \ldots, X_w\}$ such that all items in the lower-indexed subsets are pairwise preferred over all items in the higher-indexed subsets, then the former must be ranked ahead of the latter in the consensus ranking(s). XCC can be stated formally as

$$\text{if } s_{ij} > s_{ji} \quad \forall i \in X_k \ \forall j \in X_k' \ \forall k < k' \implies i \succ_{\pi} j \quad \forall i \in X_k \ \forall j \in X_k' \ \forall k < k'.$$

Truchon [57] proved that the optimal solution(s) to KEME-AGG satisfies XCC. Note that the exact ordering of the full set of items is determined by solving the separate KEME-AGG subproblems induced by the items in each subset of the partition.

Recently, Yoo and Escobedo [53] demonstrated that KEM-AGG with non-strict rankings is inconsistent with XCC, meaning that its optimal solution(s) may violate the subset orderings indicated by this property. Consequently, the authors proposed a consistent social choice property for strict and non-strict rankings, which they called the Non-strict Extended Condorcet criterion (NXCC). It can be stated formally as

$$\text{if } s_{ij} > s_{ji} + t_{ij} \quad \forall i \in X_k,\ \forall j \in X_{k'},\ \forall k < k' \implies i \succ_{\pi^*_{KS}} j \quad \forall i \in X_k,\ \forall j \in X_{k'},\ \forall k < k'.$$

Observe that when all input rankings are strict (when $t_{ij}=0,\ \forall i, j \in \mathcal{X}$), NXCC becomes XCC. It was formally proven in Yoo and Escobedo [53] that any optimal ranking of KEM-AGG is consistent with NXCC. XCC and NXCC include both the Condorcet criterion and the Smith set. In fact, the winning (losing) Smith set corresponds to the first and most preferred (last and least preferred) subset of the XCC and NXCC partitions. Moreover, the winning (losing) Smith set corresponds to the Condorcet winner (loser) when the former is a singleton.

As a convention and to distinguish partitions using CC and its variants from other existing methods—e.g., 3/4-majority rule [56]—we denote such partitions as *Condorcet partitions*. Over the past few years, various researchers have employed Condorcet partitions to facilitate the exact solution to KEM-AGG, e.g., see [53,56,58]. In the last work from this list, it is stated that NXCC can expedite the solution run time of exact formulations of this problem by up to 96 percent. Recently, Akbari and Escobedo [59] also reported that NXCC rendered up to a 25x computational improvement in the computation of lower bounds for KEM-AGG.

4.1. Generalizing condorcet partitioning schemes

XCC and NXCC have been defined only for KEM-AGG. This subsection expands the concept of Condorcet partitions to RANK-AGG(p). To that end, it first redefines the concept of pairwise preference to adapt to the nature of the generalized problem, and it introduces a novel social choice property termed the Generalized Extended Condorcet criterion (GXCC).

Definition 7. Item i is (strictly) pairwise preferred over item j under the penalty parameter $p \in (0, 1]$ if

$$s_{ij} > \max\left(\left(\frac{1-p}{p}\right)s_{ji} + t_{ij},\ s_{ji}\right),$$

and it is weakly pairwise preferred over j under the penalty parameter $p \in (0, 1]$ if

$$s_{ij} \geq \max\left(\left(\frac{1-p}{p}\right)s_{ji} + t_{ij},\ s_{ji}\right).$$

Definition 8. (GXCC) Given a fixed penalty parameter $p \in (0, 1]$, assume that $\mathcal{X}$ can be arranged into a partition $\boldsymbol{X}^{(p)} = \{X_1, X_2, \ldots, X_w\}$ such that

$$s_{ij} > \max\left(\left(\frac{1-p}{p}\right)s_{ji} + t_{ij}, s_{ji}\right) \qquad \forall i \in X_k,\ \forall j \in X_{k'},\ \forall k < k'. \tag{11}$$

GXCC specifies that $\boldsymbol{\pi}^*$ must rank all items in the lower-indexed subsets of $\boldsymbol{X}^{(p)}$ ahead of all items in the higher-indexed subsets. That is, when Eq. (11) holds, then

$$i \succ_{\boldsymbol{\pi}^*} j \qquad \forall i \in X_k,\ \forall j \in X_{k'},\ \forall k < k'.$$

GXCC contains XCC and NXCC as special cases: it becomes NXCC when $p = 1/2$, and it becomes XCC when the same penalty is used and all the input rankings are strict. Furthermore, each of these decompositions also includes the Smith set (and Condorcet winner/loser, when applicable).

The following theorem proves that the optimal solutions to RANK-AGG(p) are consistent with GXCC. This means that solving the subproblems induced by the subsets of the GXCC partition independently to optimality and then concatenating the results in the proper order (placing all items in the lower-indexed subsets ahead of all items in the higher-indexed subsets) is guaranteed to yield an optimal solution to RANK-AGG(p). To the best of our knowledge, this is the first time that an exact Condorcet partitioning scheme has been defined and applied to a problem other than KEM-AGG in its general form.

Theorem 3. *RANK-AGG(p) satisfies GXCC.*

Proof. We use contradiction. Without loss of generality, let $\boldsymbol{X}^{(p)} = \{\overline{X}, \overline{X}^c\}$ be a GXCC bipartition of $\mathcal{X}$, where $\overline{X}^c = \mathcal{X} \backslash \overline{X}$, and let $\boldsymbol{\pi}^*$ be an optimal ranking where at least one item in $\overline{X}^c$ is ranked ahead of or tied with at least one item in $\overline{X}$. Consider a ranking $\boldsymbol{\pi}'$ obtained by modifying $\boldsymbol{\pi}^*$ such that all items of $\overline{X}$ are ranked ahead of all items in $\overline{X}^c$, and the relative orderings of all items within $\overline{X}$ and $\overline{X}^c$ are as in $\boldsymbol{\pi}^*$. The difference between the cumulative $K^{(p)}$ distances (i.e., to all the input rankings) accrued with $\boldsymbol{\pi}^*$ versus $\boldsymbol{\pi}'$, denoted by Δ, is given by

$$\begin{aligned}
\Delta &= \sum_{l \in \mathcal{L}} K^{(p)}(\boldsymbol{\pi}^*, \boldsymbol{\pi}^l) - \sum_{l \in \mathcal{L}} K^{(p)}(\boldsymbol{\pi}', \boldsymbol{\pi}^l)\\
&= \sum_{i \in \overline{X}} \sum_{j \in \overline{X}^c} \sum_{l \in \mathcal{L}} K_{ij}^{(p)}(\boldsymbol{\pi}^*, \boldsymbol{\pi}^l) - \sum_{i \in \overline{X}} \sum_{j \in \overline{X}^c} \sum_{l \in \mathcal{L}} K_{ij}^{(p)}(\boldsymbol{\pi}', \boldsymbol{\pi}^l)\\
&= \sum_{i \in \overline{X}} \sum_{j \in \overline{X}^c} \sum_{l \in \mathcal{L}} K_{ij}^{(p)}(\boldsymbol{\pi}^*, \boldsymbol{\pi}^l) - \sum_{i \in \overline{X}} \sum_{j \in \overline{X}^c} (s_{ji} + p\, t_{ij}).
\end{aligned}$$

The last equation comes from the starting assumption that $\boldsymbol{\pi}'$ ranks all items in $\overline{X}$ ahead of all items in $\overline{X}^c$. Therefore, the contribution of every distinct item-pair (i, j) where $i \in \overline{X}$ and $j \in \overline{X}^c$ in $\sum_{l \in \mathcal{L}} K^{(p)}(\boldsymbol{\pi}', \boldsymbol{\pi}^l)$ is equal to $s_{ji} + p\, t_{ij}$. Observe that item-pairs from different subsets do not contribute to Δ as their relative orderings are the same in $\boldsymbol{\pi}^*$ and $\boldsymbol{\pi}'$. To determine the sign of Δ, we compare terms $\sum_{l \in \mathcal{L}} K_{ij}^{(p)}(\boldsymbol{\pi}^*, \boldsymbol{\pi}^l)$ and $s_{ji} + p\, t_{ij}$. From Eq. (6), for every distinct item-pair $(i,j),\ i\in\overline{X}$ and $j\in\overline{X}^c$, we have

$$\sum_{l\in\mathcal{L}} K_{ij}^{(p)}(\boldsymbol{\pi}^*,\boldsymbol{\pi}^l)=\begin{cases} s_{ji}+p\,t_{ij} & \text{if} i\succ_{\boldsymbol{\pi}^*} j,\\ s_{ij}+p\,t_{ij} & \text{if} j\succ_{\boldsymbol{\pi}^*} i,\\ p(s_{ij}+s_{ji}) & \text{if} i\approx_{\boldsymbol{\pi}^*} j.\end{cases} \tag{12}$$

Next, we show that for such a distinct item-pair, $s_{ji}+p\,t_{ij}$ does not exceed $\sum_{l\in\mathcal{L}} K_{ij}^{(p)}(\boldsymbol{\pi}^*,\boldsymbol{\pi}^l)$ in cases where $j\succ_{\boldsymbol{\pi}^*} i$ and $i\approx_{\boldsymbol{\pi}^*} j$ ($K_{ij}^{(p)}(\boldsymbol{\pi}^*,\boldsymbol{\pi}^l)$ equals $s_{ji}+p\,t_{ij}$ when $i\succ_{\boldsymbol{\pi}^*} j$). Based on the starting assumption that $\boldsymbol{X}^{(p)}$ satisfies GXCC, for every distinct item-pair $(i,j),\ i\in\overline{X}$ and $j\in\overline{X}^c$, the following inequalities can be derived

$$s_{ij}>\max\left(\left(\frac{1-p}{p}\right)s_{ji}+t_{ij},\,s_{ji}\right)\implies s_{ij}>\left(\frac{1-p}{p}\right)s_{ji}+t_{ij}$$
$$\implies p(s_{ij}+s_{ji})>s_{ji}+p\,t_{ij}.$$

Furthermore, we have

$$s_{ij}>\max\left(\left(\frac{1-p}{p}\right)s_{ji}+t_{ij},\,s_{ji}\right)\implies s_{ij}>s_{ji}$$
$$\implies s_{ij}+p\,t_{ij}>s_{ji}+p\,t_{ij}.$$

Therefore, for every distinct item-pair (i,j), where $i\in\overline{X}$ and $j\in\overline{X}^c$, we have

$$\sum_{l\in\mathcal{L}} K_{ij}^{(p)}(\boldsymbol{\pi}^*,\boldsymbol{\pi}^l)\geq s_{ji}+p\,t_{ij}, \tag{13}$$

and summing over all distinct item-pairs (i,j), where $i\in\overline{X}$ and $j\in\overline{X}^c$, gives

$$\sum_{i\in\overline{X}}\sum_{j\in\overline{X}^c}\sum_{l\in\mathcal{L}} K_{ij}^{(p)}(\boldsymbol{\pi}^*,\boldsymbol{\pi}^l)\geq\sum_{i\in\overline{X}}\sum_{j\in\overline{X}^c} s_{ji}+p\,t_{ij}.$$

The above inequality implies that $\Delta\geq 0$. According to the given assumption, there exists at least one item in $\overline{X}^c$ that is ranked ahead of or tied with at least one item in $\overline{X}$. Hence, Eq. (13) holds strictly for at least one item-pair, meaning that $\Delta>0$, which contradicts the optimality of $\boldsymbol{\pi}^*$. Therefore, we can conclude that all items in $\overline{X}$ must be ranked ahead of all items in $\overline{X}^c$ in the optimal ranking.

Finally, we extend the proof to the case with $|\boldsymbol{X}^{(p)}|=w>2$. Consider a GXCC bipartition $\boldsymbol{X}^{(p)}=\{\overline{X},\overline{X}^c\}$ where $\overline{X}=X_1$ and $\overline{X}^c=\{X_2,X_3,\ldots,X_w\}$. Applying the prior result, all items in X_1 must be ranked ahead of all items in $\mathcal{X}\backslash X_1$. Next, consider bipartition $\boldsymbol{X}^{(p)}=\{\overline{X},\overline{X}^c\}$ where $\overline{X}=\{X_1,X_2\}$ and $\overline{X}^c=\{X_3,\ldots,X_w\}$; from the preceding case, all items in X_1 and X_2 must be ranked ahead of all items in $\mathcal{X}\backslash(X_1\cup X_2)$, and all items in X_1 must be ranked ahead of all items in X_2. Continuing in this manner, the only way that this statement holds for all bipartitions of the form $\boldsymbol{X}^{(p)}=\{\{X_1,\ldots,X_k\},\{X_{k+1},\ldots,X_w\}\}$, where $k\in\{1,\ldots,w-1\}$, is if $\boldsymbol{\pi}^*$ ranks all items in the lower-indexed subsets of $\boldsymbol{X}^{(p)}$ ahead of all items in the higher-indexed subsets. □

To elaborate on the theoretical implications of GXCC, we review *Arrow's Impossibility Theorem* [92], a milestone in the field of voting theory and computational social choice. This theorem is motivated by the intuitive requirement that any reasonable *social welfare function* (SWF)—a function that maps the voter's ordinal preferences over a set of competing candidates into one aggregate preference order of those candidates—should simultaneously be *weakly Paretian* and *independent of irrelevant alternatives (IIA)*. The weakly Paretian paradigm states that if all the voters strictly prefer candidate *a* to candidate *b*, then *a* should be ranked strictly better than *b* in the aggregate preference order. The IIA paradigm states that the relative ordering of *a* and *b* in the aggregate preference ordering should depend only on the relative orderings of *a* and *b* in the voters' inputs, that is, not a third candidate *c*. One of the main reasons that IIA is desirable is to prevent the manipulation of results by introducing extraneous candidates. Arrow [92] proved that whenever there are three or more candidates, the only aggregation function that simultaneously is weakly Paretian and IIA is a *dictatorship*, which is a voting rule where a single fixed voter (i.e., the dictator) whose preference ordering is returned as the aggregate ordering, regardless of the other voters' inputs.

The Impossibility Theorem demonstrates that certain basic and desirable properties of voting systems are incompatible—simply put, there exists no perfect voting rule or aggregation function. Nonetheless, many researchers have developed different SWFs that satisfy a subset of these (and other) desirable properties. The SWF associated with KEM-AGG uniquely satisfies five key social choice properties simultaneously: anonymity[1], neutrality[2], unanimity[3], reinforcement[4], and local stability [4,45], which translates into various practical benefits such as the aggregate preference order being robust against outliers [23]. It is worth adding that the SWF associated with the popular Borda rule [35] satisfies all of these properties, except local stability [45], which can be interpreted as a weaker version of IIA [45]. In particular, local stability requires that the relative aggregate ordering involving an interval of "closely related" candidates—defined as a subset of the candidates whose majority pairwise relations induce a cycle with each other but not with the remaining alternatives—should not change when introducing extraneous candidates—defined as those that each of the closely related candidates is pairwise preferred over by a majority of voters. Consider an example discussed in [45], where the true ordering of three candidates *a*, *b*, and *c* from a fixed number of input preference orders is being examined. Assume that three new but inferior candidates *d*, *e*, and *f*, over which *a*, *b*, and *c* is pairwise preferred, are introduced. Ideally, the introduction of inferior candidates should not change the ordering of *a*, *b*, and *c* compared to the case where inferior candidates are absent. We remark that this property can be interpreted as a special case of XCC and NXCC—specifically, it is a requirement on any two subsets within the decomposition. Moreover, the guarantee that KEM-AGG satisfies this property, which was established by Young [45], can be equivalently induced by the fact that RANK-AGG(p) satisfies GXCC for $p\in(0,1]$—and for $p=.5$, in particular; note that GXCC is undefined for $p=0$.

4.2. Enlarged GXCC partitions

Let $\wp(\mathcal{X})$ denote the class of partitions that satisfy GXCC and consider the case when there are multiple optimal rankings. The fact that all optimal rankings must be consistent with any $\boldsymbol{X}\in\wp(\mathcal{X})$ can be viewed as a restrictive condition. It might be possible to make the partition *finer*, i.e., one with more subsets, by requiring that it respects at least one rather than all of the optimal rankings. Such partitions as known as *enlarged partitions* [57]. Schalekamp and Zuylen [58] defined a type of enlarged XCC partitions for strict rankings as follows. Assume that $\mathcal{X}$ can be arranged into a partition $\boldsymbol{X}_e=\{X_1,X_2,\ldots,X_w\}$ such that $s_{ij}\geq s_{ji}\quad\forall i\in X_k,\ \forall j\in X_{k'},\ \forall k<k'$. Then, solving the subsets of $\boldsymbol{X}_e$ independently and concatenating the results in the proper order will respect at least one of the optimal strict rankings. It is possible to extend this idea to GXCC to obtain more effective partitions.

[1] An SWF is anonymous if all voters are weighted equally \citep{brandt2016handbook}.

[2] An SWF is neutral if the names of the candidates are permuted, then their ranking is similarly permuted \citep{brandt2016handbook}.

[3] An SWF is unanimous if all voters have the same ranking, then the aggregate ranking is the unanimous ranking \citep{brandt2016handbook}.

[4] An SWF satisfies reinforcement if whenever a ranking is approved by two separate groups of voters, then it would also be approved when the votes of the two groups are pooled \citep{brandt2016handbook}.

Corollary 1. *(Enlarged GXCC) Given a fixed penalty parameter* $p \in (0, 1]$, *assume that* $\mathcal{X}$ *can be arranged into a partition* $\boldsymbol{X}_e^{(p)} = \{X_1, X_2, \ldots, X_w\}$ *such that*

$$s_{ij} \geq \max\left(\left(\frac{1-p}{p}\right)s_{ji} + t_{ij},\, s_{ji}\right) \quad \forall i \in X_k, \quad \forall j \in X_{k'}, \quad \forall k < k'. \tag{14}$$

Then, there exists at least one optimal ranking in which all items in the lower-indexed subsets of $\boldsymbol{X}_e^{(p)}$ *are ranked ahead of all items in its higher-indexed subsets. That is, when Eq. (14) holds,*

$$\exists \boldsymbol{\pi}^* \in \boldsymbol{\Pi} : i \succ_{\boldsymbol{\pi}^*} j \quad \forall i \in X_k, \quad \forall j \in X_{k'}, \quad \forall k < k'.$$

Proof. The rationale mirrors that of the proof of Theorem 3 up to the point where it concludes that $\Delta \geq 0$. Following those steps, since $\boldsymbol{\pi}^*$ is an optimal ranking according to the starting assumption and $\Delta \geq 0$, it can be concluded that $\boldsymbol{\pi}'$ is an optimal ranking as well. □

Notice that an enlarged GXCC partition requires items in the lower-indexed subsets to be only weakly pairwise preferred over items in the higher-indexed subsets.

Example 2. Consider an instance with 10 rankings of 6 items. The input rankings and the pairwise comparison matrices, $\mathbf{S} = [s_{ij}] \in \mathbb{Z}^{6\times6}$ and $\mathbf{T} = [t_{ij}] \in \mathbb{Z}^{6\times6}$, are given by

Item	Input Rankings									
	π^1	π^2	π^3	π^4	π^5	π^6	π^7	π^8	π^9	π^{10}
1	2	4	2	2	3	4	5	2	1	5
2	3	1	3	5	1	3	5	1	1	5
3	3	3	1	3	1	1	2	5	2	3
4	1	3	1	1	5	1	1	5	3	1
5	4	5	4	1	2	2	3	4	4	2
6	5	2	5	4	4	5	4	3	5	4

$$\mathbf{S} = \begin{bmatrix} 0 & 3 & 4 & 3 & 5 & 7 \\ 4 & 0 & 3 & 4 & 6 & 7 \\ 6 & 5 & 0 & 2 & 7 & 8 \\ 7 & 6 & 4 & 0 & 7 & 7 \\ 5 & 4 & 3 & 2 & 0 & 8 \\ 3 & 3 & 2 & 3 & 2 & 0 \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} 0 & 3 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The standard GXCC partitions for $p = 1/2, 3/4$, and 1 are given by $\boldsymbol{X}^{(1/2)} = \{\{1, 2, 3, 4, 5\}, \{6\}\}$, $\boldsymbol{X}^{(3/4)} = \{\{3, 4\}, \{1, 2, 5\}, \{6\}\}$, and $\boldsymbol{X}^{(1)} = \{\{3, 4\}, \{2\}, \{1, 5\}, \{6\}\}$, respectively. The enlarged GXCC partitions are given by $\boldsymbol{X}_e^{(1/2)} = \{\{3, 4\}, \{1, 2\}, \{5\}, \{6\}\}$, $\boldsymbol{X}_e^{(3/4)} = \{\{3, 4\}, \{2\}, \{5\}, \{1\}, \{6\}\}$, and $\boldsymbol{X}_e^{(1)} = \{\{4\}, \{3\}, \{2\}, \{5\}, \{1\}, \{6\}\}$.

Example 2 illustrates the improved practicality of enlarged GXCC partitions. Considering the enlarged GXCC partitions for $p = 1/2$, only the relative ordering of item-pairs (3,4) and (1,2) needs to be determined; for $p = 3/4$, only the relative ordering of item-pair (3,4) needs to be determined; and for $p = 1$, an optimal solution is trivially obtained from the partition. Clearly, this accelerates the solution to RANK-AGG(p) as each of the enlarged GXCC partitions improves their standard counterparts.

Due to the enhanced practicality of enlarged GXCC, we focus on this partitioning mechanism for the rest of the paper. To obtain an enlarged GXCC partition, we modify an algorithm introduced in Yoo and Escobedo [53], which conducts NXCC partitioning by performing sequential pairwise comparisons. The modified algorithm is presented in Algorithm 3. It starts by placing the first item in a subset of the working partition, and it adds exactly one item to it at each iteration. Let item i denote the added item at any iteration and $\boldsymbol{X}_e^{(p)} = \{X_1, \ldots, X_w\}$ denote the working partition. The algorithm compares i with all items in the first subset of the working partition, X_1, leading to three possible outcomes. If item i is weakly pairwise preferred over all items in X_1, it is placed in a new subset right before X_1; if all items in X_1 are weakly pairwise preferred over i, i is placed in a new subset right after X_1; otherwise, it is placed in X_1. Subsequently, the algorithm iteratively checks whether the current working partition is a valid enlarged GXCC partition by verifying that all items in the lower-indexed subsets are weakly pairwise preferred over all items in the higher-indexed subsets. Whenever violations are detected, the respective subsets are merged/moved until there are no violations. The algorithm has a time complexity of $O(n^2)$ [53]. Note that GXCC and enlarged GXCC rely only on parameters required by the exact formulations; this fact, coupled with the quadratic time complexity of its algorithm, makes enlarged GXCC a great and fast pre-processing step for solving RANK-AGG(p) via exact and heuristic methods.

Algorithm 3: Enlarged GXCC Partitioning.

Input : p, $[s_{ij}] \in \mathbb{Z}^{n\times n}$, $[t_{ij}] \in \mathbb{Z}^{n\times n}$
Output : Enlarged GXCC Partition

$\boldsymbol{X}_e^{(p)} = \{\{1\}\}$;
for $i = 2$ **to** $|\mathcal{X}|$ **do**
  $k = 1$;
  if $s_{ij} \geq \max\left(\left(\frac{1-p}{p}\right)s_{ji} + t_{ij},\, s_{ji}\right) \;\; \forall j \in X_1$ **then**
    Insert i in a new subset before X_1, and increment the index of subsets after $X_{\kappa(i)}$ by 1;
    $k \leftarrow 3$;
  else if $s_{ji} \geq \max\left(\left(\frac{1-p}{p}\right)s_{ij} + t_{ij},\, s_{ji}\right) \;\; \forall j \in X_1$ **then**
    Insert i in a new subset after X_1, and increment the index of subsets after $X_{\kappa(i)}$ by 1;
    $k \leftarrow 3$;
  else
    Insert i in X_1;
    $k \leftarrow 2$;
  while $k \leq |\mathcal{X}|$ **do**
    if $s_{ij} \geq \max\left(\left(\frac{1-p}{p}\right)s_{ji} + t_{ij},\, s_{ji}\right) \forall j \in X_k$ **then**
      $k \leftarrow k + 1$;
    else if $s_{ji} \geq \max\left(\left(\frac{1-p}{p}\right)s_{ij} + t_{ij},\, s_{ij}\right) \forall j \in X_k$ **then**
      if $|\kappa(i) - k| = 1$ *and* $|X_{\kappa(i)}| = 1$ **then**
        Move $X_{\kappa(i)}$ after X_k;
      else
        Merge subsets from $X_{\kappa(i)}$ to X_k;
        Decrease the index of subsets after X_k by $(k - \kappa(i))$;
        $k \leftarrow \kappa(i) + 1$;
    else if $\exists j \in X_k$ *such that* $s_{ji} > \max\left(\left(\frac{1-p}{p}\right)s_{ij'} + t_{ij'},\, s_{ij}\right)$ **then**
      Merge subsets from $X_{\kappa(i)}$ to X_k;
      Decrease the index of subsets after X_k by $(k - \kappa(i))$;
      $k \leftarrow \kappa(i) + 1$;
Return $\boldsymbol{X}_e^{(p)}$;
* $\kappa(i)$ is the index of the subset containing item i.

We close this section by contrasting GXCC and enlarged GXCC with the exact graph-based partitioning scheme of Andrieu et al. [65], which applies to the special case of RANK-AGG(p) induced by setting $p = 1$. Upon close inspection, the conditions of the latter method translate to a relaxed version of GXCC that is stricter than the enlarged GXCC. To elaborate, GXCC requires all items in the lower-indexed subsets to be strictly pairwise preferred over all items in the higher-indexed subsets; Andrieu et al. [65]'s method require all items in the lower-indexed subsets to be weakly pairwise preferred (according to Definition 7 induced by setting $p = 1$) over all items in the higher-indexed subsets except for adjacent subsets, for which a strict pairwise preference is required; conversely, enlarged GXCC requires all items in the lower-indexed subsets to be weakly pairwise preferred over all items in the higher-indexed subsets. Andrieu et al. [65]'s method respects all the optimal solutions [65]; however, enlarged GXCC respects at least one but not necessarily all optimal solutions. Nonetheless, enlarged

GXCC is guaranteed to have at least as many subsets as Andrieu et al. [65]'s method.

5. Effect of modifying the penalty parameter

The value of penalty parameter p can have a big impact on the outcome of RANK-AGG(p), as it can alter the optimal ranking and the very nature of the problem. Consider two extreme values of p over which $K^{(p)}$ is a distance metric, namely 1/2 and 1. When $p = 1/2$, this induces KEM-AGG, in which every full rank reversal of item-pair (corresponding to case 1 of the $K^{(p)}$ distance) has twice the weight of every partial rank reversal (corresponding to case 3). As the value of p increases, the weight of a partial rank reversal increases; finally, when $p = 1$, a partial rank reversal has the same weight as a full rank reversal.

Beyond the fact that $p = 1/2$ is the most frequently used value, there have been no attempts to guide the choice of p or to analyze its implications. The ensuing paragraphs provide useful insights regarding the impact of p of the resulting GXCC and enlarged GXCC partitions and on the optimal ranking.

Proposition 2. *The set of optimal objective values of RANK-AGG(p) for all values of $p \in [0, 1]$ forms a piecewise linear envelope.*

Proof. The cumulative $K^{(p)}$ distance between any solution ranking $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ and all the input rankings, i.e., $\sum_{l \in \mathcal{L}} K^{(p)}(\boldsymbol{\pi}, \boldsymbol{\pi}^l)$, can be expressed as $a^{(\boldsymbol{\pi})} + pb^{(\boldsymbol{\pi})}$, which is an affine function in terms of p, where

$$a^{(\boldsymbol{\pi})} = \sum_{(i,j) \in \boldsymbol{\Lambda}} \left(s_{ij} \mathbb{1}_{j \succ_{\boldsymbol{\pi}} i} + s_{ji} \mathbb{1}_{i \succ_{\boldsymbol{\pi}} j} \right)$$

and

$$b^{(\boldsymbol{\pi})} = \sum_{(i,j) \in \boldsymbol{\Lambda}} \left((s_{ij} + s_{ji}) \mathbb{1}_{i \approx_{\boldsymbol{\pi}} j} + t_{ij} (\mathbb{1}_{j \succ_{\boldsymbol{\pi}} i} + \mathbb{1}_{i \succ_{\boldsymbol{\pi}} j}) \right).$$

Here, the function $\mathbb{1}_v$ returns 1 if v is true, and 0 otherwise; $a^{(\boldsymbol{\pi})}$ and $\boldsymbol{\pi}$ are the number of full rank reversals and partial rank reversals between $\boldsymbol{\pi}$ and all the input rankings. More specifically, if items i and j are tied in $\boldsymbol{\pi}$, $b^{(\boldsymbol{\pi})}$ tallies the number of input rankings in which i and j are not tied; otherwise, it tallies the number of input rankings where they are tied. Since the objective function values can be expressed as a series of affine functions and the $K^{(p)}$ distance is non-decreasing in p [63], the set of optimal objective values for all values of p forms a piecewise linear envelope. □

Fig. 1 illustrates an example of RANK-AGG(p) with two items. There are three possible rankings $\boldsymbol{\pi}^1, \boldsymbol{\pi}^2, \boldsymbol{\pi}^3$, whose respective affine functions are displayed; the piecewise linear envelope is shown in red. Proposition 2 will be used to derive additional insights regarding the effect of penalty parameter p.

Corollary 2. *If $\boldsymbol{\pi}^*$ is the optimal ranking for two distinct penalty parameters p^1 and p^2 such that $0 \le p^1 < p^2 \le 1$, then $\boldsymbol{\pi}^*$ is also the optimal rankings for any $p^1 < p < p^2$. Furthermore, if $\boldsymbol{\pi}^*$ is the optimal ranking for p^1 but not for p^2, it will not be the optimal ranking for any penalty parameter $p > p^2$.*

Corollary 2 is a direct outcome of Proposition 2. This corollary can help overcome the difficulty of selecting the exact value of penalty p in certain instances. For example, if $\boldsymbol{\pi}^*$ is the optimal ranking for $p = 1/2$ and $p = 1$, then it is also the optimal ranking for every intermediate value.

Additionally, we show that using $p = 3/4$ has an interesting interpretation, as it produces a robust solution. In particular, one may also be interested in finding the optimal ranking with the minimum *average* $K^{(p)}$ distance to the input rankings over all possible values of p for which the resulting function is a metric, i.e., $\forall p \in [1/2, 1]$, instead of only one specific value.

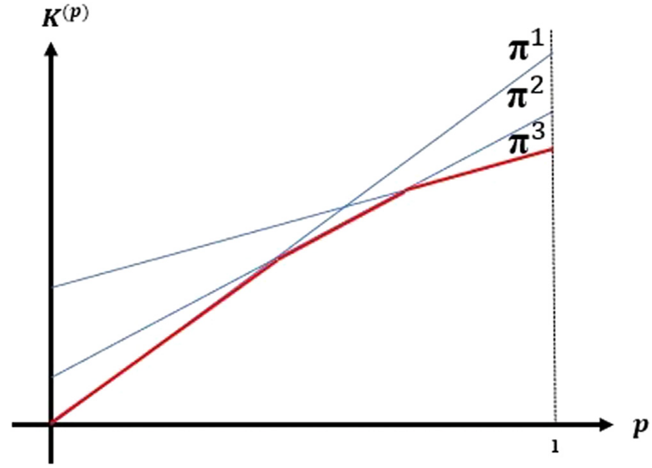

Fig. 1. Example depiction of objective function values obtained over all values of p by three different solution non-strict rankings (the piecewise linear envelope is shown in bolded red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Proposition 3. *The optimal ranking obtained by using $p = 3/4$ has the least average cumulative $K^{(p)}$ distance to the input rankings over the interval of penalty parameter p for which $K^{(p)}$ is a distance metric.*

Proof. Since all values of p are given the same weight, p can be treated as a random variable with a continuous uniform distribution over $[\frac{1}{2}, 1]$. Hence, the problem of finding a ranking with the least average cumulative $K^{(p)}$ distance to all the input rankings with respect to all values of $p \in [\frac{1}{2}, 1]$[1, 2 and 3] is equivalent to

$$\begin{aligned}
\boldsymbol{\pi}^* &= \arg\min_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \mathbb{E}_{p \in [\frac{1}{2}, 1]} \Big[\sum_{l \in \mathcal{L}} K^{(p)}(\boldsymbol{\pi}, \boldsymbol{\pi}^{\boldsymbol{l}}) \Big] \\
&= \arg\min_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \mathbb{E}_{p \in [\frac{1}{2}, 1]} \Big[a^{(\boldsymbol{\pi})} + pb^{(\boldsymbol{\pi})} \Big] \\
&= \arg\min_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \Big[a^{(\boldsymbol{\pi})} + b^{(\boldsymbol{\pi})} \mathbb{E}_{p \in [\frac{1}{2}, 1]}(p) \Big] \\
&= \arg\min_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \Big[a^{(\boldsymbol{\pi})} + 3/4 b^{(\boldsymbol{\pi})} \Big] \\
&= \arg\min_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \sum_{l \in \mathcal{L}} K^{(3/4)}(\boldsymbol{\pi}, \boldsymbol{\pi}^l).
\end{aligned}$$

□

As a last insight, when p increases, the cardinality of the GXCC and enlarged GXCC partitions may at times increase, but it cannot decrease.

Proposition 4. *Consider two fixed penalty parameters p_1, p_2, with $0 < p_1 < p_2 \le 1$. For penalty parameter p_2, the GXCC and enlarged GXCC partitions have at least as many subsets as their respective partitions with penalty parameter p_1. That is $|\boldsymbol{X}^{(p_2)}| \ge |\boldsymbol{X}^{(p_1)}|$ and $|\boldsymbol{X}_e^{(p_2)}| \ge |\boldsymbol{X}_e^{(p_1)}|$.*

Proof. For every item pair $(i, j) \in \boldsymbol{\Lambda}$, we have

$$s_{ij} > \max\left(\left(\frac{1 - p_1}{p_1} \right) s_{ji} + t_{ij},\, s_{ji} \right) \ge \max\left(\left(\frac{1 - p_2}{p_2} \right) s_{ji} + t_{ij},\, s_{ji} \right).$$

Therefore, if i is pairwise preferred over j under p_1, it will also be pairwise preferred over j under p_2. Hence, $\boldsymbol{X}^{(p_1)}$ is also a valid GXCC partition for RANK-AGG(p) using penalty parameter p_2. As a result, $\boldsymbol{X}^{(p_2)}$ will have at least as many subsets as $\boldsymbol{X}^{(p_1)}$. A parallel set of arguments can be applied to enlarged GXCC partitions. □

Proposition 4 indicates that partitioning may have more impact on large values of p. The possible effect of increasing p on the

Table 1
Solution time (in seconds) of different exact methods with and without prior GXCC partitioning for different values of p for TOC instances with $40 \leq n \leq 400$.

Instance id%	n	$p = 1/2$					$p = 3/4$					$p = 1$				
		BBP	MIP	CR	GXCC_MIP	GXCC_CR	BBP	MIP	CR	GXCC_MIP	GXCC_CR	BBP	MIP	CR	GXCC_MIP	GXCC_CR
ED-10-21	40	3.34	1.76	1.04	0.47	**0.97**	3.41	1.85	1.08	**0.13**	0.51	3.38	1.81	0.63	**0.12**	0.21
ED-10-22	40	3.79	1.74	**1.37**	1.74	**1.37**	3.41	1.75	**1.13**	1.75	**1.13**	3.36	1.73	**0.99**	1.73	**0.99**
ED-10-30	40	3.56	1.86	0.90	**0.09**	0.34	3.38	1.69	1.00	**0.09**	0.40	3.28	1.74	0.55	0.20	**0.13**
ED-10-20	41	3.87	1.93	1.41	**0.34**	0.80	3.67	1.90	1.70	**0.23**	0.90	3.70	1.94	0.83	**0.34**	0.68
ED-10-31	41	3.88	1.98	1.37	**0.18**	0.90	3.64	1.89	1.39	**0.31**	0.85	3.53	1.90	1.03	**0.18**	0.29
ED-10-4	42	4.25	2.05	1.28	1.40	**1.18**	3.83	2.05	1.24	1.42	**1.07**	3.87	2.05	1.01	**0.47**	0.76
ED-10-09	42	3.94	2.01	**0.69**	1.01	0.81	3.94	2.06	**0.73**	0.87	0.98	3.94	2.00	0.96	**0.84**	0.85
ED-10-06	43	4.65	2.17	**0.94**	2.06	1.19	4.18	2.20	1.17	2.06	**1.15**	4.18	2.29	1.35	2.06	**1.29**
ED-10-10	43	4.81	2.15	**1.02**	2.15	**1.02**	4.25	2.13	**0.97**	2.13	**0.97**	4.29	2.18	1.12	**0.55**	0.70
ED-10-08	44	4.56	2.34	1.71	**0.70**	1.13	4.53	2.32	1.28	**0.74**	1.47	4.44	2.30	1.55	0.76	**0.70**
ED-10-12	44	4.94	2.42	**0.98**	1.86	1.43	4.40	2.45	1.16	**0.36**	1.20	4.52	2.49	1.61	**0.50**	0.96
ED-10-13	44	4.42	2.5	**0.71**	0.95	0.81	4.44	2.30	**0.61**	1.02	0.78	4.42	2.39	0.99	0.97	**0.81**
ED-10-34	46	5.88	2.57	1.46	**0.68**	1.15	5.20	2.61	0.97	**0.70**	1.56	5.11	2.68	1.30	**0.25**	0.64
ED-10-07	47	6.18	2.87	0.79	2.11	**0.73**	5.69	2.79	**0.77**	2.33	1.12	5.70	2.79	1.49	0.82	**0.46**
ED-10-29	47	6.54	2.94	1.16	**0.55**	0.67	5.51	2.98	1.59	**0.38**	1.11	5.59	2.79	0.89	0.53	**0.46**
ED-10-18	49	6.84	3.20	1.46	**0.88**	1.50	6.68	3.28	1.33	**0.33**	1.43	6.35	3.26	1.41	**0.47**	0.89
ED-10-11	50	6.99	3.46	**0.67**	2.54	0.92	6.81	3.49	**1.01**	2.50	1.21	6.91	3.53	1.39	**0.68**	0.87
ED-10-02	51	7.54	3.78	0.55	2.85	**0.54**	8.22	4.09	**0.69**	1.74	1.02	8.13	4.26	1.95	2.23	**1.12**
ED-10-05	52	8.51	3.93	1.22	3.05	**0.97**	7.67	3.89	**0.70**	1.13	0.79	7.53	3.99	1.13	1.09	**0.98**
ED-10-15	52	8.40	3.94	**1.32**	2.23	1.68	7.69	3.88	**1.02**	2.31	1.45	7.65	3.93	1.98	2.23	**1.71**
ED-10-01	54	10.10	4.46	1.85	1.81	**0.89**	8.51	4.58	1.13	1.80	**1.00**	8.75	4.43	1.91	1.77	**1.18**
ED-10-03	54	10.16	5.13	**0.99**	2.59	1.99	9.40	4.26	**0.83**	2.41	1.06	8.70	4.42	1.53	1.50	**1.37**
MD-03-02	56	9.40	4.74	**0.66**	4.74	**0.66**	9.58	4.78	**0.41**	4.78	**0.41**	9.58	4.80	**0.46**	4.80	**0.46**
ED-10-16	57	11.26	5.49	1.74	**1.24**	1.61	10.00	5.42	**0.91**	1.32	1.20	10.46	5.48	1.68	**0.84**	0.95
MD-03-01	61	14.65	6.52	**0.42**	6.52	**0.42**	12.33	6.66	**0.47**	6.66	**0.47**	12.70	6.62	**0.37**	6.62	**0.37**
ED-10-17	61	13.48	6.48	1.27	5.24	**0.80**	12.70	6.37	**1.35**	1.61	1.76	12.49	6.32	1.38	1.54	**1.06**
ED-10-14	62	14.71	6.89	0.78	2.80	**0.67**	13.45	6.93	1.07	2.92	**0.74**	13.41	6.86	**1.27**	1.56	1.32
MD-03-04	63	13.86	7.12	**0.40**	7.12	**0.40**	13.66	6.99	**0.47**	6.99	**0.47**	13.73	7.33	**0.50**	7.33	**0.50**
ED-14-02	100	60.06	30.46	**0.73**	30.46	**0.73**	59.47	30.04	**0.75**	30.04	**0.75**	60.01	29.77	**0.76**	29.77	**0.76**
ED-14-03	100	60.08	30.41	**0.65**	30.40	**0.65**	59.42	29.96	**0.72**	29.96	**0.72**	60.09	29.53	**1.78**	29.53	**1.78**
MD-03-03	102	82.60	39.83	**0.59**	39.83	**0.59**	82.32	39.21	**0.69**	39.21	**0.69**	83.54	39.06	**1.02**	39.06	**1.02**
MD-03-05	103	84.11	38.35	**1.90**	38.35	**1.90**	81.71	38.35	**0.72**	38.35	**0.72**	81.48	38.92	**0.61**	38.92	**0.61**
MD-03-06	133	229.78	103.57	**1.18**	103.57	**1.18**	229.38	102.56	**1.26**	102.56	**1.26**	230.01	101.72	**1.20**	101.72	**1.20**
MD-03-08	147	305.83	136.26	**1.31**	136.26	**1.31**	307.80	136.98	**1.48**	136.98	**1.48**	303.46	137.33	**1.29**	137.33	**1.29**
MD-03-07	155	374.99	166.21	**1.64**	166.21	**1.64**	375.27	167.16	**1.57**	167.16	**1.57**	375.51	164.65	**1.46**	164.65	**1.46**
ED-10-50	170	1,144.02	207.24	152.53	202.63	**110.08**	$\geq 7,400.04$#	201.65	282.36	162.57	**159.5**	$\geq 7,392.48$&	251.31	**145.99**	252.3	192.23
ED-10-49	351	–	–	1393.73	–	**673.40**	–	–	4,260.8	–	**3,057.86**	–	–	5,956.47	–	**4,303.28**
Geometric Mean*		$\geq$ 13.51	6.33	1.16	3.41	**1.06**	$\geq$ 13.74	6.27	1.12	2.81	**1.09**	$\geq$ 13.44	6.34	1.25	2.43	**0.90**

% The instance names have been shortened. The original names include three zeros before the firs number and six zeros before the second number * The geometric mean does not include the ED-10-49 instance # The model had a relative optimality gap of 0.49% at the time of termination & The model had a relative optimality gap of 0.12% at the time of termination

Table 2
Solution time (in seconds) of different exact methods with and without prior GXCC partitioning for different values of p for the Cohen-Boulakia et al. [30] data set .

Instance	n	$p = 1/2$					$p = 3/4$					$p = 1$				
		BBP	MIP	CR	GXCC_MIP	GXCC_CR	BBP	MIP	CR	GXCC_MIP	GXCC_CR	BBP	MIP	CR	GXCC_MIP	GXCC_CR
LQTS	35	2.19	1.11	1.11	**0.17**	0.86	2.20	1.05	1.25	**0.23**	0.43	2.50	1.15	1.26	**0.18**	0.25
ADHD	45	5.73	2.52	1.27	0.48	**0.43**	6.11	2.70	2.09	**0.41**	0.56	5.92	2.61	1.18	**0.39**	0.42
Prostate Caner	218	$\geq 7,421^{*}$	1,147.64	290.29	387.15	**64.27**	3,115.36	1,231.01	167.46	436.89	**43.66**	$\geq 7,389^{\#}$	1,199.35	132.29	423.15	**40.62**
Bladder Caner	308	–	–	526.08	1,034.96	**95.54**	–	–	226.31	1,042.19	**108.72**	–	–	255.08	1,030.73	**122.69**
Breast Caner	386	–	–	**2,254.51**	–	2,545.23	–	–	4,863.50	–	**3,332.51**	–	–	1,275.95	–	**945.36**
Retinoblastoma	402	–		1,245.65	–	**652.19**	–	–	673.61	–	**466.96**	–	–	659.78	–	**547.62**
Neuroblastoma	431	–		1,502.60	–	**396.60**	–	–	1,098.71	–	**369.04**	–	–	730.46	–	**347.05**

* The model had a relative optimality gap of 32.33% at the time of termination # The model had a relative optimality gap of 98.25% at the time of termination

cardinality of the GXCC and enlarged GXCC partitions is demonstrated in Example 2, where $|\boldsymbol{X}^{(1/2)}| = 2$, $|\boldsymbol{X}^{(3/4)}| = 3$, $|\boldsymbol{X}^{(1)}| = 4$, and $|\boldsymbol{X}_e^{(1/2)}| = 4$, $|\boldsymbol{X}_e^{(3/4)}| = 5$, $|\boldsymbol{X}_e^{(1)}| = 6$.

6. Computational results

This section performs computational studies to: 1) compare the solution times of the revised Brancotte et al. [64] formulation (Formulation (7)), the proposed formulation (Formulation (9)), and the CR method; 2) investigate the effect of enlarged GXCC partitioning on the solution times of the proposed formulation and the CR method; 3) evaluate the performance of the proposed heuristic, both in terms of solution quality and run time; and 4) investigate the effect of enlarged GXCC partitioning on the solution quality and run time of LICH.

For all tested instances, we use three penalty values $p \in \{1/2, 3/4, 1\}$. All experiments herein were carried out on a PC with an Intel(R) Xeon(R) CPU E5-2680 2.40 GHz with 64 GB RAM. All optimization models were solved using CPLEX solver version 20.1, with a time limit of two-hour. The %Deviation from optimality of LICH is calculated as

$$\%Deviation = \frac{\text{objective function value of LICH - optimal objective function value}}{\text{optimal objective function value}}.$$

For the remainder of this section and the associated tables, the revised Brancotte et al. [64] binary programming formulation is denoted as BBP, and the proposed mixed-integer programming formulation as MIP. Additionally, the two-step solution method consisting of solving the partitioned problem via enlarged GXCC and then MIP is denoted as GXCC_MIP, solving the partitioned problem via enlarged GXCC and then CR is denoted as GXCC_CR, and solving the partitioned problem via enlarged GXCC and then LICH is denoted as GXCC_LICH.

Additionally, the two-step solution method consisting of partitioning the problem via enlarged GXCC and then solving via MIP is denoted as GXCC_MIP,

6.1. Data sets

The experiments consider two real-world data sets. The first is drawn from the TOC - "Orders with Ties - Complete List" data set from Preflib [60], a library of preference data. From this data set, only those instances with 40 to 351 items are used, as other instances of this data set are either too small and easy to solve or too large to be solved using exact methods. The second data set comes from a real-world application in bioinformatics provided by Cohen-Boulakia et al. [30]. Each of the seven instances of the Cohen-Boulakia et al. [30] data set contains four non-strict input rankings of genes possibly associated with Breast Cancer, Prostate Cancer, Bladder Cancer, Neuroblastoma, Retinoblastoma, ADHD (Attention Deficit Hyperactivity Disorder), and LQTS (Long QT Syndrome). Each of the input rankings is the result of querying for the genes associated with the aforementioned diseases in biological databases using four different methods. The goal of the referenced study is to alleviate the variability of information retrieval techniques by combining their outputs to obtain a more robust solution. For simplicity, henceforth, instance names are enclosed in quotation marks.

6.2. Results

First, we compare the solution times of the exact methods, beginning with the results of the TOC data set reported in Table 1; the best solution time(s) attained for each instance and each tested value of p is shown in bold. On average, MIP and CR were more

Table 3
Number of items in the enlarged GXCC partition's subsets for certain large instances.

Instance	n	$\lvert X_1\rvert, \lvert X_2\rvert, \ldots, \lvert X_w\rvert$
ED-10-50	170	1, 5, 1, 1, 1, 161
ED-10-49	351	5, 3, 3, 7, 333
LQTS	35	3, 1, 1, 1, 3, 1, 2, 1, 2, 2, 16, 2
ADHD	45	1, 2, 5, 1, 3, 1, 1, 1, 1, 25, 4
Prostate Cancer	218	1, 17, 1, 166, 15, 16, 2
Bladder Cancer	308	1, 4, 21, 13, 3, 69, 197
Breast Cancer	386	1, 362, 11, 12
Retinoblastoma	402	1, 1, 1, 33, 1, 1, 2, 4, 17, 341
Neuroblastoma	431	6, 55, 29, 9, 9, 1, 322

than 2x and 12x faster than BBP, respectively. In fact, BBP had a higher run time than MIP, and MIP than CR, for each of the tested instances and values of p. BBP failed to obtain the optimal solution of "ED-10-50" for $p = 1/2$ and $p = 1$ within the two-hour time limit; however, MIP and CR were able to solve these two cases in less than four minutes. Additionally, "ED-10-49" could not be directly solved via BBP and MIP due to out-of-memory errors, however, CR was able to solve it to optimality. Table 2 reports the solution times of the Cohen-Boulakia et al. [30] data set, where a similar pattern can be observed; the best solution time attained for each instance and each tested value of p is shown in bold. BBP had a higher run time than MIP, and MIP had a higher run time than CR for each of the tested instances and each tested value of p, except in one case. Additionally, BBP failed to obtain the optimal solution of "Prostate Cancer" for $p = 1/2$ and $p = 1$ within the two-hour time limit; however, MIP and CR were able to solve these two cases in less than 20 and 5 minutes, respectively. Additionally, "Bladder Cancer", "Breast Cancer", "Retinoblastoma", and "Neuroblastoma" could not be directly solved via BBP and MIP due to out-of-memory errors. On the other hand, CR was able to solve each of these instances within the time limit. Interestingly, all instances of the TOC data set with 100 to 155 items did not require any of the preference-transitivity constraints to be included in the optimization model, which resulted in a significant difference in the run time of MIP and CR on those instances. As a final note, the average and maximum percent of preference-transitivity constraints added by the CR method were 0.61% and 5.41% for the TOC data set, and they were 2.67% and 7.48% for the Cohen-Boulakia et al. [30] data set.

Next, we examine the impact of enlarged GXCC partitioning on the run times of MIP and CR. Beforehand, Table 3 reports the size of subsets of the enlarged GXCC partitions for the Cohen-Boulakia et al. [30] data set and the two largest instances of the TOC data set; the partitions matched for each tested value of p; other instances of TOC data set with more than 100 items were not partitionable. As Table 1 shows, enlarged GXCC partitioning was able to reduce the run times of both methods on the TOC data set for each tested value of p. Impressively, it reduced the geometric mean run

Table 4
Solution time (in seconds) and %Deviation of LICH with and without prior GXCC partitioning for different values of p for TOC instances with $40 \le n \le 400$.

Instance	n	$p = 1/2$				$p = 3/4$				$p = 1$			
		Time		%Deviation		Time		%Deviation		Time		%Deviation	
		LICH	GXCC_LICH	LICH	GXCC_LICH	LICH	GXCC_LICH	LICH	GXCC_LICH	LICH	GXCC_LICH	LICH	GXCC_LICH
ED-10-21	40	0.01	0.01	**0.57**	0.85	0.01	0.01	0.39	**0.18**	0.01	0.01	0.38	**0.17**
ED-10-22	40	0.01	0.01	**0.92**	**0.92**	0.01	0.01	**0.70**	**0.70**	0.01	0.01	**0.71**	**0.71**
ED-10-30	40	0.01	0.01	0.68	**0**	0.01	0.01	0.35	**0**	0.01	0.01	0.62	**0.08**
ED-10-20	41	0.01	0.01	**0.30**	0.54	0.01	0.01	0.43	**0.27**	0.01	0.01	0.74	**0.54**
ED-10-31	41	0.01	0.01	1.35	**0.15**	0.01	0.01	0.67	**0.23**	0.01	0.01	0.82	**0.14**
ED-10-04	42	0.01	0.01	1.35	**1.15**	0.01	0.01	1.49	**0.48**	0.01	0.01	0.96	**0.82**
ED-10-09	42	0.01	0.01	2.75	**2.62**	0.01	0.01	**1.06**	**1.06**	0.01	0.01	**0.72**	**0.72**
ED-10-06	43	0.01	0.01	**0.45**	**0.45**	0.01	0.01	**0.34**	**0.34**	0.01	0.01	**0.29**	**0.29**
ED-10-10	43	0.01	0.01	**2.71**	**2.71**	0.01	0.01	**0.22**	**0.22**	0.01	0.01	**0.19**	0.21
ED-10-08	44	0.01	0.01	2.61	**2.11**	0.01	0.01	0.63	**0.09**	0.01	0.01	0.54	**0.08**
ED-10-12	44	0.01	0.01	1.48	**1.21**	0.01	0.01	1.14	**0.24**	0.01	0.01	1.30	**0.35**
ED-10-13	44	0.01	0.01	3.29	**0.12**	0.01	0.01	0.04	**0.01**	0.01	0.01	0.03	**0**
ED-10-34	46	0.01	0.01	0.50	**0.44**	0.01	0.01	0.36	**0.25**	0.01	0.01	0.43	**0.23**
ED-10-07	47	0.01	0.01	**1.06**	**1.06**	0.01	0.01	**0.42**	**0.42**	0.01	0.01	0.35	**0.15**
ED-10-29	47	0.01	0.01	0.75	**0.21**	0.01	0.01	0.56	**0.04**	0.01	0.01	0.81	**0**
ED-10-18	49	0.01	0.01	0.42	**0.32**	0.01	0.01	0.43	**0.06**	0.01	0.01	0.38	**0.05**
ED-10-11	50	0.01	0.01	**0.38**	0.40	0.01	0.01	**0.26**	**0.26**	0.01	0.01	0.34	**0.25**
ED-10-02	51	0.01	0.01	**0.55**	**0.55**	0.01	0.01	**1.47**	**1.47**	0.01	0.01	**0.03**	**0.03**
ED-10-05	52	0.01	0.01	0.76	**0.49**	0.01	0.01	0.23	**0**	0.01	0.01	**0.14**	**0.14**
ED-10-15	52	0.01	0.01	**1.47**	**1.47**	0.01	0.01	1.02	**0.33**	0.01	0.01	0.83	**0.71**
ED-10-01	54	0.01	0.01	1.12	**0.51**	0.01	0.01	**1.60**	**1.60**	0.01	0.01	0.05	**0**
ED-10-03	54	0.01	0.01	**0.15**	**0.15**	0.01	0.01	0.29	**0.11**	0.01	0.01	0.30	**0.15**
MD-03-02	56	0.01	0.01	**0**	**0**	0.01	0.01	**0**	**0**	0.01	0.01	**0**	**0**
ED-10-16	57	0.01	0.01	**1.12**	1.72	0.01	0.01	0.14	**0.05**	0.01	0.01	0.20	**0**
MD-03-01	61	0.01	0.01	**0**	**0**	0.01	0.01	**0**	**0**	0.01	0.01	**0**	**0**
ED-10-17	61	0.01	0.01	**1.97**	**1.97**	0.01	0.01	0.53	**0.43**	0.01	0.01	0.17	**0.07**
ED-10-14	62	0.01	0.01	**0.97**	1.55	0.01	0.01	0.18	**0.06**	0.01	0.01	0.15	**0**
MD-03-04	63	0.01	0.01	**0**	**0**	0.01	0.01	**0**	**0**	0.01	0.01	**0**	**0**
ED-14-02	100	0.19	0.19	**0**	**0**	0.19	0.19	**0**	**0**	0.23	0.23	**0**	**0**
ED-14-03	100	0.20	0.20	**0**	**0**	0.18	0.18	**0**	**0**	0.23	0.23	**0**	**0**
MD-03-03	102	0.19	0.19	**0**	**0**	0.19	0.19	**0**	**0**	0.24	0.24	**0**	**0**
MD-03-05	103	0.20	0.20	**0**	**0**	0.20	0.20	**0**	**0**	0.25	0.25	**0**	**0**
MD-03-06	133	0.46	0.46	**0**	**0**	0.44	0.44	**0**	**0**	0.55	0.55	**0**	**0**
MD-03-08	147	0.60	0.60	**0**	**0**	0.60	0.60	**0**	**0**	0.60	0.60	**0**	**0**
MD-03-07	155	0.67	0.67	**0**	**0**	0.72	0.72	**0**	**0**	0.85	0.85	**0**	**0**
ED-10-50	170	0.12	0.14	2.52	**2.01**	0.09	0.12	**1.31**	1.32	0.09	0.11	1.41	**1.21**
ED-10-49	351	2.33	2.27	1.00	**1.02**	1.22	1.19	**1.35**	1.39	1.52	1.52	**0.87**	0.93
Average		0.02	0.02	0.90	**0.78**	0.02	0.02	0.48	**0.31**	0.02	0.02	0.37	**0.22**

Table 5
Solution time (in seconds) and %Deviation of LICH with and without prior GXCC partitioning for different values of p of the Cohen-Boulakia et al. [30] data set .

Instance	n	$p = 1/2$				$p = 3/4$				$p = 1$			
		Time		%Deviation		Time		%Deviation		Time		%Deviation	
		LICH	GXCC_LICH	LICH	GXCC_LICH	LICH	GXCC_LICH	LICH	GXCC_LICH	LICH	GXCC_LICH	LICH	GXCC_LICH
Long QT Syndrome	35	0.02	0.01	**0**	**0**	0.01	0.01	**0**	**0**	0.01	0.01	**3.69**	**3.69**
ADHD	45	0.01	0.01	**0**	**0**	0.01	0.01	6.74	**3.37**	0.01	0.01	**7.33**	**7.33**
Prostate Cancer	218	1.11	0.62	2.90	**2.85**	0.81	0.52	0.53	**0.51**	0.56	0.64	8.79	**0.35**
Bladder Cancer	308	2.11	0.40	0.41	**0.34**	1.99	1.46	1.09	**0.96**	2.46	1.76	0.12	**0.06**
Breast Cancer	386	5.91	4.97	**0.98**	**0.98**	4.89	3.89	**1.32**	**1.32**	2.28	1.89	37.50	**8.85**
Retinoblastoma	402	4.94	4.99	0.77	**0.05**	4.08	3.84	**0.28**	**0.28**	5.17	4.85	**0.54**	**0.54**
Neuroblastoma	431	4.59	4.58	9.30	**3.08**	5.21	4.63	4.06	**3.76**	1.29	0.41	15.83	**15.57**
Average		0.67	0.43	1.93	**1.19**	0.56	0.47	2.28	**1.66**	0.41	0.33	11.77	**5.20**

time of MIP from 6.34 to 2.43 seconds for $p = 1$. Enlarged GXCC partitioning decreased the run times of all instances with more than 62 items; however, it increased the run time of a handful of smaller instances. In fact, enlarged GXCC partitioning was able to reduce the run time of CR on “ED-10-49” approximately from 5956 to 4303 seconds, while it required only 0.02 seconds to obtain the partition. As Table 2 shows, enlarged GXCC partitioning reduced the run times of both methods on all instances of the Cohen-Boulakia et al. [30] data set for each tested value of p. It is worth adding that MIP was not able to solve “Bladder Cancer” due to an out-of-memory error; however, with the help of enlarged GXCC partitioning, MIP was able to solve this instance to optimality in approximately 1043 seconds. Most impressively, enlarged GXCC partitioning was able to reduce the run time of CR on “Bladder Cancer” approximately from 526 to 95 seconds, a 5.5x improvement, and the run time of MIP on “Prostate Cancer” approximately from 1098 to 369 seconds, close to a 3x improvement. The highest partitioning time of instances in this data set only took 0.07 seconds.

Next, we evaluate the performance of the LICH method. Table 4 reports the run time and %Deviation of the TOC data set; the best %Deviation attained for each instance and each tested value of p is shown in bold. LICH achieved an average %Deviation of at most 0.90% and a geometric mean run time of 0.02 seconds on this data set. It obtained the optimal solution in 10 instances for each tested value of p; its highest %Deviation on this data set was 2.71%. Table 5 reports the run time and %Deviation of the Cohen-Boulakia et al. [30] data set; the best %Deviation attained for each instance and each tested value of p is shown in bold. LICH achieved an average %Deviation of 1.93%, 2.28%, and 11.77% for $p = 1/2$, $p = 3/4$, and $p = 1$, respectively; its highest %Deviation was 37.50%. However, the geometric mean run time of this method was less than one second on this data set, and its highest run time was 5.91 seconds.

Finally, we investigate the effect of enlarged GXCC partitioning on the run time and solution quality of LICH. As Table 4 shows, enlarged GXCC partitioning was able to slightly reduce the average %Deviation of the TOC data set for each tested value of p while maintaining the same geometric mean run time. On the other hand, it was able to reduce both the geometric mean run time and the average %Deviation of the Cohen-Boulakia et al. [30] data set for all of the tested values of p, especially for $p = 1$. Remarkably, it reduced %Deviation of “Breast Cancer” for $p = 1$ from 37.50% to 8.85%, and %Deviation of “Prostate Cancer” for $p = 1$ from 8.79% to 0.35%.

Putting together all of these pieces, CR outperformed MIP, and MIP outperformed BBP. Additionally, enlarged GXCC partitioning reduced the run time of exact methods by up to 20x. The majority of the best run times of the exact methods were achieved by GXCC_CR. LICH achieved a near-optimal solution on most instances of the TOC data set, but it had a less commanding performance on the Cohen-Boulakia et al. [30] data set. However, enlarged GXCC partitioning reduced the run time and %Deviation of this method. Combining LICH with enlarged GXCC partitioning was shown to yield high-quality solutions in a short amount of time.

7. Conclusion and future research

This paper introduces and studies RANK-AGG(p), which contains the well-known Kemeny aggregation problem as a special case. It provides various analytical and computational contributions evaluated over two real-world data sets. It introduces a new mixed-integer programming formulation that outperforms an existing (revised) formulation over the featured instances. Additionally, it proposes a constraint relaxation technique, which was the only exact method capable of solving several large instances (with up to 431 items). Furthermore, it presents a greedy heuristic algorithm for obtaining high-quality solutions to RANK-AGG(p). The average %Deviation of this heuristic was 0.57% and 4.2% on the two tested data sets.

Additionally, this paper broadens the applicability of Condorcet criterion variants to RANK-AGG(p) by introducing a new social choice property (GXCC). It provides an algorithm for obtaining a valid GXCC partition and various analytical insights regarding the effect of the penalty parameter of the generalized Kendall-tau distance on the optimal ranking and GXCC partitions. GXCC proved to be effective in accelerating the run time of exact methods, as demonstrated by the featured experiments. It was able to decrease the run time exact and heuristic methods by up to 20x, and it improved %Deviation of the proposed heuristic by up to 19.14 percentage points.

Future research will explore the development of additional exact, approximate, and heuristic algorithms for RANK-AGG(p). Another important and interesting research direction is to compare different mathematical frameworks for aggregating non-strict rankings. Furthermore, it is possible to set different values of penalty parameter p for different indices, which can provide even more flexibility to the decision-maker on how to treat ties. However, this modification is not consistent with the assumptions of the generalized Kendall-tau distance and it would result in a new distance measure. Studying this variant can be a future research direction, as the theoretical and computational implications motivate questions beyond the scope of this work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appe ared to influence the work reported in this paper.

CRediT authorship contribution statement

Sina Akbari: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft. **Adolfo R. Escobedo:** Supervision, Methodology, Formal analysis, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

The authors gratefully acknowledge funding support from the National Science Foundation (Award 1850355).

References

[1] Chatterjee S, Mukhopadhyay A, Bhattacharyya M. A weighted rank aggregation approach towards crowd opinion analysis. Knowl Based Syst 2018;149:47–60.
[2] Kemmer R, Yoo Y, Escobedo A, Maciejewski R. Enhancing collective estimates by aggregating cardinal and ordinal inputs. In: Proceedings of the AAAI conference on human computation and crowdsourcing, vol. 8; 2020. p. 73–82.
[3] Quillet A, Saad C, Ferry G, Anouar Y, Vergne N, Lecroq T, et al. Improving bioinformatics prediction of microrna targets by ranks aggregation. Front Genet 2020;10:1330.
[4] Brandt F, Conitzer V, Endriss U, Lang J, Procaccia AD. Handbook of computational social choice. Cambridge University Press; 2016.
[5] Bar-Ilan J, Mat-Hassan M, Levene M. Methods for comparing rankings of search engine results. Comput Networks 2006;50(10):1448–63.
[6] Aledo JA, Gámez JA, Molina D, Rosete A. Consensus-based journal rankings: a complementary tool for bibliometric evaluation. J Assoc Inf Sci Technol 2018;69(7):936–48.
[7] Cook WD, Raviv T, Richardson AJ. Aggregating incomplete lists of journal rankings: an application to academic accounting journals. Accounting perspectives 2010;9(3):217–35.
[8] Losada DE, Parapar J, Barreiro A. A rank fusion approach based on score distributions for prioritizing relevance assessments in information retrieval evaluation. Information Fusion 2018;39:56–71.
[9] Şahin A, Sevim İ, Albey E, Güler MG. A data-driven matching algorithm for ride pooling problem. Computers & Operations Research 2022;140:105666.
[10] Peng Y, Kou G, Wang G, Shi Y. FAMCDM: a fusion approach of mcdm methods to rank multiclass classification algorithms. Omega (Westport) 2011;39(6):677–89.
[11] Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, et al. Wisdom of crowds for robust gene network inference. Nat Methods 2012;9(8):796–804.
[12] Puerta JM, Aledo JA, Gámez JA, Laborda JD. Efficient and accurate structural fusion of bayesian networks. Information Fusion 2021;66:155–69.
[13] Benítez-Fernández A, Ruiz F. A meta-goal programming approach to cardinal preferences aggregation in multicriteria problems. Omega (Westport) 2020;94:102045.
[14] Chen S, Liu J, Wang H, Augusto JC. Ordering based decision making–a survey. Information Fusion 2013;14(4):521–31.
[15] Liao H, Wu X. DNMA: a double normalization-based multiple aggregation method for multi-expert multi-criteria decision making. Omega (Westport) 2020;94:102058.
[16] Mohammadi M, Rezaei J. Ensemble ranking: aggregation of rankings produced by different multi-criteria decision-making methods. Omega (Westport) 2020;96:102254.
[17] Saaty TL. A scaling method for priorities in hierarchical structures. J Math Psychol 1977;15(3):234–81.
[18] Saaty TL. Decision making for leaders: the analytic hierarchy process for decisions in a complex world. RWS publications; 2001.
[19] Figueira JR, Mousseau V, Roy B. Electre methods. In: Multiple criteria decision analysis. Springer; 2016. p. 155–85.
[20] Klementiev A, Roth D, Small K. Unsupervised rank aggregation with distance-based models. In: Proceedings of the 25th international conference on Machine learning; 2008. p. 472–9.
[21] Hsu DF, Taksa I. Comparing rank and score combination methods for data fusion in information retrieval. Inf Retr Boston 2005;8(3):449–80.
[22] Desarkar MS, Sarkar S, Mitra P. Preference relations based unsupervised rank aggregation for metasearch. Expert Syst Appl 2016;49:86–98.
[23] Dwork C, Kumar R, Naor M, Sivakumar D. Rank aggregation methods for the web. In: Proceedings of the 10th international conference on World Wide Web; 2001. p. 613–22.
[24] Bolón-Canedo V, Alonso-Betanzos A. Ensembles for feature selection: a review and future trends. Information Fusion 2019;52:1–12.
[25] Sarkar C, Cooley S, Srivastava J. Robust feature selection technique using rank aggregation. Applied Artificial Intelligence 2014;28(3):243–57.
[26] Onan A, Korukoğlu S. A feature selection model based on genetic rank aggregation for text sentiment classification. Journal of Information Science 2017;43(1):25–38.
[27] Cascaro RJ, Gerardo BD, Medina RP. Aggregating filter feature selection methods to enhance multiclass text classification. In: Proceedings of the 2019 7th international conference on information technology: IoT and smart city; 2019. p. 80–4.
[28] Mehta P, Majumder P. Improving sentence extraction through rank aggregation. In: From extractive to abstractive summarization: a journey. Springer; 2019. p. 49–68.
[29] Oliveira SE, Diniz V, Lacerda A, Merschmannm L, Pappa GL. Is rank aggregation effective in recommender systems? an experimental analysis. ACM Transactions on Intelligent Systems and Technology (TIST) 2020;11(2):1–26.
[30] Cohen-Boulakia S, Denise A, Hamel S. Using medians to generate consensus rankings for biological data. In: International conference on scientific and statistical database management. Springer; 2011. p. 73–90.
[31] Aledo JA, Gámez JA, Molina D. Tackling the supervised label ranking problem by bagging weak learners. Information Fusion 2017;35:38–50.
[32] Werbin-Ofir H, Dery L, Shmueli E. Beyond majority: label ranking ensembles based on voting rules. Expert Syst Appl 2019;136:50–61.
[33] Cook WD. Distance-based and ad hoc consensus models in ordinal preference ranking. Eur J Oper Res 2006;172(2):369–85.
[34] Hare T. A treatise on the election of representatives, parliamentary and municipal. Longman, Green, Longman, & Roberts; 1861.
[35] Borda Jd. Mémoire sur les élections au scrutin. Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784) 1784.
[36] Copeland AH. A reasonable social welfare function. Tech. Rep.. mimeo, 1951 University of Michigan; 1951.
[37] Kemeny JG, Snell L. Preference ranking: an axiomatic approach. Mathematical models in the social sciences 1962:9–23.
[38] Kendall MG. A new measure of rank correlation. Biometrika 1938;30(1/2):81–93.
[39] Diaconis P, Graham RL. Spearman's footrule as a measure of disarray. Journal of the Royal Statistical Society: Series B (Methodological) 1977;39(2):262–268.
[40] Diaconis P. Group representations in probability and statistics. Lecture notes–monograph series 1988;11 i–192.
[41] Fagin R, Kumar R, Mahdian M, Sivakumar D, Vee E. Comparing partial rankings. SIAM J Discrete Math 2006;20(3):628–48.
[42] D'Ambrosio A, Iorio C, Staiano M, Siciliano R. Median constrained bucket order rank aggregation. Comput Stat 2019;34(2):787–802.
[43] Emond EJ, Mason DW. A new rank correlation coefficient with application to the consensus ranking problem. Journal of Multi-Criteria Decision Analysis 2002;11(1):17–28.
[44] Gross OA. Preferential arrangements. The American Mathematical Monthly 1962;69(1):4–8.
[45] Young HP, Levenglick A. A consistent extension of condorcet's election principle. SIAM J Appl Math 1978;35(2):285–300.
[46] Bartholdi J, Tovey CA, Trick MA. Voting schemes for which it can be difficult to tell who won the election. Soc Choice Welfare 1989;6(2):157–65.
[47] Acampora G, Iorio C, Pandolfo G, Siciliano R, Vitiello A. A memetic algorithm for solving the rank aggregation problem. In: Algorithms as a basis of modern applied mathematics. Springer; 2021. p. 447–60.
[48] Ailon N, Charikar M, Newman A. Aggregating inconsistent information: ranking and clustering. Journal of the ACM (JACM) 2008;55(5):1–27.
[49] Ailon N. Aggregation of partial rankings, p-ratings and top-m lists. Algorithmica 2010;57(2):284–300.
[50] Aledo JA, Gámez JA, Rosete A. Approaching rank aggregation problems by using evolution strategies: the case of the optimal bucket order problem. Eur J Oper Res 2018;270(3):982–98.
[51] Ding J, Han D, Yang Y. Iterative ranking aggregation using quality improvement of subgroup ranking. Eur J Oper Res 2018;268(2):596–612.
[52] Conitzer V, Davenport A, Kalagnanam J. Improved bounds for computing kemeny rankings. In: AAAI, vol. 6; 2006. p. 620–6.
[53] Yoo Y, Escobedo AR. A new binary programming formulation and social choice property for kemeny rank aggregation. Decision Analysis 2021;18(4):296–320.
[54] Azzini I, Munda G. A new approach for identifying the Kemeny median ranking. Eur J Oper Res 2020;281(2):388–401.
[55] Rico N, Vela CR, Díaz I. Reducing the time required to find the kemeny ranking by exploiting a necessary condition for being a winner. Eur J Oper Res 2022.
[56] Betzler N, Bredereck R, Niedermeier R. Theoretical and empirical evaluation of data reduction for exact kemeny rank aggregation. Auton Agent Multi Agent Syst 2014;28(5):721–48.
[57] Truchon M. An extension of the Condorcet criterion and Kemeny orders. Citeseer; 1998.
[58] Schalekamp F, Zuylen Av. Rank aggregation: Together we're strong. In: 2009 Proceedings of the eleventh workshop on algorithm engineering and experiments (ALENEX). SIAM; 2009. p. 38–51.
[59] Akbari S, Escobedo AR. Lower bounds on kemeny rank aggregation with non-strict rankings. In: 2021 IEEE symposium series on computational intelligence (SSCI). IEEE; 2021. p. 1–8.
[60] Mattei N, Walsh T. PrefLib: A library for preferences http://www. preflib. org. In: International Conference on Algorithmic DecisionTheory. Springer; 2013. p. 259–70.
[61] Milosz R, Hamel S. Exploring the median of permutations problem. J Discrete Algoritms 2018;52:92–111.
[62] Kendall MG. The treatment of ties in ranking problems. Biometrika 1945;33(3):239–51.

[63] Fagin R, Kumar R, Mahdian M, Sivakumar D, Vee E. Comparing and aggregating rankings with ties. In: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems; 2004. p. 47–58.
[64] Brancotte B, Yang B, Blin G, Cohen-Boulakia S, Denise A, Hamel S. Rank aggregation with ties: experiments and analysis. Proceedings of the VLDB Endowment (PVLDB) 2015;8(11):1202–13.
[65] Andrieu P, Brancotte B, Bulteau L, Cohen-Boulakia S, Denise A, Pierrot A, Vialette S. Efficient, robust and effective rank aggregation for massive biological datasets. Future Generation Computer Systems 2021.
[66] Moreno-Centeno E, Escobedo AR. Axiomatic aggregation of incomplete rankings. IIE Trans 2016;48(6):475–88.
[67] Fagin R, Kumar R, Sivakumar D. Comparing top k lists. SIAM J Discrete Math 2003;17(1):134–60.
[68] Lee PH, Philip L. Distance-based tree models for ranking data. Computational Statistics & Data Analysis 2010;54(6):1672–82.
[69] Kumar R, Vassilvitskii S. Generalized distances between rankings. In: Proceedings of the 19th international conference on World wide web; 2010. p. 571–80.
[70] Chee YM, et al. Breakpoint analysis and permutation codes in generalized kendall tau and cayley metrics. In: 2014 IEEE international symposium on information theory. IEEE; 2014. p. 2959–63.
[71] Fu X, Liu L, Liu L, Feng Y, Yue K. Ordinal preferences driven reputation measurement for online services with user incentive. In: 2020 IEEE international conference on Web services (ICWS). IEEE; 2020. p. 248–55.
[72] Gilbert H, Portoleau T, Spanjaard O. Beyond pairwise comparisons in social choice: asetwise kemeny aggregation problem. Theor Comput Sci 2022;904:27–47.
[73] Durand M, Pascual F. Collective schedules: Axioms and algorithms. In: Algorithmic Game Theory: 15th International Symposium, SAGT 2022, Colchester, UK, September 12–15, 2022, Proceedings. Springer; 2022. p. 454–71.
[74] Kendall MG. Partial rank correlation. Biometrika 1942;32(3/4):277–83.
[75] Rowland M, Omidshafiei S, Tuyls K, Perolat J, Valko M, Piliouras G, et al. Multiagent evaluation under incomplete information. Adv Neural Inf Process Syst 2019;32.
[76] Loreggia A, Mattei N, Rossi F, Venable KB. A notion of distance between cp-nets. In: Proc. of AAMAS; 2018. p. 955–63.
[77] Zhang Y, Bouadi T, Martin A. A clustering model for uncertain preferences based on belief functions. In: International conference on big data analytics and knowledge discovery. Springer; 2018. p. 111–25.
[78] Fiorini S, Fishburn PC. Weak order polytopes. Discrete Math 2004;275(1–3):111–27.
[79] Grötschel M, Jünger M, Reinelt G. A cutting plane algorithm for the linear ordering problem. Oper Res 1984;32(6):1195–220.
[80] Yoo Y, Escobedo AR, Skolfield JK. A new correlation coefficient for comparing and aggregating non-strict and incomplete rankings. Eur J Oper Res 2020.
[81] Glover F, Woolsey E. Converting the 0–1 polynomial programming problem to a 0–1 linear program. Oper Res 1974;22(1):180–2.
[82] Davenport A, Kalagnanam J. A computational study of the Kemeny rule for preference aggregation. In: AAAI, vol. 4; 2004. p. 697–702.
[83] Pedings KE, Langville AN, Yamamoto Y. A minimum violations ranking method. Optimization and Engineering 2012;13(2):349–70.
[84] Dantzig G, Fulkerson R, Johnson S. Solution of a large-scale traveling-salesman problem. Journal of the operations research society of America 1954;2(4):393–410.
[85] Dantzig GB, Fulkerson DR, Johnson SM. On a linear-programming, combinatorial approach to the traveling-salesman problem. Oper Res 1959;7(1):58–66.
[86] Marquis de Condorcet MJA. Essai sur l'application de l'analyse a la probabilite des decisions: rendues a la pluralite de voix. De l'Imprimerie royale; 1785.
[87] Young HP. Condorcet's theory of voting. American Political science review 1988;82(4):1231–44.
[88] Dodgson C. A method of taking votes on more than two issues. The theory of committees and elections 1876.
[89] Young HP. Extending condorcet's rule. J Econ Theory 1977;16(2):335–53.
[90] Tideman N. Collective decisions and voting: the potential for public choice. Routledge; 2017.
[91] Smith JH. Aggregation of preferences with variable electorate. Econometrica: Journal of the Econometric Society 1973:1027–41.
[92] Arrow KJ. Social choice and individual values. New Haven, CT: Cowles Foundation; 1951.