ATLANTIS PRESS

Atlantis Studies in Uncertainty Modelling, volume 3

Joint Proceedings of the 19th World Congress of the International Fuzzy Systems Association (IFSA), the 12th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), and the 11th International Summer School on Aggregation Operators (AGOP)

The Borda Count as an Initial Threshold for Kemeny Ranking Aggregation

***Noelia Rico**[a] and **Camino R. Vela**[b] and **Raúl Pérez-Fernández**[c] and **Irene Díaz**[d]
[a]Department of Computer Science, University of Oviedo, noeliarico@uniovi.es
[b]Department of Computer Science, University of Oviedo, crvela@uniovi.es
[c]Department of Statistics and O.R. and Mathematics Didactics, University of Oviedo, perezfernandez@uniovi.es
[d]Department of Computer Science, University of Oviedo, sirene@uniovi.es

Abstract

The need of establishing a consensus ranking from the preferences expressed by different voters arises in several contexts. The method proposed by Kemeny to this purpose is famously known due to its numerous fulfilled properties and intuitive interpretation, as it minimizes the number of pairwise comparisons in which the consensus ranking disagrees with the preferences given by the voters. Nevertheless, this method has a main drawback regarding its execution time, thus preventing its use in practice. There exist some other methods, such as the Borda Count, that can be computed in polynomial time, even though they do not fulfill as many intuitive properties as the Kemeny method. Here, we propose to use the Borda Count ranking as the initial solution of a Branch-and-Bound algorithm for reducing the execution time of the Kemeny method. The presented experiments show that this approach leads to an improvement in execution time.

Keywords: Ranking aggregation, Kemeny method, Borda Count, Optimization, Computational Social Choice.

1 Introduction

The situation in which different alternatives need to be ranked according to the preferences of some voters arises frequently in many real-life contexts. Different methods for obtaining this ranking have been studied over the years in the field of social choice theory [7]. Here, we focus on the most classical case in which the preferences of the voters are provided in the form of a ranking of the alternatives.

Back in the 18th century, Borda [6] proposed a method based on exploiting the positional information given by the rankings. This method has been used in many real-life situations due to its simplicity and its ease of computation. However, this proposal was already heavily criticized by Borda's contemporary: Condorcet [8]. According to Condorcet [18], only the pairwise information provided by the rankings may be exploited and, for any two alternatives, the preferred one should be the one that beats the other one by simple majority, i.e., the one that is preferred to the other one by more than half of the voters. A ranking of the alternatives obtained in such a way is referred to as the Condorcet ranking. Unfortunately, the Condorcet ranking might not exist since the simple majority relation may lead to cycles (known as the voting paradox). Different methods for avoiding the voting paradox have been studied in the field of social choice theory.

In 1959, Kemeny [12] proposed a method that naturally extends Condorcet's proposal in the sense that, in case the Condorcet ranking exists, the ranking obtained by the Kemeny method coincides with it. Otherwise, the Kemeny method searches for the ranking that is the closest to all rankings given by the voters. Unfortunately, the computation of the Kemeny ranking is known to be NP-hard [5], as the number of possible rankings to explore increases factorially in relation to the number of alternatives. In this paper, we propose to reduce the execution time of the computation of the Kemeny ranking by limiting the search space using a threshold distance related to the ranking obtained by the Borda Count.

The remainder of this paper is organized as follows. In Section 2, the ranking aggregation methods used in this work are presented. Section 3 discusses how branch and bound algorithms can be applied to compute the Kemeny ranking. Section 4 presents a proposal for reducing the execution time of the branch and bound algorithm detailed in the previous section. Experimental results are shown in Section 5. Final remarks are given in Section 6.

2 The Borda Count and Kemeny methods

2.1 Aggregation of rankings

Consider a set of n alternatives $\mathscr{A} = \{a_1, \ldots, a_n\}$. The voters express their preferences over $\mathscr{A}$ in the form of a **ranking**, which is a complete, reflexive and transitive relation. The ranking can be partitioned into a strict order relation ($a_i \succ a_j$) and an equivalence relation ($a_i \sim a_j$). The list of all the rankings on the set of n candidates expressed by the m voters is called the **profile of rankings**, denoted by π_m^n. As some voters may agree on their ranking, there can be repeated rankings in the profile of rankings. For the purpose of this work, the profile of rankings is summarized into a compact representation with $m' \leq m$ unique rankings, where each $r_i \in \pi_m^n$ is associated with a weight w_i corresponding to the number of voters that expressed the ranking r_i. Hence, it holds that $m = \sum_{i=1}^{m'} w_i$.

The **outranking matrix** is a summarized representation of the profile of rankings based on the pairwise comparison of all the alternatives in $\mathscr{A}$. This representation is typically given in the form of a matrix $\mathbf{O}$, where each element o_{ij} equals the number of times that $a_i \succ a_j$ plus one half of the number of times that $a_i \sim a_j$ in the profile of rankings. Therefore, all non-diagonal elements are such that $o_{ij} + o_{ji} = m$. An example of a profile of rankings and its corresponding outranking matrix is shown in Table 1.

Number of voters	Ranking
3	$a_4 \succ a_2 \succ a_1 \succ a_3$
4	$a_3 \succ a_4 \succ a_2 \succ a_1$
2	$a_4 \succ a_1 \succ a_2 \succ a_3$
1	$a_1 \succ a_2 \succ a_3 \succ a_4$

	a_1	a_2	a_3	a_4
a_1	0	3	6	1
a_2	7	0	6	1
a_3	4	4	0	5
a_4	9	9	5	0

Table 1: Profile of rankings π_{10}^4 given by ten voters on the set of four alternatives $\mathscr{A} = \{a_1, a_2, a_3, a_4\}$.

A **ranking aggregation function** (or ranking rule) is a function that maps the profile of rankings into a unique ranking that summarizes the profile of rankings. This ranking is usually referred to as the *winning* or *consensus ranking*. We often refer to the process of applying a ranking aggregation function to a profile of rankings as a (ranking aggregation) method.

For the purpose of this paper, we distinguish two different types of methods. On the one hand, there are methods whose computation is straightforward based on the calculation of a score/value for each alternative (as in the Borda count method) or for each pair of alternatives (as in the Condorcet ranking). On the other hand, there are methods whose computation involves the calculation of a score/value for each possible ranking on the set of alternatives. A prototypical family of the latter type of methods is that of *distance-based* methods, for which it is necessary to define a distance δ between rankings and the consensus ranking is the one that minimizes the sum of distances to the rankings in the profile of rankings. The most prominent example of this family of distance-based methods is the one proposed by Kemeny [12].

2.2 The Borda Count method

The Borda Count method [6] ranks the alternatives according to the number of times that another alternative is ranked at a worse position. More specifically, the Borda score α_{a_i} of each alternative a_i is computed as $\alpha_{a_i} = \sum_{j=1}^{n} o_{ij}$, where o_{ij} represents the element at the i-th row and j-th column of the outranking matrix. Subsequently, the alternatives are sorted by decreasing order of their scores to obtain the consensus ranking. Note that, as two alternatives may have the same Borda score, some alternatives might be tied in the obtained consensus ranking.

For example, given the profile of rankings shown in Table 1, the Borda score of each alternative is $\alpha_{a_1} = 10$, $\alpha_{a_2} = 14$, $\alpha_{a_3} = 13$ and $\alpha_{a_4} = 23$. The Borda Count ranking is: $a_4 \succ a_2 \succ a_3 \succ a_1$.

2.3 The Kemeny method

Condorcet stated that, if voters are just expressing rankings, the only valid information that can be exploited is that given by the pairwise comparisons between the alternatives. Therefore, according to Condorcet, an alternative a_i should be ranked at a better position than another alternative a_j in the consensus ranking if $o_{ij} > o_{ji}$. Unfortunately, the relation defined in this way is not necessarily transitive, i.e., it might happen that $a_i \succ a_j$, $a_j \succ a_k$ and $a_k \succ a_i$.

To avoid this so-called voting paradox, Kemeny proposed a distance-based method for the aggregation of rankings. Firstly, he defined a distance between two rankings [12], which is currently known as the **Kemeny distance**. Formally, given two rankings r_i and r_j on the set of alternatives $\mathscr{A}$, the Kemeny distance annotates for each pair of alternatives $a_k, a_\ell \in \mathscr{A}$ a certain number of points $d_{r_i,r_j}(a_k, a_\ell)$ according to their order in both rankings, as shown in Table 2.

	$a_k \succ_j a_\ell$	$a_k \sim_j a_\ell$	$a_\ell \succ_j a_k$
$a_k \succ_i a_\ell$	0	1	2
$a_k \sim_i a_\ell$	1	0	1
$a_\ell \succ_i a_k$	2	1	0

Table 2: Points annotated for the Kemeny distance for all rankings r_i and r_j of two alternatives a_k and a_ℓ.

By summing these values for every pair of alternatives, the Kemeny distance $\delta(r_i, r_j)$ between two rankings is defined as

$$\delta(r_i, r_j) = \sum_{k=1}^{n} \sum_{\ell=k+1}^{n} d_{r_i,r_j}(a_k, a_\ell) \,. \tag{1}$$

As an example, we illustrate the computation of the distance between the rankings $r_1 = a_4 \succ a_3 \succ a_1 \succ a_2$ and $r_2 = a_2 \succ a_3 \sim a_4 \succ a_1$. The comparison of the alternative a_1 with all other alternatives yields $2+0+0 = 2$, as the order of a_1 and a_3 and of a_1 and a_4 coincides in both rankings but the order of a_1 and a_2 is the opposite. Afterwards, a_2 is compared with a_3 and a_4, which adds $2+2 = 4$ more points. Lastly, a_3 and a_4 are compared, adding 1 more point as they are tied just in one of the rankings. This gives a final distance of, $\delta(r_1, r_2) = (2+0+0)+(2+2)+(1) = 7$.

The Kemeny distance of a ranking s to the profile of rankings π_m^n is defined as the sum of the Kemeny distances of s to all the rankings $r_i \in \pi_m^n$, as shown in Equation (2):

$$\delta(s, \pi_m^n) = \sum_{i=1}^{m'} w_i \cdot \delta(s, r_i) \,. \tag{2}$$

Equivalently, the Kemeny distance from one ranking s to the profile of rankings can be given in terms of the outranking matrix $\mathbf{O}$, as follows:

$$\delta(s, \pi_m^n) = 2 \sum_{i=1}^{n} \sum_{j=1}^{n} o_{ji} \cdot x_{ij} \,, \tag{3}$$

where $x_{ij} = 1$ if $a_i \succ^s a_j$ and 0 otherwise; which is the sum for each alternative a_i of all the elements o_{ji} at the i-th column of the matrix, such that the alternative a_i is ranked at a better position than a_j.

For example, the distance of the ranking $a_4 \succ a_2 \succ a_1 \succ a_3$ to the profile of rankings in Table 1 can be computed as follows. Firstly, the sum of all the elements at the fourth column of the matrix $(1+1+5 = 7)$ represents the disagreement with a_4 being ranked at the first position. Thus, all the rankings of the form $a_4 \succ \ldots$ will have at least a distance to the profile of rankings of 7. Secondly, the sum of all the elements at the second column of the matrix but the fourth one $(3+4 = 7)$ represents the disagreement with a_2 being ranked at the second position (bearing in mind a_4 is ranked at the first position). Finally, the third element of the first column (4) represents the disagreement with a_1 being ranked at the third position (bearing in mind a_4 is ranked at the first position and a_2 is ranked at the second position). This results in a Kemeny distance of $2((1+1+5)+(3+4)+(4)) = 36$.

In order to determine the Kemeny ranking, it is necessary to explore all the possible rankings on the set of alternatives and find the ranking that minimizes the Kemeny distance to the profile of rankings. Young [18] observed that the Kemeny method returns the Condorcet ranking in case it exists and claimed that this method was actually what Condorcet had in mind (but was not able to formalize) back in the 18th century.

Notice that the Kemeny distance is sometimes divided by two in order to agree with the so-called Kendall distance between rankings (without ties). As the use of this constant does not have an impact on the Kemeny ranking, the Kemeny distance divided by 2 is used for the remainder of this paper.

3 A branch and bound algorithm for computing the Kemeny ranking

3.1 Branch and bound algorithms

The main drawback of the Kemeny method is that it requires to explore all possible rankings on the set of alternatives $\mathscr{A}$ to determine the one(s) which is(are) the closest to the profile of rankings. Therefore, for a set of n alternatives, a total of $n!$ rankings[1] should be explored, thus affecting the computation in terms of execution time and memory storage. The computational complexity of this method has been proved to be NP-hard by Bartholdi (see Theorem 2 in [5]).

The Kemeny method can be framed as a combinatorial optimization problem, as it is based on finding an optimal solution from a finite set of tentative solutions. The set of all tentative solutions is referred to as the *search space* and denoted by S. In order to solve combinatorial optimization problems for which exhaustive search is not tractable, it is necessary to reduce the search space by finding a way to discard some of the possible solutions even before they are explored.

Branch-and-bound (B&B) algorithms aim to optimize the value of an *objective function* by recursively splitting the search space into different branches and subsequently pruning the branches that can be proven not to contain an optimal solution. In the context of the Kemeny problem, the search space S is the set of all possible rankings s on the set of alternatives and the objective function is the Kemeny distance $\delta(s, \pi_m^n)$ from the ranking s to the profile of rankings. The objective function needs to be minimized in order to find the Kemeny ranking(s).

Different branch-and-bound algorithms have been pro-

[1] In this work we consider that the Kemeny ranking does not contain ties, which is a common practice in the literature.

posed to find the Kemeny ranking. Edmond and Manson proposed to split the tree considering for each pair of alternatives all the three possible pairwise relations (better, worse, tied) for the pair of alternatives, subsequently associating each branch with a correlation measure and penalty [10]. This idea was later expanded in [2]. The exploration of prefixes of rankings (i.e., rankings starting from the same alternatives) was firstly introduced in [4]. These prefixes are also used in [14], where the author provides a lower bound for the Kemeny distance associated with each prefix by adding the values $\min(o_{ij}, o_{ji})$ for every two alternatives that have not been fixed by the prefix.

3.2 B&B based on prefixes

In this work, we focus on the branch and bound algorithm that associates with each node a *prefix* ρ that represents the subset of the search space containing all the possible rankings beginning with that prefix. Each prefix has an associated partial distance δ_ρ to the profile of rankings, meaning that all the rankings beginning with that prefix are at least at this distance to the profile.

Given a profile of rankings on a set of n alternatives, the search space is thought of as a rooted tree structure of n levels where each level $\ell \in \{1,\dots,n\}$ of the tree contains nodes representing all different prefixes of length ℓ. The root of the tree (level 0) contains the empty prefix $\emptyset$ associated with the whole search space. The first level of the tree contains n nodes, where each node has a prefix containing one different alternative. Each node of the tree has $n-\ell$ successors, that are obtained by adding to the prefix ρ one of the alternatives in $\mathscr{A}$ that had not been added yet to the prefix ρ.

Let us denote by δ_ρ the partial distance from the prefix ρ to the profile of rankings and by δ^* the best distance for a complete solution (i.e., a ranking) $s \in S$ found until the moment in the exploration of the search space. This distance δ_ρ increases every time that a new alternative is added to the prefix. Therefore, we may use δ_ρ as a *lower bound* for the Kemeny distance of any ranking with the prefix ρ to the profile of rankings. Thus, only the nodes such that δ_ρ does not exceed the *upper bound* δ^* (i.e., $\delta_\rho \leq \delta^*$) need to be explored.

The nodes are explored in a depth-first fashion, i.e., all the children of a node that can lead to a potential solution are explored before its siblings. The algorithm is outlined in the following steps:

1. Set the initial distance as $\delta^* = \infty$.
2. Initialize the *fringe* that stores the list of nodes to explore to the first node to be studied, which corresponds to the first node in $\ell = 1$.
3. Explore the first node ρ in the fringe. Compute the distance $\delta_\rho = \delta(\rho, \pi_m^n)$.
   - If $\ell = n$ and $\delta_\rho = \delta^*$, then add this solution to the list of solutions.
   - If $\ell = n$ and $\delta_\rho < \delta^*$, then overwrite the solution list with the current ranking and update $\delta^* = \delta_\rho$.
   - If $\ell < n$ and $\delta_\rho \leq \delta^*$, then add at the beginning of the fringe all the possible children of this node.
4. Repeat recursively *Step 3* until the fringe is empty.

In the first level of the tree, each node a_i represents all the possible rankings starting with a_i. As every node at the next level must extend its parent node, this means that the algorithm has initially n branches, each of them containing all the possible solutions starting with a different alternative. These nodes are expanded keeping track on the upper bound given by the distance of the best solution found until the moment δ^* and the lower bound of the node being considered δ_ρ. These bounds are used to prune the branches of the search space, eliminating nodes that would create solutions that cannot lead to an optimal solution. Therefore, δ^* is updated with the distance of the first s explored and updated every time that a ranking that minimizes the distance to the profile is found. Every time that the algorithm visits a new node, it checks whether the lower bound computed for that node is smaller than, greater than or equal to the current upper bound found. If the lower bound is greater than the current best solution, represented by the upper bound, then the branch associated with that node can be discarded as it cannot lead to an optimal solution.

4 The Borda Count for reducing the influence of the initial random ranking

A problem of the branch and bound algorithm presented in the previous section is that the execution time heavily relies on where the ‘good’ solutions are placed on the tree. If the ‘good’ solutions are discovered at the end of the exploration, then the execution time will be higher than in the case in which the ‘good’solutions are discovered at the beginning of the tree. This is because a better current solution allows to prune more branches from the beginning and, thus, to reduce the execution time.

In order to minimize the number of tentative solutions to explore, some authors have suggested to initialize the algorithm with a random ranking [3]. However,

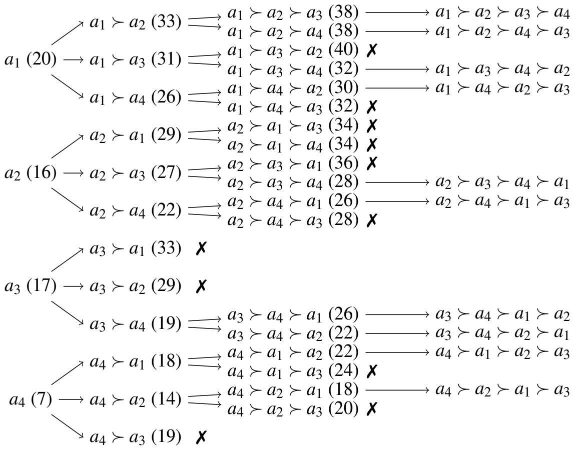
a_1 (20)
$a_1 \succ a_2$ (33)
$a_1 \succ a_2 \succ a_3$ (38)
$a_1 \succ a_2 \succ a_3 \succ a_4$
$a_1 \succ a_2 \succ a_4$ (38)
$a_1 \succ a_2 \succ a_4 \succ a_3$
$a_1 \succ a_3$ (31)
$a_1 \succ a_3 \succ a_2$ (40) ✗
$a_1 \succ a_3 \succ a_4$ (32)
$a_1 \succ a_3 \succ a_4 \succ a_2$
$a_1 \succ a_4$ (26)
$a_1 \succ a_4 \succ a_2$ (30)
$a_1 \succ a_4 \succ a_2 \succ a_3$
$a_1 \succ a_4 \succ a_3$ (32) ✗
a_2 (16)
$a_2 \succ a_1$ (29)
$a_2 \succ a_1 \succ a_3$ (34) ✗
$a_2 \succ a_1 \succ a_4$ (34) ✗
$a_2 \succ a_3$ (27)
$a_2 \succ a_3 \succ a_1$ (36) ✗
$a_2 \succ a_3 \succ a_4$ (28)
$a_2 \succ a_3 \succ a_4 \succ a_1$
$a_2 \succ a_4$ (22)
$a_2 \succ a_4 \succ a_1$ (26)
$a_2 \succ a_4 \succ a_1 \succ a_3$
$a_2 \succ a_4 \succ a_3$ (28) ✗
a_3 (17)
$a_3 \succ a_1$ (33) ✗
$a_3 \succ a_2$ (29) ✗
$a_3 \succ a_4$ (19)
$a_3 \succ a_4 \succ a_1$ (26)
$a_3 \succ a_4 \succ a_1 \succ a_2$
$a_3 \succ a_4 \succ a_2$ (22)
$a_3 \succ a_4 \succ a_2 \succ a_1$
a_4 (7)
$a_4 \succ a_1$ (18)
$a_4 \succ a_1 \succ a_2$ (22)
$a_4 \succ a_1 \succ a_2 \succ a_3$
$a_4 \succ a_1 \succ a_3$ (24) ✗
$a_4 \succ a_2$ (14)
$a_4 \succ a_2 \succ a_1$ (18)
$a_4 \succ a_2 \succ a_1 \succ a_3$
$a_4 \succ a_2 \succ a_3$ (20) ✗
$a_4 \succ a_3$ (19) ✗

Figure 1: Explored nodes when δ^* is set to $+\infty$ or at the worst case scenario when randomly choosing a ranking.

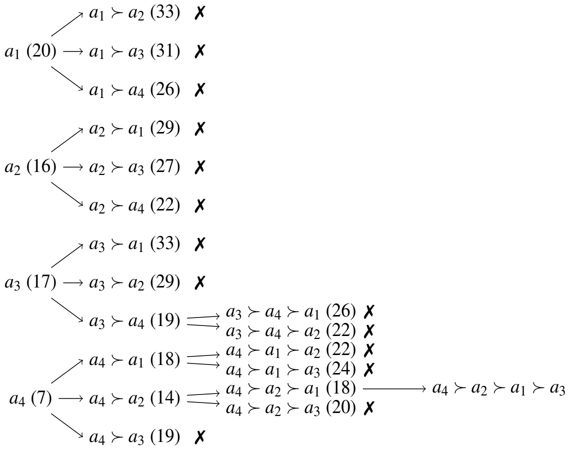

Figure 2: Explored nodes when using the distance of the consensus ranking given by the Borda Count method $a_4 \succ a_2 \succ a_3 \succ a_1$ with distance 20 to the profile of rankings.

due to the randomness of this initialization, there is no guarantee on the quality of the randomly chosen ranking. Actually, in the worst case scenario, where the random ranking is the one that is the farthest to the profile of rankings, the exploration of the tree is as inefficient as the case in which no initial bound is given and, therefore, there would be no guarantee to find the optimal solution in a reasonable execution time.

In this work, we propose to initialize the B&B algorithm with a value of δ^* equal to the distance of the ranking obtained by the Borda Count. This choice is justified since, even though the Borda Count is not a Condorcet method (i.e., it does not need to agree with the Condorcet ranking in case it exists), the Borda Count assures that the Condorcet winner is ranked at a better position than the Condorcet loser in case both of them exist [15]. Furthermore, even if there is no Condorcet winner or loser, it has been proven that the Borda Count always ranks the Kemeny winner at a better position than the Kemeny loser [16]. Conversely, the Kemeny method always ranks the Borda Count winner at a better position than the Borda Count loser. Moreover, it has already been acknowledged that the Borda Count method provides a good approximation [9] of the Kemeny method. Due to these results concerning the rankings obtained by the Borda count and the Kemeny method, it can be presumed that the use of the distance of the Borda Count ranking to the profile of rankings should give better results in general than that given by a randomly-chosen ranking.

An example of the explored nodes for the profile of rankings of Table 1 in the case in which δ^* is initialized as $+\infty$ (or, equivalently, we are at the worst case scenario in which the chosen random ranking is the one that maximizes the distance to the profile of rankings) is shown in Figure 1. Figure 2 shows the nodes that are explored in the case in which δ^* is initialized as the distance of the Borda Count ranking to the profile of rankings. Note that this approach could be useful not only to branch and bound algorithms but also to other types of recursive algorithms [3] that may reduce the execution time when constraining the search space by adding a precondition based on the distance to the profile of rankings.

5 Experiments

5.1 Experimental setting

To measure if there is any improvement when using the proposed initialization in practice, we have developed an implementation of the described branch and bound algorithm (in Python 3.8) that receives an initial distance threshold δ^*, which is used to prune the branches that cannot lead to an optimal solution. For the range of alternatives $n \in \{8,9,10,11,12\}$, we have considered 20 different profiles of rankings with n alternatives, having a total of 100 different profiles of rankings. For each profile of rankings, the algorithm is executed in two different forms: 1) initializing the algorithm with δ^* as the distance of the Borda Count ranking to the profile of rankings; 2) initializing the algorithm with $\delta^* = +\infty$. Recall that the Borda Count ranking may contain ties and, if this is the case, the linear extension of the ranking obtained after ordering the tied alternatives in lexicographical order is considered.

The execution time of the algorithm for each profile of rankings with the two described initializations has been measured. As the execution of the algorithm for some values of n only takes a few seconds, this measurement may be affected by other tasks being performed by the computer. To make this measurement as fair as possible, for the range of $n \in \{8, 9, 10, 11\}$, for each modality of execution the average execution time after running the algorithm three different times is considered. For the case in which $n = 12$, each profile of rankings has been measured only once due to its higher execution time. Afterwards, for each value of n, the average time obtained for the twenty different profiles of rankings with n alternatives is given.

In [1], the authors stated that the performance of the algorithms does not depend only on the value n, but also on the distribution of the votes expressed in the profile of rankings. They state that the difficulty of finding the winning rankings depends on the concentration of the data around the Kemeny ranking, which has a strong influence on the running time. They established levels of difficulty for the profiles of rankings based on different indices related to the margins of the pairwise victories, the average rank entropy and variance and the Borda score.

In this work, in order to measure the level of agreement between the voters of the profile of rankings, we considered a value based on the margin between the pairwise comparisons of each pair of alternatives in $\mathscr{A}$. The outranking matrix gathers the information of the $\frac{n(n-1)}{2}$ possible pairwise comparisons of the alternatives in $\mathscr{A}$. Due to the constant sum property $o_{ij} + o_{ji} = m$, for each pair (o_{ij}, o_{ji}), one of the elements must be greater than or equal to half of the number of voters and, consequently, the other element must be smaller than or equal to half of the number of voters. This implies that the absolute difference $|o_{ij} - o_{ji}|$ always yields a value in the interval $[0, m]$.

The higher the margin given by $|o_{ij} - o_{ji}|$, the more confident the voters are concerning the victory of the first alternative over the second one. The situation that leads to the maximum possible disagreement between the voters is that in which each of the alternatives is preferred to another one by exactly $\frac{m}{2}$ voters if m is even or $[1 + \frac{m}{2}]$ if m is odd. The lower bound for the margin is thus 0. On the other hand, the most desirable case is that in which for every pair of alternatives the margin is maximum. Therefore, the agreement of the voters is proposed to be measured by

$$\sigma(\pi_m^n) = \sum_{i=1}^{n} \sum_{j>i} |o_{ij} - o_{ji}| \tag{4}$$

for a profile of rankings with n alternatives and m voters, which leads to a value $\sigma \in [0, m \cdot \frac{n(n-1)}{2}]$. The higher the agreement between the voters, the greater the value of σ. In order to ease the comparison of the results for different values of n, we have normalized σ to be in the interval $[0, 1]$.

Furthermore, it is evident that the execution time will be affected by the position of the winning ranking in the search space. When considering the B&B algorithm without initialization, profiles of rankings having as Kemeny winner the first alternative on the set $\mathscr{A}$ will be naturally faster, as an optimal solution is found at the beginning of the exploration, which allows to prune more branches in the remaining of the execution of the algorithm and thus reducing the execution time. On the contrary, when the solution is placed at the end of the search space, the algorithm is more likely to explore more branches until it founds a good lower bound.

As the dimension of the outranking matrices is not altered by the number of voters, the execution time of the algorithm should not be largely affected by this number of voters. For this reason, the number of voters has been kept constant and set to 10.

5.2 Results

To analyze the performance of our proposal, we compared the results of the algorithm in the worst case scenario and when initializing the algorithm with δ^* equal to the distance of the Borda Count ranking to the profile of rankings.

Note that, in the case in which δ^* is initialized as the distance of the Borda Count ranking to the profile of rankings, also the time of computing the Borda Count must be taken into account. However, the computation of the Borda Count method takes around 10^{-4} seconds, even for the profiles of rankings where $n = 12$, making this additional time not significant in the final comparison.

n	Borda			Worst			average
	min	max	average	min	max	average	difference
8	0.17	1.31	0.52	0.23	1.40	0.60	0.08
9	0.90	6.77	3.43	0.95	7.90	3.88	0.45
10	4.00	62.85	23.03	4.64	64.44	25.75	2.72
11	15.74	358.84	97.97	18.44	366.26	111.52	13.55
12	76.91	1755.75	496.73	81.59	1961.70	577.09	80.36

Table 3: Execution time in seconds of the B&B algorithm initialized with the distance of the Borda Count ranking vs. the worst case scenario.

The results of both approaches are shown in Table 3 for each number of alternatives, together with the average difference obtained by both initializations. All the results have been obtained using an Apple M1 processor with 16GB RAM.

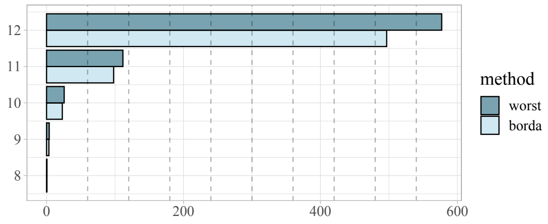

Figure 3: Average execution time in seconds of the branch and bound algorithm, which increases exponentially as the number of alternatives n increases.

The average time for each value of n is presented in Figure 3, which shows that the average execution time increases exponentially as the number of alternatives increases. Dashed vertical lines represent minutes, showing that the average time for the profiles of rankings with $n \leq 11$ alternatives is less than 2 minutes, whereas for $n = 12$ alternatives the average time goes up to almost 10 minutes.

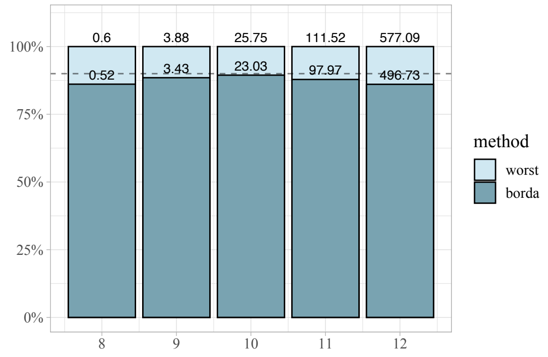

Figure 4: Percentage of execution time reduced with the B&B algorithm when it is initialized with the distance given by the Borda Count ranking.

Figure 4 shows the percentage of execution time taken by the algorithm when initialized using the Borda Count in relation to the average execution time without initialization. The execution time is expressed in seconds. It is shown that, for all the values of n, the time that takes to compute all the Kemeny rankings is reduced by at least 10% (showed by the dashed horizontal line in the plot) when the Borda Count ranking is used to initialize the algorithm.

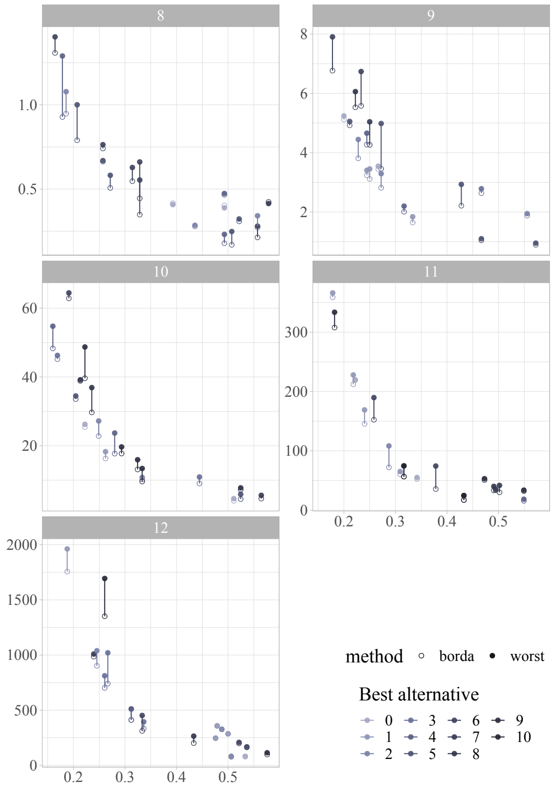

Figure 5: Plot showing the execution time for number of alternatives $n \in \{8, 9, 10, 11, 12\}$ and different levels of agreement between the voters.

The level of agreement between the voters has a strong influence on the execution time taken by the algorithm. Figure 5 shows the execution time for each profile of rankings, where each subplot corresponds to a different number of alternatives, the x axis represents the level of agreement σ (see Equation (4)) between the voters (smaller values are linked to a smaller level of agreement) and the y axis represents the execution time taken by the algorithm (each subplot has each own scale). Filled points represent the execution time taken by the algorithm when considering $\delta^* = +\infty$. Each point is connected to the corresponding execution time for the same profile of rankings when the algorithm is initialized with the distance given by the Borda Count ranking, which is represented by not-filled points. The plots show the influence of the level of agreement, as the execution time increases for all the values of n when the level of agreement decreases. Furthermore, the points are colored according to the alternative at the first position of the winning ranking (for the profiles with multiple winning rankings that have different alternatives at the first position, the first alternative in $\mathscr{A}$ in terms of the lexicographical order has been considered). Darker points indicate that the first Kemeny ranking found for that profile starts with a candidate that is explored later in the tree due to its position in the set of alternatives $\mathscr{A}$. Notice how, at the same level of disagreement, darker points take more execution time than lighter points for the algorithm without initialization, as they take more time to find the first optimal solution. Moreover, darker points tend to have larger lines in connection to their corresponding execution time when initialized with the Borda Count, as using the Borda Count makes the pruning of the branches that exceed the bound easier from the beginning.

6 Conclusions

We have studied the use of the Borda Count ranking to define an initial bound for the computation of the Kemeny ranking within a branch and bound algorithm. The results have shown that, as expected, the execution time is reduced by at least 10% in average for all the number of alternatives studied. Furthermore, there is some evidence that the level of agreement between the voters of the profile strongly influences the execution time. Moreover, we believe it could be interesting to examine other initializations of the algorithm (e.g., based on linear extensions of the transitive closure of the simple majority relation [11, 13] or on the ranked pairs ranking [17]) in order to study whether the execution time could be further reduced.

Acknowledgement

This research has been supported by Grant TIN2017-87600-P from the Spanish Government and PID2019-106263RB-I00. Noelia Rico is supported by the Severo Ochoa program (PA-20-PF-BP19-167).

References

[1] A. Ali, M. Meilă, Experiments with Kemeny ranking: What works when?, Mathematical Social Sciences 64 (1) (2012) 28 – 40, Computational Foundations of Social Choice.

[2] S. Amodio, A. D'Ambrosio, R. Siciliano, Accurate algorithms for identifying the median ranking when dealing with weak and partial rankings under the Kemeny axiomatic approach, European Journal of Operational Research 249 (2) (2016) 667 – 676.

[3] I. Azzini, G. Munda, A new approach for identifying the Kemeny median ranking, European Journal of Operational Research 281 (2020) 388–401.

[4] J. P. Barthelemy, A. Guenoche, O. Hudry, Median linear orders: Heuristics and a branch and bound algorithm, European Journal of Operational Research 42 (1989) 313–325.

[5] J. Bartholdi, C. A. Tovey, M. A. Trick, Voting schemes for which it can be difficult to tell who won the election, Social Choice and Welfare 6 (2) (1989) 157–165.

[6] J. C. Borda, Mémoire sur les Élections au Scrutin, Histoire de l'Académie Royale des Sciences, Paris, 1781.

[7] F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. D. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, 2016.

[8] M. Condorcet, Essai sur l'Application de l'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix, De l'Imprimerie Royale, Paris, 1785.

[9] D. Coppersmith, L. K. Fleischer, A. Rurda, Ordering by weighted number of wins gives a good ranking for weighted tournaments, ACM Transactions on Algorithms 6 (3) (2010) 1–13.

[10] E. J. Emond, D. W. Mason, A new rank correlation coefficient with application to the consensus ranking problem, Journal of Multi-Criteria Decision Analysis 11 (1) (2002) 17–28.

[11] K. Inada, The simple majority decision rule, Econometrica 37 (3) (1969) 490–506.

[12] J. G. Kemeny, Mathematics without numbers, Daedalus 88 (4) (1959) 577–591.

[13] K. O. May, A set of independent necessary and sufficient conditions for simple majority decision, Econometrica 20 (1952) 680–684.

[14] S. V. Muravyov, Ordinal measurement, preference aggregation and interlaboratory comparisons, Measurement 46 (2013) 2927–2935.

[15] D. Saari, Which is better: The Condorcet or Borda winner?, Social Choice and Welfare 26 (2006) 107–129.

[16] D. G. Saari, V. R. Merlin, A geometric examination of Kemeny's rule, Social Choice and Welfare 17 (3) (2000) 403–438.

[17] T. N. Tideman, Independence of clones as a criterion for voting rules, Social Choice and Welfare 4 (3) (1987) 185–206.

[18] H. P. Young, Condorcet's theory of voting, American Political Science Review 82 (4) (1988) 1231–1244.