On the complementarity of order lifting and social ranking: retrieving individual rankings from partial lifted preferences

Experimental results

Ariane Ravier

April 2025

Contents

1 A study of the impact of missing coalitions on performance

We have shown that, given a complete ranking over the powerset of a population X, every social ranking manages to recover the initial ranking over elements from which the ranking over the powerset has been inferred. This result holds for all the studied order lifting methods (minmax, maxmin, leximin, leximax and Borda-sum) as well as for all the studied social ranking methods (lexcel, CP-majority and ordinal Banzhaf).

We now consider the case in which some coalitions may be missing from the ranking over the powerset. For each order lifting method, we randomly generate a ranking over items, from which we infer a ranking over coalitions. We remove a varying number of coalitions from the latter ranking, so that only a given percentage of the coalitions (among the 2^n) remains available: this constitutes the partial ranking given as input to each social ranking method. We then observe the output of each social ranking method and compare it to the correct initial ranking over items.

1.1 Retrieving the exact initial order

We start by studying the impact of missing coalitions on each social ranking method's ability to recover the correct initial order. In this section, we count as a success each occurrence in which a social ranking method has managed to recover the initial ranking over items without any error or imprecision *(e.g. addition of equivalence).*

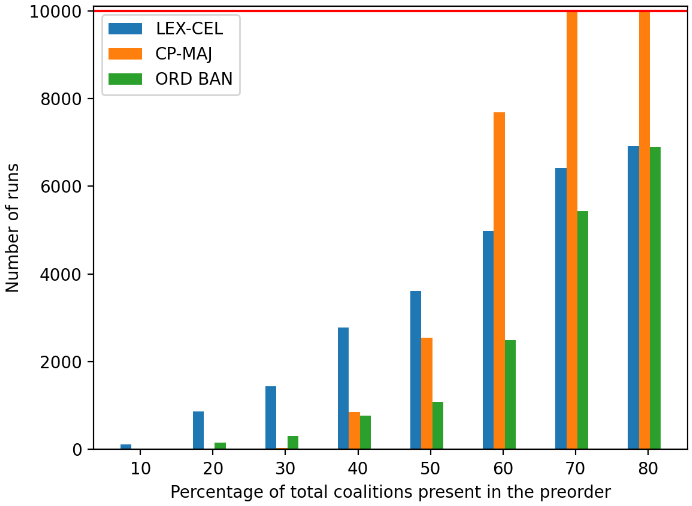
Figure 1 displays experimental results for a population of size 4; Figure 2 for a population of size 8. We already notice changes in conclusions: lexcel performs particularly well over a small population when a relatively small percentage of coalitions are available, while CP-majority appears to be systematically more efficient over a larger population.

Naturally, as the total number of coalitions in the powerset increases, we observe that it is harder for all social ranking methods to determine the correct order based on only a small percentage of all the coalitions. This result is however different when it comes to the Borda-sum extension *(cf. Figure 2e)*, which succeeds in retrieving the correct initial order for over half of the runs with as little as 20% of all the coalitions. This result is due to the weakly additive nature of Borda-sum, to which CP-majority reacts rather well.

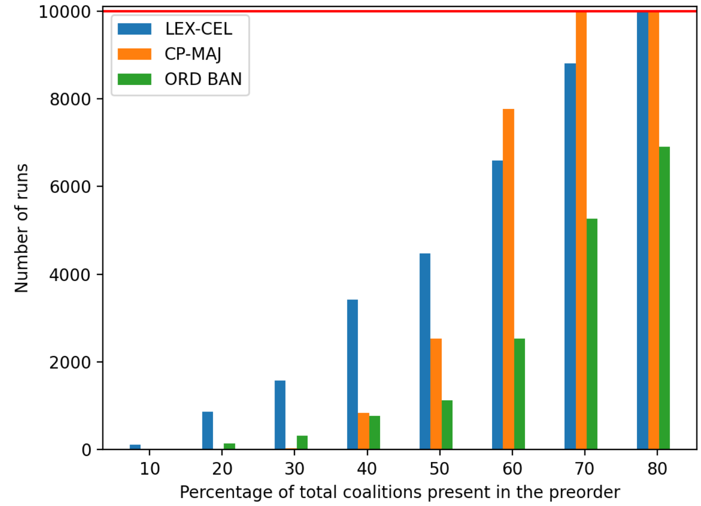
Additionally, these experimental results are also coherent with our theoretical findings, displaying ordinal Banzhaf's poor reaction to preferences extended using Borda-sum *(cf. Figures 1e and 2e)*. What's more, it appears that ordinal Banzhaf is always suboptimal, regardless of the type of extension considered.

Finally, we observe that, over a small population, the performance of each social ranking method (bar ordinal Banzhaf for Borda-sum) remains somewhat similar over each extension. That is not the case, however, when the size of the population increases, as Figures 2a and 2b seem to highlight an impossibility for any social ranking method to find the correct initial order with fewer than 60% of all coalitions - an observation which does not hold for the other extensions.

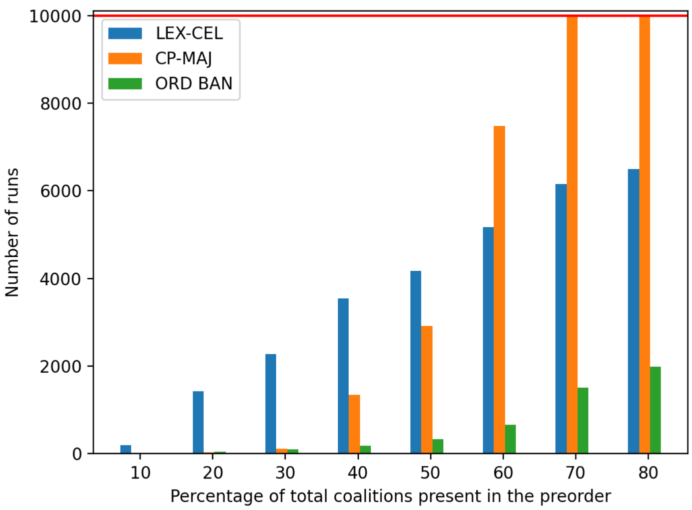
To get a better sense of the difference in performance between all three social ranking methods, we then focus on the Kendall-Tau distance between the ranking uncovered by a social ranking method and the correct initial order. Recall that the Kendall-Tau distance between two rankings $\succ_a$ and $\succ_b$ over an identical population X is the number of pairs $(a, b) \in X$ such that $(a, b) \in \succ_a$ but $(a, b) \notin \succ_b$.



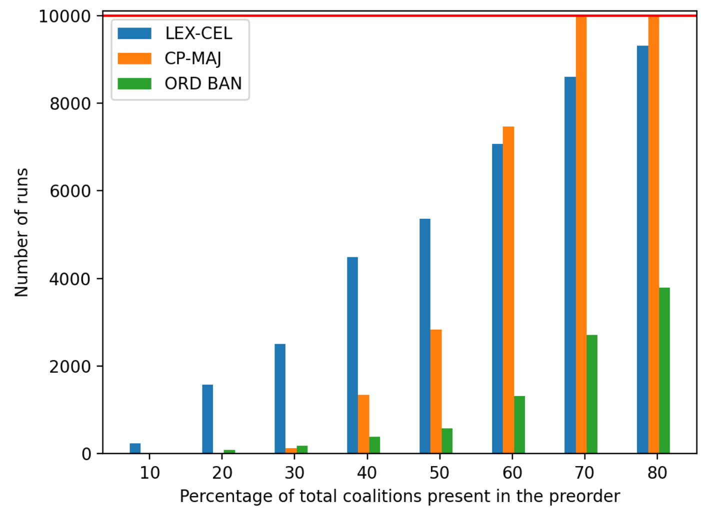
(a) Minmax extension



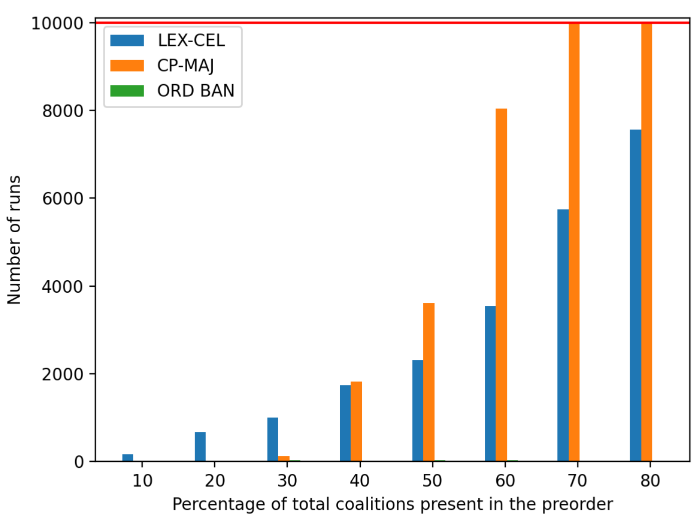
(b) Maxmin extension



(c) Leximin extension

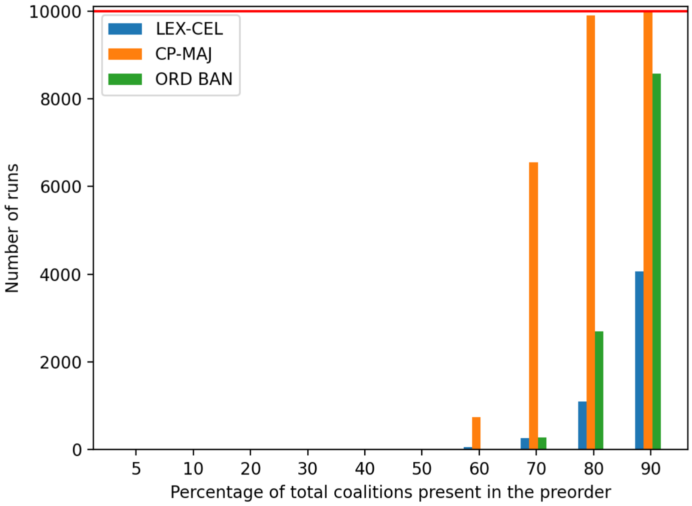


(d) Leximax extension

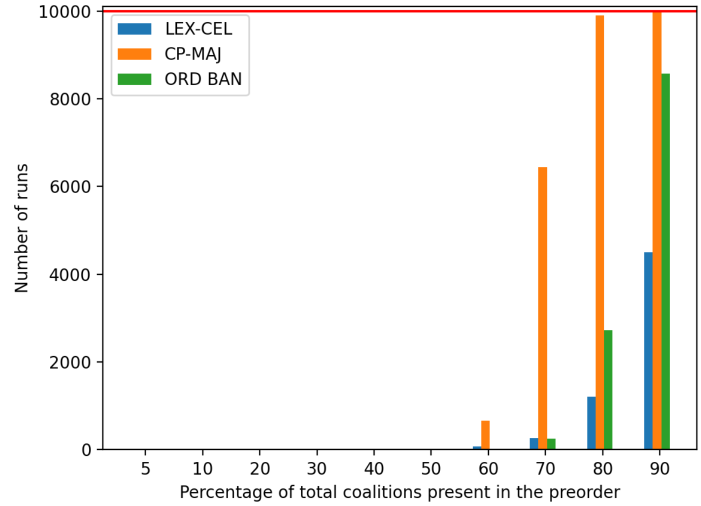


(e) Borda-sum extension

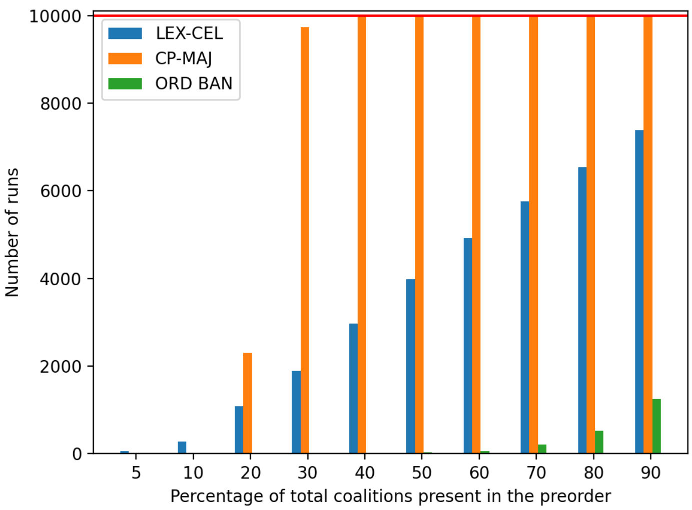
Figure 1: Number of times where each social ranking method finds the correct initial order (over 10000 runs with $|X| = 4$) for each extension



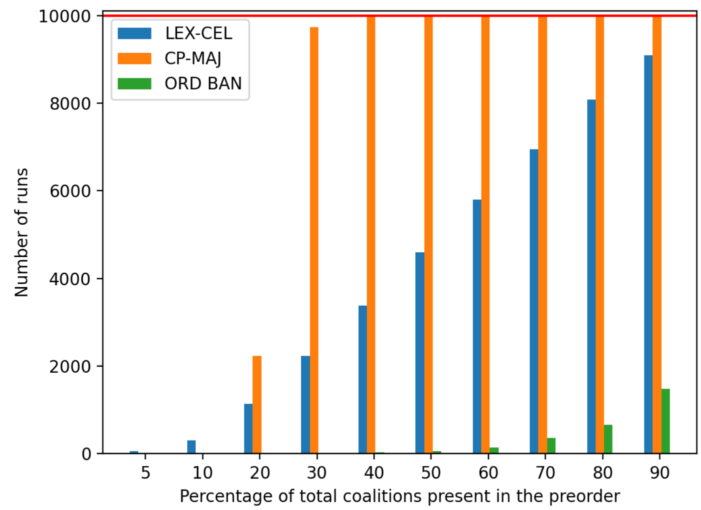
(a) Minmax extension



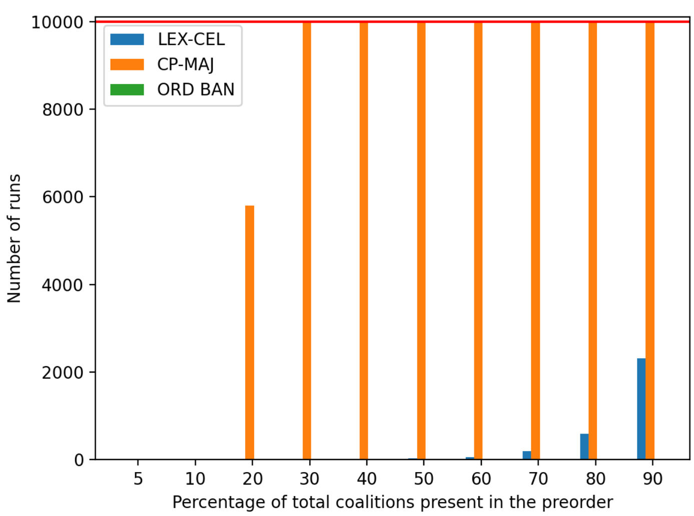
(b) Maxmin extension



(c) Leximin extension



(d) Leximax extension



(e) Borda-sum extension

Figure 2: Number of times where each social ranking method finds the correct initial order (over 10000 runs with $|X| = 8$) for each extension

1.2 A study of Kendall-Tau distance to the correct ranking

We now focus on measuring the distance between the output ranking over items and the initial ranking.

Figure 3 displays experimental results of each social ranking method for a population of size 8.

A surprising observation concerns CP-majority, for which we have shown that no incorrect preference may be returned, *i.e.* if a is preferred to b in the initial ranking, then CP-majority cannot return that $b\ P_{CP}\ a$. Yet, for every extension, and every population size, we observe that CP-majority systematically displays the worst Kendall-Tau distance to the truth when only a few coalitions are available. This is explained by the fact that the Kendall-Tau distance considers equivalence to be a difference: if a is preferred to b in one order, but another order considers a to be equivalent to b, then the pair (a, b) is counted by the Kendall-Tau method. These figures therefore show that CP-majority returns many equivalences, particularly when only a small percentage of coalitions are available.

While this seems a reasonable phenomenon to observe, as the more information is lacking, the harder it will be to determine a strict preference with certainty, we note that lexcel maintains what appears to be a low Kendall-Tau distance, providing distinct preferences between the items.

This leads to a dual conclusion, relative to one's position regarding equivalence and exactitude of results. Indeed, if one favours the robustness of the resulting ranking over its guaranteed expressiveness, then CP-majority appears to be the best method. If, however, one wishes to get clear preferences, with some tolerance for possible errors, then it would appear best to favour lexcel - at least when only a small percentage of all coalitions is available.

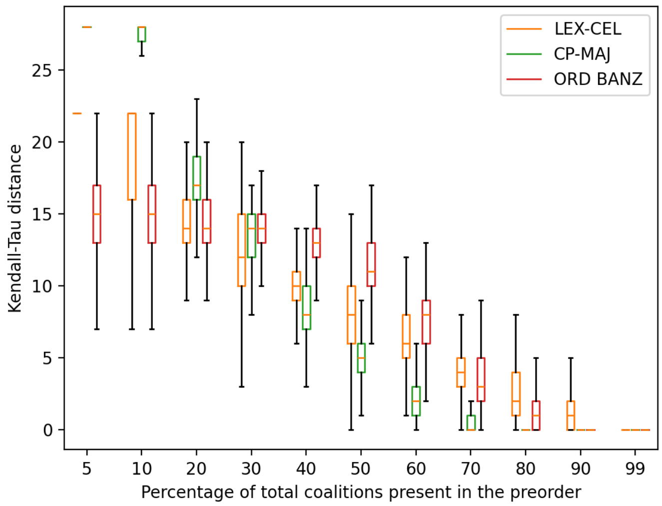
It is worth noting that some pairs of items over which lexcel returns an incorrect strict preference may be such that their strict preference relation is returned correctly by CP-majority. This leads us to introduce another method altogether, which combines the guarantee of CP-majority's strict preference results and lexcel's low Kendall-Tau distance. We choose to combine the two rules, starting from the ranking determined using CP-majority, then using lexcel as a tie-breaker for any pair of items determined equivalent. We also study a version where ordinal Banzhaf is used as a tie-breaker, to compare the performance of both combinations.

Figure 4 displays experimental results of each combination of social ranking methods for a population of size 8.

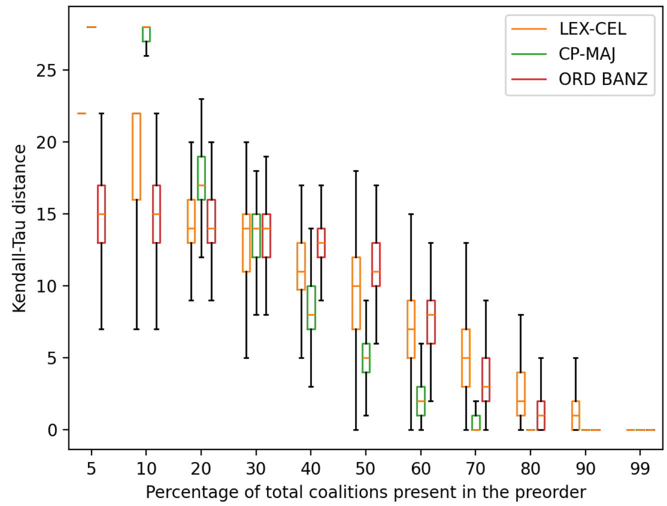
Figures 4a and 4b seem to indicate that the combination of CP-majority and Banzhaf is optimal under the maxmin-based extension, while the combination of CP-majority and lexcel is the optimal method for all other extensions. We also observe that, in any case, both combinations yield results with a lower Kendall-Tau distance to the truth than any social ranking method by itself. This reinforces our assumption that the tie-breaking combination improves the overall result, and leads us to conclude that it is always better to opt for one of these combinations.

All in all, while it appears that CP-majority yields the overall best results by itself, it seems to display a weakness in its propensity to return equivalence when faced with insufficient information. While this may be a reasonable weakness, as one may consider that, in situations of severe lack of information, it is reasonable to answer that strict preferences over items are not retrievable, it can also present as a setback if one prefers to obtain strict preferences with a tolerance for some errors. In the latter case, we observe that using another social ranking method as a tie-breaker on CP-majority's results allow for a more decisive resulting ranking *(in that it will contain more strict preferences)* while maintaining the lowest Kendall-Tau distance to the truth, therefore guaranteeing the highest possible level of correctness of strict preferences among all other methods.

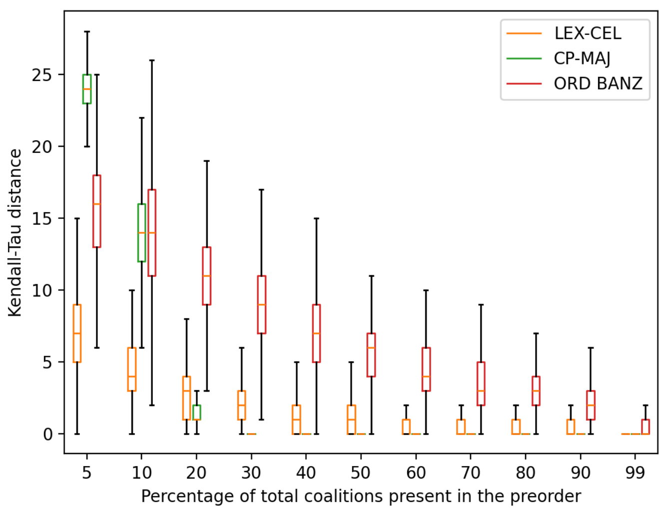
In the final section of this document, we focus on cases where the ranking is not expressed over all the coalitions of the powerset, but only on coalitions of a given size k.



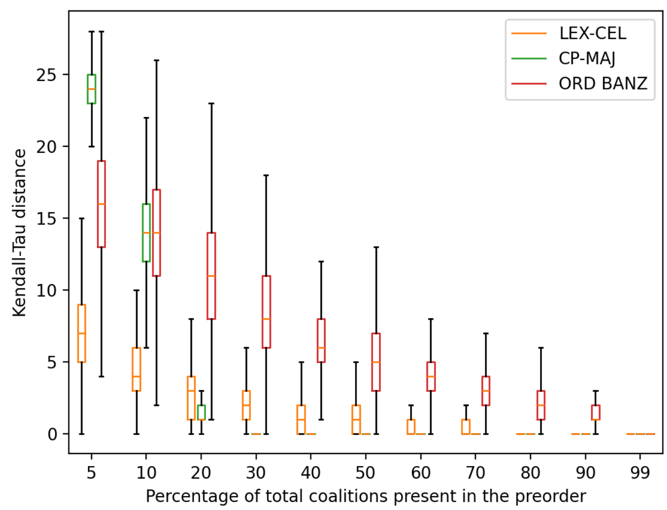
(a) Minmax extension



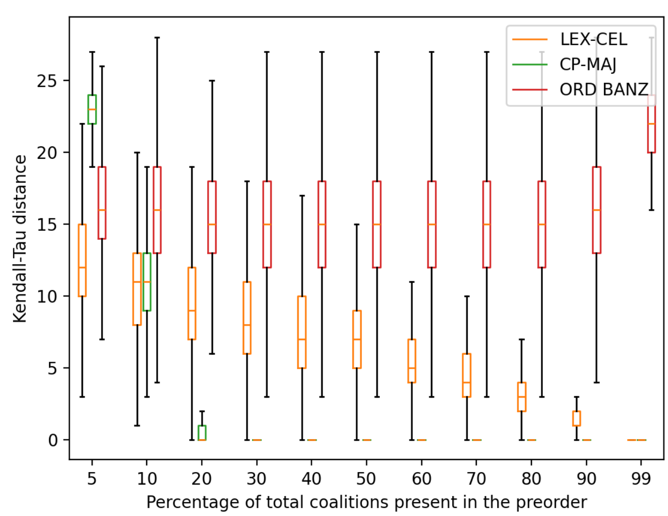
(b) Maxmin extension



(c) Leximin extension

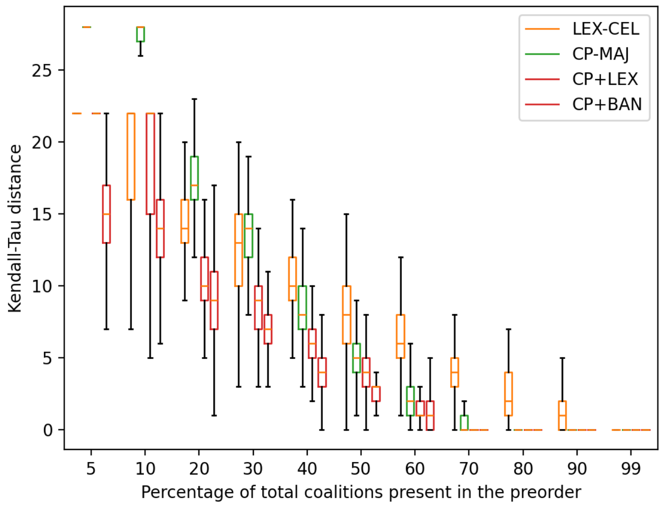


(d) Leximax extension

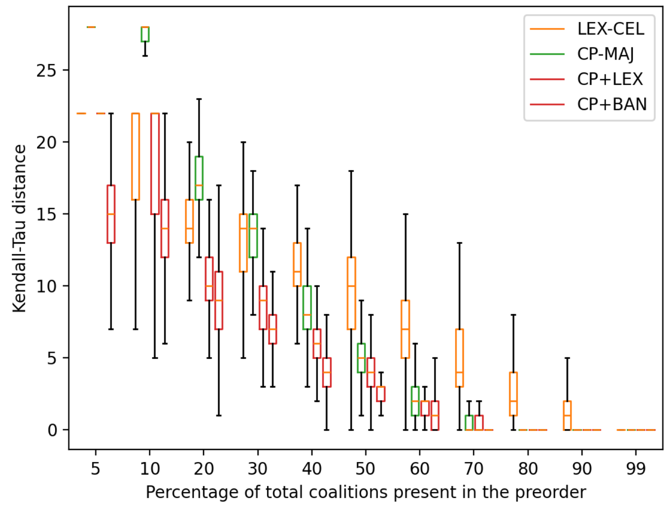


(e) Borda-sum extension

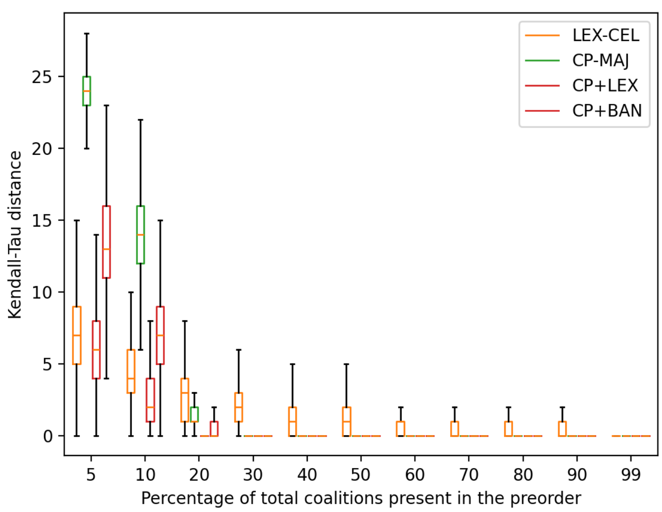
Figure 3: Kendall-Tau distance to the initial order (over 10000 runs with $|X| = 8$) for each method and each extension



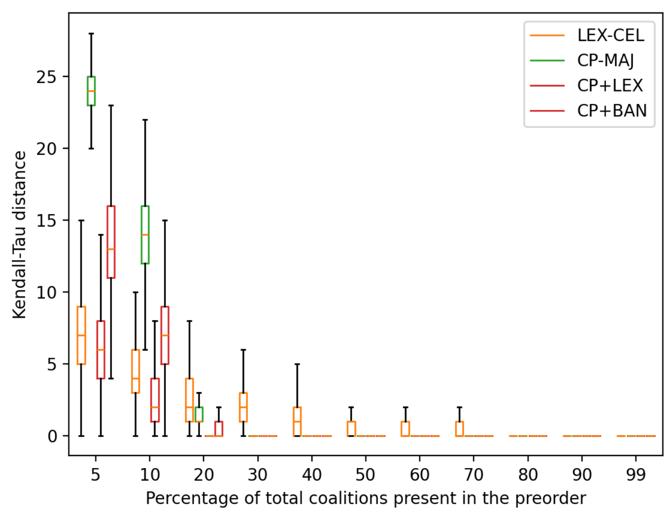
(a) Minmax extension



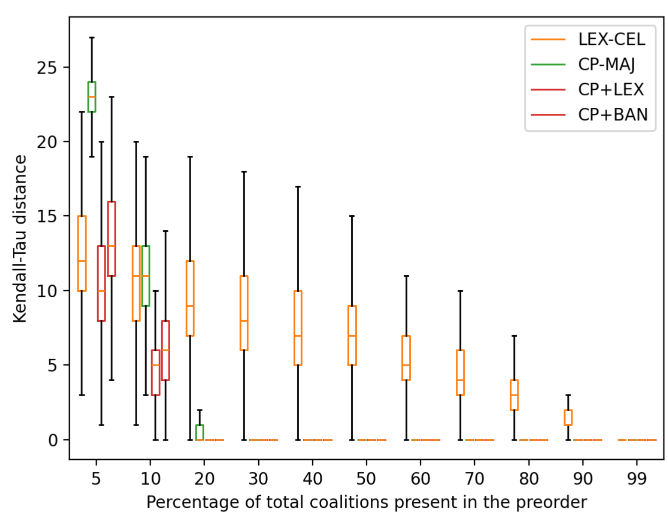
(b) Maxmin extension



(c) Leximin extension



(d) Leximax extension



(e) Borda-sum extension

Figure 4: Kendall-Tau distance to the initial order (over 10000 runs with $|X| = 8$) for each combination of methods and each extension

2 The case of k-sized coalitions

Preferences over coalitions may be limited by structural constraints or hypotheses: the decider may be tasked with forming a team of k candidates, or selecting a bundle of k items. In this particular case, the only available ranking will be over coalitions of size k. We will consider that $k \in \{2, \ldots, |X| - 1\}$, as the reader can easily convince themself that if $k = 1$, then the problem becomes trivial, while if $k = |X|$, it becomes impossible to solve, as there is only one coalition of size $|X|$: X itself.

Recall that ordinal Banzhaf becomes inefficient in this particular scenario: as such, it will not be considered in this section.

Table 1 seems to indicate that, the closer k gets to $\frac{n}{2}$, the easier it gets for each method to determine the correct ranking over items, even with access to only partial preferences over the coalitions. That is particularly true for the minmax and maxmin extensions (respectively rules 1 and 2 in the tables).

This may be explained by the fact that, as k gets closer to $\frac{n}{2}$, the number of coalitions of size k (*i.e.* $\binom{n}{k}$) increases. Therefore, as there are more coalitions to choose from, 10% of that set will represent a larger amount of useable coalitions.

We also note that, while lexcel seems to be (slightly) more efficient with very little information, CP-majority rapidly outperforms it as the amount of accessible coalitions increases.

Finally, while we have only proven the impossibility for minmax and maxmin to determine the total order when $k > \lceil \frac{n}{2} \rceil$, we observe that leximin and leximax seem to also present poor results in some scenarios, such as when $k = n - 1$.

		rule	SR										
N=8	**k=2 (28 coals)**	rule	SR	10% (3)		20% (6)	30% (9)	40% (12)	50% (14)	60% (17)	70% (20)	80% (23)	90% (26)
		1	lexcel	0		25	349	1162	1833	3324	4909	6617	8538
			CP	0		0	0	8	198	4029	9187	10 000	10 000
		2	lexcel	0		3	27	92	180	514	1221	2848	6103
			CP	0		0	0	16	250	3958	9222	10 000	10 000
		3	lexcel	0		23	321	1120	1906	3313	4846	6693	8575
			CP	0		0	0	5	194	4039	9221	10 000	10 000
		4	lexcel	0		1	20	97	180	508	1194	2762	6052
			CP	0		0	0	21	207	3984	9231	10 000	10 000
	k=3 (56)	rule	SR	5% (3)	10% (6)	20% (12)	30% (17)	40% (23)	50% (28)	60% (34)	70% (40)	80% (45)	90% (51)
		1	lexcel	0	77	1799	5331	10 000	10 000	10 000	10 000	10 000	10 000
			CP	0	0	2457	9866	10 000	10 000	10 000	10 000	10 000	10 000
		2	lexcel	0	31	591	3111	10 000	10 000	10 000	10 000	10 000	10 000
			CP	0	0	2362	9860	10 000	10 000	10 000	10 000	10 000	10 000
		3	lexcel	0	17	210	483	972	1623	2629	3840	5364	7529
			CP	0	0	0	16	3421	8816	9973	10 000	10 000	10 000
		4	lexcel	0	4	36	97	226	463	957	1979	3344	6198
			CP	0	0	0	15	3349	8816	9977	10 000	10 000	10 000
	k=4 (70)	rule	SR	5% (4)	10% (7)	20% (14)	30% (21)	40% (28)	50% (35)	60% (42)	70% (49)	80% (56)	90% (63)
		1	lexcel	7	429	7259	10 000	10 000	10 000	10 000	10 000	10 000	10 000
			CP	0	0	10 000	10 000	10 000	10 000	10 000	10 000	10 000	10 000
		2	lexcel	7	261	7329	10 000	10 000	10 000	10 000	10 000	10 000	10 000
			CP	0	0	10 000	10 000	10 000	10 000	10 000	10 000	10 000	10 000
		3	lexcel	3	13	79	220	493	916	1529	2648	4169	6558
			CP	0	0	0	282	6526	9836	10 000	10 000	10 000	10 000
		4	lexcel	3	20	100	215	511	911	1540	2723	4141	6533
			CP	0	0	0	263	6491	9859	9998	10 000	10 000	10 000
	k=5 (56)	rule	SR	5% (3)	10% (6)	20% (12)	30% (17)	40% (23)	50% (28)	60% (34)	70% (40)	80% (45)	90% (51)
		3	lexcel	0	4	35	95	228	471	964	1986	3473	6217
			CP	0	0	0	17	3315	8810	9979	10 000	10 000	10 000
		4	lexcel	0	15	211	473	987	1594	2540	3826	5281	7571
			CP	0	0	0	21	3317	8826	9976	10 000	10 000	10 000
	k=6 (28)	rule	SR	10% (3)		20% (6)	30% (9)	40% (12)	50% (14)	60% (17)	70% (20)	80% (23)	90% (26)
		3	lexcel	0		4	31	86	182	527	1217	2801	6165
			CP	0		0	0	13	218	4052	9125	10 000	10 000
		4	lexcel	0		16	361	1135	1945	3251	4915	6662	8624
			CP	0		0	0	19	210	4064	9158	10 000	10 000
	k=7 (8)	rule	SR	30% (3)				40% (4)	60% (5)		70% (6)	80% (7)	
		3	lexcel	0				0	0		0	1235	
			CP	0				0	0		0	0	
		4	lexcel	0				0	0		0	1245	
			CP	0				0	0		0	0	

Table 1: Number of times (out of 10 000 runs) each SR method finds the exact correct order over parts of the k-sized coalitions, depending on the extension rule (1=minmax, 2=maxmin, 3=leximin, 4=leximax) for $n = 8$