How to Make Condorcet Cycles Vanish and Obtain Grades without Combinatorial Exposure?

Antoine Bernard, Ph.D.
École Polytechnique de Montréal, Montréal (Québec), Canada
CIRRELT – Interuniversity Research Center on Enterprise Networks, Logistics and Transportation, Montréal (Québec), Canada
antoine.bernard@polymtl.ca
438-388-3998

Stéphane Rouillon, Eng., M.Sc.A., Ph.D.
Concordia University
stephane.rouillon@sympatico.ca
514-914-8002

Keywords: Condorcet's Paradox, Arrow's Impossibility Theorem, Median of Differences, Combinatorial Exposure, Incomplete Ballots, Computer Networks, Clustering.

Brief overview: We use rated ballots rather than preferential ones to address Arrow's impossibility theorem, introducing a new method that eliminates Condorcet cycles and obtains a coherent sorted list that can produce grades, even for a huge number of ballots.

Abstract: In elections using preferential ballots, Condorcet cycles can lead to inconsistencies in winner selection. Our approach redefines pair-wise comparisons between candidates by using the median of differences as a new metric. Starting from the median rank for both odd and even sets of ballots, we iteratively resolve pair-wise comparisons, raising the rank when cycles or ties occur, until a coherent result is achieved. In the worst case, this process generates a fully ranked list that reflects the arithmetic average of all ratings, ensuring a single winner except in the case of pure ties. This method renders traditional cycle-breaking techniques like Schulze, Tideman, Minimax, and others obsolete, as it always guarantees coherence. We also present an efficient extension to handle incomplete ballots, enabling scalable grading of candidates even in elections with a huge number of participants. In computer network clusters, the need for an efficient and fair method to elect a leader is paramount, even in the presence of incomplete or inconsistent data from different nodes. This makes our approach highly applicable in leader election protocols for decentralized networks, sensor networks, and other scenarios where robustness and scalability are essential. This combines the teachings from election theory and distributed computing, offering an interesting framework for decision-making in a variety of contexts.

1) Introduction and previous work

In a previous work [1], using rated ballots to evaluate the different options or candidates in an election, we proposed a way to make Condorcet's cycles disappear and guarantee a single winner, with the exception of pure ties obviously. The general idea was to redefine the criterion used to apply pair-wise comparisons between two options. Instead of margins (or relative margins or winning votes), we used a criterion named median of differences.

Starting with a median of rank one (for odd occurrences) or two (refer to [2] for a definition of the median for an even number of occurrences), we try to resolve the pair-wise comparisons. If a cycle appears, we raise the rank and try again until the pair-wise comparisons produce no cycle and no tie. In the worst case, the process will produce a coherent sorted list that represents the arithmetic average of all rates produced by each option. Because a coherent result is guaranteed, the choice between the different cycle-breaking Condorcet methods (Schulze, Tideman, Minimax, Nanson, Kemeny-Young, Dodgson, Copeland, etc...) becomes obsolete.

The result can be extended to produce grades or scores for every option or candidate. However, this extension modifies the algorithm in the extremely rare case when winner(s) would have a lower score than loser(s). This is not a surprise as we know that the preferences of a group of voters can be inherently incoherent as soon as three or more options are available. Arrow's impossibility theorem [3] has shown that a group can (not must) be inherently inconsistent if we refer to individual preferences as a matter of preferences: this is called Condorcet's paradox. It is not surprising then that our algorithm could propose different ordered set as solution of the election. Usually, we stop at the first coherent ordered set. But when the inconsistency $score(winner) < score(loser)$ is the only that remains, if we can ensure that it would be removed in the worst case, why not pursue the algorithm further?

For a complete set of ballots, we know that for full rank, we can obtain at worst $score(winner) = score(loser)$ by averaging all evaluations. Thus, for a complete set of ballots it is recommended to continue the algorithm as long as the inconsistency $score(winner) < score(loser)$ persists. However, for an incomplete set of ballots, we could keep the first obtained ordered set even if it ends up with scores equals for all options, or could continue, if necessary, until the maximal rank equals the number of ballots without any guarantee of removing the inconsistency. We will build examples to show that, however improbable, these cases can happen and can be handled. In fact, a set containing incomplete ballots can produce an ordered set based on averaging all evaluations for each candidate option, with a different number of occurrences per option. In this case, it is based directly on the evaluations and not on the differences between evaluations. Note that for a matter of consistency, this opens the door to complete in an "appropriate way" the set of incomplete ballots to produce an "equivalent" set of complete ballots and then apply the more reliable algorithm with the median of differences. We will suggest several ways to deal with a set of incomplete ballots.

The previous method to determinate scores tried to identify the smallest subset of ballots that would produce the different ranked median of differences. This process, in the case of multiple ballots containing the same pair-wise difference value could be very time consuming: not only several ballots with the same values for the two options or candidate involved in a specific pair-wise comparison can exist, but others with different values but with the same difference can be present too. In addition, the multiplication of all the other values proposed for other options contained in all these ballots is a source of great variation for the averages, an instability that increase the probability to reverse the monotonicity of the score obtained at the extremes of the path of the ordered set. Some ballots could contain only one of the targeted pair-wise comparison, others could contain several. In consequence, the search of the minimal subset of ballots with the needed properties could end up in a huge combinatorial exploration. We want to avoid such situation.

First, we will start with an interpretation of the median of difference method: as we increase the rank of the median to remove Condorcet's cycles, the truncated averages identify a subset of the ballots to compute the new pair-wise differences until we obtain a coherent result. This subset of ballots corresponds to a subset of voters from the electorate that we treat as specialists. Because these voters contribute to at least one median of the coherent final ordered set, we consider the values they proposed as good estimates to compute the final score of each option, as we select a jury of specialist from all the voters at the same time.

Previously, we were taking the median of coherent rank of the evaluations among the smallest subset of eligible ballots proposed by the subset of specialist to compute the final scores of the options. This time, we suggest using only the values of the identified pair-wise values to determinate extreme scores. It simplifies the process and reduces the number of evaluations considered. In real elections with thousands of voters, in almost all cases, winner(s) will have a greater score than loser(s). In this case, winner(s) and loser(s) receive the winning and losing scores respectively. A final mapping, following the coherent path from the winner(s) to the loser(s) and using the medians of the differences of the final rank, establishes the final scores proportionally. This method avoids exposing the algorithm to the risk of combinatorial explosion.

Finally, among many possible applications to everyday life we mentioned the following situations where not only a winner, but the entire ordered set of each candidate option could require a useful score as measure of its position. It is generally a matter of selection, not representation. For example, a process of selections for "democratic" applications could be appropriate for these:

- selection of players or teams for a sport pool (e.g., hockey, soccer, football).
- prioritization of expenses between co-owners.
- awarding of several research grants by university professors.
- assignment of workers to tasks without other constraints.
- designation of winners in Olympic diving or figure skating competitions.
- nominations for a competition, contest or gala awards (e.g., Oscars, Grammies).
- awarding of several similar contracts by a city administration.
- determination of priority tasks for robots on a space mission.
- election of candidates for a multiple-winner election or primary.

As mentioned before, such a method avoids the effects of exaggeration and identifies efficient members of the jury. For example, it could be used at the Olympic diving and figure skating competitions: compared to the actual method [4], in addition of providing grades or scores, it would guarantee the selection – not election, as we evaluate performances – of a Condorcet winner and identify reliable jury members. Beyond traditional election and selection scenarios, this approach holds interesting potential in new situations where we aim to combine different interpretations of a problem from various specialized AI agents. This includes an application in computer networks, particularly in distributed systems where selection of a cluster head plays a critical role in achieving consensus and ensuring reliable coordination. For example, it could identify a leader in cluster swarms such as drones or other vehicular networks. Coherent scoring would be a great tool to build consensus over collective knowledge in many other situations like autonomous driving.

2) The median of differences criteria – Methodology for complete ballots

To generalize understanding, this paper will use examples different than those from previous work [1], but we will apply the same logic with some of the previous examples to compare the results. Before presenting the algorithm, one should note that it works whether the inputs – grade ballots – are normalized or not. However, to promote a fair process, organizers should limit all voters to the same maximal value, typically one hundred is public practice. This does not force all voters to use both maximal (100) and minimal (0) values when evaluating all different options: a typical example is the usage of a reference scale by professors to note students. Fair evaluators have a higher probability to have an influence on the results: in facts we will show some examples where exaggeration has no consequence at all, except identifying the voter as a bad evaluator.

a) Review of the generic method to eliminate incoherent cycles

We propose to use Tideman's locking pair-wise method [5] with the median of differences. Starting with a median of rank one or two, we try to resolve the pair-wise comparisons. If a cycle appears, we raise the rank and try again until the pair-wise comparisons produce no cycle and no tie. For a complete set of ballots, in the worst case, the process can produce a coherent sorted list that represents the arithmetic average of all rates produced by each option. To show the mechanism, we use this first simple example (Table 1):

	Coefficient	A	B	C	
Vote #1	1	100	56	0	(1)
Vote #2	1	0	100	20	(2)
Vote #3	1	62	0	100	(3)

Table 1: Simple cycle example

This set of ballots produces the following Condorcet cycle:

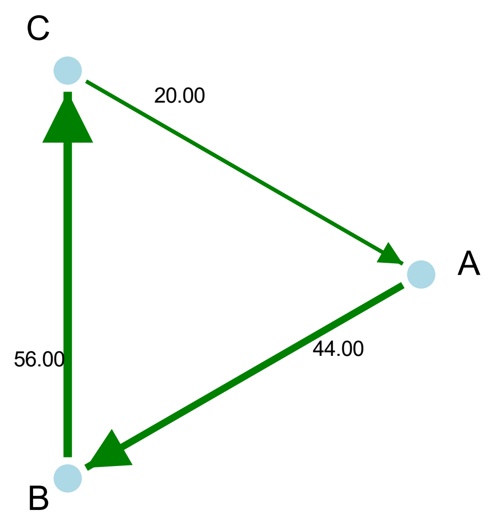

Figure 1: Cycle illustration from ballot extracted from Table 1 – rank 1

This oriented graph (Figure 1) implies that the median evaluator estimates that A beats B by 44 points, B beats C by 56 points and C beats A by 20 points. We write m_k the mean of the k median values, and we call it the median – or truncated average if you prefer – of rank k. As there is a cycle, we will raise the rank of the median of differences to 3. Since we reach maximal rank equal to the number of ballots, the algorithm becomes equivalent to use an average method. The Condorcet cycle hence vanishes to produces a coherent ordered set $A > B > C$. The following oriented graph (Figure 2) shows the median of differences of rank three.

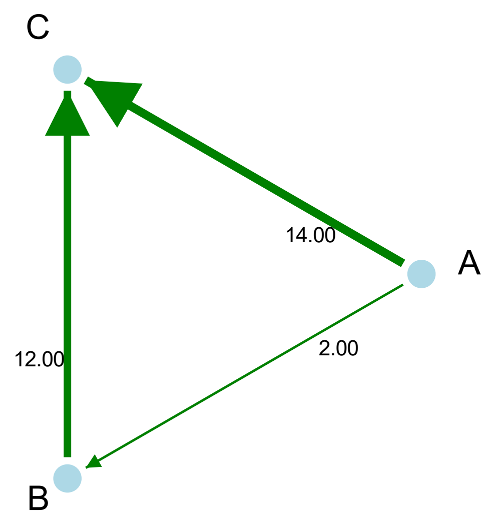

Figure 2: Higher rank illustration from ballot extracted from Table 1 – rank 3

Our new method retains all pair-wise comparisons from each ballot for both extreme values of A and C, thus $A\ (54)$ and $C\ (40)$. Using the coherent path with the final median of each pair-wise comparison, we complete the intermediate score by mapping. With these proportions, all final scores are $A\ (54)$, $B\ (52)$ and $C\ (40)$ as would produce an average method.

b) Obtain an ordered set to identify a coherent path

For our second example (Table 2), we will use one from our previous work where there is no cycle. Let us assume three voters with the following respective sincere satisfaction levels for options A, B and C:

	Coefficient	A	B	C	
Vote #1	1	100	25	0	(4)
Vote #2	1	0	25	100	(5)
Vote #3	1	75	100	0	(6)

Table 2: Ranking illustrative example

Any preferential method that meets the Condorcet criterion would select option B as the winner because it wins both pair-wise comparisons against options A and C with a margin of: $2 - 1 = 1$ preference. However, both average and median methods would select option A as the winner. In the case of the average method, $175/3$ for option A is greater than $150/3$ or $100/3$ for the other options. In this example, the pair-wise comparisons produce this oriented graph (Figure 3):

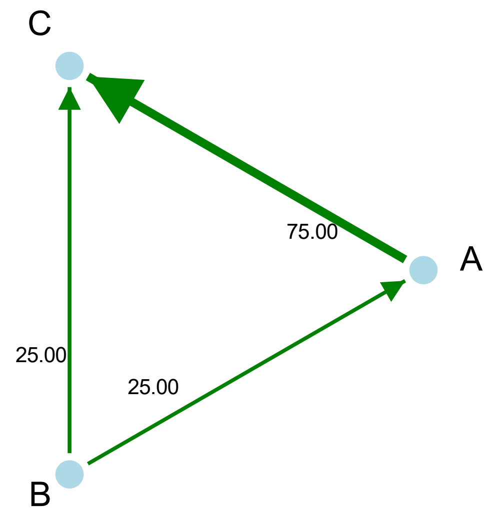

Figure 3: Example graphical illustration of ranking ballots from Table 2

Following our method, as rank one is enough to lock pair-wise comparisons $A > C$ (75), $B > A$ (25) and $B > C$ (25), it produces a coherent ordered set $B > A > C$. Previous method was identifying ballot (6) as the smallest subset of ballot that generates $B > A$ (25) and $A > C$ (75) as median of differences along the coherent path. This provided the following scores: B (100), A (75) and C (0).

c) Values retained from the extreme pairs of the coherent path

Our new method retains both ballot (5) and (6) for the $B > A$ (25) pair-wise comparison and ballot (6) for the last $A > C$ pairwise-comparison. To evaluate the score of B we average both respective B values 25 and 100. To evaluate C only one value is available. Thus, we obtain scores for the extreme options of the coherent path: B (62.5) and C (0). We finalize again the intermediate score with that same mapping, using the coherent path with the median of each pair-wise comparison. Using these proportions, all final scores are B (62.5), A (46.875) and C (0). Although these results are like the previous ones when we compare normalized values, we showed here how to reduce the size of the subset of values used to assign a score to each option corresponding to its final position: the process would consider a lot less combinations when there are millions of ballots.

d) Ordered set producing an extreme incoherent score comparison

We will now consider the rare case where we obtain – apparently – an ordered set but with $score(winner) < score(loser)$ The treatment in the case of a set of ballots containing some incomplete ballots is described later, but in the case of a set of complete ballots we suggest augmenting the rank to pursue the algorithm. We know that in that case, apart from pure ties, the algorithm will produce an ordered set when maximal rank reaches the total number of ballots: the final ordered set might be the same as the previous one or a new one. Table 3 is an example to show that, however improbable, this case can happen.

	Coefficient	A	B	C	
Vote #1	1	2	0	4	(7)
Vote #2	1	7	8	4	(8)
Vote #3	1	6	5	10	(9)

Table 3: Incoherent score example table

Rank one is enough to lock pair-wise comparisons $C > B$ (4), $C > A$ (2) and $A > B$ (1), generating a coherent ordered set $C > A > B$. We retain respectively ballots (7) and (9) for the $C > A$ (2) and $A > B$ (1) pairwise-comparisons.

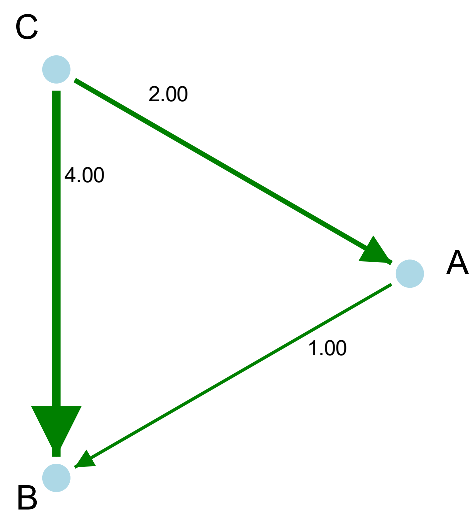

Figure 4: First rank illustration example from Table 3

However, the resulting scores for the extreme options of the coherent path are not coherent: C (4) and B (5)! Pursuing the algorithm for rank three, the ordered set is maintained but the new extreme scores become coherent: C (6) and B (13/3). We finalize again the intermediate score to obtain A (5), producing the same grades as the average method for all options.

Let us use this previous example to build one where the inconsistent scores of the extreme options of the ordered set produces finally a different coherent ordered set (Table 4).

	Coefficient	A	B	C	
Vote #1	1	2	0	4	(10)
Vote #2	1	7	19	4	(11)
Vote #3	1	6	5	10	(12)

Table 4: Incoherent score example table with different final coherent ordered set

Rank one locks the same pair-wise comparisons generating the same coherent ordered set $C > A > B$ with identical resulting scores for the extreme options that are not coherent: C (4) and B (5). Pursuing the algorithm for rank three, a new ordered set $B > C > A$ is produced with coherent extreme scores: B (8) and A (5). We finalize again the intermediate score to obtain C (6), producing the same grades as the average method for all options.

3) An example without strategy

Another aspect of the method based on the median of differences is that it reduces the opportunities for strategic voting. To illustrate this, here is an example without tie to simplify for 11 voters, where no strategic voting was left with our method. These are the sincere ballots (Table 5):

	Coefficient	A	B	C	
Vote #1	4	100	25	0	(13)
Vote #2	2	48	100	0	(14)
Vote #3	1	0	100	51	(15)
Vote #4	2	53	0	100	(16)
Vote #5	2	0	46	100	(17)

Table 5: Sincere ballot example

A is the Condorcet winner, using margin or the median of differences criterion. Groups $V3$ and$V5$ cannot change the result alone. Group $V4$ could make B win but it is not coherent with their sincere preferences. As group $V1$ has no interest into changing the result, only group $V2$ could have some interest in strategic voting. In fact, they can create a cycle if they vote like the $V3$ member. With margin: $B > C$ by 3, $A > B$ by 1 and $A > C$ by 1 becomes margin: $B > C$ by 3, $C > A$ by 3 and $A > B$ by 1. Any cycle breaking method would finally dismiss the $A > B$ result and thus elect B , a preferable outcome for the $V2$ voters.

However, using the median of differences criterion the cycle would maintain as the rank increases, until rank equals five enforces a coherent sorted list from the average of evaluations. Worst case tempered ballots is $V2*$, 2: $B\,(100) > C\,(1) > A\,(0)$. (Table 6, eq. (19))

	Coefficient	A	B	C	
Vote #1	4	100	25	0	(18)
Vote #2*	2	0	100	1	(19)
Vote #3	1	0	100	51	(20)
Vote #4	2	53	0	100	(21)
Vote #5	2	0	46	100	(22)

Table 6: Altered ballot example

Despite full knowledge and 2 voters strategic voting: $A > B\,(89/5)$, $B > C\,(46/5)$ and $A > C\,(4/5)$. For scores: $A\,(206/5 = 41.2) > B > C\,(100/5 = 20)$ and A remains the Condorcet winner. Please note that incentive for strategic voting of at least one group is present with FPTP ($V5$ could obtain B as winner), Approval ($V4$ could obtain C as winner), Borda ($V2$ could obtain B as winner), and any Condorcet cycle-breaking method ($V2$ could obtain B as winner).

In addition, this example shows how the method using the median of differences is resistive to exaggeration in comparison to the classic average method: group $V5$ cannot influence the results even by voting $C\,(1000) > B\,(??) > A\,(-1000)$.

4) Incomplete sets and mixed parities

With a set of complete ballots every pair-wise comparison has several evaluations equal to the number of ballots. However, in the case of a set that contains some incomplete ballots, the main issue is that every pair-wise comparison might have a different number of occurrences. First, we will suppose that there is at least one comparison for any pair-wise comparison because obviously if no joint evaluator rated two options, it is hard to justify their comparison on any basis. Next, we will assume that the highest lower available rank is the best approximation when there is no way to obtain one for the exact rank k: either because parity does not fit or either because there are no more occurrences to estimate a median of higher rank.

a) Interpretation of the absence of evaluation

This brings us to the fundamental question of the interpretation of the lack of evaluation. What meaning to give when there is no evaluation? To treat consequently incomplete ballots, one should understand the different senses of such behaviour. Different interpretations were suggested under the Election Method discussion list [6] for equal ranks and empty ballots. In previous work [7], for truncated ranked ballots, the "all bad options" interpretation seemed the most appropriate. For an unrated evaluation, essentially the interpretations summarize to three options: "no impact" on the pair-wise comparison(s), "all equal" candidates or "all bad" candidates. Since "all bad options" is easy to express by putting zero (0) as a grade and “all equal options” can too be expressed with identical grades, we will retain the "no impact" interpretation for unrated ballots. Any of these interpretations is acceptable: what is important is that voters should know in advance which of these interpretations will be given before voting!!

b) How to treat mixed parities

Because of incomplete ballot sets, both odd and even occurrences must be managed. This mixed parity forces us to redefine the initial rank and the step for rank increase. As long as ties or incoherent cycles remain and some differences are not taken into account to determine the median, we will increase the rank by this step. We then apply the algorithm using both parities median definition. If there are no more available occurrences of a pair-wise comparison we keep the last median evaluation. This continues until a coherent ordered set is obtained or maximal rank is reached. If any pair-wise comparison has even occurrences, initial rank should be two instead of one. Different choices for rank increase lead to different algorithms for a set containing at least one incomplete ballot: it is possible to raise the lowest of odd or even rank once at a time (rank increase step of 1) or both in the same time (rank increase step of 2). At first glance, none has an advantage.

To illustrate this case, we will use an incomplete table of evaluations (Table 7) similar to the one presented in previous work [1]. In our fictive example, professors Valérie (V), Walter (W), Xavier (X), Yoshua (Y) and Zhou (Z) harmonized their scales – thus grades are not normalized to a [0,100] range – and evaluated the students Alice (A), Bernard (B), Carole (C), Daniel (D) and Ernest (E). Let us assume that the process could lead to the awarding of one, two or three grants. Note also that the professors agreed that each student would receive about the same number of evaluations, i.e. four:

Professors \ Students	Alice	Bernard	Carole	Daniel	Ernest
Valérie	**60**	**90**	**20**	**20**	**10**
Walter	**70**	**60**	**80**	**30**	
Xavier			**60**	**10**	**30**
Yoshua	**30**	**50**	**40**		**90**
Zhou	**30**	**20**		**30**	**70**

Table 7: Evaluations of some students by professors – Incomplete ballot case

We will show the resolution for both choices. Computing median of differences for the lowest ranks produces the results from Figure 5:

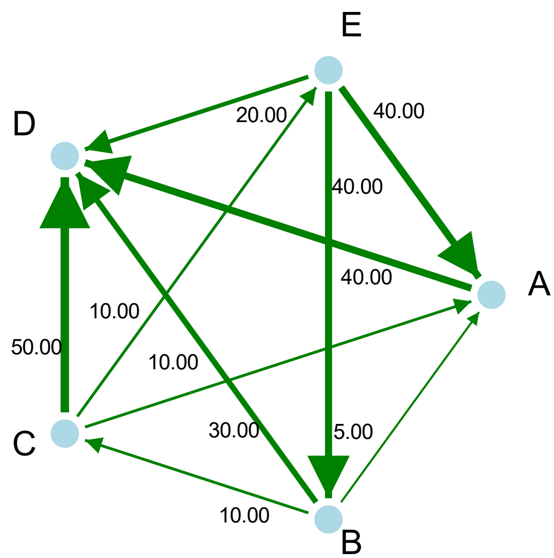

Figure 5: Scoring illustration from Table 7 – Incomplete ballots case

A cycle ($E > B > C > E$) is present, thus we need to increase the rank to obtain a coherent set. First, note that even if every student received four evaluations the number of differences varies between four for the $A - B$ pairwise comparison or three for the others. Starting with rank increase step of 1, the cycle vanishes and we obtain a coherent ordered set: $E > B > A > C > D$. Our new method retains all pair-wise comparisons from each ballot for both extreme values of E and D , thus E (56.67) and D (20). With the proportions of the final median of each pair-wise comparison – $E > B$ (3.33), $B > A$ (5), $A > C$ (6.67) and $C > D$ (33.33), we complete the intermediate scores along the coherent path: E (56.67), B (54.14), A (50.35), C (45.29) and D (20). The next graph (Figure 6) illustrates the pairwise-comparison for all medians of the differences of rank two and three.

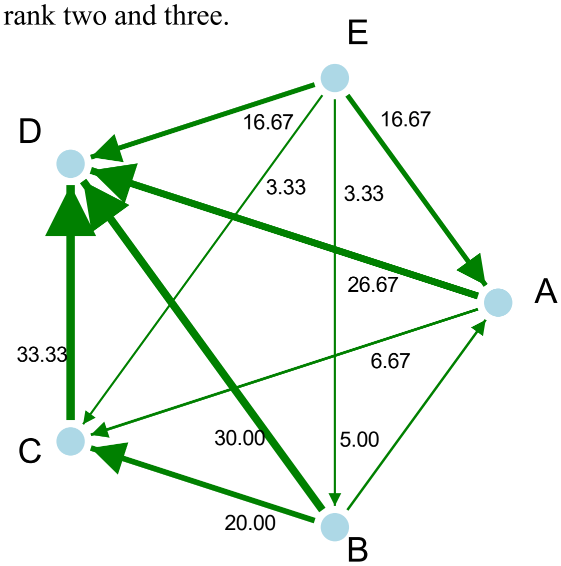

Figure 6: (r2,r3) illustration for incomplete ballot from Table 7

By comparison, the graph is almost identical with the other method when the rank increase step is two and ranks are three and four (Figure 7):

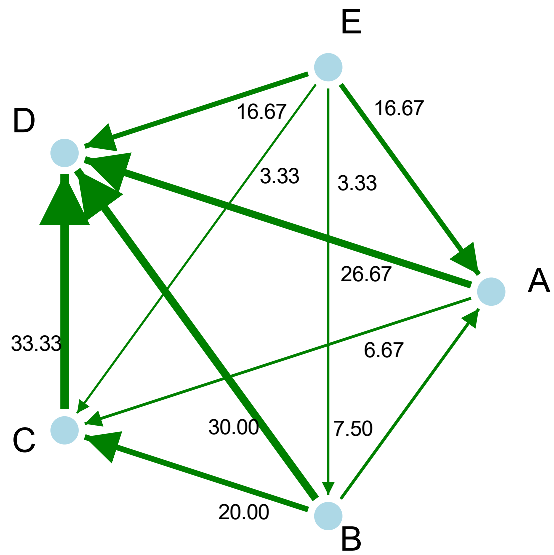

Figure 7: (r3,r4) illustration for incomplete ballot from Table 7

Only the $B > A$ (7.5) pairwise comparison changes. The ordered pair locking method with median of differences still results in the same coherent graph in the next iteration: $E > B > A > C > D$. However, because the $B > A$ pairwise comparison results in a different value with rank four, scores along the coherent path are slightly different: E (56.67), B (54.26), A (48.85), C (44.04) and D (20). This second algorithm is faster to reach maximum rank, as we avoid some intermediate steps. However, there is no justification to prefer any algorithm to the other. Both results are different than what would produce an average method.

c) Convert to an equivalent problem with even pair-wise comparisons

Another approach doubles all ballots. All number of pair-wise comparisons are now even. This allows to work with only even ranked median of preferences, providing a unique new result to the election. However, all pair-wise differences do not have the same number of occurrences, thus there is no guarantee that $score(winner) > score(loser)$ at the end. The double ballot approach uses a rank increase step of 2. The ordered pair locking method with median of differences as comparison criterion results in the same results and cycle $E > B > C > E$ at first for a rank of two. But at the next iteration, for a rank of four, a tie appears (Figure 8):

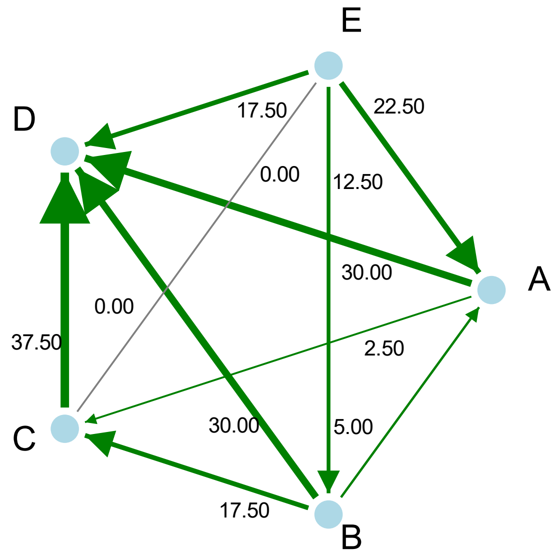

Figure 8: rank 4 illustration for incomplete ballot from Table 7 after doubling all ballots counts

Thus, the algorithm continues with rank equal to six (Figure 9), we obtain a coherent ordered set: $E > B > A > C > D$:

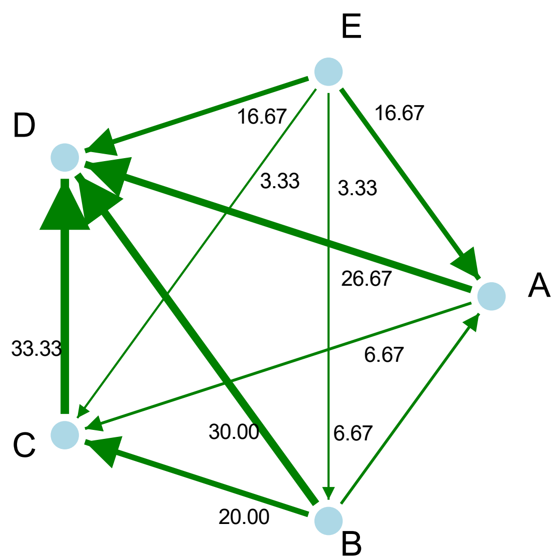

Figure 8: rank 6 illustration for incomplete ballot from Table 7 after doubling all ballots counts

Again, this method retains all pair-wise comparisons from each ballot for both extreme values of E and D, thus E (56.67) and D (20). Using the proportions of each pair-wise comparison for a rank of six – $E > B$ (3.33), $B > A$ (6.67), $A > C$ (6.67) and $C > D$ (33.33), we complete the intermediate scores along the coherent path: E (56.67), B (54.22), A (49.33), C (44.44) and D (20).

However, with these previous methods for an incomplete set of ballots, the last graph of preferences could still contain cycles and/or pure ties and/or incoherent extreme values. Hence, a tie-breaking procedure is necessary: we still recommend a generic, fair, reproducible and safe tiebreaker [8]. For a partial set of evaluations, the incoherency might remain until maximal rank. In this case, we would keep the resulting ordered set… and accept that all options will receive the same final score averaged from the extremes winning and losing scores. This is not a satisfying result. In fact, none of the last three algorithm for an incomplete set of ballots can ensure that we would obtain a complete ordered set as result.

d) Complete ballot equivalence from a set of incomplete ballots

The tendency to make cycles disappear is well highlighted. In general, even with an incomplete set, a coherent ordered set of options should appear as soon as we increase enough the rank of medians. With an incomplete set of evaluations, the tendency is present, but the absence of cycles is not guaranteed. We prefer to avoid such situations and guarantee the absence of cycles.

When the interpretation of incomplete ballots and their proportion permit it, we prefer to complete ballots using the median of all other evaluations [9] for a missing estimation. Hence, the resulting ballot set becomes complete, thus offering a guarantee that a coherent ordered set will be obtained. Let us return to our example and complete our table with the median of the other evaluations:

Professors \ Students	Alice	Bernard	Carole	Daniel	Ernest
Valérie	**60**	**90**	**20**	**20**	**10**
Walter	**70**	**60**	**80**	**30**	*50*
Xavier	*45*	*55*	**60**	**10**	**30**
Yoshua	**30**	**50**	**40**	*25*	**90**
Zhou	**30**	**20**	*50*	**30**	**70**

Table 8: A complete set of evaluations of the students

This approach allows us to apply the complete algorithm and to obtain directly a coherent ordered set as result for rank one: $C > B > A > E > D$.

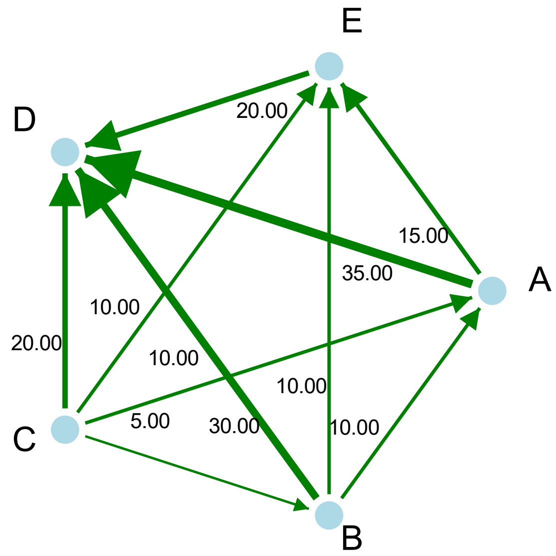

Figure 9: Completed set of ballots from Table 8

Our new method retains the opinion of professor Xavier for the $C > B$ (5) pair-wise comparison and the opinions of both professors Walter and Xavier for the last $E > D$ (20) pairwise-comparison. Thus, we obtain scores for the extreme options of the coherent path: C (60) and D (20). Using the proportions along the coherent path, all final scores are C (60), B (56), A (48), E (36) and D (20).

5) Applicability in other domains and further exploration

Although originally developed for elections, this method generalizes to any setting requiring a robust ranking or selection process. For instance, in distributed systems (e.g., leader election or consensus protocols), it yields stable outcomes despite partial data or inconsistent inputs across nodes. It also applies in multi-criteria decision-making (e.g., awarding research grants, selecting sports teams, ranking competing proposals) by producing coherent scores, even if evaluators use different scales or some ballots are incomplete. Its use of median-based comparisons makes it resilient to exaggeration or manipulation, ensuring fair rankings in various real-world scenarios where reliable, consistent results are paramount.

However, it mostly depends on one's definition of best candidate. For this work we defined the best candidate as the one that best represents the median opinion in comparing duels of candidates. In rare specific cases, where for example the number of candidates and voters are sparse and scores greatly differ, some computer systems might implement other election paradigms using averages instead of medians, which could yield different results.

Finally, computer systems allow for fine-grained delegation, allowing us to create a ranking between hundreds, or even thousands of devices, and to assign tasks based on this ranking, determined through peer evaluation. Additionally, further processing could also help create a possibly even more reliable system, for example by extracting cycles from the graph and running our algorithm on these cycles only instead of improving median ranks over the whole system. Table 9 and 10 show possible results from our system with 99 ballots where rank increase show variations in ranking.

a) Cycle and reverse cycle examples

Table 9 shows an example with 3 candidates and 99 voters. Using our method, we observe a cycle for the first 23 ranks. At rank 24, A becomes the winner with $A > B > C$. The alternative method of taking averages, which is equivalent in our case to exploring the maximum rank (rank 99), would give a similar result in this specific case. However, we could construct election results with a winner in term of median of differences becoming loser when exploring higher ranks.

Exploring further ranks in the example from Table 9 also shows us different orders, with some cycles $A > B > C > A$ and $A > C > B > A$; and some alternative ranking $A > C > B$. In our method, we simply retain the first rank without cycle, in that case rank 24 ($A > B > C$).

	Coefficient	A	B	C
Vote #1	10	10	5	0
Vote #2	11	0	10	6
Vote #3	12	4	0	10
Vote #4	11	100	0	50
Vote #5	12	60	100	0
Vote #6	10	0	40	100
Vote #7	12	0	4	2
Vote #8	10	3	0	4
Vote #9	11	4	1	0

Table 9: Ballot cycles example (3 candidates)

b) Cycle and reverse cycle examples –Extension to 4 candidates with leader changes

Table 10 expands on the above example with adding candidate D. Candidate D is a leader for this set of ballots, despite the cycle, and ends up a definitive winner at rank 24 with the ranking $D > A > B > C$. But we see that when exploring higher ranks, which would be ignored as the winner was already found, the ranking would be changed by the extreme values with D ending up last in the competition from rank 60 and even becoming loser from rank 92 with ranking $A > C > B > D$.

In this case, we observe different results between the correct median of differences from our method and the results from exploring higher ranks up to averaging the differences (here at rank 99). Higher ranks are more sensitive to extreme values.

	Coefficient	A	B	C	D
Vote #1	10	10	5	0	11
Vote #2	11	0	10	6	11
Vote #3	12	4	0	10	11
Vote #4	11	100	0	50	10
Vote #5	12	60	100	0	10
Vote #6	10	0	40	100	10
Vote #7	12	0	4	2	1
Vote #8	10	3	0	4	1
Vote #9	11	4	1	0	1

Table 10: Ballot cycles example (4 candidates)

c) Consensus building in computer networks

Our scoring method can be applied to select a leader or to build consensus in a computer network. It is the case for example with an autonomous vehicule having to take critical decisions in an emergency situation: damaged sensors could need a fix to make sense of unreliable data. That approach can also be used to determine a leader in a computer cluster. Cluster head determination is a subject studied by applying algorithms expanding from real world problem such as the byzantine generals problem [10] from which algorithms were developed such as PAXOS [11] or RAFT [12]. In reliable scenarios, extending our approach to build a coherent scoring is feasible.

In autonomous wireless network however, where devices usually leave and join the network frequently, computing votes might prove challenging. However, we could still apply the scoring method to rank voters opinion about situations such as determining actions or building consensus in autonomous drone swarms. Comparing votes from partial view over a situation measured by drone sensors or vehicular cameras, then determining a course of action based on the consensus built over the network. We could also expand clustering methodology from this approach by grouping devices with similar view to determine complementary mission based on this kind of sensing and determine mission leaders based on the vote, leveraging them as both a clustering and a cluster head election solution.

6) Conclusions

The criterion of median of differences allows us to resolve Condorcet’s paradox using grade ballots. By raising the rank of the median, we can ensure to obtain a coherent ordered set with score(winner) >= score(loser). The coherence property provides the method with several qualities. The process is much less sensitive to strategic voting, especially the exaggeration tactic. Essentially, it reduces strategic voting opportunities to cases where hackers could obtain a full knowledge of the ballots. It can identify specialists among the jury. Finally, this latest implementation avoids the risk of combinatorial explosion when determining the values used to calculate the scores of the winner(s) and loser(s). A set of complete evaluations produces an ordered set with only pure ties to determine the single or multiple winner(s).

In the case of a set of ballots containing at least an incomplete ballot, several solutions are suggested. Even though the method could still work, we recommend that at least one evaluator gave two grades to both options of any pair-wise comparison. Obviously, if any pair of two options was not rated by any voter, it is hard to justify their comparison because each evaluator uses his or her own scale. Within a set containing at least one incomplete ballot, both odd and even occurrences can be present. This mixed parity suggests a different algorithm for both steps of rank increase, one and two. A third algorithm is obtained by duplicating all ballots. The result is an "equivalent" problem with even pair-wise comparisons that can be resolved with a step of rank increase equal to two. Whichever algorithm used, it continues until a coherent ordered set is obtained or maximal rank is reached. However, these three algorithms cannot guarantee a coherent result. Thus, the last algorithm that fills the blanks to obtain full ballots is our favorite because it produces an ordered set and coherent scores for all options. These scores can represent the grades for students, the satisfaction of the electorate concerning some election or some probability of success of a decision regarding a mission. This solution could also be applied to other applications. AI agents in complex systems could use this kind of voting approach [13, 14] to collaborate on a problem and then vote on the solutions based on their own output. This could, for example, apply to combine various experts in a mixture of expert [15] approach in detecting deepfakes in images, with each expert voting on the possibility of images being deepfake based on their partial view of the problem and leveraging our partial ballot approach.

Bibliographical references

[1] S. Rouillon, A. Bernard, Why Break Condorcet Cycles when we Can Make them Disappear?, MPSA 80th Annual National Conference, Chicago, 2023 (April).
[2] https://en.wikipedia.org/wiki/Median#Finite_data_set_of_numbers
[3] Kenneth J. Arrow (1963), Social Choice and Individual Values.
[4] https://www.eurosport.com/figure-skating/beijing-2022/2022/beijing-olympics-2022-how-are-figure-skating-scores-judged-who-chooses-outfits_sto8786724/story.shtml

[5] Nicolaus Tideman, Collective Decisions and Voting, Virginia Polytechnic and State University, USA, 360 pages.
[6] S. Rouillon, A Preferential and Proportional System Without Geographical Divisions, MPSA 65th Annual National Conference, Chicago, 2007 (April), http://www.mpsanet.org/Conference/ConferencePaperArchive/tabid/681/q/Rouillon/Default.aspx
[7] S. Rouillon, Juho Laatu, Defeat strength, Winning Votes vs. Margins, what to do with equal-ranks on the ballot?, [EM] Election Methods discussion list, http://lists.electorama.com/pipermail/election-methods-electorama.com/2019-May/002126.html
[8] S. Rouillon, From Preferential Input to Proportional Output, MPSA 76th Annual National Conference, Chicago, 2018 (April), http://www.votebook.ca/pdf/PrefInputPropOutput-Complete.pdf
[9] https://rangevoting.org/MedianVrange.html, M. Balinski & R. Laraki: Majority judgment, MIT Press, Cambridge MA, 2011.
[10] Lamport L., Shostak R., Pease M. (1982), "The Byzantine Generals Problem" (PDF). ACM Transactions on Programming Languages and Systems. 4 (3): 382–401. CiteSeerX 10.1.1.64.2312. doi:10.1145/357172.357176.
[11] Oki B., Liskov B. (1988). "Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems". PODC '88: Proceedings of the seventh annual ACM Symposium on Principles of Distributed Computing. pp. 8–17. doi:10.1145/62546.62549.
[12] Ongaro D., Ousterhout J., In search of an understandable consensus algorithm. In 2014 USENIX annual technical conference (USENIX ATC 14) 2014 (pp. 305-319).
[13] I. Partalas, I. Feneris, I. Vlahavas, "Multi-agent Reinforcement Learning Using Strategies and Voting," 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), Patras, Greece, 2007, pp. 318-324, doi: 10.1109/ICTAI.2007.15.
[14] Jeremy Pitt, Lloyd Kamara, Marek Sergot, Alexander Artikis, Voting in Multi-Agent Systems, The Computer Journal, Volume 49, Issue 2, March 2006, Pages 156–170, https://doi.org/10.1093/comjnl/bxh164
[15] Masoudnia, S., Ebrahimpour, R. Mixture of experts: a literature survey. Artif Intell Rev 42, 275–293 (2014). https://doi.org/10.1007/s10462-012-9338-y