8-1-2015

Project Ranking Using Partial Ranks

Jiali Ju
Portland State University

Pei Zhang
Portland State University

Tim R. Anderson
Portland State University, tim.anderson@pdx.edu

Citation Details

Jiali Ju; Pei Zhang; Anderson, T. (2015) Management of Engineering and Technology (PICMET), 2015 Portland International Conference on p. 472-477

Project Ranking Using Partial Ranks

Jiali Ju, Pei Zhang, Timothy Anderson
Department of Engineering and Technology Management, Portland State University, Portland, OR - USA

***Abstract*--A competition was recently held for new elder care technologies and a method was needed for selecting an audience favorite project. Since each of the eight projects received a half hour presentation time slot spread out over a full day, attendance varied significantly with morning projects having a much lower attendance. An algorithm was desired that was robust with respect to varying evaluator. A new algorithm for aggregating ranks from a large number of incomplete judgment was developed and applied to the projects to select a winner. This paper presents the new algorithm, tests the new approach against others in the application, and discusses relative tradeoffs.**

I. INTRODUCTION

Voting systems have a long history, and are still being researched. This time, referring to a competition event, we were able to design a new voting system for the ranking process. After designing the paper ballots, distribute to the audiences and collect the data, the new algorithm had been tested.

Regarding to the situation, the new algorithm should be able to deal with partial ranking, tie rankings and determine the full ranking rather than simply pick single winner. The algorithm we designed to address these features, and refer to it as a "Sequential Pair Rank" system, which is a pairwise-comparison-based voting system. We will introduce the algorithm, and study its abilities. This system has also been coded on R platform, and the computational performance was also examined.

The Sequential Pair Rank system appears to be new among the single-winner voting systems. Warren D. Smith defines a single winner in an election as "the candidate maximizing the total sum (over all voters) of benefit, wins [1]." When people think of voting, several system are frequently thought of, including Majority Voting [2] (the candidate gets the most vote wins), Borda Count [3] (award N-K points to the Kth-ranked candidate in the vote with the preference order, picking the maximum points as the winner), and Weighted Voting [4] (award weight points from Rank 1 to N, count the number of ranks that each candidate gets and pick up the maximum score as the winner).

The voting system provides a social choice function that takes as input a profile of cast ballots and produces as output the name of the election winner[5]. This new algorithm is based on the principle of the "systems in which each vote is a preference ordering of the candidates" [1]. The preference orders from each judge is then compared against every permutation of the eight projects (candidates). By providing the most-fitted full ranks for the candidates, which would provide a new angle of view on different voting systems for the future studies.

II. LITERATURE REVIEW

There is a great range of voting systems developed over the centuries. Warren D. Smith has divided them into five categories of single-winner voting systems [1]:

1. Systems that ignore the votes;
2. Systems in which each vote is the name of a single candidate;
3. Systems in which each vote is a preference ordering of the candidates;
4. Systems in which each vote is a real N-vector;
5. Sarvo-Range voting.

The new algorithm – the "Sequential Pair Rank" would fit in the third category, "Systems in which each vote is a preference ordering of the candidates". Even within just this one category, there are about 30 different single-winner voting systems with several very different kinds of approaches.

Several systems were based on pairwise-comparison, which provide the comparing processes between either a random pair of the candidates, or among each pair of the candidates. Our Sequential Pair Rank system is also based on this methodology. Some popular voting systems with the pairwise comparison are: Gibbard Random Pair (Allan Gibbard, 1973) [6], Black's System (Duncan Black, 1958) [7], Improvement of Dodgson's System, A. H. Copeland's System (A. H. Copeland, 1951) [8], Arrow-Raynaud Pairwise Elimination (Kenneth J. Arrow & Herve Raynaud, 1986) [9], Smith Sets, and Banks Sets (J. Banks, 1985) [10].

Several systems were associated with a scoring system by awarding different scores to different rank orders, and pick the candidate with the maximum score as the winner. Some popular voting systems with this procedure are: Nauru (Benjamin Reilly, 2002) [11], Borda Count (1781) [3], Condorcet Least-reversal System (1785) [12], and Nanson-Baldwin Elimination (Edward J. Nanson, 1882) [13].

Several of the systems were using the majority voting, considering the candidate which gets the most votes as the winner. The usages of probabilities were extended the application ways of the majority voting in practices. Some widely used voting systems with the majority voting procedures contains: Instant Runoff Voting (Nicolaus

Tideman, 1995) [14], Coombs STV System (C.H. Coombs, 1954) [15], Bucklin, and Woodall's Descending Acquiescing Coalitions method (Douglas Woodall, 1997) [16].

There are also some existing algorithms with similar logic to the Sequential Pair Rank. One example would be Gibbard Random Pair (Allan Gibbard, 1973) [6]. In this system, each vote is a preference order among the candidates. Select 2 candidates at random (all pairs equally likely), then perform a 2-candidate election among them by ignoring the other N-2 candidates in each preference ordering [1]. Similar to our system, Gibbard Random Pair uses a pairwise-comparison procedure for the final winner. The key difference is that Gibbard Random Pair selects only 2 candidates out of the N candidates. From the literature review, we haven't found any algorithm which could tolerate partial rank, ties and determine the full ranks at the same time as the Sequential Pair Rank System. The detailed procedure of this algorithm would be described in the following part.

III. ALGORITHM: SEQUENTIAL PAIR RANK

As it is mentioned, each ballot is a preference order list of all candidates. If there are N candidates, the number of the full permutation of all possible votes would be N!. Comparing the orders in the ballots with each permutation's order of N candidates by matching the preference order of each sequential pair. The next step is to accumulate the number of preference order fit for each permutation. The goal is to find the maximum fit number among all permutations and return the optimal permutation(s).

Procedure:

1. Generate the permutations of all possible votes, P .
2. V is the list of preference order from ballots. V_j^k is the order/rank of the *k*th candidate given by the *j*th voter. P_m^k is the rank/order of the *k*th candidate in the *m*th permutation. F is the fitness matrices for each candidate's preference order. $F_m^k = 1$ means the preference order of *k*th candidate and *(k+1)th* candidate in the permutation does fit with the preference order of the same adjacent pair in the ballot (i.e., either both in ascending order or both in descending order). So a non-fit, $F_m^k = 0$, is defined as the rank order of pair in permutation violates the order of same pair in the ballot. If any rank in each pair is 0, in the case of partial ranking ($V_j^k \cap V_j^{k+1} = 0$), then the order is treated as a non-fit. If the rank is a tie in each pair, the order is also considered as a non-fit.

   If $V_j^k \cap V_j^{k+1} \neq 0$ and $\dfrac{V_j^k - V_j^{k+1}}{P_m^k - P_m^{k+1}} > 0$, $F_m^k = 1$; otherwise, $F_m^k = 0$.

   K = 1, 2, 3… N-1; m = 1, 2, 3… N!
3. The optimal ranking is the maximize sum value of F_m:

   $$\max_m \sum_{n=1}^{n} F_m$$

   n = 1, 2, 3…N-1
4. Return the *M*th permutation if F_M is the maximize value of Step 3.

Example:

There are 3 candidates and 4 voters. N=3, N! = 6. The tables below show the process to calculate the fit value F. The sum fit of permutation 1 is 4. Then repeat the process for the full 6 permutations to find the maximum sum fit value.

Step 1: Generate the permutations of 3 candidates ranks

TABLE 1A: RANK 3 CANDIDATES BY 4 VOTERS

	Candidates		
Voters	N1	N2	N3
1	1	2	3
2	1	1	2
3	0	1	3
4	1	3	2

TABLE 1B: PERMUTATIONS OF RANKING 3 CANDIDATES

	Candidates		
Permutations	N1	N2	N3
1	1	2	3
2	1	3	2
3	3	1	2
4	3	2	1
5	2	3	1
6	2	1	3

Step 2: Compare Voter matrix (Table 1a) with Permutation 1 to determine the fit

The rule of fit is to find the same preference orders of the same adjacent pair in both voters' choices and in the permutations. In the fitness matrices, an element equals to 1 represents as the preference order fit and 0 indicates the preference order does not fit. We examine the preference order of each adjacent candidate pair in each permutation with the pairs ranking from all 4 voters. The follow chart are the examples that how to determine fitness.

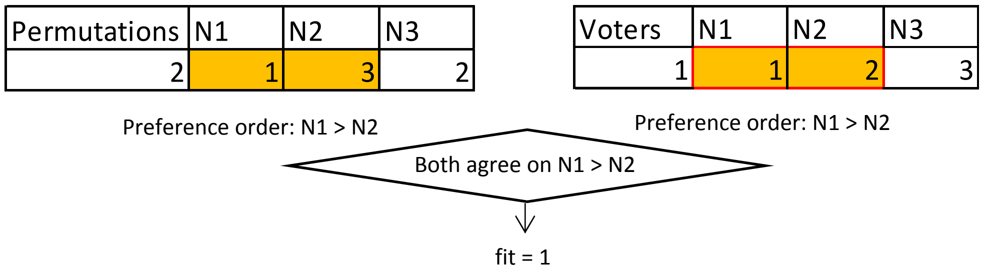

Figure 1a: preference order fit example

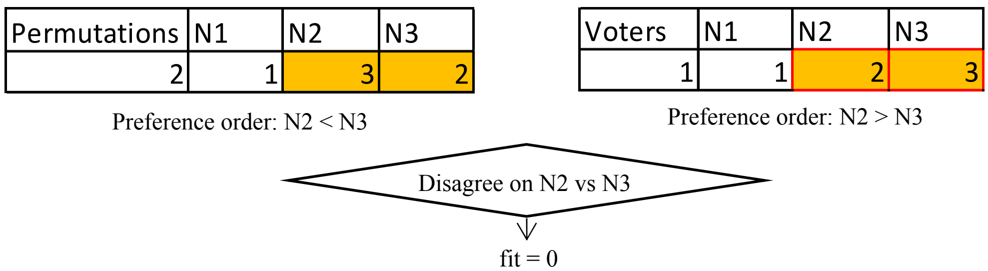

Figure 1b: preference order does not fit example

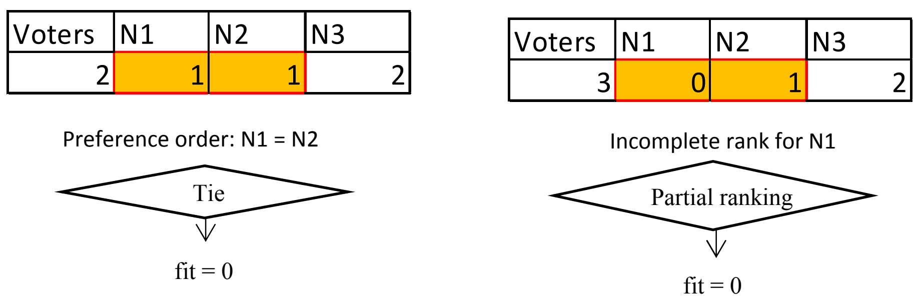

Figure 2a and 2b: tie, and partial ranking examples

As shown in the above, Voter 1 prefers N1 to N2 (N1 > N2). In Permutation 2: the rank of N1 is also higher than N2. Therefore, the preference order fit of N1 vs N2 is 1. Similarly, Voter 1 prefers N3 to N2 (N2 < N3). But the rank of N2 is higher than N3 in the Permutation 2 (N2 > N3). So the fit of N2 vs N3 is 0 as the preference order does not fit with each other. If there is a tie between N1 and N2 (from Voter 2, N1 = N2), the fit value of N1 vs N2 is 0 as the permutation will not give indifference preference order. In the case of incomplete votes, Voter 3 only gave ranks to N2 and N3. The rank of N1 is entered as 0. The preference order fit of N1 vs N2 is 0 as it is possible to find the missing preference order in the permutation. As there is no N4 candidate, we only have 2 columns in the fitness matrices.

Following the same procedure for the preference orders in Permutation 1 against 4 voters, we got the 4x2 matrices shown in Table 2. Figure 3 demonstrates the 6 fitness matrices for the full permutations by continually comparing the preference orders for each adjacent pair in each permutation against the preference orders given by 4 voters.

TABLE 2: FITNESS ORDER MATRICES FOR PERMUTATION 1 AGAINST 4

Voters	Preference order fit of N1 vs N2	Preference order fit of N2 vs N3
1	1	1
2	0	1
3	0	1
4	1	0

$$\begin{bmatrix}1 & 1\\0 & 1\\0 & 1\\1 & 0\end{bmatrix}$$

Fitness for Permutation 1

$$\begin{bmatrix}1 & 0\\0 & 0\\0 & 0\\1 & 1\end{bmatrix}$$

Fitness for Permutation 2

$$\begin{bmatrix}0 & 1\\0 & 1\\0 & 1\\0 & 0\end{bmatrix}$$

Fitness for Permutation 3

$$\begin{bmatrix}0 & 0\\0 & 0\\0 & 0\\0 & 1\end{bmatrix}$$

Fitness for Permutation 4

$$\begin{bmatrix}1 & 0\\0 & 0\\0 & 0\\1 & 1\end{bmatrix}$$

Fitness for Permutation 5

$$\begin{bmatrix}0 & 1\\0 & 1\\0 & 1\\0 & 0\end{bmatrix}$$

Fitness for Permutation 6

Figure 3: Fitness Matrices for all permutations

Step 3: Calculate the sum fits for each permutation.

TABLE 3: TOTAL FITNESS

Permutations	Total Fitness
1	5
2	3
3	3
4	1
5	3
6	3

Step 4: Find the maximum fitness and return the corresponding permutation (Optimal Solution).

TABLE 4: OPTIMAL SOLUTION

Permutations	N1	N2	N3
1	1	2	3

In this example, the ranking for the 3 candidates which appears in the Permutation 1 is the optimal solution to fit with voters' preference order among 3 candidates.

IV. APPLICATION

Given the comparison of the permutations, this algorithm is one of the single-winner voting systems that allows partial ranking. We implemented the algorithm in a real event to find the winner by using R. R is one of the popular language and open source platforms to explore, understand, and analyze data [18]. There were 8 teams participating in the event and 69 ballots in total including 10 ties. Table 5 is the example of the ballot including tie ranks. The input data is the ranking order for the teams, from rank 1 as the best to rank 8 as the worst.

TABLE 5: EXAMPLE OF TIE RANKS

Team 1	**Team 2**	**Team 3**	**Team 4**	**Team 5**	**Team 6**	**Team 7**	**Team 8**
3	5	2	3	1	4	6	7
3	3	2	3	2	1	1	1
0	3	2	2	1	0	0	0
4	5	2	1	4	3	2	6
6	4	4	3	4	3	3	3
1	2	2	2	1	2	1	2
3	3	1	4	2	6	4	4

In this event, the ballots we gathered include incomplete votes, as not every voter were able to attend the whole event. As shown in the table 4, we also had tie in the ballot. Our algorithm did allow these ballots enter as input and handle them accordingly. The permutations of 8 candidates are 40320. The maximum fitness in this event is 235 and it gave us about 1% of full permutations multiple optima to help us find the winner and full rank.

V. COMPARISON WITH BORDA COUNT AND BUKLIN

After using the sequential pair rank for the real dataset for a ranking result with the ballet data, there would also be a necessary comparison on the result with the existing ranking methods, especially with the classical ones. Here we picked up the two most popular ones – Borda Count [3] and Buklin [16]. Through the calculations, the ranking results for the 8 projects from the competition is listed as the following:

TABLE 6: COMPARISON RESULTS WITH BORDA COUNT AND BUKLIN

Ranking	Project No.#		
	Borda Count	Buklin	Sequential Pair Rank
1	6	6	3
2	7	7	1
3	3	3	4
4	5	5	2
5	4	4	7
6	1	1	6
7	2	2	5
8	8	8	8

From the comparison above, it is showing that to the real dataset we were using, the results from Borda Count [3] and Buklin [16] had shown a consistent result, while they are very different from the results of the Sequential Pair Rank. This difference on the ranking result is not only showing the different focusing perspectives from the different methods, but also showing the different application areas of the different methods. And there were other disadvantages showing on Borda Count [3] and Buklin [16] while testing out this comparison results.

Firstly, Sequential Pair Rank has a different designing perspective from the 2 classical methods. For Borda Count and Buklin, they were focusing on every single candidate and looking for the true winner from the competition. Under this assumption of these methods, the relative positions among each of the candidates would be ignored because the result is only relevant for the winner. On the other hand, Sequential Pair Rank is looking for the closest ranking order from the votes comparing with the full permutation, which means the target of Sequential Pair Rank is looking for the whole ranking order rather than the winner candidate. The results from Sequential Pair Rank may different from other methods because it is seeking for the most relevant relative order which could serve the overall opinion from the voters to the whole set of candidates which got the most agreements from all the voters.

Secondly, Sequential Pair Rank has the advantage on strategic planning for picking up the right sequence order of all the projects. From the management perspective, if the company would like to pick up the only project to do in the period of time, they may need to pick up the winner through the voting. For most of the time, for the project managers in the company, they need to "undertake detailed planning to ensure that the activities performed during the execution phase of the project are properly sequenced, resourced, executed and controlled. [20]" So it might be more useful for the project planers to know the priority orders of all the possible projects along with the consideration of the agreements from the voters, because the voters are from different departments with different concerns of their limited resources at the period. Furthermore, we were not really looking for get rid of the ties at this stage. The tie rankings would provide more options for the project planners to take a deep thought on that. The resources, personal skills, management environments, and the real-time situations may be changed along the times, so they could have the flexibilities to adjust the overall plans.

During the comparison process, the disadvantages of Borda Count and Buklin method are also coming out along with the dataset. For Borda Count, the setup of the weight scores would have impact on the final orders, especially on the first place winner. For Buklin method, there would be same probabilities for different candidates as the same order places, which would increase the difficulty on picking up the right one for the winner if there would not allow any ties. If using the Sequential Pair Rank methods, there would be these kinds of problems.

VI. DISCUSSIONS

From the previous application case, the team was also be able to output the final rank. By revisiting the ballots, there were several concerns.

First of all, the technics on interoperating data from the ballots would affect the dataset, especially with ties. The original assumption of the Sequential Pair Rank system was inputting different ranks to different candidates. In the real situation, there were judges that gave out the same rank to different candidates, or even different ranks to a same candidate. These ballots would be considered as a bias for the data entry.

Secondly, considering about the partial ranks, we could not simply apply the majority rule. During the competition, since the ballots were collected in the afternoon, the audience may just provide partial ranks for the projects they could remember. There were more audiences voted for the projects in the afternoon rather than the projects in the morning, and Sequential Pair Rank could tolerate this situation.

Thirdly, the design on paper ballots should be improved. If just given out the index number of the candidates, the ranks from 1~8 as the best to the worst, then let the public judges just fill in a single index number to the position of ranks, the data entry could be more accurate. This may also reduce the ties appears from the ballots.

VII. FUTURE WORK

Firstly, the single-direction pairwise comparison makes the last project listed never get its pairwise computation with another project, which increased the number of multiple optima. Using a comparison with each pair of the candidates could be a possible solution. Secondly, if part of the audiences only gave ranks to the morning projects, while another part only gave ranks to the afternoon projects, the whole dataset would be divided into two parts with two winners – one for the morning and one for the afternoon. In this case, an additional tie-breaking function should be added.

Furthermore, through some more literature researches, there were some scholars tried to make extensions on existed methods in order to make them work for partially voting, especially for the Borda Count. Through the research of David Polett of Bard College [21], he was trying to do a mathematical analysis of Borda Count with the Non-Linear Preferences and partially voting. They "have successfully developed an extension of Linear Borda which allows voters to use partially ordered ballots while still allocating ballot Borda Scores in a manner that is perfectly compatible with linear, bucket and graded poset ballots. [21]" by using "Structure Chains" which referred to the logical comparison relationships among the candidates and re-structured scores for the voting. Although the extension on Borda Count would satisfy the situation, but there are still leak of proofs on if it could be used effectively for non-linear ballots through Borda Count. Also other scholars also researched on the computational impact of partial votes [22]. They also claimed that, "with an elimination style voting rule like single transferable vote, partial voting does not change the situations where strategic voting is possible; with scoring rules and rules based on the tournament graph, partial voting can increase the situations. [22]" As the consequence, "the computational complexity of computing a strategic vote can change [22]" with the partial order. So generally, the complexity of the computing process would be increased significantly through different kinds of voting algorithm. A further comparing research on the processes with the partial votes would also be a future research topic.

VIII. CONCLUSION

Regarding to the application case and analysis, the most creative points of Sequential Pair Rank system are:

1. Comparing the ranking orders of each pair of the candidates to the full permutations and the final ranking result would never show ties;
2. Not focusing on picking a single-winner, but on determining the full ranks;
3. Providing multiple optimal ranking orders that have the same maximum fit value to each judge's rank;
4. Tolerating the partial ranks and tie ranks.

The system could be potentially used for the project management purpose, as a method on scanning the potential projects at the very front end. The value to have the multiple optima would provide more potential options, and increasing the flexibilities of project management process. Furthermore, the Sequential Pair Rank system could potentially be developed and associated with other high-resolution decision making tools.

REFERENCES

[1] W. Smith, “Descriptions of single-winner voting systems,” *math. temple. edu/~ wds/homepage/votedesc. pdf. …*, pp. 1–34, 2006.
[2] J. Gans and M. Smart, “Majority voting with single-crossing preferences,” *J. Public Econ.*, vol. 59, no. 2, pp. 219–237, Feb. 1996.
[3] H. . Young, “An axiomatization of Borda’s rule,” *J. Econ. Theory*, vol. 9, no. 1, pp. 43–52, Sep. 1974.
[4] D. K. Gifford, “Weighted voting for replicated data,” *Proc. seventh Symp. Oper. Syst. Princ. - SOSP ’79*, pp. 150–162, 1979.
[5] R. Rivest and E. Shen, “An optimal single-winner preferential voting system based on game theory,” *Proc. 3rd Int. Work. Comput. …*, 2010.
[6] A. Gibbard, “Manipulation of Voting Schemes: A General Result,” *Evonometrica*, vol. 41, no. 4, p. 587~601, 1973.
[7] D. Black, *The Theory of Communittees and Elections*. Cambridge Univ. Press, 1958.
[8] A. H. Copeland, “A reasonable social welfare function,” in *University of Michigan Seminar on Applications of Mathematics to the social sciences*, 1951.
[9] K. J. Arrow and H. Raynaud, “Social choice and multicriterion decision-making,” *MIT Press Books*, vol. 1, 1986.
[10] J. S. Banks, “Sophisticated voting outcomes and agenda control,” *Soc. Choice Welfare*, vol. 1, no. 4, pp. 295–306, 1985.
[11] B. Reilly, “Social Choice in the South Seas: Electoral Innovation and the Borda Count in the Pacific Island Countries,” *Int. Polit. Sci. Rev.*, vol. 23, no. 4, pp. 355–372, Oct. 2002.
[12] H. P. Young, “Condorcet’s theory of voting,” *Am. Polit. Sci. Rev.*, vol. 82, no. 04, pp. 1231–1244, 1988.
[13] E. J. Nanson, “Methods of election,” 1882.
[14] N. Tideman, “The single transferable vote,” *J. Econ. Perspect.*, pp. 27–38, 1995.
[15] C. H. Coombs, “Social choice and strength of preference.,” 1954.
[16] D. R. Woodall, “Monotonicity of single-seat preferential election rules,” *Discret. Appl. Math.*, vol. 77, no. 1, pp. 81–98, 1997.
[17] M. Schulze, “A new monotonic and clone-independent single-winner election method,” *Voting matters*, vol. 17, no. 1, pp. 9–19, 2003.
[18] R. Kabacoff, *R in Action*. Manning Publications Co., 2011.
[19] J. N. Bearden, A. Rapoport, and R. O. Murphy, “Experimental studies of sequential selection and assignment with relative ranks,” *J. Behav. Decis. Mak.*, vol. 19, no. 3, pp. 229–250, 2006.
[20] J. Westland. *The Project Management Life Cycle: A Complete Step-By-Step Methodology for Initiating, Planning, Executing & Closing a Project Successfully*. Phylidelphia, PA: Kogan Page Publishers, 2007.
[21] D. Pollet, and A.O. Hudson, "Empowering the voter: a mathematical analyiss of borda count elections with non-linear preferences," *A Senior Project submitted to The Division of Science, Mathematics, and Computing of Bard Collage Annandale-on-Hudson,* 2010.
[22] N. Narodytska, and T. Walsh, "The computional impact of partial votes on strategic voting," *arXiv preprint arXiv:*1405.7714, 2014.