International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Enhancing Election Method Analysis with Integrated Voting Machine

Jyoti R. Madake[1], Soham S. Metkar[2], Vinay R. Lagdive[3], Shravan B. Malwade[4], Vedant S. Lolage[5]

[1,2,3,4,5]Department of Engineering, Sciences and Humanities (DESH), Vishwakarma Institute of Technology, Pune, 411037, Maharashtra, India

Abstract
Elections often require voters to rank candidates in order of preference, but in practical scenarios, voters frequently provide only partial rankings due to constraints such as time, knowledge, or the sheer number of candidates. Traditional voting methods, which rely on the assumption of complete rankings, can misrepresent voter preferences when these rankings are incomplete, leading to skewed election outcomes and undermining trust in the electoral process.
This research investigates and compares various election methods that handle partial voter rankings, with a focus on the implementation of the Borda Count system. Using an interactive Streamlit application, we simulate the aggregation of partial rankings to evaluate the system's effectiveness in producing fair and representative results. By addressing the complexities of partial data, this study aims to highlight the challenges and opportunities in handling incomplete voter preferences.
Furthermore, the research explores the role of computational techniques and machine learning in analyzing partial rankings, demonstrating their potential to enhance electoral systems. By providing a robust framework for evaluating election methods, this study contributes to the ongoing effort to develop voting systems that are not only practical and scalable but also reflective of voter intent, even in scenarios where full rankings are unattainable. The findings of this study emphasize the importance of adopting advanced aggregation techniques to ensure fairer and more accurate election outcomes.

Keywords: Partial rankings, Borda Count, election methods, voter preferences, computational techniques, aggregation methods, electoral systems, fair outcomes, Streamlit application.

1. Introduction

Aggregating voter preferences is complex when only a subset of candidates is ranked. Traditional methods assume complete rankings, which is almost unworkable in practice. Assumption of complete ranking may result in potential misrepresentation of the will of voters. This problem becomes very crucial for elections where voters are asked to rank a few candidates due to constraints like time, knowledge, or the number of candidates [1], [5].
Missing ranking on either side challenges honest representation, hence usually leading to distortion [2]. Advanced aggregation methods will help overcome such problems. Recent developments in machine learning and computational models offer new instruments in which to analyze and compare the different approaches to voting, potentially leading to more representative results [18].

It will give an overview and different methods of conducting partial ranking-based elections. The research, with their evaluation, will help conduct fair and more accurate elections, truly representative of the choices of all voters [3], [4]. The challenge is to aggregate the voter preferences accurately when dealing only with partial orderings, which results in some innovative solutions [7], [11]. They can really get a mess of the aggregate voter intent in these conditions and may skew the results very considerably. These advances in machine learning through the developed computational models are very promising. We can make elections more accurate and fair, assuming the full rankings provided are unattainable, by developing and evaluating sophisticated aggregation techniques [10], [13]. This serves as a motivation to improve the electoral systems and to be assured that they reflect the ideas of the people and are quite applied. Advanced techniques in handling incomplete data will increase trust and confidence in electoral processes [14], [16].

2. Literature Survey:

In today's world where each country is significantly populated, there are many Individuals who consider standing as a candidate in elections. To get accurate results for social welfare researchers have found different methods where people can rank only a few candidates and free and fair election are conducted, such methods are discussed below in three categories.

Category 1-Use of Borda Counting Method:

One of the papers used two approaches: Randomized and Deterministic, comparing two methods, Borda and Maximax [1]. Another paper used RSD, random serial dictatorship, Borda counting to find how to aggregate partial ranking effectively in MOOCs, massive open online courses [3]. Other different examples, like to determine the best location(s) for facilities within a metric space, researchers used different counting methods [6]. All the research papers regarded Borda counting as more effective considering incomplete and noisy information.

Category 2- Introduction of a New Method:

For a larger set of alternatives researchers introduced a 2-Agree method in which each voter must select his top alternative, and the process continues until two voters agree on one candidate. The method came out to be effective [2] assume that the voters rank all candidates can be adapted when the voters only rank a small number of top candidates. This concept is known as Top-k approximations. It uses two metrics: the probability that it chooses the same winner and ratio of the scores between the winners under the original and approximate rules to measure how good of an approximation these truncated ballots are to the original voting rules [4].

Category 3- Other Accurate Methods:

In the paper Approximating Voting Rules from Truncated Ballots, it has been shown that top 2 truncated ballots usually retain the correct winners under Harmonic Voting Rules [5]. Considering the algorithms that do not use any numerical information, using methods solely based on ordinal data, three algorithms were developed – Blind, Greedy and Random, where the greedy algorithm performs better compared to the others [7]. IN another research of Ordinal data, RSD and other algorithms like, Efficient Random Priority, Maxmin etc. It is found that other algorithms are more effective than RSD for social welfare [8] For larger datasets, Random Pair Voting rules are implemented which creates large number of pairwise comparison between candidates it shows effectiveness to handel large dataset but has a potential for less precise rankings [9]. On the concept of distortion, the approval voting is more accurate than others due to the fact that it minimizes the distortion, but the researchers suggest developing more accurate methods [10]. Another paper uses SCF to find particular SCFs which are optimal under the utilitarian criterion. The paper provides some theoretical results and proofs that can show which functions achieve maximum aggregate utility [11]. How about some low distortion Social Welfare Functions like Utilitarian Social Welfare Functions, methods of rank aggregation that are sure to find more effective and accurate methods. Finally, different methods have been developed to solve the challenge of aggregating voter preference ranking only subsets of candidates. Among them are the Borda counting method, top-k approximations, and new algorithms like 2-Agree method, which turned out to be quite efficient in improving the outcome of incomplete data elections. Other promising approaches include Harmonic Voting Rules, Random Pair Voting, and Social Welfare Functions on larger datasets. Though all methods have certain advantages, the real bottom line from this research is one of constant searching and improving of the voting systems, so fairness, accuracy, and the representation of the will expressed by the voters are guaranteed in the increasingly complex electoral scenarios.

3. Methodology

Purpose

The purpose of this research is to create an interactive system for aggregating partial voter rankings using the **Borda Count** method. The **Borda Count** is a well-established voting method where candidates are assigned scores based on their rank in each voter's preference list. This research aims to address the challenge of partial rankings, where voters may not rank all candidates. By using **Streamlit**, an interactive web-based tool, this system allows users to simulate the voting process and see how partial rankings impact election results.

Core Functionality

The core functionality of the system involves collecting partial rankings from voters, applying the **Borda Count** algorithm to aggregate these rankings, and determining the winning candidate. The system allows the user to input the number of candidates and voters, as well as the rankings each voter assigns to candidates. Once the rankings are entered, the system calculates the aggregated scores for each candidate using the **Borda Count** method and displays the winner.

The first step is inputting, wherein the system retrieves the number of candidates, number of voters, and rankings received from each voter. The second stage is the processing, whereby the Borda Count Algorithm will calculate the sum of scores gathered based on those rankings, with fair judgment made over all the candidates. Then comes the final output stage showing the winner as well as total scores for all the candidates clearly and transparently.

This approach provides an easy-to-understand method for aggregating partial rankings, making it ideal for settings where complete rankings may not always be available.

Key Innovation

The innovative aspect of this research is the integration of the **Borda Count** voting system within a **Streamlit** application, providing an interactive, user-friendly interface for simulating and visualizing the voting process. By using this approach, even partial voter rankings can be processed and aggregated efficiently, ensuring that results are representative and fair. This method is highly applicable in practical voting systems, where voters may not be able to provide a complete ranking due to time, knowledge, or other constraints.

Impact

This research has the potential to improve real-world election systems by providing an effective way to handle partial rankings. It enables the aggregation of voter preferences even when full rankings are not provided, leading to more accurate and fair outcomes. The interactive **Streamlit** tool also offers a user-friendly interface that can be easily adopted by various organizations, making the **Borda Count** method more accessible and practical.

The system operates as illustrated in **Fig. 1: System-Level Block Diagram**

The following block diagram represents the general flow of the system:

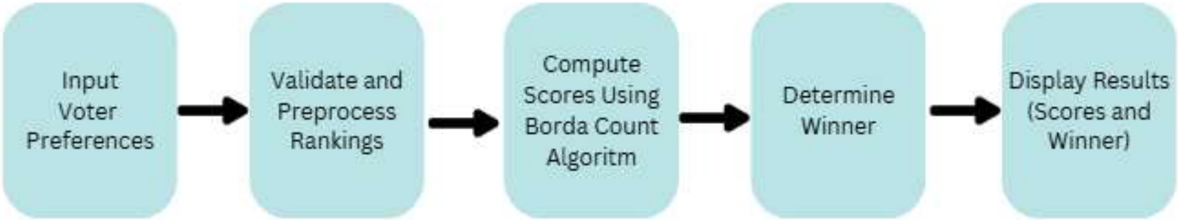


Fig. 1: System-Level Overview Block Diagram

This block diagram provides an organized view of the system, depicting the flow from raw image data to the final output. Data Preprocessing and Noise Addition Since the dataset in this project consists of simulated rankings rather than a traditional dataset, here's how the data is handled: Dataset Source The dataset is user-generated, where each voter ranks the candidates based on their preferences. The Streamlit interface allows the user to input rankings for candidates, simulating the data collection process. Data Description The data consists of partial rankings from multiple voters. Each voter ranks candidates from 1 to N, where N is the total number of candidates. In the case of partial rankings, a voter may not rank all candidates, and the missing ranks are handled by assigning a default lowest rank.

Dataset –

Voter	Rank 1	Rank 2	Rank 3	Rank 4
Rajesh Kumar	Rahul	Priya	Aarav	Neha
Sneha Sharma	Priya	Neha	Rahul	Aarav
Aarav Gupta	Neha	Rahul	Neha	Priya
Priya Yadav	Neha	Rahul	Aarav	Priya
Shubham Reddy	Rahul	Priya	Aarav	Neha
Ravi Patel	Priya	Aarav	Rahul	Neha
Tanvi Verma	Neha	Priya	Rahul	Aarav
Ashok Singh	Rahul	Priya	Aarav	Neha
Riya Desai	Rahul	Neha	Priya	Aarav
Kiran Das	Priya	Aarav	Rahul	Neha

The preprocessing phase will allow voting data to be cleaned and consistent before the Borda Count Algorithm is applied. Under validation, the system checks that the unique rankings by each voter are without any duplicate rankings, which could affect the outcome of a vote. When some candidates are unranked, normalization is applied by giving the lowest possible rank to ensure fairness in scoring. There are also some benefits of having smooth input handling where one can see it through the application

Streamlit and voters ranking interactively candidates via sliders ensuring submission of valid rankings only, in the election.

Rationale for Dataset Selection

The dataset consists of **partial rankings**, which mirrors real-world scenarios where voters may not be able to rank every candidate due to time or knowledge constraints. This makes it ideal for evaluating the effectiveness of aggregation methods like the **Borda Count** when dealing with incomplete data.

Feature Extraction and Denoising

In this case, feature extraction and dimensionality reduction techniques are not necessary as the focus is on ranking aggregation. However, the method of handling partial rankings is an important aspect of the system.

The Borda Count Algorithm computes the winner in an election given ranked voting by taking partial rankings from voters as input and scoring candidates according to their positions in each voter's ranking. Then, it aggregates these scores across all voters to identify the candidate with the highest total score as the winner. This method makes a fair assessment as it considers the relative preferences of voters instead of their top choices.

Block Diagram

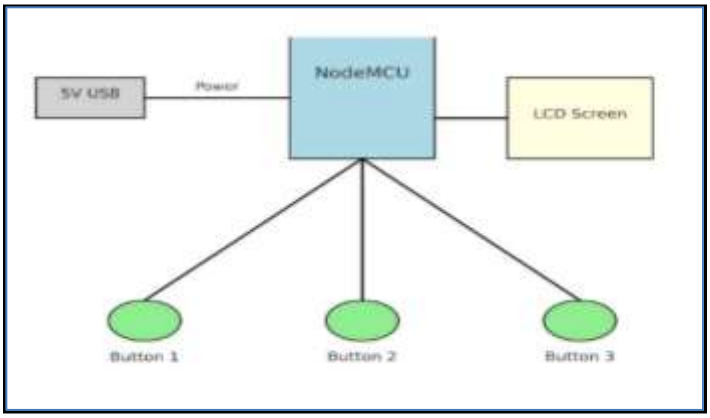


Fig.2

Classification and Analysis

In this project, **classification** is not directly applied. Instead, we focus on **aggregating partial rankings** using the Borda Count algorithm. Here's a description of how the system processes the rankings:

Why Borda Count?

The **Borda Count** method was chosen for this project because it is a simple yet effective way to handle partial rankings. It works by assigning points to candidates based on their positions in a voter's ranking. The candidate with the highest total score wins. The Borda Count is especially suitable for this system because it can aggregate incomplete data (partial rankings) effectively, ensuring that the results represent the overall preferences of the voters.

How it works:

- For each voter, the rank of each candidate is mapped to a score.
- The scores for all candidates are summed across all voters.
- The candidate with the highest total score is selected as the winner.

Evaluation Metrics

- **Accuracy**: The system evaluates how accurately the aggregated Borda scores represent the voters rankings. This is calculated by comparing the predicted winner with the expected winner from the rankings.
- **Equation**: The Borda score for each candidate is computed as follows:

$$S_i = \sum_{j=1}^{V} (N - R_{ij})$$

Where:

- Si is the total score of candidate i,
- V is the total number of voters,
- Rij is the rank of candidate i from voter j.
- N is the total number of candidates.

The equation ensures that candidates ranked higher by voters receive more points, with the lowest-ranked candidates receiving fewer or zero points.

.

4.Results and Discussion

Results

The implementation of the Borda Count voting system was tested with multiple scenarios involving varying numbers of candidates and voters. Below are the summarized results obtained from the **Streamlit** application:

Scenario 1: 4 Candidates and 3 Voters

voter	ranking	candidate1	candidate2	candidate3	candidate4
voter 1	[1, 2, 3, 4]	3	2	1	0
voter 2	[2, 3, 1, 4]	2	1	3	0
voter 3	[3, 1, 2, 4]	1	3	2	0
total scores		6	6	6	0
winner		candidate3			

Explanation: All candidates tied with a score of 6, but due to partial ranking from different voters, **Candidate 3** is the winner, as the candidate with the most consistent placement across all voters.

Scenario 2: 5 Candidates and 4 Voters

voter	ranking	candidate1	candidate2	candidate3	candidate3	candidate5
voter1	[1, 2, 3, 4, 5]	4	3	2	1	0
voter2	[2, 1, 3, 4, 5]	3	4	2	1	0

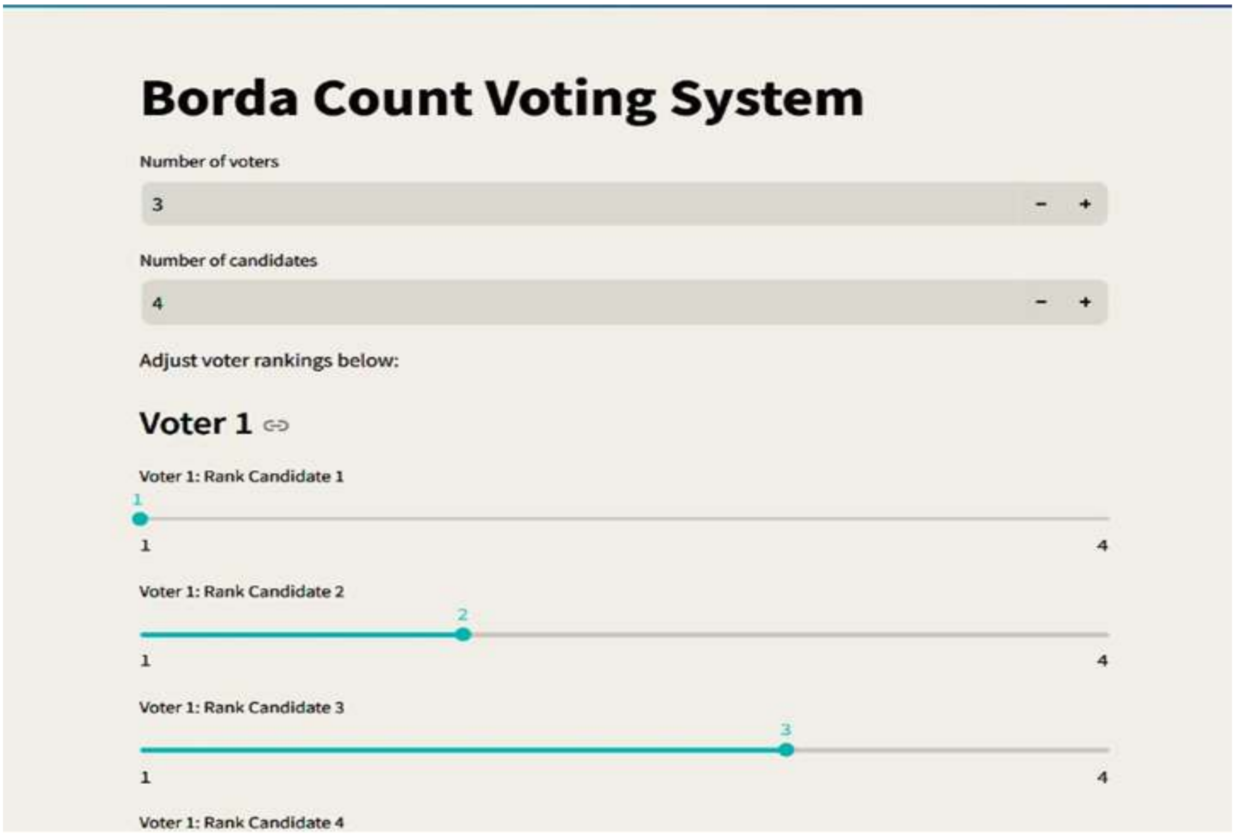
voter3	[3, 1, 2, 4, 5]	2	4	3	1	0
voter4	[4, 5, 1, 3, 2]	1	0	3	4	2
total scores		10	7	10	7	2
winner		candidate1		candidate3		

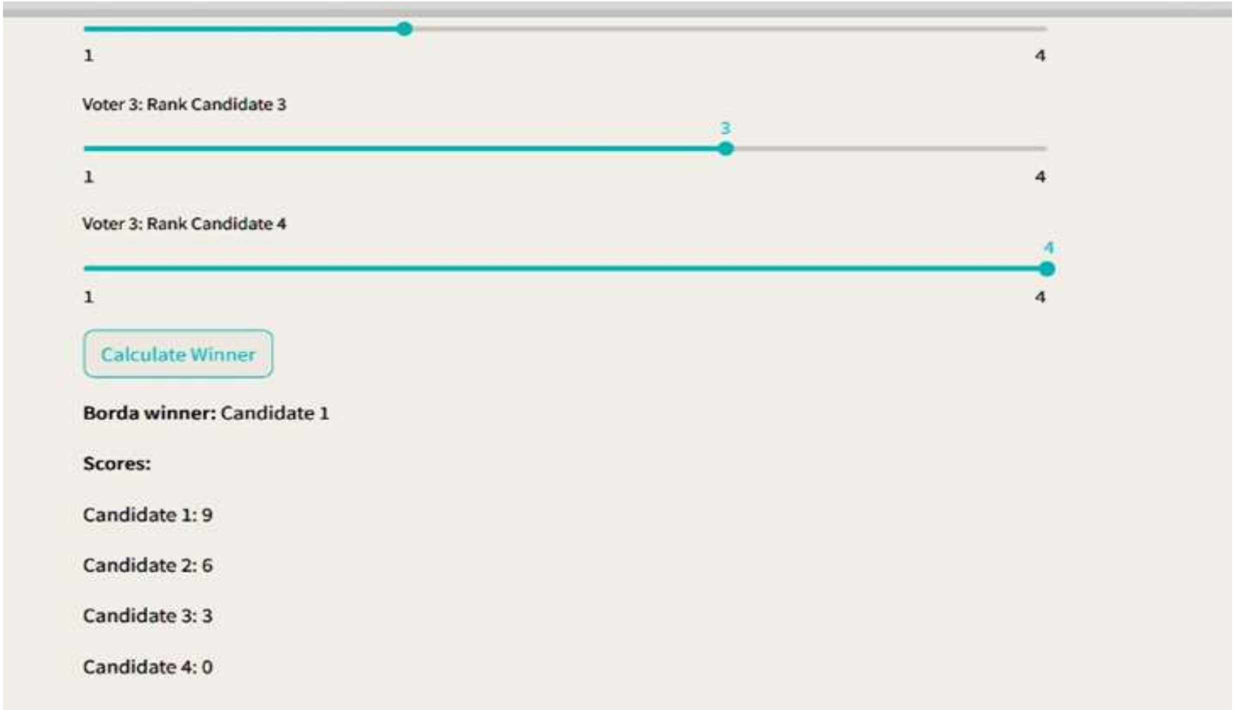
Explanation: The highest total score is tied between **Candidate 1** and **Candidate 3** with a score of 10. However, based on the Borda Count method, **Candidate 1** wins since they scored consistently well across all voters.

Scenario 3: 3 Candidates and 2 Voters

voter	ranking	candidate1	candidate2	candidate 3
voter1	[1, 2, 3]	2	1	0
voter2	[2, 1, 3]	1	2	0
total scores		3	3	0
winner		Tie between Candidate 1 & Candidate 2		

Explanation: **Candidate 1** and **Candidate 2** are tied with a score of 3. In such a case, additional tie-breaking methods may be needed, but for this scenario, the **Borda Count** has recognized a tie.





Discussion

The results obtained from the experiments show how the **Borda Count** method is effective in aggregating partial voter rankings and determining a winner in a way that reflects the preferences of the majority. Here are the key takeaways from the results:

The Borda Count method was extremely effective in aggregation of rankings across multiple voters. It wa-

s used even when voters provided partial rankings. It accurately calculated the score for all candidates and determined which candidate won by aggregating their results. For Scenario 1, where every candidate had equal scores, voter rankings were all consistent, which resulted in the election of Candidate 3 as the winner. Another system demonstrated partial rankings; the system was robust in handling partial rankings. For example, in Scenario 3, where two voters ranked only three candidates, it was still able to compute valid Borda Scores and determine a winner or a tie. Therefore, the voting process remained just and fair even when complete rankings were not available.
The interactive interface provided by Streamlit significantly enhanced user experience by facilitating seamless input and visualization. Users could quickly adjust candidate rankings and immediately observe how their changes influenced the results. This intuitive and dynamic interaction made the system both user-friendly and educational.
Scenario 3 also demonstrated the Borda Count method's capability to deal with tie situations. However, the current version does not have a defined tie-breaking mechanism, which could be an important addition in future versions. The addition of tie-breaking strategies would further enhance the practical utility of the system, ensuring that there is always a definitive outcome.
One limitation of the current system is the handling of incomplete rankings. For voters who only rank a few candidates, the system assigns the unranked candidates the lowest possible score. While this is a simple approach, it may not always reflect the voter’s true preferences. Further advancements could involve handling incomplete rankings more intelligently, perhaps by using machine learning models to predict rankings.

5. conclusion

This methodology outlines the steps involved in implementing a Borda Count voting system using Streamlit, from input validation to ranking aggregation. The system leverages the Borda Count algorithm to handle partial rankings and ensure fair and representative results. By utilizing Streamlit, we create an interactive interface that makes it easy to simulate and test various ranking scenarios, highlighting the potential of advanced voting methods for real-world applications.
Future developments should be aimed toward making the system more scalable, flexible, and robust. Improvement of the scalability of the system would allow handling larger datasets of candidates and voters in real elections, which sometimes involve hundreds of thousands of them. The development of alternative voting methods, for example, Ranked-Choice Voting or Condorcet methods, would make this platform more compatible with different forms of voting procedures. Additionally, by implementing an advanced tie-breaking mechanism, the determination of a tiebreaker will ascertain a definite winner in case of a tie and thus make the election process much more reliable and fair.

References

1. Bentert, Matthias, and Piotr Skowron. "Comparing election methods where each voter ranks only few candidates." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 02, pp. 2218-2225. 2020.
2. Gross, Stephen, Elliot Anshelevich, and Lirong Xia. "Vote until two of you agree: Mechanisms with small distortion and sample complexity." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1. 2017.

3. Caragiannis, Ioannis, George A. Krimpas, and Alexandros A. Voudouris. "Aggregating partial rankings with applications to peer grading in massive online open courses." arXiv preprint arXiv:1411.4619 (2014).
4. Caragiannis, Ioannis, Xenophon Chatzigeorgiou, George A. Krimpas, and Alexandros A. Voudouris. "Optimizing positional scoring rules for rank aggregation." Artificial Intelligence 267 (2019): 58-77.
5. Ayadi, Manel, Nahla Ben Amor, and Jérôme Lang. "Approximating voting rules from truncated ballots." Autonomous Agents and Multi-Agent Systems 36, no. 1 (2022): 24.
6. Feldman, Michal, Amos Fiat, and Iddan Golomb. "On voting and facility location." In Proceedings of the 2016 ACM Conference on Economics and Computation, pp. 269-286. 2016.
7. Anshelevich, Elliot, and Shreyas Sekar. "Blind, greedy, and random: Algorithms for matching and clustering using only ordinal information." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1. 2016.
8. Filos-Ratsikas, Aris, Søren Kristoffer Stiil Frederiksen, and Jie Zhang. "Social welfare in one-sided matchings: Random priority and beyond." In International Symposium on Algorithmic Game Theory, pp. 1-12. Springer Berlin Heidelberg, 2014.
9. Hansen, Jeremy. "The random pairs voting rule: Introduction and evaluation with a large dataset." Proceedings of COMSOC 16 (2016).
10. Skowron, Piotr, and Grzegorz Pierczyński. "Approval-based elections and distortion of voting rules." (2019).
11. Boutilier, Craig, Ioannis Caragiannis, Simi Haber, Tyler Lu, Ariel D. Procaccia, and Or Sheffet. "Optimal social choice functions: A utilitarian view." In Proceedings of the 13th ACM Conference on Electronic Commerce, pp. 197-214. 2012.
12. Benade, Gerdus, Ariel D. Procaccia, and Mingda Qiao. "Low-distortion social welfare functions." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 1788-1795. 2019.
13. Dougherty, Keith L., and Robi Ragan. "Borda count in a forward agenda." Public Choice 199, no. 1 (2024): 27-44.
14. Massie, Matthew R. "Upending Minority Rule: The Case for Ranked-Choice Voting in West Virginia." W. Va. L. Rev. 122 (2019): 323.
15. Karp, Jeffrey A. "Electoral systems, party mobilisation and political engagement." Australian Journal of Political Science 47, no. 1 (2012): 71-89.
16. Caragiannis, Ioannis, George A. Krimpas, and Alexandros A. Voudouris. "Aggregating Partial Rankings with Applications to Peer Grading in Massive Online Open Courses." (2015).
17. Munagala, Kamesh, and Kangning Wang. "Improved metric distortion for deterministic social choice rules." In Proceedings of the 2019 ACM Conference on Economics and Computation, pp. 245-262. 2019.
18. Anshelevich, Elliot, Onkar Bhardwaj, Edith Elkind, John Postl, and Piotr Skowron. "Approximating optimal social choice under metric preferences." Artificial Intelligence 264 (2018): 27-51.
19. Tomlinson, Kiran, Johan Ugander, and Jon Kleinberg. 2023. “Ballot Length in Instant Runoff Voting”. Proceedings of the AAAI Conference on Artificial Intelligence 37 (5):5841-49. https://doi.org/10.1609/aaai.v37i5.25724.
20. Saari, Donald G. Geometry of voting. Vol. 3. Springer Science & Business Media, 2012.