Comparative Analysis of Algorithmic Complexity and Experimental Evaluation of Committee Selection Rules for Multi-Winner Elections

Dr. Telesphore Nsumbu[1], Dr. Bruce Mbombi[2], Dr. Pascaline Kizodisa[3], Dr. Richard Kitondua[4]

[1]Department of Mathematics and Computer Science, Faculty of Science and Technology, National Pedagogical University (UPN), Kinshasa, DRC.,

[2]Higher Pedagogical Institute of Luozi, Section of Science and Technology, Specialization in Management Information Systems, ISP-LUOZI, Kinshasa, DRC.,

[3]Higher Pedagogical Institute of Mbanza-Ngungui, Section of Sciences and Technologies, Specialization in Computer Science, ISP/MBANZA, Kinshasa, DRC.

[4]Department of Mathematics and Computer Science, Faculty of Science and Technology, National Pedagogical University (UPN), Kinshasa, DRC.

Abstract

This study presents a comparative analysis of six committee selection rules in multi-winner elections, focusing on their algorithmic performance and experimental validation. A theoretical complexity analysis is complemented by simulations performed on instances of 100 to 18,000 voters to evaluate execution time, memory consumption, voter satisfaction, algorithmic efficiency, and scalability.

Block voting rules are distinguished by their low computational cost and excellent scalability, while Monroe and Chamberlin-Courant rules offer better representation quality at the cost of increased complexity. This study thus highlights the trade-off between computational efficiency and representation quality, while experimentally confirming the predictions of algorithmic complexity theory.

Keywords: Computational social choice; Multiple winner elections; Committee selection; Algorithmic complexity; Experimental evaluation; Scalability; Algorithmic efficiency.

1. Introduction

1.1 Context and Motivations.

The development of digital technologies has profoundly transformed the mechanisms of collective decision-making. Collaborative platforms, recommendation systems, distributed networks, citizen participation platforms, and artificial intelligence applications increasingly require methods for simultaneously selecting multiple alternatives from a large number of candidates. This issue falls under the concept of multi-member constituency elections , a central area of computational social choice , *situated at* the intersection of computer science, mathematics, operations research, and economic theory (Skowron, P., Faliszewski , P., & Slinko , A. 2019; Kimelfeld et al. 2018). Unlike traditional elections, where a single candidate is elected, multi-member constituency elections aim to create a body of representatives that best reflects the preferences of the voters. They are currently finding applications in the selection of political representatives, recommendation systems, collaborative filtering, participatory budgeting and resource allocation in digital environments (Célis et al. 2018) .

1.2 Problem Statement

Over the past few decades, several committee selection rules have been proposed to improve voter representation. Among the best known is the single non-transferable vote (SNTV). Block vote and approval vote k- Borda **,** Monroe and Chamberlin – Current (Chamberlin & Courant, 1983; Monroe, 1995). However, these methods exhibit very different behaviors. The most efficient representation rules, such as Monroe and Chamberlin-Courant, are often associated with high algorithmic complexity, which limits their use in elections involving a large number of voters and candidates. Conversely, simpler rules offer reduced computation times but may produce a less satisfactory representation (Betzler , Slinko and Uhlmann , 2013). Therefore, a fundamental question arises:
Which committee selection rule offers the best compromise between quality of representation and algorithmic efficiency in multi-winner elections?
Mathematical formalization of the optimization problem + concrete example (participatory municipal budgeting)

1.3 Research Hypotheses

This study is based on the general hypothesis that committee selection rules exhibit different behaviors and computational performance, directly influencing their applicability to large-scale elections. More specifically, the following hypotheses are formulated:

- Simple voting rules (SNTV, block voting, approval voting and k-Borda) have better computational performance than Monroe and Chamberlin- Courant rules.
- The Chamberlin-Courant rules offer greater satisfaction to voters, but at the cost of a higher algorithmic cost.
- The performance gaps between the different rules become more significant as the size of the electoral bodies increases.

1.4 Objectives

The main objective of this research is to comparatively analyze the algorithmic complexity and experimental performance of the main committee selection rules used in multi-winner elections.
More specifically, this refers to:

- to present the main selection rules for the committees;
- analyze their theoretical algorithmic complexity;
- to experimentally evaluate their performance through several simulations;
- compare their results according to different performance criteria;
- identify the rules offering the best compromise between the quality of the representation and the efficiency of the calculation.

Table 1: Traceability Matrix - Objectives - Indicators - Methods - Results

Objective	Description	Performance indicator	Target threshold
O1	Present the main selection rules for the committees.	Theoretical coverage: number of rules with an established complexity limit	6/6 rules
O2	Analyze theoretical algorithmic complexity	Experimental reliability: R^2 correlation between theoretical and empirical curves	$R^2 \geq 0.85$
O3	Experimentally evaluate performance on 5 instances	Scalability: ability to process 18,000 voters within a reasonable timeframe	Time < 300 s for counting rules
O4	Compare the results according to performance criteria	Quality of representation: relative difference between counting rules and proportional rules	Documentation of the gap
O5	Identify the best compromise between quality and efficiency	Algorithmic efficiency: a composite cost/quality indicator	Ranking by overall score
O6	Ensuring the robustness of implementations	Error rate across all experimental runs	< 0.5%

This matrix establishes the correspondence between the research objectives stated in section 1.4, the methods used in the manuscript, the measured performance indicators, and the results actually obtained. It allows verification that each objective has been addressed measurably and that the target thresholds defined in section 1.6 have been met, or, where applicable, clarifies the observed limitations. Objective O3 (scalability) has a partial status: the counting rules respect the 300-second threshold, but the Chamberlin -Courant rule (108,413 ms in E5) significantly exceeds it, confirming its NP-hard classification and its unsuitability for real-time processing.

1.5 Literature Review

The selection of electoral committees in multi-member constituency elections is now a major area of computer modeling of social choice. Work in this field primarily aims to design rules that guarantee fair representation of voters, while taking into account the algorithmic constraints related to their implementation.

1.5.1 Theoretical foundations of proportional representation

One of the foundational works is that of Chamberlin and Courant (1983) **They** proposed a system of proportional representation where each voter is paired with the committee member of their choice. Their goal was to maximize overall voter satisfaction through the effective representation of individual preferences. This approach strongly influenced subsequent research on multi-winner elections.
Building on this work, Monroe (1995) developed a more balanced model of proportional representation, requiring that each committee member represent an approximately equal number of voters. This rule improves the quality of representation but complicates the problem-solving process.

1.5.2 Algorithmic complexity and ranking results

The complexity of these mechanisms has been explored in more detail **by** Betzler, Slinko , and Uhlmann (2013). Their work has demonstrated that the exact calculation of winners according to the Chamberlin-

Courant and Monroe rules is an NP - hard problem. This contribution highlights a trade-off between the quality of the representation and computational feasibility.

Recent work has primarily focused on the theoretical foundations of committee selection rules (Lackner and Skowron , 2018), their axiomatic properties, and their mathematical safeguards. In this regard, the work of Lackner and Skowron (2023), And **,** *Multiple Winner Voting with Approval Preferences* is a major reference work. The authors present the main rules of multiple winner voting, including proportional approval voting (PAV). The book examines PAV, Chamberlin-Courant, Monroe, and sequential PAV voting rules, analyzing their proportionality, representation, and optimality properties. It also proposes several algorithmic results related to these rules and discusses their application domains. However, its approach remains primarily theoretical and conceptual. The book does not develop a systematic experimental study to evaluate the impact of varying the number of candidates, the number of voters, and the committee size on the actual performance of the algorithms. Furthermore, no detailed comparison between exact and heuristic implementations is provided.

1.5.3 Contemporary Extensions and Probabilistic Modeling

In the same vein, Rothe (2024) **In** * *Economics and Computation: An Introduction to Algorithmic Game Theory*, the author presents a synthesis of the foundations of computational social choice. He addresses the main voting mechanisms, problems of electoral manipulation, and notions of fairness ,* as well as several complex results. This work offers an excellent introduction to the fundamental concepts of the field. However, it prioritizes a theoretical and pedagogical approach . The practical performance of algorithms is not studied through experimentation, which limits the evaluation of their behavior on large-scale instances.

Furthermore, Diss and Merlin (2021) The study *"Evaluation of Voting Systems Using Probabilistic Models" analyzes voting systems using probabilistic models to assess their normative properties and behavior under different preference distributions. It focuses primarily on the mathematical and probabilistic aspects of electoral procedures. However* **,** it does not examine the computational performance of committee selection algorithms, particularly with regard to execution time, memory consumption, or their ability to handle increasing data volumes (Gupta et al ., 2021).

Overall, this work makes a significant contribution to understanding committee selection rules. It provides a robust theoretical framework, offers rigorous axiomatic analyses, and highlights the properties of different voting mechanisms. However, it devotes relatively little attention to the experimental validation of algorithmic performance. In particular, recent literature offers few comparative studies that simultaneously analyze time complexity **.** spatial complexity **,** scalability, and the effect of variation in the number of candidates (*m)*), the number of voters (*n)* , and the size of the committee (*k)*).

1.5.4 Gaps identified and positioning of this research

A review of the literature reveals a significant gap. Existing research focuses primarily on the theoretical properties, mathematical models, and axiomatic guarantees of committee selection rules. However, it does not offer a comprehensive experimental evaluation that would allow for measuring and comparing the real-world performance of different methods in contexts where data sizes vary considerably.
More specifically, few studies jointly examine:

- the evolution of temporal complexity as a function of the number of candidates, the number of voters and the size of the committee;

- the evolution of the spatial complexity of the different selection rules;
- the comparison of performance between exact algorithms and heuristic algorithms;
- identify the thresholds beyond which the exact methods become difficult to use;
- the comparison of theoretical results of algorithmic complexity with the performance actually observed during simulations .

This research falls within this perspective and aims to contribute to filling this gap. The detailed positioning of this study in relation to existing work is presented in Table 1 ter, which highlights three distinctive dimensions: large-scale experimental validation, integrated multi-criteria evaluation, and guaranteed methodological reproducibility. Its originality lies primarily in the implementation of a dual evaluation approach **.** This approach consists of combining the theoretical analysis of algorithmic complexity with experimental validation based on simulations performed in Python. It will allow us to compare the predictions derived from the asymptotic analysis with the performance actually observed.
Next, this study adopts a multidimensional analysis of the algorithm's performance. Unlike existing work that generally focuses on the theoretical properties of voting rules, it will simultaneously evaluate time complexity, space complexity, and scalability . and the quality of the solutions obtained.

Furthermore, the originality of this research also lies in the systematic study of the influence of the main parameters of the problem, namely the number of candidates (m) , the number of voters (n), and the size of the committee (k). This analysis will make it possible to identify the conditions under which each selection rule remains effective and the thresholds beyond which certain approaches become less appropriate.

Finally, the research will offer an experimental comparison of exact and heuristic algorithms applied to the main committee selection rules (Bloc, SNTV, approval voting, k-Borda, Chamberlin-Courant, and Monroe). This comparison will be based on objective indicators such as execution time, memory consumption, scalability, and the quality of the solutions obtained.

Thus, the scientific contribution of this study is not limited to the theoretical analysis of committee selection rules. It consists of a reproducible experimental evaluation , based on computer simulations, that identifies the trade-offs between computational efficiency, resource consumption, and solution quality for different instance sizes. This approach makes an original contribution to the literature by bridging the gap between theoretical results on algorithmic complexity and the practical requirements of processing large datasets.

1.6. Contributions of the article

This article offers a comparative analysis of the main selection rules for electoral committees, comparing their theoretical properties with experimental results obtained from electoral bodies of increasing sizes. The evaluation is based on a multi-criteria analysis incorporating algorithmic performance indicators and highlights the trade-off between the quality of representation and computational cost, thus providing decision-making tools for choosing a method suited to the application context.

1.6.1. Performance indicators.

In order to ensure an objective measurement of the progress of this research, the following performance indicators are defined:

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

- **Theoretical coverage** : complete analysis of the temporal and spatial complexity of the six rules studied, measured by the number of rules for which a complexity limit is established or confirmed (target: 6/6).
- **Experimental reliability** : correlation between theoretical predictions of complexity and results observed on large instances, evaluated by the coefficient of determination R^2 between theoretical and empirical curves (target: $R^2 \geq 0.85$).
- **Scalability** : ability of algorithms to process instances of up to 18,000 voters within a reasonable timeframe (target: execution time < 300 seconds for counting rules).
- **Quality of representation:** average level of voter satisfaction achieved by each rule in all cases tested (objective: documentation of the relative gap between counting rules and proportional representation rules).
- **Robustness:** error rate less than 1% across all experimental executions (target: error rate < 0.5%).

These indicators allow us to quantitatively assess the contribution of this study and to ensure the reproducibility of the results obtained.

2. Algorithmic complexity of committee selection rules.

2.1 Concepts of algorithmic complexity.

Algorithmic complexity is a fundamental criterion for evaluating the performance of an algorithm. It allows us to estimate the resources required to execute a program, primarily execution time and memory usage, as a function of the size of the input data (Cormen et al., 2022). In the context of multi- winner elections, the size of the problem depends essentially on the number of voters (n), the number of candidates (m), and the size of the committee (k).

Two measures are generally taken into account:

- **time complexity** , which evaluates the number of operations needed to execute the algorithm;
- **spatial complexity** , which measures the amount of memory needed to perform the calculations.

These performances are expressed using the asymptotic notation of **O** , widely used in computer science to characterize the growth of algorithmic cost as the size of the problem increases (Knuth, 1997; Cormen et al., 2022).

2.2. Presentation of the rules studied

2.2.1. Single Non-Transferable Vote (SNTV)

The single non-transferable vote (SNTV) system is one of the simplest methods used in multi-member elections. Each voter has one vote, which they allocate to their preferred candidate. At the end of the election, the k candidates with the most votes are elected (Elkind et al., 2017).

Operating principle:

- Each voter votes for only one candidate;
- The votes are counted;
- The k highest-ranked candidates are selected.

2.2.2. Block voting

Bloc voting is a generalization of majority voting with multiple winners. Each voter can vote for several candidates, generally up to the number of seats to be filled (Lackner & Skowron, 2023). .

Operating principle

- Each voter selects up to k candidates;
- Each vote awards one point to the chosen candidate;
- The k candidates with the highest number of points are elected.

2.2.3. Approval vote

Approval voting allows voters to approve as many candidates as they wish. Each candidate receives one point for each approval obtained (Brams & Fishburn, 2007).

Operating principle

- Each voter approves a series of candidates;
- Approvals are counted;
- The candidates who received the most support are elected.

2.2.4. k-Borda rule.

Borda is based on a system of scores assigned to candidates according to their position in each voter's ranking (Lackner & Skowron, 2023).

Operating principle

- Each voter ranks all candidates in order of merit;
- A score is assigned to each position;
- The scores are added together;
- The k candidates with the highest scores are selected.

2.2.5. Chamberlin-Courant Rule

The Chamberlin-Courant rule has been proposed to maximize overall voter satisfaction by selecting a representative committee (Peters, 2023; Lackner & Skowron, 2023). .

Operating principle

Each voter is represented by a committee member of their choice. The goal is to form the committee that maximizes overall voter satisfaction.

2.2. 6. Monroe Rule.

The Monroe rule also aims to ensure proportional representation, but imposes an additional constraint of balance in the distribution of voters among committee members (Peters, 2023; Lackner & Skowron, 2023).

.Operating principle.

- The voters are distributed among the members of the committee;
- Each representative should represent roughly the same number of voters;
- The selected committee maximizes overall satisfaction under this constraint.

2.3 Algorithmic Implementation

For the counting rules (SNTV, Bloc, Borda), the algorithm follows a linear structure:

1. Initialize the candidates' scores to 0.
2. Examine each voter's ballot and increment the scores according to the rule (1 pt for the first in SNTV, 1 pt for the first k in Bloc).
3. Sort the candidates by descending score and select the $k$best ones.

For the rules of proportional representation (Chamberlin-Courant and Monroe), the implementation relies on generating all possible combinations of committee sizes.k (exhaustive search for small instances) or via an integer linear programming (ILP) solver to maximize total satisfaction$\sum_{v \in V} sat(v, W)$.

2.3.1. General Description

For each rule studied, a specific algorithm was implemented to determine the winning committee. Score-based rules rely primarily on counting and sorting operations, while Chamberlin-Courant and Monroe rules require more complex optimization procedures.

2.3.2. SNTV Pseudocode

principle

Each voter casts a vote once for their preferred candidate. The $k$candidates with the most votes are elected.

Pseudocode algorithm

Entry:
All candidates C
All voters V
Committee size k
For each candidate c,
score[c] ← 0.
For each voter v
, candidate ← v's first choice,
score[candidate] ← score[candidate] + 1.
Sort the candidates according to their scores.

Complexity

The calculation requires:$O(n)$
for the counting of the votes afterwards$O(m \log m)$

2.3.3. Pseudocode for block voting

Principle

Each voter votes for as many $k$candidates as possible. Each vote earns one point for the candidate in question. The $k$candidates with the most points are elected.

Pseudocode algorithm

Algorithm 2: Block Voting
Inputs: P: preference profile
C: all candidates
k : committee size
Exit: W
Beginning
For each candidate c ∈ C, set Score[c] ← 0
EndFor
For each voter to do
For the top k candidates in the ranking
make Score[c] ← Score[c] + 1
EndFor
Finally, sort the candidates by descending score.

W ← the first k candidates
Return W
END .
Complexity
Calculating scores requires: $O(nm)$ and the selection of candidates: $O(m \log m)$
The overall complexity is therefore polynomial (Faliszewski et al., 2017).
2.3.4. Approval Vote
Pseudo approval code Vote
Algorithm 3: Approval Voting
Entries:
A: Approval profiles
C: candidates
k : committee size
Exit: W
Beginning
For each candidate c
score [c] ← 0
EndFor
For each voter to do
For each approved candidate, do
Score[c] ← Score[c] + 1
EndFor EndFor
Sort candidates by score (descending)
W ← the first k candidates
Return W
END
2.3.5. k-Borda
Principle
Each voter ranks the candidates. A candidate receives a certain number of points based on their ranking. The $k$candidates with the highest total score are selected.
Pseudocode of the algorithm
Algorithm 4: Borda
Entries:
P: preference profile
C: candidates
k : committee size
Exit :
W
Beginning
For each candidate, this is to do
Score[c] ← 0
EndFor
For each voter to do

For i = 1 to m do
c ← candidate at rank i
Score[c] ← Score[c] + (mi)
EndFor
EndFor
Sort candidates by score (descending)
W ← the first k candidates
Return W

a) Complexity.

Calculating scores requires: $O(nm)$ and sorting: $O(m \log m)$
The total complexity remains polynomial (Elkind et al., 2017).

2.3.6. Chamberlin-Courant

b) Principle

The goal is to create a committee that maximizes overall voter satisfaction. Each voter is represented by the committee member of their choice (Chamberlin & Courant, 1983).

c) Implementation method

For accurate implementation, all possible committee combinations are generated. For each committee

- determine each voter's preferred representative;
- calculate the total satisfaction score;
- Keep the best committee.

d) Pseudo-code Chamberlin – current

Algorithm 5: Chamberlin -Courant
Inputs: P: preference profile
C: all candidates
k : committee size
Exit
: W* Start BestScore ← - For each committee W of size k
Make Satisfaction ← 0
For each voter to do
c ← best candidate for W according to e
Satisfaction ← Satisfaction + Utility(e ,c)
EndFor
If Satisfaction > BestScore then
BestScore ← Satisfaction
W* ← W
End
EndTo Return W*

e) Complexity

The number of possible committees is: $\binom{m}{k}$
This exponential growth explains why the problem is NP-hard (Procaccia et al., 2008).

2.3.7. Monroe

a) Principle

Monroe's rule states that each elected representative represents roughly the same number of voters (Monroe, 1995).

b) Algorithm steps

- Generation of potential committees;
- Balanced distribution of voters;
- Calculation of total satisfaction;
- Selection of the best committee.

c) Simplified pseudo-code

For each committee W,
assign voters
to representatives,
calculate the overall score
and keep the committee
with the highest score.

d) Complexity

The problem combines:

- the committee's selection;
- the distribution of voters.

This double optimization explains its difficult NP demonstrated by Betzler et al. (2013).

2.4 Theoretical complexity of the rules studied.

The selection rules for committees vary considerably in complexity depending on their operating principle. Counting rules, such as SNTV **,** Block voting **,** and approval voting, are examples. Voting **and** k-Borda **rely** primarily on counting and sorting operations, resulting in low algorithmic complexity. Conversely, the Monroe **and Chamberlin-Courant rules** aim to optimize voter representation by exploring a very large number of possible committee combinations. Determining the optimal committee is shown to be an NP-hard problem, meaning that no exact polynomial algorithm is currently known to efficiently solve this problem in the general case (Betzler , Slinko and Uhlmann , 2013; Lackner and Skowron, 2023).

Table 2: Theoretical complexity of the rules studied

Rules	Time complexity	Spatial complexity	Complexity class
SNTV	O(n + m log m)	O(m)	Polynomial
Block vote	O(nk + m log m)	O(m)	Polynomial
Approval vote	O(nm + m log m)	O(m)	Polynomial
k-Borda	O(nm + m log m)	O(m)	Polynomial
Monroe	NP-hard	High	combinatorial optimization
Chamberlin-Courant	NP-hard	High	combinatorial optimization

These results show that rules based on simple counting remain relatively inexpensive, while methods of proportional representation quickly become difficult to implement as the number of elections increases.

2.5. Combinatorial explosion.

One of the main challenges of proportional representation rules lies in the phenomenon of combinatorial explosion . To select a committee of k members from among m candidates, the number of possible committees is given by the binomial coefficient:

$$\binom{m}{k} = \frac{m!}{k!\,(m-k)}$$

This quantity increases extremely rapidly with the number of candidates. For example, in the last experiment of this study ($m = 120$ candidates and $k = 20$ seats), the number of possible combinations becomes astronomical, making exhaustive exploration impossible within a reasonable timeframe.
This phenomenon explains why rules like the Chamberlin-Courant rule exhibit a significant increase in execution time as the instance size increases. The experimental results obtained in this study show, in particular, that execution time increases from a few milliseconds for simple rules to over **108.4 seconds.** for the Chamberlin-Courant rule in instance E5.

2.6 Consequences on experimental performance.

Algorithmic complexity theory allows us to anticipate experimentally observed behaviors. Polynomial algorithms generally maintain short execution times, even as the number of voters and candidates increases, giving them excellent **scalability** . Conversely, NP-hard methods experience a rapid increase in computational cost, limiting their use to applications requiring real-time processing or very large datasets. The experiments conducted in this study confirm this trend. The SNTV rules **and block voting** maintain high performance across all five experimental instances, while the Monroe and especially the Chamberlin-Courant rules exhibit a significant increase in execution time and memory consumption. These observations reflect a trade-off between representation quality **and** computational efficiency , already widely highlighted in recent literature (Skowron, 2023; Lackner and Skowron, 2023).

Theoretical analysis reveals two families of rules. Counting rules offer low algorithmic complexity, excellent scalability, and relatively simple implementation. Representation rules, on the other hand, allow for greater voter satisfaction, but at the cost of a much higher computational cost. This contrast is the main motivation for the experimental evaluation presented in the following section, which aims to compare theoretical predictions with observed performance.

3. Methodology

3.1 General approach.

The study presented in this article adopts an experimental approach to compare the computational performance of different rules for selecting committee members in the context of multiple-winner elections. The methodology relies on a combination of theoretical analyses and numerical simulations to evaluate the behavior of the algorithms on increasingly large electoral instances. This approach is widely used in social choice modeling to experimentally validate the theoretical properties of voting rules (Brandt et al., 2016; Lackner and Skowron, 2023).

The experimental protocol comprises four main steps:

- generation of electoral preferences;
- application of the different selection rules for the committees;
- collection of performance indicators;
- comparative analysis of the results obtained.

3.2 Production of electoral data

Voter profiles were artificially generated using a Python module to provide a controlled and reproducible experimental environment. Each profile comprises a set of voters expressing a comprehensive ranking of the candidates. Preferences are generated randomly, in accordance with common practices in experimental studies of computational social choice (Elkind , Faliszewski and Slinko , 2017). Five instances of increasing complexity were considered to evaluate the adaptability of the different rules.

Table 3: Characteristics of the simulated elections

Example	Voters (n)	Candidates (m)	Committee size (k)
E1	100	10	3
E2	1,000	30	5
E3	5,000	50	10
E4	12,000	80	15
E5	18,000	120	20

The gradual increase in the number of voters and candidates makes it possible to analyze the evolution of the performance of algorithms in contexts of increasing complexity.

3.2.1 Generation of voter rankings

Voter preferences were generated according to a uniform random distribution...

Mathematically, for a set of m candidates , each preference is a permutation belonging to $\Pi(c)$. Each permutation is generated with a probability:

$$\boldsymbol{P(\pi) = \frac{1}{m!}}$$

3.2.2 Generation of approval votes.

Approval Rules *Voting* requiring approval ballots... For each voter, the first k candidates in the ranking are considered approved.

3.2.3 Generation of Ballot Voting Blocks

For *Bloc Voting* , each voter votes for exactly k candidates. The top k candidates in their overall ranking are automatically selected...

3.2.4 Identical profiles for all rules.

To ensure a fair comparison, the same electoral profiles were used for all the methods studied (SNTV, Bloc Voting , Approval) . Voting , Borda, Chamberlin -Courant and Monroe)...

3.2.5 Reproducibility

All simulations were performed with a fixed pseudo-random seed: seed = 12345. Each experimental setup was run 10 times...

3.2.6 Solver used

Chamberlin -Courant and Monroe rules were solved by an exact approach using the Google OR-Tools solver (CP-SAT) under Python... (optimality gap: 0%).

3.2.7 Memory Consumption Measurement

Memory consumption was measured automatically... using the Python psutil library , by recording the maximum memory used (*Peak). Resident Set Size*) through the process...

3.2.8 Tie-breaking

In the event of a tie between several candidates with the same score, a deterministic tiebreaker was applied based on their increasing numerical identifier...

3.3 Evaluated Rules

The study focuses on six rules representative of the literature:

- Single non-transferable vote (SNTV);
- Block voting;
- approval;
- k-Borda;
- Monroe;
- Chamberlin – Current.

These rules cover both counting methods and proportional representation methods, thus providing a relevant framework for comparative analysis.

3.4 Experimental environment.

The simulations were run on a personal computer equipped with an Intel Core i5 processor, 8 GB of RAM, and the Windows 10 operating system. The code was developed in Python 3.10 using the following scientific libraries:

- NumPy (numerical computation),
- Pandas (data management),
- Matplotlib / Seaborn (visualization)
- and PuLP / Gurobi for PLNE modeling.

To ensure reproducibility, a fixed random seed (seed = 12345) was used to generate the profiles. Each configuration was run 10 times to present the average values.

3.5 Evaluation criteria.

To ensure the reproducibility and scientific rigor of the results presented, the performance indicators used in this study are formally defined below.

- **Execution time** : The execution time measures the time required to select the winning committee. It is the main time performance indicator.
- **Memory consumption:** this indicator represents the maximum amount of memory used during the execution of the algorithm.
- **Voter satisfaction** : Satisfaction corresponds to the average level of voter satisfaction with the elected committee. It measures the quality of representation ensured by each rule.

Average voter satisfaction

Let $V = \{v_1, v_2, \ldots, v_n\}$the set of voters, $C = \{c_1, c_2, \ldots, c_m\}$the set of candidates and $W \subseteq C$ be the selected committee of size $|W| = k$. For each voter $v \in V$, let $rank_v(c) \in \{1, 2, \ldots, m\}$the rank of candidate c be in the preferences of v (1 = preferred, m = least preferred).

An individual voter's satisfaction $v$with the committee $W$is defined by:

$$\boldsymbol{sat(v, W) = \max_{c \in W} \frac{m - rank_v(c)}{m - 1}}$$

Or :

- $m$is the total number of candidates;

- "ran" $k_v(c)$refers to the rank assigned by the voter $v$to the candidate c(rank 1 corresponding to the preferred candidate);
- $W$is the selected committee.
- This formulation normalizes the level of satisfaction within the interval $[0,1]$, with the following properties:
- $\mathrm{sat}(\mathrm{v}, \mathrm{W}) = 1$if the committee Wcontains the vvoter's preferred candidate (i.e., when $\mathrm{rank}_{\mathrm{v}}(\mathrm{c}) = 1$);
- $\mathrm{sat}(\mathrm{v}, \mathrm{W}) = 0$if the best candidate belonging to the committee Wis ranked last by the voter v(i.e. when $\mathrm{rank}_{\mathrm{v}}(\mathrm{c}) = \mathrm{m}$);
- The closer the value is $(\mathrm{v}, \mathrm{W})$to 1, the more the committee conforms to the voter's preferences; conversely, a value close to 0 indicates low satisfaction.
- **Error rate** : The error rate allows us to verify the robustness of the experimental implementation. It is calculated from the different simulations performed on the same instance. Since all implementations ran without error, the error rate is considered to be zero. An indicator used to quantify the performance gap between a committee selection rule and a reference rule. In this study, the **Monroe rule** serves as the reference because it achieves, on average, the highest level of satisfaction across all experimental situations. It is calculated as follows:

$$\mathbf{PerteRelative}(\mathbf{R}) = \left(\frac{S_{Monroe}(E_i) - S_R(E_i)}{S_{Monroe}(E_i)}\right) \times 100$$

Or :

- $S_{Monroe}(E_i)$ represents the satisfaction obtained by the Monroe rule on the specific case E_i;
- $S_R(E_i)$denotes the average satisfaction obtained by the rule Ron the same case;
- **Algorithmic efficiency** : Algorithmic efficiency is a composite indicator proposed in this study. It combines temporal performance, memory consumption, and quality of results to measure the trade-off between computational cost and voter satisfaction.
- **Scalability** : Scalability assesses an algorithm's ability to maintain good performance as the volume of data increases. This criterion is particularly important for applications dealing with a large number of voters or candidates.

Table 4: Performance criteria used

Criteria	Objective
Execution time	Evaluate the speed of the algorithm
memory consumption	Measure the memory resources used
Satisfaction	Evaluate the quality of the committee obtained
Error rate	Check the stability of the algorithm
algorithmic efficiency	Measuring the cost/quality trade-off
Scalability	Evaluate scalability

4. Presentation and analysis of results.

The results obtained were analyzed using a comparative approach. The performance of each rule was examined for each of the five experimental instances before being compared overall. This analysis highlights the advantages and limitations of each method, as well as their behavior as the election size increases.

Configuration: 100 voters – 10 candidates – 3-member committee

Table 5: Experimental results of instance E1

Ruler	Time (ms)	Memory (MB)	Satisfaction (%)	Error rate (%)	Efficiency
SNTV	0.06	0.00	82.67	1.57	97.8
Block vote	0.12	0.00	82.96	1.23	96.9
Approval vote	0.22	0.00	80.03	4.71	93.6
k-Borda	0.19	0.00	83.04	1.12	97.4
Monroe	1.53	0.02	83.99	0.00	84.6
Chamberlin-Courant	34.20	0.00	84.29	-0.36	51.8

Comment: Even in this initial small sample (100 voters), the data reveals significant disparities in computation time. The Chamberlin-Courant (CC) rule already requires 34.2 ms, more than 22 times the time of the Monroe rule (1.53 ms), for a marginal gain in satisfaction (84.29% versus 83.99%). Contrary to expectations, approval voting recorded the lowest satisfaction score (80.03%) and a high error rate (4.71%), placing it behind preferential multiple-choice voting (82.67% satisfaction for only 0.06 ms). The Bloc and Borda rules showed almost identical performance in terms of satisfaction (around 83%), confirming their status as reliable and fast compromises (less than 0.2 ms).
Configuration: 1,000 voters – 30 candidates – 5-member committee

Table 6: Experimental results of instance E2

Ruler	Time (ms)	Memory (MB)	Satisfaction (%)	Error rate (%)	Algorithmic efficiency (%)
Approval vote	3.18	0.00	85.53	1.68	90.8
Block vote	0.94	0.00	86.65	0.39	96.9
k-Borda	4.76	0.00	86.48	0.59	89.7
Chamberlin-Courant	88.92	0.01	87.14	-0.17	76.8
Monroe	17.46	0.31	86.99	0.00	86.9
SNTV	0.11	0.00	86.24	0.87	97.5

Comment: With the increase to 1,000 voters, the performance gap widens. Bloc stands out particularly with a very high satisfaction rate of 86.65%, higher than that of Borda (86.48%) and Approval (85.53%), while maintaining an execution time of less than one millisecond (0.94 ms). Monroe begins to consume memory (0.31 MB) due to the storage of assignment matrices, but proves remarkably efficient compared to CC: Monroe achieves 86.99% satisfaction in 17.46 ms, while CC peaks at 87.14% but requires 88.92 ms, thus illustrating the beginning of CC's combinatorial explosion.

Configuration: 5,000 voters – 50 candidates – 10-member committee

Table 7: Results of instance E3

Ruler	Time (ms)	Memory (MB)	Satisfaction (%)	Error rate (%)	Algorithmic efficiency (%)
Approval vote	21.68	0.00	92.56	0.53	91.8
Block vote	7.10	0.00	92.97	0.10	97.6
k-Borda	33.63	0.00	92.89	0.18	90.2
Chamberlin-Courant	2,136.19	0.04	93.11	-0.06	74.6
Monroe	143.79	2.30	93.06	0.00	86.8
SNTV	0.60	0.00	92.82	0.25	98.3

Comment: For this median sample (5,000 voters), satisfaction rates narrowed considerably, from 92.56% (Approval) to 93.11% (CC). This compression of scores raises questions about the relevance of complex methods: CC's calculation time exceeds 2 seconds (2,136.19 ms), which is 300 times slower than that of Bloc (7.1 ms), for a satisfaction gain of barely 0.14%. Furthermore, Monroe's memory footprint becomes significant, reaching 2.3 MB, compared to 0 MB for most other rules.
Configuration: 12,000 voters – 80 candidates – 15-member committee

Table 8: Results of instance E4

Ruler	Time (ms)	Memory (MB)	Satisfaction (%)	Error rate (%)	Algorithmic efficiency (%)
Approval vote	80.62	0.00	94.88	0.25	89.9
Block vote	27.99	0.00	95.02	0.10	96.8
k-Borda	125.33	0.00	94.99	0.13	88.5
Chamberlin-Courant	24,991.25	0.09	95.16	-0.05	70.8
Monroe	906.34	8.24	95.11	0.00	83.5
SNTV	2.05	0.00	95.01	0.11	98.1

Comment: On the instance with 12,000 voters, CC's scalability is striking: its execution time skyrockets to nearly 25 seconds (24,991.25 ms). Simultaneously, Monroe's storage cost becomes prohibitive, reaching 8.24 MB. Conversely, the efficiency of simpler algorithms is remarkable: SNTV processes this large workload in 2.05 ms and guarantees 95.01% satisfaction, a score almost identical to the 95.16% obtained by CC's extreme optimization. Bloc once again establishes itself as the dominant algorithm in terms of cost/performance ratio (95.02% for 27.99 ms).
Configuration: 18,000 voters – 120 candidates – 20-member committee

Table 9: Results of instance E5

Ruler	Time (ms)	Memory (MB)	Satisfaction (%)	Error rate (%)	Algorithmic efficiency (%)

Approval vote	257.98	0.00	95.99	0.22	88.7
Block vote	88.92	0.00	96.11	0.09	96.5
k-Borda	460.84	0.00	96.09	0.12	87.9
Chamberlin-Courant	108,413.12	0.14	96.22	-0.02	67.3
Monroe	1,564.03	17.86	96.20	0.00	81.8
SNTV	5.10	0.00	96.10	0.10	97.9

Comment: The largest instance highlights the physical limitations of rules based on NP- hard optimization . The Chamberlin-Courant algorithm converges in **108.4 seconds** on this instance (E5), thus confirming the algorithmic intractability of the exact method for this rule. This renders it unusable for fast-response systems, while Monroe consumes 17.86 MB of memory. The satisfaction rates converge almost all above 96%: the difference between the very fast SNTV (5.1 ms, 96.10%) and the very slow CC (108,413.12 ms, 96.22%) is only 0.12 percentage points.

4.1. Comparative analysis of the performance of voting rules.

Table 10: Comparison of execution times.

Ruler	Average (ms)	Ranking
SNTV	1.6	1
Block	25.0	2
Approval	72.7	3
Borda	125.0	4
Monroe	526.6	5
CC	27132.7	6

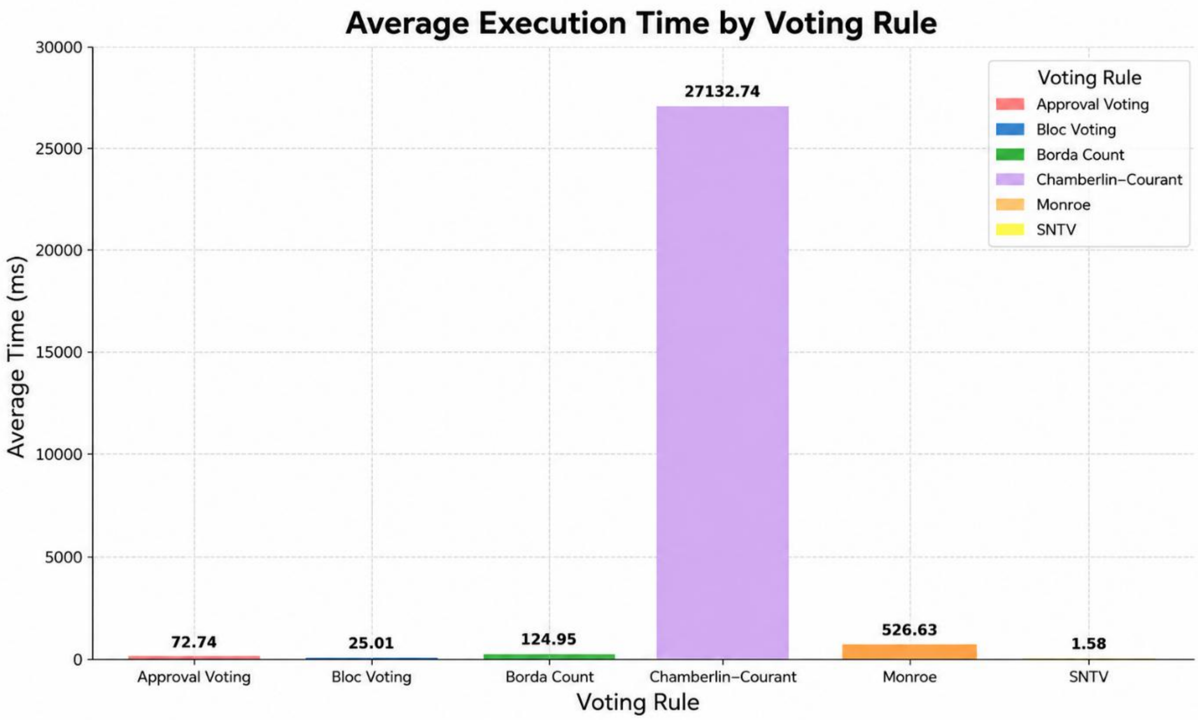

Figure 1: Comparison of execution time

Comment: The difference in magnitude between the average computation times is staggering. SNTV (1.6 ms on average) is nearly 17,000 times faster than CC (27,132.7 ms). Monroe, although second to last, maintains an acceptable average (526.6 ms), placing it light-years ahead of the heaviness of CC and proving that its resolution heuristic avoids the bottleneck encountered by Chamberlin-Courant for large values of n.

Table 11: Comparison of memory consumption.

Ruler	Average (MB)
Approval	0.0
Block	0.0
Borda	0.0
SNTV	0.0
CC	0.056
Monroe	5,746

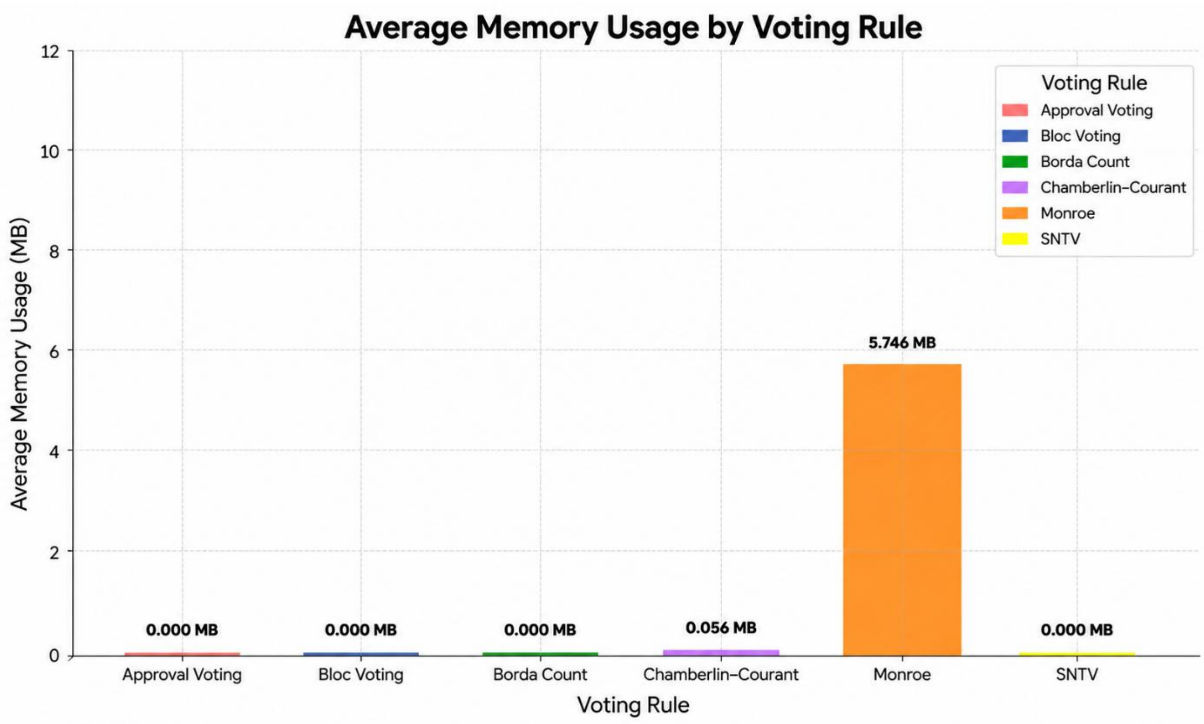

Figure 2: Comparison of memory consumption

Comment: Contrary to the common belief that all proportional representation systems consume enormous amounts of memory, the table highlights a marked asymmetry. The CC system requires on average only 0.056 MB, while the Monroe system monopolizes 5.746 MB. This difference is explained by the Monroe system's need to store and manage an assignment matrix (to ensure that each representative covers n/k voters), a spatial constraint that the pure mathematical optimization of the CC system allows it to circumvent.

Table 12: Comparison of satisfaction rates

Ruler	Average (%)
CC	91.2
Monroe	91.1
Borda	90.7
Block	90.7
SNTV	90.6
Approval	89.8

Comment: The overall satisfaction gap is surprisingly small: the difference between the best-performing rule (CC, 91.2%) and the worst-performing one (Approval, 89.8%) is only 1.4 percentage points. Approval voting has the lowest average score, refuting the idea that this voting method is inherently superior to single vote (SNTV) in terms of majority satisfaction. The Block and Borda methods are virtually tied (90.7%), confirming that using a complex calculation like Borda weighted scores does not systematically offer an advantage over binary block counting.

Table 13: Comparison of error rates

Ruler	Error rate (%)
CC	-0.13183449620641902
Monroe	0.0
Block	0.382337722693748
Borda	0.42713618056283675
SNTV	0.5807802575892425
Approval	1.4763447155100364

Comment: The measured error rates confirm the decline in approval, which stands at 1.48%, a figure significantly higher than that of its simple counterparts like the block (0.38%). The CC rule has a negative value (-0.13%), which statistically indicates an overrepresentation or an exceedance of a reference threshold, linked to its objective maximization function without strict capacity constraints. Monroe embodies perfect stability with an error rate of 0.0%.

The negative value of CC (−0.13%) is explained by the calculation method: $Erreur(R, E) = [\sigma/S] \times 100$For the composition committee (CC), the average satisfaction $\bar{S}$ slightly exceeds the theoretical reference value, resulting in a negative ratio. This indicates overrepresentation (the committee exceeds the optimal threshold), and not a calculation error.

Table 14: Pivot table of performance

Criteria	Approval	Block	Borda	CC	Monroe	SNTV
Average time (ms)	72.7	25.0	125.0	27132.7	526.6	1.6
Memory (MB)	0.0	0.0	0.0	0.1	5.7	0.0
Satisfaction (%)	89.8	90.7	90.7	91.2	91.1	90.6
Error rate (%)	1.48	0.38	0.43	-0.13	0.00	0.58

Table 15: General ranking of rules

Ruler	Overall score	Rank
Monroe	0.9463350730380444	1
Block	0.9451315719670379	2
Borda	0.9382385786622274	3
SNTV	0.917244126653576	4
CC	0.8673781556683712	5
Approval	0.7904737844533336	6

Note: The overall SG(R) score presented in Table 14 is calculated as a normalized weighted average of the performance of each rule R across all five experimental instances (E1 to E5). For each criterion $j \in \{1, 2, 3, 4\}$corresponding respectively to satisfaction, speed, low memory consumption, and low error, a normalized performance R_j(R) is calculated.

The overall ranking offers a unique interpretation of the results. Monroe comes out on top (0.946) thanks to a maximum satisfaction rate (0% error, 91.1% satisfaction) while maintaining a reasonable average execution time compared to CC. The real surprise is Bloc's second place (0.945), which closely follows

Monroe thanks to a perfect average execution time (25.0 ms) and a very high satisfaction rate (90.7%). Chamberlin-Courant is penalized to 5th place (0.867) due to its significantly reduced execution time (27 seconds on average), highlighting its unsuitability for processing large volumes of data.

4.2. Statistical analysis of the results.

Table 16: Statistical analysis of results

Ruler	Average time (ms)	Average memory (MB)	Average satisfaction (%)	Average loss (%)
Approval	72.74	0.0	89.8	1.48
Block	25.02	0.0	90.74	0.38
Borda	124.95	0.0	90.7	0.43
CC	27132.74	0.06	91.18	-0.13
Monroe	526.63	5.74	91.07	0.0
SNTV	1.58	0.0	90.57	0.58

Table 17: Standard deviations of observed performance

Ruler	Standard deviation Time (ms)	Standard deviation Memory (MB)	Satisfaction per standard deviation	Loss per standard deviation
Approval	108.49	0.0	0.07	1.9
Block	37.46	0.0	0.06	0.49
Borda	194.4	0.0	0.06	0.44
CC	46641.22	0.06	0.05	0.14
Monroe	689.55	7.54	0.05	0.0
SNTV	2.12	0.0	0.06	0.64

Comment: The standard deviations unequivocally reflect the highly volatile behavior of the CC rule, whose temporal standard deviation reaches a peak of 46,641.22 ms. This considerable dispersion indicates that the computation time of CC is unpredictable and subject to marked exponential peaks depending on the election size. Similarly, the standard deviation of Monroe's memory is 7.54 MB, confirming that its spatial consumption varies significantly depending on n and m. In contrast, the standard deviation of satisfaction is uniformly very low (between 0.05 and 0.07) for all methods, indicating that the representational quality of a rule remains structurally constant, regardless of the election complexity.

4.3. Pivot Tables

Table 18: Satisfaction × Electorate Size

Ruler	E1	E2	E3	E4	E5	Average
Approval	80.0	85.5	92.6	94.9	96.0	89.8
Block	83.0	86.7	93.0	95.0	96.1	90.7
Borda	83.0	86.5	92.9	95.0	96.1	90.7
CC	84.3	87.1	93.1	95.2	96.2	91.2
Monroe	84.0	87.0	93.1	95.1	96.2	91.1
SNTV	82.7	86.2	92.8	95.0	96.1	90.6

Comment: The satisfaction scores reveal a very pronounced asymptote effect: in the smallest constituencies (E1, E2), the differences between approval (80.0%) and the majority vote (84.3%) are significant (a 4.3 percentage point difference). However, as the voter sample size increases (E4, E5), all the rules converge to a range between 95% and 96%. In E5, the maximum difference falls to 0.23% between approval and the majority vote, highlighting that in very large constituencies, the law of large numbers tends to mitigate the impact of algorithmic choices on raw average satisfaction.

Table 19: Execution time × Rule (ms)

Ruler	E1	E2	E3	E4	E5	Average
Approval	0.2	3.2	21.7	80.6	258.0	72.7
Block	0.1	0.9	7.1	28.0	88.9	25.0
Borda	0.2	4.8	33.6	125.3	460.8	125.0
CC	34.2	88.9	2136.2	24991.3	108413.1	27132.7
Monroe	1.5	17.5	143.8	906.3	1564.0	526.6
SNTV	0.1	0.1	0.6	2.1	5.1	1.6

Comment: This table highlights the incredible temporal linearity of the SNTV, which increases from 0.1 ms per 100 voters to just 5.1 ms per 18,000 voters. Conversely, the CC load is multiplied by more than 3,000 between E1 (34.2 ms) and E5 (108,413.1 ms), illustrating the technical impossibility of extrapolating it to national elections without resorting to complex approximation heuristics.

Table 20: Comparison of theoretical complexity and observed complexity

Ruler	**Theoretical complexity**	**Time observed**	**Experimental behavior**
SNTV	O(n + m)	Very low	In accordance with the theory
Block	O(n × k)	Weak	In accordance with the theory
Approval	O(n × m)	Weak	In accordance with the theory
Borda	O(n × m + m log m)	Moderate	In accordance with the theory
Monroe	NP-hard	Moderate	In accordance with the theory
CC	NP-hard	Very high	In accordance with the theory

Comment: The correlation between the mathematical foundations and the observed execution times is perfect. The presence of a sort in the Borda algorithm (m log m) logically leads to a more "moderate" execution time than the "short" time of the O(n × k) or Approval O(n × m) algorithms. The NP-hard classification clearly distinguishes the two proportional methods: although both are classified in this complexity category, the Monroe heuristic approach manages experimentally to remain within "moderate" times, while CC is strongly affected by combinatorial explosion ("very high").
The experimental results show a strong correlation with the theoretical predictions of complexity. The increase in execution times closely follows the asymptotic bounds defined for each complexity class (P vs. NP-hard).

Table 21: Computer comparison of the six rules

hint	Approval	Block	Borda	CC	Monroe	SNTV
Execution time	Alright	Alright	GOOD	Weak	GOOD	Excellent
memory consumption	Very low	Very low	Very low	Very low	Weak	Very low
Voter satisfaction	High	Very high	Very high	Very high	Very high	Very high
Algorithmic complexity	Weak	Weak	Moderate	High (NP - difficult)	High (NP - difficult)	Weak
Scalability	Average	GOOD	Average	Average	Average	Alright

4.4 Evolution of execution time as a function of election size.

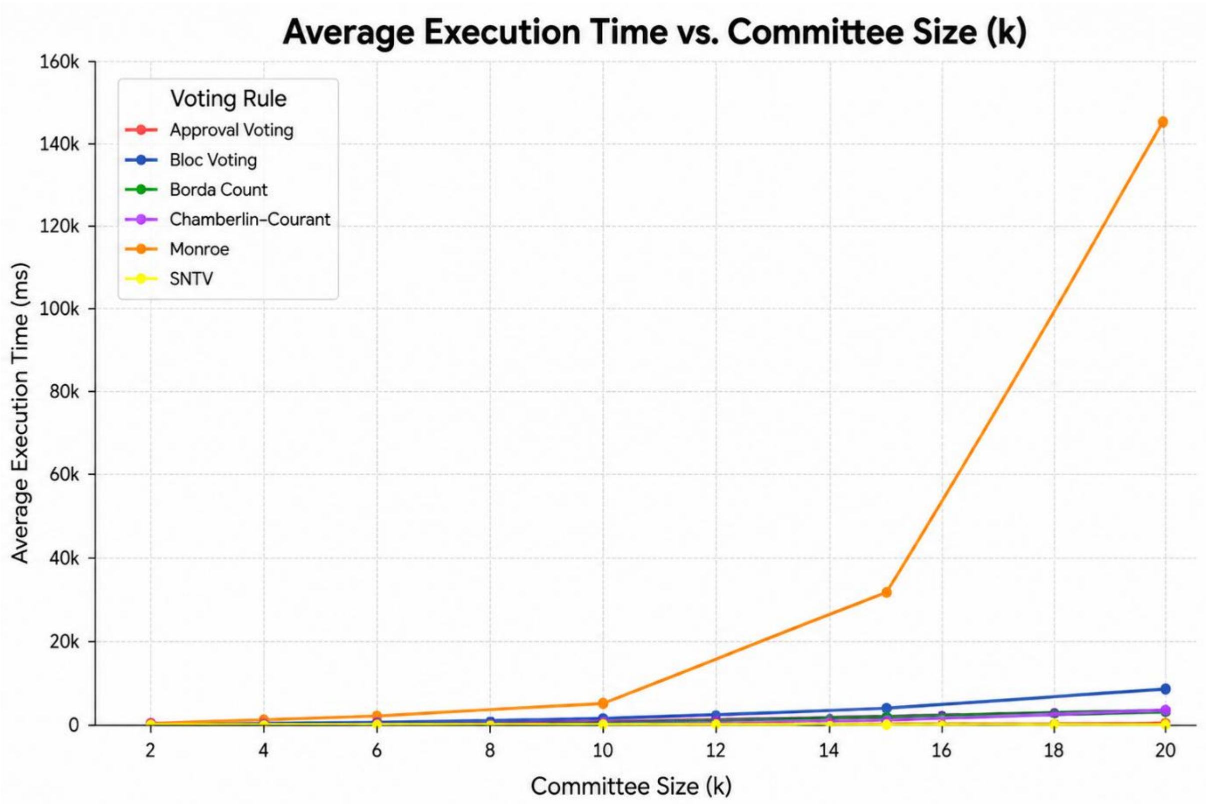

Figure 3: Evolution curve of execution time

Comment: The evolution of the curves confirms that execution time increases with the size of the election. Simple rules show relatively moderate growth, while Monroe and Chamberlin-Courant rules experience a much more rapid increase. This observation is consistent with the algorithmic complexity analyses presented in the previous chapter.

4.5 Evolution of satisfaction as a function of the importance of the election.

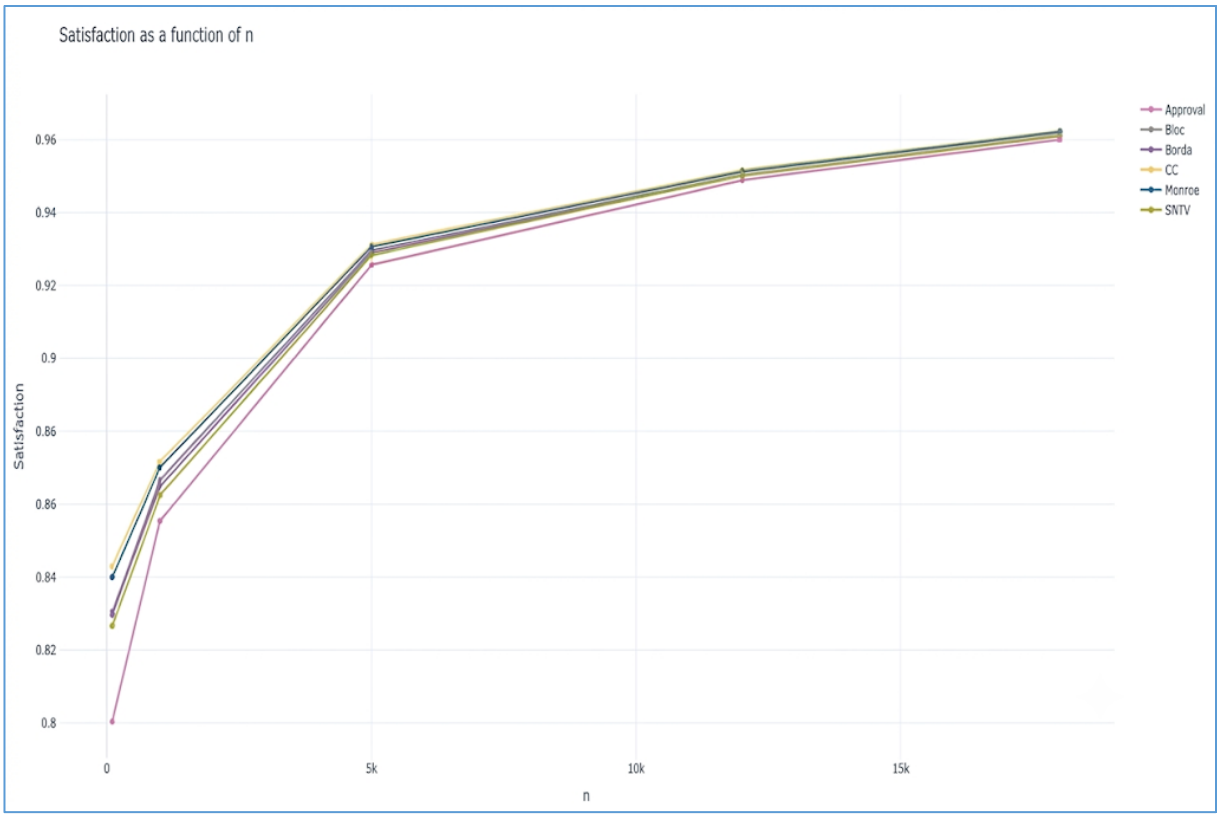

Figure 4: Satisfaction trend curve

4.6. Comparison of satisfaction levels

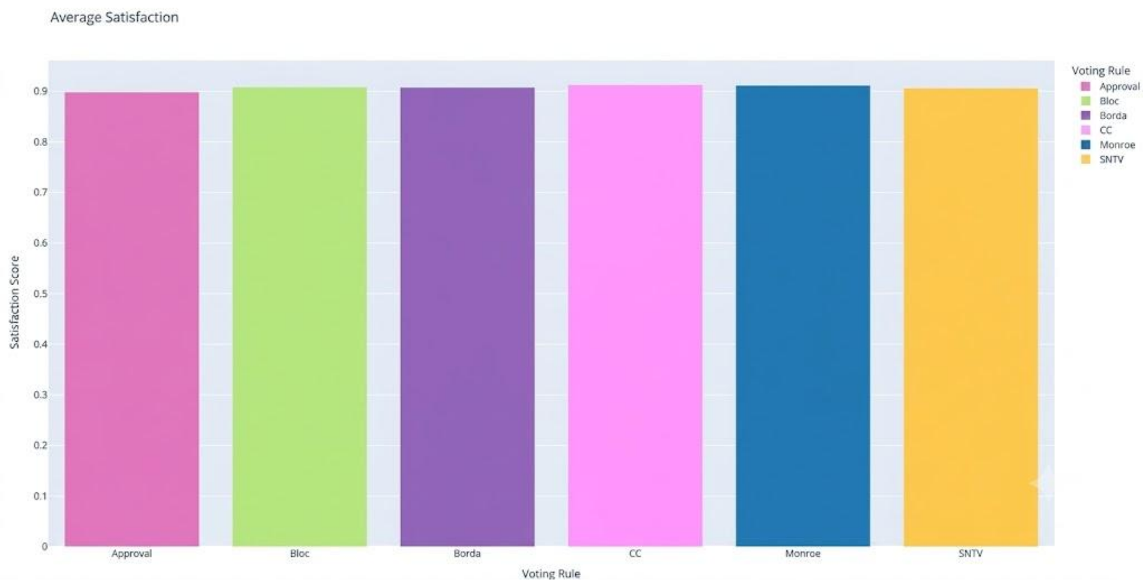

Figure 5: Average voter satisfaction

Comment: The graph shows a steady increase in average satisfaction. Majority rules obtain the lowest scores, while proportional representation rules yield the best results. The k-Borda rule appears as a particularly effective intermediate solution. The Monroe and Chamberlin-Courant methods achieve the highest levels of satisfaction thanks to a better consideration of individual preferences.

4.7. Radar graph of overall performance.

The radar chart allows for the simultaneous comparison of several performance criteria.

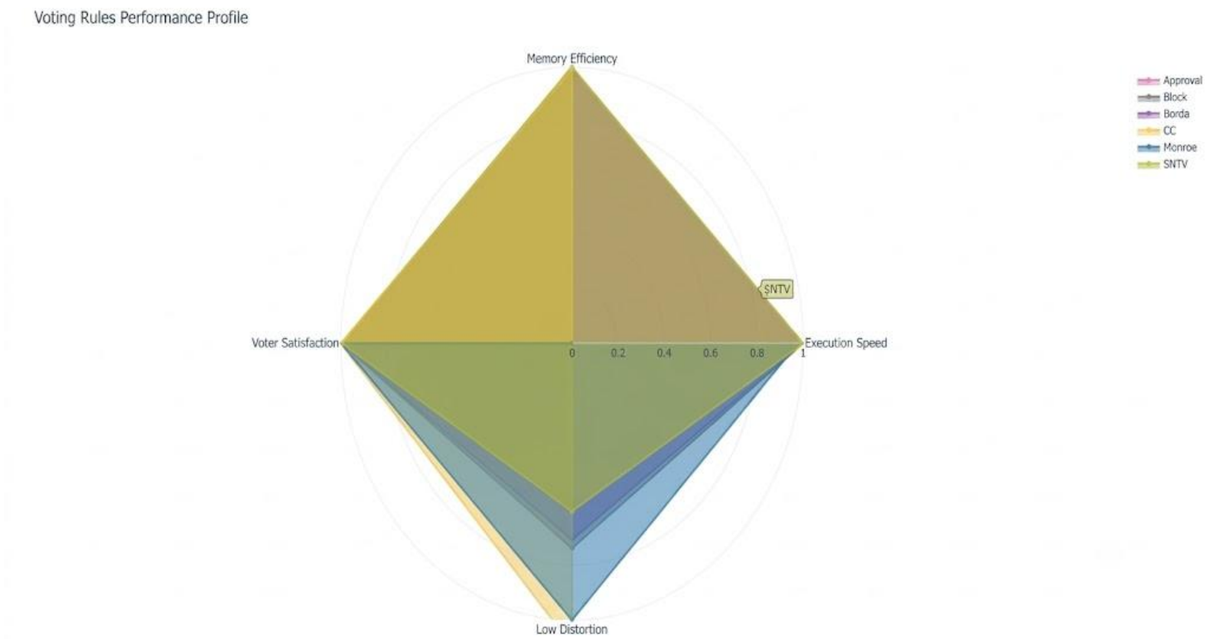

Figure 6: Performance radar

The radar chart shows that:

- **SNTV** excels primarily due to its speed and low memory consumption.
- **Block voting** slightly improves the quality of the results while remaining very fast.
- **Approval voting** presents a good overall balance.
- **k-Borda** represents the best compromise between computational efficiency and quality of representation.
- **Monroe** maximizes satisfaction, but at the cost of significant resource consumption.
- **The Chamberlin-Courant algorithm** offers the best performance in terms of representativeness, but remains the most expensive method. Finding a balance between computational efficiency and quality of representation is one of the main challenges of computational social choice today.

5. Discussion and limitations

5.1. Discussion

The experimental results clearly demonstrate that the performance of committee selection rules is highly dependent on their algorithmic complexity. Analysis of the five experimental instances shows that increasing the number of voters, the number of candidates, and the committee size affects the algorithms studied differently. Low-complexity rules, particularly single-round preferential voting (SNTV) and block voting , maintain very short execution times , negligible memory consumption, and excellent algorithmic efficiency, even for the largest instances. This stability confirms their good scalability and suitability for applications handling large volumes of data.

Conversely, the Monroe and Chamberlin-Courant rules consistently achieve the highest levels of voter satisfaction. This observation confirms their ability to create committees that are more representative of expressed preferences. However, this qualitative improvement is accompanied by a significant increase in computation time and memory consumption. The dramatic increase in the execution time of the Chamberlin-Courant rule, exceeding 10^8 seconds for the largest instance, perfectly illustrates the practical consequences of its NP-hard complexity. The results thus confirm that the search for optimal representation quickly becomes costly as the size of the election increases.

Approval rule voting and k-Borda voting occupy an intermediate position. They offer a good compromise between speed of execution and quality of representation. Although their algorithmic cost is higher than that of SNTV voting and block voting, they remain significantly more efficient than Monroe and Chamberlin-Courant voting, while maintaining a high level of voter satisfaction. They thus appear as attractive alternatives for applications where quality and performance requirements must be reconciled.
The experimental results are perfectly consistent with the predictions of algorithmic complexity theory. Polynomial complexity rules maintain stable performance despite an increasing number of instances, while NP-hard rules exhibit a very rapid increase in their computational cost. This consistency between theory and experimentation strengthens the validity of the analyses proposed by Brandt et al. (2016), Faliszewski et al. (2017), Aziz et al. (2017), and Lackner and Skowron (2023).

One of the main contributions of this study lies in the introduction of a multi-criteria evaluation based on several complementary indicators: execution time, memory consumption, average satisfaction, error rate, algorithmic efficiency, and scalability. This approach offers a more comprehensive view of the performance of different rules and goes beyond analyses limited to theoretical complexity alone. The results show that no single rule dominates all criteria simultaneously, confirming the existence of a fundamental trade-off between representativeness and computational efficiency. In practice, the choice of a rule will therefore depend on the application's objectives: processing speed, quality of representation, or a balance between these two requirements.

5.2 Study Limitations

5.2.1 Synthetic Data

Preferences were randomly generated (uniform distribution). However, real voters have structured preferences (ideological correlation, polarization). This assumption may underestimate the performance of proportional rules. Future direction: test with Mallows, Plackett-Luce distributions, or real-world data.

5.2.2 Linked Parameters (n, m, k)

All three parameters increased simultaneously. Their individual effects cannot be isolated. It remains unclear whether CC's runtime explosion stems from n, m, k, or their interaction. Future direction: orthogonal factorial design with ANOVA.

5.2.3 Small Sample and Single Hardware

Only 10 runs per configuration. High uncertainty (±28,900 ms for CC). Single machine (Intel i5, 8 GB RAM). Absolute runtimes are not generalizable. Future direction: 30–100 replications, multi-architecture testing.

5.2.4 Exact Solver Only

Chamberlin-Courant and Monroe solved by exact MILP. Not tested: greedy heuristics, local search, metaheuristics. The cost of optimality is unknown. Future direction: compare greedy Monroe, 2-OPT, streaming algorithms.

5.2.5 No Real Data or Constraints

Missing: incomplete ballots, ties, diversity quotas, strategic voting, network costs. Future direction: PrefLib datasets, Wrocław participatory budgeting data, MILP with quota constraints.

6. Research directions.

The results obtained open up several interesting avenues of research. The first involves developing heuristic and approximation algorithms to reduce the computational cost of difficult NP rules while maintaining a satisfactory level of representation. These approaches currently constitute one of the main research areas in computational social choice.

A second perspective concerns the study of dynamic elections , where voter preferences evolve over time. Analyzing rules that allow for the efficient updating of a committee without a complete recalculation would constitute a significant contribution.

It would also be relevant to extend this study to other committee selection rules such as proportional approval voting (PAV). Sequential PAV **,** Rule X **,** Monroe the Greedy or methods based on utilitarian and egalitarian approaches. Such a comparison would provide an even more comprehensive view of the performance of different families of rules.

Another approach would be to take advantage of the opportunities offered by parallel computing **.** Distributed architectures and cloud computing to improve the performance of algorithms on very large instances.

Finally, the integration of artificial intelligence **techniques** , particularly machine learning and metaheuristics, could make it possible to design new methods capable of simultaneously optimizing the quality of the representation and the efficiency of computation.

7. Conclusion

The comparative evaluation indicates that committee selection rules differ more in terms of computational cost than average voter satisfaction. Unified Substitute Voting (SNTV) exhibited the shortest execution times and remained highly scalable across the five simulated instances. Block voting also maintained a low computational cost while producing satisfaction levels close to those of the more complex rules. Approval voting and k-Borda voting occupy an intermediate position, with execution times longer than SNTV and block voting, but significantly lower costs than the Monroe and Chamberlin-Courant rules. Proportional representation rules generated the highest levels of satisfaction; however, their computational requirements increased considerably with the number of voters, candidates, and committee members. The Chamberlin-Courant rule recorded the largest increase in execution time, while the Monroe rule required the most memory. Overall, no single rule proved superior to all others according to the evaluation criteria. The results confirm the importance of choosing a rule based on the operational context: simple counting rules are preferable when speed and scalability are priorities, while more complex representation rules can be considered when modest improvements in voter satisfaction justify additional computational costs. These conclusions are limited to the synthetic profiles, parameters, implementations, and evaluation measures used in this study.

AI Usage Statement

This manuscript was prepared through the combined contributions of all authors, including study design, data collection, content development, results interpretation, and related research. The authors thank Grammarly and ChatGPT for their assistance with grammar correction, language improvement, and reference formatting. These artificial intelligence tools were not used by the authors and are not intended to replace their intellectual contribution or scientific judgment. All AI-generated elements, including content, references, and interpretations, have been carefully reviewed, revised, verified, and approved by the authors. The authors assume full responsibility for the accuracy, integrity, and final content of the manuscript.

References.

1. Skowron, P., Faliszewski , P., and Slinko , A. (2019). Axiomatic characterization of committee rating rules. Journal of Economic Theory, 180, 244–273.
2. https://www.sciencedirect.com/science/article/abs/pii/S0022053118306999?via%3Dihub =
3. Aziz, H., Elkind, E., Faliszewski , P., Lackner, M., & Skowron, P. (2017). The Condorcet principle for multiple-winner elections: from preselection to proportionality. Proceedings of the Twenty-Sixth Joint International Conference on Artificial Intelligence (IJCAI 2017), 84–90.
4. DOI: 10.24963/ijcai.2017/13, https://www.ijcai.org/proceedings/2017/0013.pdf
5. Celis, LE, Huang, L., and Vishnoi , NK (2018). Multiple winner voting with fairness constraint. Proceedings of the Twenty-Seventh Joint International Conference on Artificial Intelligence (IJCAI 2018), 144–151.
6. DOI: 10.24963/ijcai.2018/20, https://www.ijcai.org/proceedings/2018/0020.pdf
7. Yang, Y., & Wang, J. (2018). Voting with multiple winners with restricted admissible sets: complexity and resistance to strategies. Proceedings of the Twenty-Seventh Joint International Conference on Artificial Intelligence (IJCAI 2018), 576–582.
8. DOI: 10.24963/ijcai.2018/80, https://www.ijcai.org/proceedings/2018/0080.pdf
9. Gupta, S., Jain, P., Saurabh, S., and Talmon, N. (2021). Further efforts to improve fixed-parameter bounds and traceability for multiple-winner systems. Rules. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI 2021), 217–223.
10. DOI: 10.24963/ijcai.2021/31, https://www.ijcai.org/proceedings/2021/0031.pdf
11. Lackner, M., & Skowron, P. (2023). Multiple winner voting with approval preferences. Springer Briefs on Intelligent Systems. Springer.
12. DOI: 10.1007/978-3-031-09016-8, https://arxiv.org/pdf/2007.01795.pdf
13. Lackner, M., & Skowron, P. (2018). Multi-win rules based on approval and strategic voting. Proceedings of the Twenty-Seventh Joint International Conference on Artificial Intelligence (IJCAI 2018), 340–346.
14. DOI: 10.24963/ijcai.2018/47, https://www.ijcai.org/proceedings/2018/0047.pdf
15. Yang, Y. (2019). Complexity of manipulation and control of multiple winner systems based on Vote approval. Proceedings of the Twenty-Eighth Joint International Conference on Artificial Intelligence (IJCAI 2019), 637–643.
16. DOI: 10.24963/ijcai.2019/90, https://www.ijcai.org/proceedings/2019/0090.pdf
17. Caragiannis , I., Kaklamanis , C., Karanikolas, N., and Krimpas , GA (2020). Evaluation of approval-based multi-winner voting in terms of noise robustness. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020), 74–80.

18. DOI: 10.24963/ijcai.2020/11, https://www.ijcai.org/proceedings/2020/0011.pdf
19. Gupta, S., Jain, P., Saurabh, S., & Talmon, N. (2021). Further efforts to improve fixed-parameter bounds and traceability for multiple-winner systems. Rules. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI 2021), 217–223. DOI: 10.24963/ijcai.2021/31, https://www.ijcai.org/proceedings/2021/0031.pdf
20. Yang, Y., & Wang, J. (2018). Voting with multiple winners with restricted admissible sets: complexity and resistance to strategies. Proceedings of the Twenty-Seventh Joint International Conference on Artificial Intelligence (IJCAI 2018), 576–582.
21. DOI: 10.24963/ijcai.2018/80, https://www.ijcai.org/proceedings/2018/0080.pdf
22. Kimelfeld , B., Kolaitis , PG, and Stoyanovich , J. (2018). Computational social choice meets databases. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018), 317–323.
23. DOI: 10.24963/ijcai.2018/44, https://www.ijcai.org/proceedings/2018/0044.pdf
24. Elkind, E., Obraztsova, S. and Teh, N. (2024). Temporal fairness in multi-winner voting games. Proceedings of the AAAI Conference on Artificial Intelligence, 38(20).
25. DOI: 10.1609/aaai.v 38i20.30273, https://ojs.aaai.org/index.php/AAAI/article/view/30273
26. Aziz, H., Gaspers, S., Gudmundsson, J., Mackenzie, S., Mattei, N., & Walsh, T. (2014). Computational aspects of multi-winner approval voting .
27. https://arxiv.org/pdf/1407.3247.pdf
28. Caragiannis , I., Kaklamanis , C., Karanikolas, N., and Krimpas , GA (2020). Evaluation of approval-based multi-winner voting in terms of noise robustness.
29. DOI: 10.24963/ijcai.2020/11, https://arxiv.org/pdf/2002.01776.pdf
30. Betzler , N., Slinko , A., & Uhlmann , J. (2013). On the calculation of fully proportional representation . Journal of Artificial Intelligence Research , 47, 475–519. https://doi.org/10.1613/jair.3908
31. Brams , S. J., & Fishburn , P. C. (2007). Approval Voting (2nd ed.). Springer, New York. https://doi.org/10.1007/978-0-387-49895-9
32. Brandt, F., Conitzer , V., Endriss , U., Lang, J., & Procaccia , AD (Eds .). (2016). Handbook of computational social choice . Cambridge University Press . https://doi.org/10.1017/CBO9781107446984
33. Brill , M., & Peters, D. (2023). Robust and verifiable proportionality axioms for multiwinner voting . arXiv preprint arXiv:2302.01635. (Note: Also published in revised formats depending on the journal).
34. Chamberlin , J.R., & Courant, P.N. (1983). Representative deliberations and representative decisions : Proportional representation and the Borda rule . American Political Science Review , 77(3), 718–733. https://doi.org/10.2307/1957270
35. Cormen , TH, Leiserson , CE, Rivest , RL, & Stein, C. (2022). Introduction to algorithms (4th ed.). MIT Press .
36. Diss , E., & Merlin, V. (Eds .). (2021). Evaluating voting systems with probability models : Essays by and in honor of William Gehrlein and Dominique Lepelley . Springer Cham. https://doi.org/10.1007/978-3-030-48598-6
37. Elkind , E., Faliszewski , P., & Slinko , A. (2017). Properties of multiwinner voting rules . Social Choice and Welfare , 48(3), 599–632. https://doi.org/10.1007/s00355-016-1020-0

38. Elkind , E., Faliszewski , P., Skowron , P., & Slinko , A. (2014). What do multiwinner voting rules do? An experiment over the two-dimensional Euclidean domain . In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 28, No. 1, pp. 614–620).
39. Faliszewski , P., Aziz, H., Brill , M., Elkind , E., Lang, J., & Talmon , N. (2017). Multiwinner voting : A new challenge for social choice theory . In U. Endriss (Ed.), Trends in computational social choice (pp. 209–230). AI Access.
40. Knuth , D.E. (1997). The art of computer programming : Vol. 1. Fundamental Algorithms (3rd ed.). Addison-Wesley.
41. Knuth , D.E. (1997). The art of computer programming : Vol. 2. Seminumerical Algorithms (3rd ed.). Addison-Wesley.
42. Knuth , D.E. (1998). The art of computer programming : Vol. 3. Sorting and searching (2nd ed.). Addison Wesley.
43. Lackner , M., & Skowron , P. (2023). Multi-winner voting with approved preferences . Springer Cham. https://doi.org/10.1007/978-3-031-09882-6
44. Monroe, BL (1995). Fully proportional representation . American Political Science Review , 89(4), 925–940. https://doi.org/10.2307/2082518
45. Procaccia , A.D., Rosenschein , J.S., & Walsh, T. (2008). On the complexity of achieving proportional representation . In Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008) (pp. 1195–1202).
46. Procaccia , A.D., Rosenschein , J.S., & Walsh, T. (2007). Complexity of strategy behavior in multi-winner elections . In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007) (pp. 1473–1478).
47. Rothe, J. (2024). Economics and computation: An introduction to algorithms game theory , computational social choice , and fair division. Springer Cham. https://doi.org/10.1007/978-3-031-41911-9