Voting Systems: From Method to Algorithm

A Thesis Presented to

The Faculty of the Computer Science Department

California State University Channel Islands

In (Partial) Fulfillment

of the Requirements for the Degree

Masters of Science in Computer Science

by

Christopher R. Devlin

Advisor: Michael Soltys

December 2019

APPROVED FOR MS IN COMPUTER SCIENCE

December 23, 2019

Advisor: Dr. Michael Soltys — Date

Dec. 18, 2019

Name: Dr. Bahareh Abbasi — Date

Dec. 18, 2019

Name: Dr. Vida Vakilian — Date

APPROVED FOR THE UNIVERITY

1/6/20

Name — Date

Acknowledgements

I'd like to thank my wife, Eden Byrne for her patience and support while I completed this Masters Degree. I'd also like to thank Dr. Michael Soltys for his guidance and mentorship. Additionally I'd like to thank the faculty and my fellow students at CSUCI who have given nothing but assistance and encouragement during my time here.

Voting Systems: From Method to Algorithm

Christopher R. Devlin

December 18, 2019

Abstract

Voting and choice aggregation are used widely not just in politics but in business decision making processes and other areas such as competitive bidding procurement. Stakeholders and others who rely on these systems require them to be fast, efficient, and, most importantly, fair. The focus of this thesis is to illustrate the complexities inherent in voting systems. The algorithms intrinsic in several voting systems are made explicit as a way to simplify choices among these systems. The systematic evaluation of the algorithms associated with choice aggregation will provide a groundwork for future research and the implementation of these tools across public and private spheres.

Contents

List of Figures

1 Introduction

This thesis will examine common voting systems and restate them as algorithms for the purpose of clarity and transparency. The necessity of this stems from the ubiquitous presence of algorithms in general in modern life and in creating an understanding between the application of these algorithms and end users. The use of preference aggregation has applications beyond the well known political sphere and is increasingly used in artificial intelligence and multiagent systems. In situations where a group of people needs to decide between several alternatives there will generally be disagreements on which alternatives are the most and least desirable. This will necessitate the aggregation of individual preferences into a preference that is most acceptable the largest number of stakeholders. The importance of this rests on the idea that all voting stakeholders feel that their preferences were considered and dealt with fairly.

The use of voting in terms of politics covers not only who will wield the powers of the state but also how they will be allowed to allocate the vast sums of money and other resources. It seems then only natural to expect, and this is born out by history, that some groups will seek to manipulate or control the outcome of elections. As a formal rule, manipulation, also called strategic voting, exists when a voter has the ability to unilaterally change their ballot to secure a desired outcome [1]. This rule also assumes that the voter in question has perfect knowledge of how all others have or will vote and that this voter will mark their choices in an insincere manner to secure their own preferred result. Control, as an alternative, is the use of election procedures by those running the election to change the outcome without changing who individuals may choose to vote for. Often this control is used to decide who can vote or run for office in the first place as opposed to who those voters can vote for.

As a system the idea of popular rule has obvious advantages over dictatorial systems in which decisions are made by a king or some ruling class, or at least advantages for anyone who is not the king or a member of said ruling class. While it may be easy for an individual or small group to come to a decision, there needs to be some formal system to account for the varying opinions of large groups of equals. Additionally that system needs to be computationally efficient and easy to explain to the stakeholders using it for purposes of transparency. This would eliminate systems such as the one developed by Schulze [26] which uses a pairwise matrix to count and a directed graph to visualize the strongest candidate's path to victory.

The importance of securing the integrity of elections is obvious, in fact Defense Advanced Research Projects Agency (DARPA) was tasked with building a prototype voting system from the ground up, both hardware and software [29]. The System Security Integrated Through Hardware and Firmware (SSITH) program is seeking to secure against hardware vulnerabilities that are exploited through software of electronic systems. The final demonstration of which will be released in 2020 and will be a documented, open source architecture that will include verifiable paper balloting systems for the voting booth, ballot box which will provide both digital and physical representations of votes cast. There are additional security procedures that are also widely used such as election monitoring that are extremely effective at preventing or at least detecting interference. All of these procedures put together make it extremely difficult but not impossible to manipulate or control an election.

Our purpose here is to work towards a system that has defenses against interference built in. We will be using the term manipulability in a formal sense of voters using various strategies to control the outcome. We will also consider how the number of candidates affect the outcome and strategies used.

Of voting systems and politics we are already familiar with several commonly used systems. Plurality voting is the most common system, often referred to as "first past the post", and the declared winner is the candidate with the most votes regardless of whether that number represents a majority. The next most widely known system would be instant runoff, where the count is conducted iteratively with the candidate receiving the fewest votes—or sometimes the candidate with the fewest first place votes—being dropped after each iteration until a single winner is left remaining. We shall also be looking at a type of approval voting where a voter may cast a vote for all of the candidates that they approve of and the winner being the candidate with the most votes. We will look at this in more detail when discussing our modifications to the Borda count.

The Borda count is a well know consensus building system that also has several flaws which make it susceptible to strategic voting schemes. The greatest strike against Borda is its potential for compromising and burying candidates [2]. In a given hypothetical election suppose there are two candidates that are perceived to be equally likely to win—assuming a strategically minded voter would not have sincerely placed either of the two candidates first or last—this voter can maximize the individual power of their vote by simply ranking their more preferred of the two candidates first and the least preferred of the two dead last thus insincerely *compromising* and *burying* the candidates. Oddly enough in a two party system if both parties use this strategy, it opens up room for third party candidates to win.

Arrow's impossibility Theorem [3] and the Gibbard-Satterthwaite Theorem [4], [5] both deal with the realities of social choice theory when an election has more than one candidate. Both theorems examine elections with more than two candidates and require that the final winner not be chosen by a single person or entity—which is considered to be a non-dictatorship. While Gibbard-Satterthwaite says that any voting system is going to be vulnerable to manipulation, Arrow lays out several qualities of preference aggregation that will not be satisfied simultaneously. For our purposes, we are concerned with the independence of irrelevant alternatives—that a preference between a and b should only be determined by the individual preference between a and b—and Pareto efficiency [1], which states that a voter can not improve the position of one candidate without worsening the position of at least one other candidate.

That plurality voting would lead to undesired results and that an entire ordering of a voter's preferences would be required for making social decisions was Borda's starting point [3]. Unfortunately for us it is not possible to develop a system that is completely immune from manipulation and control as any system that allows for more than two candidates leaves itself open to either dictatorial control or strategic voting [4], [5]. At the same time we are also prevented from creating a ranked choice system that communicates individual preferences into an aggregated community wide ranking [3]. Because of these theorems which establish the impossibility of creating a system that is completely devoid of manipulability or control-ability, we are left attempting to design a system that would at least make interference not solvable in polynomial time or in other words an NP-complete problem.

Because the Borda count takes into account a voter's preferences over several if not all candidates it offers advantages over plurality voting such as resistance to the spoiler effect and the problem of vote splitting between similar candidates. That the 2000 election in the United States would have turned out differently had a ranked choice system been in place is, at this point, well worn territory [9]. But what is less well known is that were a system such as the Borda count put in place the 2000 election may well have been a race between the two more liberal candidates [2].

In practice, the Borda count is only used in a few political elections worldwide, in Slovenia it is used to elect representatives for groups of ethnic minorities, while the Parliament of Nauru uses a modified Borda count for electing multi seat constituencies. Outside politics many academic institutions use Borda, such as the student government at the University of Michigan. And in sports where baseball's Most Valuable Player award, and college football's Heisman Trophy are both chosen by Borda count. The very popular Eurovision Song Contest uses a slightly modified Borda count designed to favor a clear winner. Historically a version of the Borda count was used by the Roman Senate in the second century.

Beginning in 1971 the island nation of Nauru has been using a modified version of the Borda count called the Dowdall [25] method in elections seeking the top two, three, or four candidates. Named for its creator and at the time Nauru's Secretary for Justice—the system was thought to be easier to count than the Altervative voting—also called Instant-runoff voting—that had been inherited from Australia [16]. The system works thusly, voters list their candidate preferences and those lists are tallied so that candidates receive $\frac{1}{k}$ points for being the k^{th} ranked choice. In 2019, for example, candidate Timothy Ika received 585.869 points from a total of 924 votes [17]. Unlike the traditional Borda count where the scores given to a candidate varies based on the total number of candidates—the Dowdall method always begins at 1 and decreases from there [25].

When we discuss manipulation what we are saying is that a voter has misrepresented their true preference or preferences in order to gain a perceived benefit because of that misrepresentation. But one could also say that any submitted ranked choice ballot is representative of that voters wishes regardless of any other motives by the voter. If the voters preferred candidate receives a benefit from that misrepresentation that they would not have received had the voter voted sincerely than the manipulation is considered successful. There are four ways that a voter can potentially change their votes to gain an advantage for a preferred candidate. The first two, compromising and burying are considered the Borda counts greatest weaknesses—Condorcet himself wrote about them in his 1790 work, *Essay on the application of probability analysisis to majority decisions* which is discussed in Szpiro [9]. Compromising is the act of elevating a less favored candidate to get them elected—very common in first past the post elections and is evidence of Duverger's law—that "the simple majority single ballot system favours the two party system" [21]. An often cited example comes from the 2000 election of George Bush Jr. over Al Gore when the supporters of Ralph Nader were blamed for not compromising their votes in order to elect Nader voters presumed second choice Gore. Compromising in a Borda count system consists of elevating a second choice candidate over a first choice in order to beat a competitive third choice candidate. While burying is essentially the opposite—a voter insincerely places a competitive candidate at the bottom of their list in order to minimize that candidate's chances of winning the election. Often these strategies are combined—if there are two candidates who are considered front runners a strategic voter might simultaneously place the favored (but not favorite) candidate first and place the less favored (but not least favorite) candidate last—thus compromising a favorite for a less favored but still preferred candidate and burying an opposed but not least favored candidate. The third form of tactical voting, mischief voting, is common in primary or runoff elections and is used to elevate a candidate that is perceived to be easier to defeat to face off against favorite. This was on display in 2016 when the Hillary Clinton campaign hoped that Donald Trump would win the Republican primary as he was believed to be easier to defeat in the general election [22].

The last form of potential vote manipulation is bullet voting. Bullet voting simply means that a voter may cast votes for more than one candidate but only votes for one—this may not necessarily be due to strategy but simply because the voter only finds one candidate to be acceptable. When used as a form of tactical voting the bullet vote can be massed to increase the odds for a favored candidate—the city of Philadelphia uses the limited vote method to elect members to the city council and in a report published in 2015 found that of the top five Democratic party nominees to receive bullet votes in the primary three made it into the general election [23].

One of the problems of discussing tactical voting is that tactics used by voters to sway an election also align with our values. In the case of bullet voting the Oklahoma Supreme Court found in Dove v. Oglesby 1926 that forcing voters to rank a number of candidates in order for their votes to count is to "prevent the free exercise of the right of suffrage" [20]—as far as U.S. courts are concerned it seems that no one should be forced to rank a candidate that they find unacceptable. Similarly if a group of voters can build a coalition that votes tactically than our system of free association should allow it—what is politics if not coalition building.

There are some objective comparisons that can be conducted between voting systems. The cornerstone of social choice theory is Arrow's Impossibility Theorem and the idea that no ranked choice system can satisfy all its criteria. First of these criteria, that there may be no dictator that usurps the will of the greater populace for their own. That is, there can be no individual whose preference is always reflected in the outcome in spite of the preferences of the majority [3]. A voting system that uses secret ballots and has more than one voter is able to satisfy this condition. The second criteria, unrestricted domain, simply states that there will not be limitations placed on how voters may rank their preferences. Those individual preferences are then aggregated into a complete ranking of societal preferences. Additionally for the unrestricted domain criteria to be met that final societal ranking must be created deterministically—that is the results must be the same whenever the same input is given. If the unrestricted domain is replaced with a system which finds the preference with the highest median score than this criteria can be said to have been satisfied [24]. The third criteria Arrow proposed is known as independence of irrelevant alternatives (IIA) [3] and is considered to be the most difficult criteria to overcome whenever there are more than two candidates. Lastly Arrow cited Pareto efficiency or in this context that a candidate is considered the winner unless there is an alternative candidate that every voter prefers to it. Or if every voter v votes in the manner that expresses a preference of x over y then $x > y$ and x should be declared the winner.

Running parallel to Arrow's Impossibility Theorem is the Gibbard-Satterthwaite Theorem (G-S Theorem). Independently published by Gibbard in 1973 [4] and shortly later by Satterthwaite [5] in 1975, the theorem shows three rules for voting systems, one of which must apply. First, like Arrow's Impossibility Theorem, deals with dictatorships and plainly states that if a voting rule can have three or more possible outcomes and if that rule is also non-manipulable then that system is a dictatorship. Second, the theorem shows that manipulability is not an issue when the number of candidates is limited to two. In this case the choice becomes a simple majority outcome. Any voter when presented with only two candidates will always be best served by communicating their choice sincerely, and if the majority rules then that can be said to be a non-dictatorship. Finally, the theorem states that tactical voting will always come into play when a voter is given more than two options to choose from and that this cannot be mitigated by simply allowing a voter to rank all of the candidates. Ultimately the G-S Theorem shows that a voting system will always be manipulable unless the race is limited to two candidates or the winner is chosen by dictatorship. In short, a dictator erodes all other outcomes.

1.1 Contribution of this Thesis

The purpose of designing clear and concise algorithms of the common voting systems is to open and expose the internal workings of the selected choice aggregating systems. The importance of this is threefold. First, in order for a system to be considered fair, users must be able to understand how the output is derived from the input. Second, from a legal standpoint, it must be assured that the same input will always return the same result. Third, those results must be delivered in a consist and timely manner. We will elucidate these three points by examining the algorithms and their runtimes. In an example that may be thought of as trivial but had real world consequences, at the 1995 World Championship of figure skating the first, second, and third place standings were set when Michelle Kwan stepped on the ice—at the end of her heat she had received a score placing her in fourth place—the odd thing is that her fourth place finish caused the second and third place competitors to switch their rankings. This would come to be called the "Great Flip-Flop" and would cause a scandal in the figure skating world, and is also an example of independence of irrelevant alternatives. It was believed by fans of the sport that Michelle Kwan's performance should not have effected the final scores of the other two skaters and the International Skating Union (ISU) attempted to fix the problem with community outreach and education. The scoring method was placed in competition programs and technical liaisons were made available to answer questions, in the end the ISU would scrap the old scoring system in favor of a system that would guarantee that "If you are in front, you will stay in front" [2]. Arrow's Impossibility Theorem would prove them wrong. The works of Arrow, Gibbard, and Satterthwaite all assume that voters will rank their choices, although these rules still apply even if only the first choice is counted. Any voting system must work thusly—voters mark their ballots, those ballots are tallied one step at a time by a precise algorithm, and the winner is output. What is paramount is that those who use these systems are able to make informed choices about each systems strengths and weaknesses—and we will show those here. Additionally, choice aggregation must not be a black box between input and output, with the use of big data being an added confounding factor. In the twenty-first century all of the largest corporations use some kind of sorting algorithm to make, for example, human resource decisions—though in reality almost all company data must be sorted by computer before it is useful for human consumption. It was only a year ago that Amazon had to cease using a recruiting tool because it was biased against female applicants [27]. That these systems be transparent is the only way to keep them fair, but conversely, transparency makes them prone to manipulation.

In evaluating voting systems there are several perceived factors that can be weighed to determine the desirability of a given system. For the purposes of this thesis we will prefer systems that allow the ranking of one or all candidates, produces a preferred outcome, and is easy to implement and explain to voters. When we say preferred outcome we mean that the winners of a given election are those that accurately reflect the will of the voters. In this case if a candidate receives a majority of votes then it seems natural that they be declared the winner. Through restating these voting methods as algorithms we also open up the techniques used for proving algorithm correctness and ensuring that voting expectations are met. We can, applying these methods, establish preconditions, postconditions, and show that the vote, when satisfying the precondition, will also satisfy the postcondition.

2 Background

We should look briefly at what we mean by election, and formally define some key terms along with our voting systems themselves. First we consider some number of candidates that for our purposes will always be greater than two. How these candidates are compiled will not be discussed in this work. Second, each voter is presented with that list of candidates and asked to rank their preferred candidates from highest to lowest. We will consider this list to be transitive to avoid problems such as the Condorcet paradox where $A > B > C > A....$ Finally all voter preferences are collected and tabulated into a final result where for our purposes there will be only one winner. Let $V = v_1, ..., v_j$ be the finite set consisting of all voters and $C = c_1, ..., c_k$ be the finite set of all candidates. Given the cardinality of set C equals $|C|$ then a voters n^{th} choice is given $|C| - n$ points. As an example, given an election where $|V| = 10$ and $|C| = 4$, the individual ballots can be viewed as lists of voter preferences, so voters v_1 and v_2 have given an order of $c_1 > c_2 > c_3 > c_4$, voters v_3 through v_6 have given an order of $c_2 > c_3 > c_1 > c_4$, voters v_7 through v_9 have a preference order of $c_3 > c_2 > c_4 > c_1$, and finally voter v_{10} has a preference of $c_4 > c_3 > c_2 > c_1$. The figure 6 below gives candidate c_2 a victory with 23 points.

In setting up comparisons of voting rules we could compare the vectors which represent the weighted scores given to candidates. If we set the total number of candidates to $n = 3$ then Plurality voting would be represented by the vector $(1, 0, 0)$ the sum of which will always be 1 for any n. The Borda count by comparison would be represented by the vector $(2, 1, 0)$ the sum of which for any n is $n[\frac{(n-1)}{2}]$. Under the Dowdall method the vector looks like $(1, \frac{1}{2}, \frac{1}{3})$ and the sums of vectors under this system are harmonic numbers represented by the form $\sum_{k=1}^{n} \frac{1}{k}$.

Let us quickly formalize some of the shorthand of choice validation theory that we will be using. Given a finite set of candidates, A we can denote the number of candidates in A with $|A|$ and given two candidates $x, y \in A$, if xPy then we can say x is prefered to y or $x > y$, while if xIy then x is indifferent to y or $x = y$ [1].

IIA works as follows, consider the set $S = [x, y]$ and the majority preference over set S is xPy, if candidate z is added to S to become $[x, y, z]$ and the majority preference changes to $yPxPz$ then IIA can be said to have been violated. The Borda count is especially vulnerable to cloning, which is used to diminish rivals of a favorite candidate [25]. Like unrestricted domain IIA can be mitigated with the median voter theorem [24].

Below we will briefly explain the voting systems whose algorithms will be presented.

1. Approval Voting: Approval Voting ballot look identical to those submitted under a standard plurality system i.e. the ballots are a list of candidates—what differentiates the Approval from the Plurality system is that voters may place a mark next to as many candidates as they choose. As with plurality the candidate with the most marks wins.

2. Baldwin Method: The Baldwin Method is a modification of the Borda count that includes an iterative removal process. After the first round of Borda scores are tallied that candidate with the lowest score is removed and the Borda scores are recounted as though the removed candidate had never been in the race. This process is continued until two candidates remain with the winner chosen between them.

3. Borda Count: When using the Borda count each voter submits a ballot consisting of a complete ranking of candidates. Points are awarded to each candidate based on their placement on the ballot—if their are n candidates then the candidate ranked first receives $n - 1$ points, the candidate ranked second receives $n - 2$ points, and so on with the last ranked candidate receiving $n - n$ or 0 points. The candidate with the highest Borda score is the winner.

4. Bucklin Method: With the Bucklin method voters may rank as many candidates as they choose. If one candidate has a clear majority after counting the first choice on all ballots then they are declared the winner, if not the second choice votes are added to the first choice, this is continued until a majority winner is reached. It is possible for more than one candidate to have obtained a majority after the first round, in that case the candidate with the largest score is declared the winner.

5. Condorcet: The Condorcet method is run as a series of two candidate elections designed to find a candidate which defeats all others which

is termed the Condorcet winner. Voters submit ranked choice ballots which are tallied in a round robin fashion with that candidate ranked higher considered to be victor over those ranked below. Scores are kept in a pairwise comparison matrix and the winner of each individual contest is then declared victor in that contest. If there is no Condorcet winner other methods must be employed to break any ties or cycles that may exist. Other systems designed to find Condorcet winners when one exists are often called Condorcet methods, even when they use entirely different methods of counting.

6. Coombs Rule: The Coombs Rule takes ranked choice ballots and counts first choice preferences. If there is a candidate with a majority then they are declared the winner—if not then that candidate with the most last place or unranked marks is dropped from the candidate list and the ballots are recounted as if the dropped candidate had not been in the race. This is continued until a majority is reached.

7. Exhaustive Ballot: In the Exhaustive Ballot method the voters are asked to do the iterative process themselves. Voters choose a single candidate from the list and if a candidate has a majority then they are the winner, else, the lowest vote getter is dropped from the candidate list and voters return to the polls to vote again. This is continued until there is a majority winner.

8. Instant Runoff Voting: IRV takes ranked choice ballots and tallies the

first choice votes, if there is candidate with a majority they are declared the winner, else, the candidate with the fewest first choice votes is dropped from the candidate list and the first choices are counted again. This process continues until a candidate with a clear majority emerges.

9. Kemeny-Young: Kemeny-Young is based on the Condorcet method and seeks to establish a most agreed upon ranking of candidates, i.e. to maximize the number of voters who agree with that ranking. Like the Condorcet method it uses a pairwise matrix and takes the additional step of sorting pairwise comparisons from greatest to closest victories.

10. Majority Judgment: The Majority Judgment system uses a grading scale to rank choices on a ballot. This scale may be numbered such as $1-10$, a lettered grading scale e.g. $A, B, C, etc.$, or use a common vocabulary such as $Excellent, Good, Average, Bad$. Once voters have given each candidate a grade the scores are tallied and the candidate with the highest median grade is declared the winner. In the event of a tie one median grade is removed from each tied candidate, this is continued until there is only one candidate with the highest median grade.

11. Nanson's Method: Nanson's Method is another modification of the Borda Count—in this case after the Borda scores are counted all candidates whose scores are below the mean of all scores are removed. The ranked ballots are recounted without those candidates and this process

continues until a single winner remains.

12. Plurality: This is the common system in which voters are allowed to cast ballots for a single candidate, that candidate with the plurality of votes wins.

13. Range Voting: Also called Score voting, uses a ratings ballot grading scale of $1 - 10$ in which voters may give candidates any or no score. After ballots are collected and counted that candidate with the highest mean score is declared the winner.

14. Tideman Method: The Tideman method is conducted by first tallying ranked choice ballots in a way which gives a candidate one point for each pairwise victory against the other candidates. Those pairwise scores are then ranked from largest majority to smallest. The next step creates a directed graph in which a path from the victor to loser is created from each pairwise comparison from largest victory to smallest. This continues unless the drawing of a path creates a cycle, in which case that pairing is dropped. The winner is found to be the source of the directed graph that passes through all candidates without creating a cycle.

15. Schulze Method: The Schulze method uses a directed graph with the weight of the paths being the deciding factor. First ranked choice ballots are counted and pairwise comparison scores are tallied. The

paths are drawn between winners and losers and the source of the acyclic directed graph with strength is declared the winner i.e. that path with the strongest weak link is the winner.

16. Veto System: Voters may cast single vote for a candidate that they most disapprove of, that candidate with the lowest score is declared the winner.

There are also several voting criteria that are useful for comparing voting systems. The following criteria are used to show that if ballots are to be counted in a certain way than there will be candidates that must or must not win based on that count. Firstly the majority criterion simply states that if a given candidate is ranked as the first preference by a majority of voters then that candidate must win. There is an inverse to this called the majority loser criterion which states that if a majority of voters rank a candidate last then that candidate must not win. Voting systems that pass both the majority criterion and the majority loser criterion include Nanson and IRV [31]—the Borda count only clears the majority loser criterion [30]. The mutual majority criterion takes the majority criterion one step further by stating that if there is a subset S of candidates from set of all candidates C and that a majority of voters prefer every candidate in S to every candidate outside of S then the winner must come from the set of S. Among systems passing this criteria are Bucklin, Copeland, and IRV pass while Borda and majority judgement do not. The Condorcet criterion is met when the ultimate winner is able to win in pairwise contests against all other candidates in the field. Among the systems that meet this criteria are Instant Runoff Voting, Bucklin, Kemeny-Young, and Borda when either the Nanson or Baldwin counting methods are used. Conversely the Condorcet loser is the candidate that loses to all other candidates when paired—the systems that ensure this losing candidate does not ultimately win include the Borda count, Copeland's method, IRV, and Kemeny-Young. As we discussed with Arrow's impossibility theorem the Independence of Irrelevant Alternatives shows whether the addition of a candidate will or will not affect the outcome and that an individuals preference between x and y will only factor in x and y. Voting systems that satisfy this condition include majority judgement and score voting. Under the independence of clones criteria we are able to show if an election method is susceptible to the outcome changing with the addition of candidates that are similar to a candidate already in the race. The effect may be positive or negative to a candidate depending on the system used. For example in a plurality system similar candidates often cause vote splitting—often preventing either candidate from winning. Conversely under the Borda count cloning can elevate a preferred candidate by putting more distance between less preferred candidates and the top. Instant runoff voting and majority judgement methods are both resistant to the addition of clone candidates.

3 Algorithms for Voting Systems

In this section a number of voting systems will be explained in detail. For all of the following examples the set of candidates C will consist of c_1, c_2, c_3, c_4 and the set of voters V will consist of $v_1, v_2, ..., v_{10}$ although these ten voters will not be examined individually.

3.1 Approval voting

Approval voting was created in 1971 by Weber and published as part of his Ph.D. thesis. Under the approval voting each voter is given a ballot listing all candidates for a position—the voter may then place a mark next to as many of those names as they choose, signaling their "approval" of the candidate for that position. After the ballots are tallied the winner is simply the candidate with the most approval votes [12]. When approval voting is used a voter may cast a single vote for all candidates that they would find acceptable for a given position. You could imagine a ballot with a listing of candidates and voters could simply place a checkmark next to the name of whomever they believe could do a satisfactory job. Ballots of this type could be looked at like as two lists, one list consisting of supported candidates and one consisting of not supported candidates. This is also useful if we do not need a fixed number of winners—the baseball Hall of Fame uses such a system with the members of the Baseball Writers' Association of America electing for admission former players who receive above a predetermined number of votes.

Figure 1: Approval Voting Algorithm

```
Input  : List of candidates C and ballots from all voters in V
Output: A single winner from C with a plurality of approval votes
while there are ballots to be counted do
    count ballots;
    for i from 1 to C do
        if c_i has ✓ then
            c_i score ←+1
        end
    end
end
Sort candidates by scores.
return c with highest score.
```

Algorithm 1: Approval Voting

Figure 2: Approval Voting Example

Approval voting example

In this example we can see that the approval vote tally has three candidates that have crossed majority approval, in this case c_2 is returned as the winner with nine votes.

Candidates \ Number of Voters	4	3	2	1
c_1	✓		✓	
c_2	✓	✓	✓	
c_3	✓	✓		✓
c_4				✓

Candidate	Vote Total Value
c_1	6
c_2	9
c_3	8
c_4	1

Because Approval voting does not convey any additional information to rank preferences and assumes that candidates are ranked equally, it becomes incumbent upon the voter to know when to cease approving. Accordingly it has been shown that an unstrategic voter who simply approves of any candidate considered to be "good" or better actually provides more resistance to compromising strategies than voters who strategically only vote for "very good" candidates [13]. Approval voting is in essence a plurality voting system and carries many of that system's disadvantages—among them a production of wasted votes, a lack of minority representation, and two party domination of elections. [14]

3.2 Baldwin Method

The Baldwin method is a hybrid of the Borda count and instant-runoff procedures—and closely related to Nanson's method which we will look at shortly. Formalized by Joseph M. Baldwin in 1926, voters provide a strict order preference that may or may not include all candidates—the ballots are iteratively counted and recounted with the lowest Borda scoring candidate removed after each round. One of the benefits of both Baldwin's and Nanson's methods is that a Condorcet winner will be chosen when one exists [15]. One advantage both Baldwin's and Nanson's methods both share is that elimination style rules are computationally more difficult to manipulate the the Borda count from which they originate [28].

Figure 3: Baldwin Method Algorithm

```
Input  : List of candidates C and ballots from all voters in V
Output: A single winner from C with > |V|/2 + 1 support
while there are ballots to be counted do
    create two arrays of size |C|, c[|C|] for Borda points, and
     cFirst[|C|] for 1^st choice picks.
    count ballots;
    for i from 1 to |C| do
        if c_i is 1^st choice then
            [c_i First] ← [c_i First] + 1
        else
            [c_i] score ←|C| − k + 1 for k^th choice
        end
    end
end
if ∃c ∈ C with > |V|/2 + 1 First choice support then
    return Winning c
else
    Remove lowest scoring candidate.
    lowestScore ←bordaPoints[1]
    lastCandidate ← 1
    for i from 1 to |C| do
        if bordaPoints[i] < lowestScore then
            lowestScore ← bordaPoints[i]
            lastCandidate ← i
        end
        Remove from ballots c_lastCandidate
    end
    return to top
```

Algorithm 2: Baldwin Method

Figure 4: Baldwin Method Example

Baldwin Example *Round* 1

The first round consists of tallying Borda points as seen here.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice* 3 points	c_2	c_3	c_1	c_4
2^{nd}*choice* 2 points	c_3	c_2	c_2	c_3
3^{rd}*choice* 1 point	c_1	c_4	c_3	c_2
4^{th}*choice* 0 points	c_4	c_1	c_4	c_1

Candidate	Vote Total Value
c_1	10
c_2	23
c_3	21
c_4	6

Baldwin Example *Round* 2

In the second round c_4 has been removed and Borda points are recounted.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice* 2 points	c_2	c_3	c_1	c_3
2^{nd}*choice* 1 points	c_3	c_2	c_2	c_2
3^{rd}*choice* 0 point	c_1	c_1	c_3	c_1

Candidate	Vote Total Value
c_1	4
c_2	14
c_3	12

Baldwin Example *Round* 3

Finally, in the third round only two candidates are left and c_2 is returned as the winner.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice* 1 points	c_2	c_3	c_2	c_3
2^{nd}*choice* 0 points	c_3	c_2	c_3	c_2

Candidate	Vote Total Value
c_2	6
c_3	4

3.3 Borda Count

Jean-Charles de Borda devised his voting system in June of 1770 as a way of fairly electing members to the French Academy of Sciences. Developed long before Kenneth Arrow would prove that no ranking method could be designed perfectly, the Academy would use Borda's method from 1784 until 1800 when a new Academy member named Napoléon Bonaparte would demand cessation of its use [2]. In it's simplest terms involving an election with n candidates, a voters first preference receives $n - 1$ votes and the k^{th} preference receives $n - k$ votes until all ranks are chosen with the final choice receiving 0 points. The benefit of this is that it considers and weights a voters entire range of preferences as opposed to only considering the first choice as in the plurality vote. The drawback of this is that a voter who votes insincerely may affect the outcome. An extremely effective example is the act of compromising and burying preferences wherein a voter switches their first and second choices in order to harm a third option, and by moving a likely crowd favorite to the bottom of their ballot regardless of their feeling towards the competitors. We will examine this in more detail later. Among uses outside of politics the Borda count has been used to successfully conduct metasearches on aggregated results of multiple search engines [6]. It has also been used in image recognition [7], and in resource price negotiations in e-markets [8]. The applications of the Borda count to choice aggregation will continue to be used in artificial intelligence and multi agent system applications.

Figure 5: Borda Count Algorithm

```
Input  : List of candidates C and ballots from all voters in V
Output: A single winner from C with the greatest number of Borda
        points
while there are ballots to be counted do
    create an array called bordaPoints[|C|] to contain Borda points
    count ballots;
    for i from 1 to |C| do
        if c_i is ranked then
            bordaPoints[i] ←bordaPoints[i] + |C| − k + 1 points for
              k^th choice
        end
    end
end
return candidate with highest score in bordaPoints
```

Algorithm 3: Borda Count

Figure 6: Borda Count Example

Borda Example $points = |C| - k$

In our example Borda Count after our tally we see that c_2 is returned as the winner.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice* 3 points	c_2	c_3	c_1	c_4
2^{nd}*choice* 2 points	c_3	c_2	c_2	c_3
3^{rd}*choice* 1 point	c_1	c_4	c_3	c_2
4^{th}*choice* 0 points	c_4	c_1	c_4	c_1

Candidate	Vote Total Value
c_1	10
c_2	23
c_3	21
c_4	6

3.4 Bucklin method

Bucklin voting rules entail voters submitting ranked choice ballots which are iteratively tallied by counting first choice votes—if there is no majority the second choice votes are added to the first—this is continued with third, forth, and so on until a candidate gains majority support—which is also the highest median Borda score. After the first round it is likely to have more votes than voters and thus it is very possible for more than one candidate to have a majority in which case the candidate with the highest total wins.

Figure 7: Bucklin Method Algorithm

```
Input  : List of candidates C and ballots from all voters in V
Output: A single winner from C with > |V|/2 + 1 support
k ← 1
while there are ballots to be counted do
    count ballots;
    for i from 1 to C do
        if c_i is k^th choice then
            c_i score ← +1
        end
    end
    if ∃ c ∈ C with > |V|/2 + 1 support then
        return Candidate with highest score.
    else
        k ← k + 1
    end
end
```

Algorithm 4: Bucklin Method

Figure 8: Bucklin Method Example

Here we see that first round scores are identical to what would be produced using a plurality system. In the second round when second choice picks are added to first c_2 is returned as the winner.

Bucklin Example

Preference Number \ Number of Voters	4	3	2	1
$1^{st}choice$	c_2	c_3	c_1	c_4
$2^{nd}choice$	c_3	c_2	c_2	c_3
$3^{rd}choice$	c_1	c_4	c_3	c_2
$4^{th}choice$	c_4	c_1	c_4	c_1

Candidate	Round 1 Total	Round 2 Total
c_1	2	1
c_2	4	9
c_3	3	8
c_4	1	1

3.5 Condorcet

The Condorcet method, and sometimes just referred to a the pairwise comparison method, imagines a race in which each candidate faces every other candidate in the race individually. The goal of these contests is to find that candidate that beats all others in head-to-head contests—a candidate that can do this is called the Condorcet winner—but may not always exist in every election. Like other methods, voters submit ballots consisting of their ranked candidate choices. Ballots are often counted in a matrix who's intersecting rows and columns show the result of that pairwise comparison—in the case below we can see that candidate c_2 has won each contest with their opponent.

Figure 9: Condorcet Algorithm

```
Input   : List of candidates C and ballots from all voters in V
Output: A single winner from C and a matrix of pairwise
        comparisons
Create a matrix of M size |C| × |C|
while there are ballots to be counted do
    count ballots;
    for i from 1 to |C| − 1 do
        for j from 2 to |C| do
            if c_i > c_j then
                M[c_i, c_j] ←M[c_i, c_j] + 1
                j ← j + 1
            else
                M[c_j, c_i] ← M[c_j, c_i] + 1
                j ← j + 1
            end
            i ← i + 1
        end
    end
end
return Candidate with greatest number of wins.
```

Algorithm 5: Condorcet Method

Figure 10: Condorcet Example

Condorcet Example

After running pairwise comparisons and creating a matrix of results we can see from that matrix that c_2 is our Condorcet winner defeating all other candidates.

Preference Number \ Number of Voters	4	3	2	1
$1^{st} choice$	c_2	c_3	c_1	c_4
$2^{nd} choice$	c_3	c_2	c_2	c_3
$3^{rd} choice$	c_1	c_4	c_3	c_2
$4^{th} choice$	c_4	c_1	c_4	c_1

Candidate \ Opponent	c_1	c_2	c_3	c_4
c_1	x	2	2	6
c_2	8	x	6	9
c_3	8	4	x	9
c_4	4	1	1	x

3.6 Coombs Rule

In the Coombs rule voters cast ranked choice votes—if there is a candidate that has received majority support then that candidate is declared the winner—else, the candidate receiving a plurality of last place votes is removed and ballots are retallied as if the removed candidates had not been on the ballot. This method can be used for choosing multiple winners if you stop iterating at the desired number of winners. Coombs is a simple system that has the added advantages of picking Condorcet winners while avoiding losers and a resistance to strategic voting.

Figure 11: Coombs Rule

```
Input  : List of candidates C and ballots from all voters in V
Output: A single winner from C with ≥ |V|/2 + 1 support
while there are ballots to be counted do
    count ballots;
    create array firstChoice;
    create array lastChoice;
    for i from 1 to |C|
     do
        if c_i is ranked first. then
            firstChoice[i] ← firstChoice[i] + 1
        end
        else if c_i is ranked last. then
            lastChoice[i] ← lastChoice[i] + 1
        end
        i ← i + 1
    end
end
if ∃c ∈ C with ≥ |V|/2 + 1 first choice votes then
    return Candidate with highest score.
else
    Remove from C and ballots the candidate with the greatest
     lastChoice score.
    return to top for next round.
end
```

Algorithm 6: Coombs Rule

Figure 12: Coombs Example

Coombs Example *Round* 1

In the first round of the Coombs example we see that only two candidates have been placed last with c_4 have the greatest number and thus being removed from the running.

Preference Number \ Number of Voters	4	3	2	1
$1^{st} choice$	c_2	c_3	c_1	c_4
$2^{nd} choice$	c_3	c_2	c_2	c_3
$3^{rd} choice$	c_1	c_4	c_3	c_2
$4^{th} choice$	c_4	c_1	c_4	c_1

Candidate	Last Place Votes
c_1	4
c_2	0
c_3	0
c_4	6

Coombs Example *Round* 2

In round two, again two candidates have been ranked last, this time c_1 is removed.

Preference Number \ Number of Voters	4	3	2	1
$1^{st} choice$	c_2	c_3	c_1	c_3
$2^{nd} choice$	c_3	c_2	c_2	c_2
$3^{rd} choice$	c_1	c_1	c_3	c_1

Candidate	Last Place Votes
c_1	8
c_2	0
c_3	2

Coombs Example *Round* 3

Here c_2 gains a majority of first preference votes and is declared the winner.

Preference Number \ Number of Voters	4	3	2	1
$1^{st} choice$	c_2	c_3	c_2	c_2
$2^{nd} choice$	c_3	c_2	c_3	c_3

Candidate	Last Place Votes
c_2	4
c_3	6

3.7 Exhaustive Ballot

Exhaustive Ballot is similar to Coombs with the exception that voters only cast one vote at a time for their most preferred candidate and after the removal of the least preferred candidate the vote is repeated until one candidate has a majority. The exhaustive ballot is used in a number of real world applications including the choosing of the Speaker of the British House of Commons, and by the International Olympic Committee to choose Olympic host cities—because the process involves voters casting ballots several times it is not practical for large scale elections.

Figure 13: Exhaustive Ballot Algorithm

```
Input  : List of candidates C and ballots from all voters in V
Output: A single winner from C with ≥ |V|/2 + 1 support
while there are ballots to be counted do
    count ballots;
    for i from 1 to |C|
     do
        if c_i is chosen. then
            c_i ← c_i + 1
        end
    end
end
if ∃ a c in C with ≥ |V|/2 + 1 support then
    return Candidate with highest score.
else
    Remove from C and ballots the candidate with the lowest score.
     Voters vote again with new smaller candidate field and ballots
     are counted.
end
```

Algorithm 7: Exhaustive Ballot

Figure 14: Exhaustive Ballot Example

Exhaustive Ballot Example *Round* 1

Because each round is run independently the ballot changes after each rount. In round one c_4 is removed with the fewest votes.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice*	c_2	c_3	c_1	c_4

Candidate	Results
c_1	2
c_2	4
c_3	3
c_4	1✓

Exhaustive Ballot Example *Round* 2

In round two the ballot only has three candidates and now c_2 is removed.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice*	c_2	c_3	c_1	c_3

Candidate	Results
c_1	2✓
c_2	4
c_3	4

Exhaustive Ballot Example *Round* 3

Finally, in the third round their are only two candidates remaining and c_2 is returned as the winner.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice*	c_2	c_3	c_2	c_3

Candidate	Results
c_2	6✓
c_3	4

3.8 Instant Runoff Voting

IRV or Instant Runoff Voting is a majoritarian system that is used around the world including for most elections in Australia. IRV was developed in the 1870's by Massachusetts Institute of Technology professor William Robert Ware and is based on the Single Transferable Vote which itself was used to choose multiple winner proportional representation. The system works as follows—voters cast ranked choice ballots giving a preference number to one or all of the candidates. On the first count if no candidate holds a majority that candidate with the fewest number of first preference votes is eliminated and the ballots are recounted. While IRV is quite resistant to tactical voting it fails to always find the Condorcet winner.

Figure 15: Instant Runoff Voting

```
Input  : List of candidates C and ranked choice ballots from all
         voters in V
Output: A single winner from C with ≥ |V|/2 + 1 support
while there are ballots to be counted do
    count ballots;
    for i from 1 to |C|
     do
        if c_i is voter first choice then
            c_i ← c_i + 1
        end
    end
end
if ∃ a c in C with ≥ |V|/2 + 1 support then
    return Candidate with highest score.
else
    Remove from C and ballots the candidate with the fewest first
     choice votes.
    return to top for next round of counting
end
```

Algorithm 8: Instant Runoff Voting

Figure 16: Instant Runoff Voting Example

IRV Example *Round* 1

In this example of IRV candidate c_4 is removed after receiving the fewest number of first choice votes.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice*	c_2	c_3	c_1	c_4
2^{nd}*choice*	c_3	c_2	c_2	c_3
3^{rd}*choice*	c_1	c_4	c_3	c_2
4^{th}*choice*	c_4	c_1	c_4	c_1

Candidate	First Choice Votes
c_1	2
c_2	4
c_3	3
c_4	1✓

IRV Example *Round* 2

In round two candidate c_1 is removed.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice*	c_2	c_3	c_1	c_3
2^{nd}*choice*	c_3	c_2	c_2	c_2
3^{rd}*choice*	c_1	c_1	c_3	c_1

Candidate	First Choice Votes
c_1	2✓
c_2	4
c_3	4

IRV Example *Round* 3

Finaly, in round three there are two candidates left and c_2 is returned as the winner.

Preference Number \ Number of Voters	4	3	2	1
1^{st}*choice*	c_2	c_3	c_2	c_3
2^{nd}*choice*	c_3	c_2	c_3	c_2

Candidate	First Choice Votes
c_2	6
c_3	4

3.9 Kemeny-Young

The Kemeny-Young system is another pairwise comparison method of counting, in this case voters are also permitted to rank candidates at the same preference level or leave candidates unranked altogether. The tallying of votes is conducted in two steps—first a table counting the pairwise preferences is created—then a score is given based on the the percentages of each winning pairwise comparison. The rankings which have those winning pairwise comparisons have that percentage added to their ranking score with the candidate at the top of the ranking with the highest score being the winner. One problem that the Kemeny-Young method has is that in a worst case scenario the calculations to find a winner can be NP-hard to calculate and can potentially take an impractically long time to find the victor [18].

Figure 17: Kemeny-Young Algorithm

```
Input  : List of candidates C and ballots from all voters in V
Output: A single winner from C
Create a matrix of M size |C| × |C|
while there are ballots to be counted do
    count ballots;
    for i from 1 to |C| − 1 do
        for j from 2 to |C| do
            if c_i > c_j then
                M[c_i, c_j] ←M[c_i, c_j] + 1
                j ← j + 1
            else
                M[c_j, c_i] ← M[c_j, c_i] + 1
                j ← j + 1
            end
            i ← i + 1
        end
    end
end
for i from 1 to |C| do
    for j from 1 to |C| do
        [c_i, c_j] ← [c_i,c_j] / |V|
    end
end
create an array S to tally scores;
for i from 1 to |C| do
    for j from 1 to |C| do
        if M[c_i, c_j] ≥ .5 then
            S[i] ← S[i] + M[c_i, c_j]
        end
    end
end
return Candidate with the highest score in S
```

Algorithm 9: Kemeny-Young

Figure 18: Kemeny-Young Example

Kemeny-Young Example

Here in the example we see how the tabulated ballots are turned into pairwise comparisons and sorted by magnitude of victory. After which those magnitude scores are tabulated to find the strongest rankings and return the winner c_2.

Preference Number \ Number of Voters	4	3	2	1
$1^{st} choice$	c_2	c_3	c_1	c_4
$2^{nd} choice$	c_3	c_2	c_2	c_3
$3^{rd} choice$	c_1	c_4	c_3	c_2
$4^{th} choice$	c_4	c_1	c_4	c_1

Kemeny-Young as pairwise comparison

Pair \ Preference	X over Y	X = Y	Y over X
X $= c_1$ Y $= c_2$	20%	0%	80%
X $= c_1$ Y $= c_3$	20%	0%	80%
X $= c_1$ Y $= c_4$	60%	0%	40%
X $= c_2$ Y $= c_3$	60%	0%	40%
X $= c_2$ Y $= c_4$	90%	0%	10%
X $= c_3$ Y $= c_4$	90%	0%	10%

The score for the ranking $c_2 > c_3 > c_1 > c_4$ is calculated by adding together the comparison preference percentages for each pairwise contest—which in this case totals to 460 and giving this ranking the highest score.

Figure 19: Kemeny-Young Ranking Scores

$$c_2 > c_4 = 90\%$$
$$c_3 > c_4 = 90\%$$
$$c_2 > c_1 = 80\%$$
$$c_3 > c_1 = 80\%$$
$$c_2 > c_3 = 60\%$$
$$c_1 > c_4 = 60\%$$

Kemeny-Young Ranking Scores

This table shows which ranking has the highest score, in this case c_2, c_3, c_1, c_4.

First Choice	Second Choice	Third Choice	Forth Choice	Ranking Score
c_1	c_2	c_3	c_4	340
c_2	c_3	c_1	c_4	460
c_3	c_2	c_4	c_1	420
c_4	c_3	c_2	c_1	260

3.10 Majority Judgement

The Majority Judgment is a candidate grading system that has many unique qualities that were built in by design. For one—a letter or number grade scale, or a list of descriptive words or phrases is used to rank candidates. This could consist of a letter from a high of "A" to a low of "E", a number from 10 to 0—or a wider scale of 100 to 0 if necessary—or a word scale ranging from "excellent" to "bad". The purpose is to give a common shared scale of evaluation beyond the relative evaluation of rankings. This method of grading candidates also gives voters a way to convey the merits of a candidate individually and relative to competitors. The winner is found by calculating the candidate with the highest median grade—in the likely event of a tie—the tied candidates will have a single median score removed reiteratively until there is a single candidate with the highest median grade [19]. For finding the median in a given set $V = [v_1, v_2, ..., v_n]$, if v_n is an odd number then the median grade is $v_{(n+1)/2}$ while for an even v_n the median is $v_{n/2}$.

Figure 20: Majority Judgment Algorithm

```
Input   : List of candidates C and ballots from all voters in V
Output: A single candidate from C with the highest median score
while there are ballots to be counted do
  | for i from 1 to |C| do
  |   | if c_i has been scored then
  |   |   | [c_i] ←[c_i]+ score
  |   | end
  | end
end
Sort candidates by score.
return Candidate with highest score.
```

Algorithm 10: Majority Judgment

Figure 21: Majority Judgment Example

Majority Judgment

In the first round there are three candidates with the highest median vote and candidate c_4 is removed.

Grades \ Candidate	c_1	c_2	c_3**end**	c_4
Excellent	2	4	3	1
VeryGood	1	1	1	1
Good	$2 \leftarrow$	$3 \leftarrow$	$2 \leftarrow$	1
Acceptable	1	1	3	$2 \leftarrow$
Poor	4	1	2	5

Majority Judgment after removing 2 median votes

Median votes are removed one at a time until only one candidate has the highest median vote, in this case after two removals c_2 is returned as the winner.

Grades \ Candidate	c_1	c_2	c_3
Excellent	2	4	3
VeryGood	1	1	1
Good	0	$1 \leftarrow$	0
Acceptable	$1 \leftarrow$	1	$3 \leftarrow$
Poor	4	1	2

3.11 Nanson's Method

Similar to and sometimes confused with Baldwin's method, the Nanson method also combines the Borda count with iterative elimination counting.

In this case those candidates who have scores below the mean of all scores are removed and the ballots are recounted as though the removed candidates had never been placed on the ballot. This counting and removing is repeated until a single winner is chosen.

Figure 22: Nanson's Method Algorithm

```
Input  : List of candidates C and ballots from all voters in V
Output: A single winner from C with > |V|/2 + 1 support
while there are ballots to be counted do
    create two arrays of size |C|, [c] for Borda points, and [cFirst] for
     1^st choice picks.
    count ballots;
    for i from 1 to |C| do
        if c_i is 1^st choice then
            [c_i First] ← [c_i First] + 1
        else
            [c_i] score ← |C| − k + 1 for k^th choice
        end
    end
end
if ∃c ∈ C with > |V|/2 + 1 First choice support then
    return Winning c
else
    Find mean score.
    meanScore ← 0
    for i from 1 to |C| do
        meanScore ← meanScore + [c_i]
    end
    meanScore ← meanScore / |C|
end
for i from 1 to |C| do
    if c_i < meanScore then
        remove c_i from C and ballots
    end
end
return to top
```

Algorithm 11: Nanson's Method

Figure 23: Nanson's Method Example

Nanson Example *Round* 1

In round one the Borda scores are tallied and the candidate with the lowes score is removed, in this case c_4.

Preference Number \ Number of Voters	4	3	2	1
1^{st} *choice* 3 points	c_2	c_3	c_1	c_4
2^{nd} *choice* 2 points	c_3	c_2	c_2	c_3
3^{rd} *choice* 1 point	c_1	c_4	c_3	c_2
4^{th} *choice* 0 points	c_4	c_1	c_4	c_1

Candidate	Vote Total Value
c_1	10
c_2	23
c_3	21
c_4	6

Nanson Example *Round* 2

Again points are tallied and the lowest scoring candidate removed, this time candidate c_1.

Preference Number \ Number of Voters	4	3	2	1
1^{st} *choice* 2 points	c_2	c_3	c_1	c_3
2^{nd} *choice* 1 points	c_3	c_2	c_2	c_2
3^{rd} *choice* 0 point	c_1	c_1	c_3	c_1

Candidate	Vote Total Value
c_1	4
c_2	14
c_3	12

Nanson Example *Round* 3

Finally in the third round candidate c_2 is returned as the winner.

Preference Number \ Number of Voters	4	3	2	1
1^{st} *choice* 1 points	c_2	c_3	c_2	c_3
2^{nd} *choice* 0 points	c_3	c_2	c_3	c_2

Candidate	Vote Total Value
c_2	6
c_3	4

3.12 Plurality

Plurality, or first-past-the-post, is the system used in most of the United States and is one of the simplest voting systems in use. Under plurality each voter casts a single ballot for a single candidate—those ballots are counted and the winner is that candidate with the most votes regardless of whether that number constitutes a majority. That it would be the best choice for elections consisting of two candidates is obvious as it would be a simple majority rule. When that candidate count is increased to three or more, however, the plurality system suffers from a number of disadvantages especially encouraging voters not to vote sincerely because the spoiler effect is so prevalent.

Figure 24: Plurality Algorithm

```
Input  : List of candidates C and ballots from all voters in V
Output: List of candidates sorted by score.
while there are ballots to be counted do
    for i from 1 to |C| do
        if c_i is preference choice then
            [c_i] ←[c_i] + 1
        end
    end
end
Sort candidates by scores.
```

Algorithm 12: Plurality

Figure 25: Plurality Example

Plurality Example

In this example we can see that candidate c_2 is returned as the winner as they are “first past the post”.

Number of Voters	4	3	2	1
Candidate	c_2	c_3	c_1	c_4

3.13 Range Voting

Range Voting Average is similar to the majority judgement method but uses the mean score instead of the median to eliminate candidates. In this system voters rank candidates on a scale of one to ten for example—with voters permitted to rank multiple candidates at the same level. After tallying all the votes the scores are averaged and the highest score wins. This system is prone to some of the same issues as the Borda count such as compromising and burying.

Figure 26: Range Voting Algorithm

```
Input  : List of candidates C and ballots from all voters in V
Output: List of ranked candidates sorted by score
while there are ballots to be counted do
    for i from 1 to |C| do
        if c_i has been scored then
            [c_i] ←[c_i]+ score
        end
    end
end
return Candidate with the highest score
```

Algorithm 13: Range Voting

Figure 27: Range Voting Example

Range Voting

In this first example the mean score is found and candidates with scores lower than that mean are removed, in this case c_4.

Grades \ Candidate	c_1	c_2	c_3	c_4
Excellent	2	4	3	1
VeryGood	1	1	1	1
Good	$2 \leftarrow$	$3 \leftarrow$	$2 \leftarrow$	1
Acceptable	1	1	3	$2 \leftarrow$
Poor	4	1	2	5

Range vote after removing the candidate with a score below the mean

After scores are recalculated, c_2 is returned as the winner with the highest mean.

Grades \ Candidate	c_1	c_2	c_3
Excellent	2	4	3
VeryGood	1	1	1
Good	0	$1 \leftarrow$	0
Acceptable	$1 \leftarrow$	1	$3 \leftarrow$
Poor	4	1	2

3.14 Tideman method

The Tideman method is also called Ranked Pairs and was developed by Tideman in 1987 and is another system satisfying the Condorcet criterion.

Ranked Pairs like the Schulze method uses a directed graph to determine the winner after tallying and sorting candidates based on the magnitude of their victory. Because of the difficulty in explaining this method Tideman would also create the Tideman alternative method which disposes of the graph to determine the winner.

Figure 28: Tideman Method Algorithm

Input : List of candidates C and ballots from all voters in V
Output: List of candidates from largest victor to smallest
Create a matrix of size $|C| \times |C|$
while *there are ballots to be counted* **do**
    count ballots;
    for i *from* 1 *to* $|C|$ **do**
        for j *from* 1 *to* $|C|$ **do**
            if $c_i > c_j$ **then**
                $[c_i, c_j] \leftarrow [c_i, c_j] + 1$
            else
                $[c_j, c_i] \leftarrow [c_j, c_i] + 1$
            end
        end
    end
end
Calculate win percentages which become the edges of the graph.
for i *from* 1 *to* $|C|$ **do**
    for j *from* 1 *to* $|C|$ **do**
        if $i \neq j$ **then**
            $[c_i, c_j] \leftarrow \frac{[c_i, c_j]}{|V|} \times 100$
        end
    end
end
Sort the edges $e_1 \geq e_2 \geq ... \geq e_m$. # Kruskal's Algorithm
$T \leftarrow \emptyset$
for i *from* 1 *to* m **do**
    if $T \cup \{e_i\}$ *has no cycle* **then**
        $T \leftarrow T \cup \{e_i\}$
    end
end

Algorithm 14: Tideman method

Figure 29: Tideman Method Example

Tideman Example

In the example pairwise comparisons are made and the resulting win percentages are caluclated.

Preference Number \ Number of Voters	4	3	2	1
$1^{st} choice$	c_2	c_3	c_1	c_4
$2^{nd} choice$	c_3	c_2	c_2	c_3
$3^{rd} choice$	c_1	c_4	c_3	c_2
$4^{th} choice$	c_4	c_1	c_4	c_1

Tideman pairwise comparisons

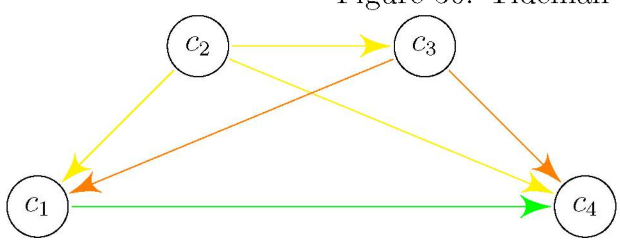
Pair \ Preference	X over Y	X = Y	Y over X
X $=c_1$, Y $=c_2$	20%	0%	80%
X $=c_1$, Y $= c_3$	20%	0%	80%
X $=c_1$, Y $= c_4$	60%	0%	40%
X $=c_2$, Y $= c_3$	60%	0%	40%
X $=c_2$, Y $= c_4$	90%	0%	10%
X $= c_3$, Y $= c_4$	90%	0%	10%

From the pairwise comparisons above the winners are sorted by the largest majority to the smallest.

c_2 defeats c_4 by 80%
c_3 defeats c_4 by 80%
c_2 defeats c_1 by 60%
c_3 defeats c_1 by 60%
c_1 defeats c_4 by 20%
c_2 defeats c_3 by 20%

From the list of winners we can draw our directed graph—if a cycle exists it is omitted.

Figure 30: Tideman Graph



And from this graph we can determine that c_2 is the Condorcet winner defeating all three challengers.

3.15 Schulze Method

The Schulze method is a relative newcomer, created in 1997 by Markus Schulze. It can be used to produce a single winner or multiple winners if needed which are computed by finding the strongest path on a directed graph. It has a particular popularity in the computing community and is used by organizations such as Debian, GNU PG, Haskell, and Ubuntu [26]. It is also used by the Pirate Party chapters around the world and even by Mtv to choose the order music videos are played. Much like the Tideman method, Schulze uses a directed graph, this time however, the weights of the victories are included in the calculation of the victor. Once the graph is drawn a search for the strongest path begins. Schulze's method is actually better thought of as two algorithms, first of which is a Condorcet pairwise comparison followed by a search for the strongest path which signifies the victor. When the edges are constructed by margin they are calculated as such, $N[a,b]-N[b,a]$ where $N[a,b] > N[b,a]$ gives edge ab a weight of $(N[a,b], N[b,a])$. The edge strengths could also be calculated by ratio, e.g., $N[a,b]/N[b,a]$ or by winning or losing votes which is the support or opposition of $N[a,b]$ but we will not be examining these here. Schulze calculates the strongest path from candidate a to candidate b, $P[a,b]$, through candidates $c_1, ..., c_n$, with the strength of that path being $min(N[a,b], N[b,a]), (N[c_i, c_{i+1}], N[c_{i+1}, c_i]) | i = 1, ..., (n-1)$; or the strength of the weakest edge. Schulze uses the Floyd-Warshall algorithm to calculate the strongest path between candidates a and b which has a runtime of $\mathcal{O}(C^3)$ with C being the total number of candidates.

Figure 31: Schulze Method Algorithm: Stage 1[26]

```
Input  : List of N[i,j] pairwise comparisons
Output: P[i,j] the strongest path from i to j
while there are ballots to be counted do
    count ballots;
    for i from 1 to |C| do
        for j from 1 to |C| do
            if i ≠ j then
                P[i,j] ← (N[i,j], N[j,i])
                pred[i,j] ←i
            end
        end
    end
end
```

Algorithm 15: Schulze method: Stage 1 (initialization)

Figure 32: Schulze Method Algorithm: Stage 2[26]

```
for i from 1 to |C| do
    for j from 1 to |C| do
        if i ≠ j then
            for k from 1 to |C| do
                if i ≠ k then
                    if j ≠ k then
                        if P_D[j,k] < min{P_D[j,i], P_D[i,k]} then
                            P_D[j,k] ← min{P_D[j,i], P_D[i,k]}
                            if pred[j,k] ≠ pred[i,k] then
                                pred[j,k] ← pred[i,k]
                            end
                        end
                    end
                end
            end
        end
    end
end
```

Algorithm 16: Schulze method: Stage 2 (calculating the strengths of the strongest paths)

Figure 33: Schulze Method Algorithm: Stage 3[26]

```
for i from 1 to |C| do
    winner[i] ← true
    for j from 1 to |C| do
        if i ≠ j then
            if P_D[j, i] > P_D[i, j] then
                ji ∈ W
                winner[i] ← false
            else
                ji ∉ W
            end
        end
    end
end
```

Algorithm 17: Stage 3: Finding a unique winner or set of potential winners

Figure 34: Schulze Method Example

Schulze Example

Preference Number \ Number of Voters	4	3	2	1
$1^{st}choice$	c_2	c_3	c_1	c_4
$2^{nd}choice$	c_3	c_2	c_2	c_3
$3^{rd}choice$	c_1	c_4	c_3	c_2
$4^{th}choice$	c_4	c_1	c_4	c_1

3.16 Veto System

Veto system is also called the anti-plurality voting system and voters will votes against candidates. In this system the candidate with the lowest score is the winner.

```
Input   : List of candidates C and ballots from all voters in V
while there are ballots to be counted do
    count ballots;
    i ← 1
    for all c_i in C do
        if c_i is preference choice then
            c_i score ← c_i + 1
        end
    end
end
Sort candidates by scores.
return Candidate with the lowest score.
```

Algorithm 18: Veto

Veto example

Here candidate c_2 is returned the winner with the fewest votes.

Number of Voters	8	6	7	9
Candidate	c_1	c_2	c_3	c_4

4 Conclusion and future work

4.1 Conclusion

This paper has clearly and simply explained the inner workings of voting systems for the purpose of transparency and fairness. We conclude from the work creating algorithms for the above voting methods that there are clear and substantive differences that effect the speed and clarity of results. It should not be understated that the safety of an election not only depends on those cybersecurity methods that focus on verifying and securing hardware and software, but also on the ability to reasonably and effectively explain those results and how they were achieved to the voters. Because many of the above systems provide the same result it would be natural to ask how to choose between them. If we use the works of Kenneth Arrow as a guide and work within the framework of his Impossibility Theorem we can see which voting systems have what shortcomings and make our necessary compromises accordingly. We have shown that Approval voting is not computationally expensive but lacks some of the finer nuance that can be afforded from other systems. The Borda count has an easy to understand and implement system it is surpassed by other methods when reaching a desired outcome. While Nanson's and Baldwin's methods can convey that additional voting information from the voter and still pick the Condorcet winner when one exists—and doing so in a manner that is also easy to explain to the layperson. The implementation of either of these systems while not trivial is relatively easy given that they are iterative implementations of the Borda count. They are also more secure from manipulation because of their properties of eliminating low scoring candidates. Bucklin's method is in essence an iterative implementation of plurality voting that takes in ranked choice ballots. It is also very easy for potential voters to understand as it is simply adding rounds of voting until a candidate obtains a clear majority. The Condorcet method while simple in appearance is more complex in its implementation. While the concept of the round-robin is well known in practice when there are potentially thousands of voters and factors such as cycling are considered the implementation becomes more difficult. That race of four candidates requires six pairwise comparison might be considered trivial those comparisons balloon to ten comparisons with 5 candidates, 15 comparisons with 6 candidates, and 21 comparisons with seven candidates. Coombs and the Instant-runoff methods do not require a complex implementation in that both merely count two placements of a ranked choice ballot and iterate accordingly. That Coombs tends to deliver the more desired outcome is evident by IRV's procedure of dropping the lowest first choice at the risk of eliminating a very popular second choice. This gives Coombs the edge between the two. The exhaustive ballot method is simple as it is a simple plurality vote conducted successively. It is that simplicity that makes it impractical for elections with more than a few hundred voters. Kemeny-Young loses practicality simply by the fact that a worse case scenario finding a winner may fall within NP-C problems. While the implementation is not exceedingly difficult the fact that it is within the realm of possibility that no winner may be found eliminates it from widespread use. Majority Judgment and Range voting are two systems that pick their winners by finding either the highest median or mean scores respectively. One benefit of these systems is their resistance to tactical voting as it would require perfect knowledge of all ballots cast. The implementations of these algorithms need not be computationally expensive as Range voting requires a simply dividing Borda count scores by the number of voters and Majority judgement counts all each 'grade' and either returns that candidate with the highest median or repeatedly eliminates individual median grades until a winner is found. Plurality and Veto systems are exceedingly easy to implement and an explanation of how it works consists simply of whoever gets the most votes wins. There are however many instances where the most desired candidate will not be the winner, especially due to the spoiler effect. While this is the most common real world voting system, it is apparent that there are better options available. Finally Tideman's and Schulze's methods lose the simplicity argument by requiring directed graphs in order to find winners. As we have shown, the algorithms are also quite complex to implement and execute. The difficulty of being able to explain these systems broadly coupled with the computational expense eliminates them from the running of widespread viability.

That a compelling case could be made for many of these systems is evident. But each system comes with an element of uncertainty that must be addressed. We have worked to make that uncertainty more transparent and give decision makers those tools necessary to explain and implement the fairest systems available.

In order to ease selection a table is presented showing some comparisons of election criteria and runtimes of popular voting systems. The selection of the Condorcet winner for cases where it is imperative to find that candidate who can defeat all others. Methods that can claim independence of irrelevant alternatives will allow candidates to be rated individually without regard to their competitors, while clone proof methods are useful in cases where there are many similar candidates. For runtime complexity we show how the number of voters V and the number of candidates C can independently affect runtimes, and in the cases of Ranked Pairs and Schulze, where there are two distinct steps, first creating the pairwise comparison matrix which has a runtime of $\mathcal{O}(V \cdot C^2)$ plus, as in the case of Tideman's ranked pairs, Kruskal's Algorithm, which has a runtime of $\mathcal{O}(E \log C)$ where E is the number of edges created which is $\frac{C^2 - C}{2}$.

Voting System	Condorcet Winner	IIA	Clone Proof	Ballot	Runtime
IRV	no	no	yes	ranking	$\mathcal{O}(V \cdot C^2)$
Majority Judgment	no	yes	yes	scored	$\mathcal{O}(V \cdot C)$
Nanson's Method	yes	no	no	ranking	$\mathcal{O}(V \cdot C^2)$
Ranked Pairs	yes	no	yes	ranking	$\mathcal{O}(V \cdot C^2 + E \log C)$
Schulze	yes	no	yes	ranking	$\mathcal{O}(V \cdot C^2 + C^3)$[26]

4.2 Future Work

While we have made inroads towards adding to general understanding of some of the most common voting systems there are many questions that may still be answered. Because of the broad use of voting systems and choice aggregation some of these questions fall outside of the realm of mathematics or computer science. While it remains impossible to get around the Impossibility Theorem there is still much work that can get satisfactorily close. The search will require a broad approach that unite politics, the law, and humanities with math and computing. One potential way to evaluate these systems using real world data would be through the performance of exit polling. This would be especially useful in the U.S. where few locations allow for ranked choice voting. This would obviously require a great deal of work in gathering data and then processing it through the implemented voting systems. Another evaluation method should be examining proofs of correctness for these algorithms. The application of Hoare logic with precondition and postcondition can prove total and partial correctness thus setting expectations of how the algorithms will run.

Appendices

In these appendices we will show the formal statements of the two most important theorems behind ranked choice theory. This will give us the opportunity to examine some formalizations of social choice systems as presented by Arrow, Gibbard, and Satterthwaite. As said previously, the statement x is preferred to y can be stated as xPy. We can also show the weak ordering, or no preference between x and y by stating xRy. Additionally, C is our set of three or more candidates, v is called our agenda in voting theory, or the subset from C which contains the winning candidate, n is a positive integer representing the number of voters, S represents our voting system, and P is our profile, or the set of ballot preferences, with P_i being the preference of our i^{th} voter.

Arrow's Theorem for Social Choice Functions[1]

Independence of Irrelevant Altervatives: $For\, every\, two\, (C,n)-profiles\; P\; and\; P',$ $and\ every\ agenda\ v \subseteq C,\ if\ R_i|v = R'_i|v\ for\ every\ i,\ then$
$S(P)(v) = S(P')(i)$

Nondictatorship: $There\, is\, no\, i\, with\, the\, following\, property:\ For\ every\ (C,n)-profile\ P,\ every\ agenda\ v \subseteq C,\ and\ every\ pair\ of\ candidates\ x, y \in v,\ if\ xP_iy\ for\ this\ particular\ i\ then\ y \notin S(P)(v)$

Pareto: $For\ every\ (C,n)-profile\ P,\ every\ agenda\ v \subseteq C,\ and\ every\ pair\ of\ alternatives x, y \in v,\ if\ xP_iy\ for\ every\ i,\ then\ y \notin S(P)(v)$

Gibbard-Satterthwaite Theorem[4]

As the original Gibbard-Satterthwaite was presented by Gibbard as a corollary to Arrow's theorem [4], we may state simply that any voting system is either limited to only two candidates, dictatorial, or manipulable.

Dictatorship: *There is a dictator k where* $yP_kx \implies S(P)(v = y)$ *when* $xP_iy \, \forall \, i$.

Manipulable: *There is a voter k where* xP_ky *may be changed to* $yP'_kx \implies S(P)(v = y)$.

References

[1] "Social Choice and the Mathematics of Manipulation", by Alan D. Taylor, *Cambridge University Press, New York, 2005*

[2] "Gaming the vote: Why elections aren't fair: And what we can do about it." by William Poundstone *Hill and Wang, New York, 2008*

[3] "Social Choice and Individual Values. 3rd ed." by Kenneth Joseph Arrow *Yale University Press, New Haven, 2012*

[4] "Manipulation of Voting Schemes: A General Result", by Allan Gibbard, *The Econometric Society, vol. 41, no. 4, New York, 1973*

[5] "Strategy-Proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions", by Mark Allen Satterthwaite, *Journal of Economic Theory, vol. 10, is. 2, 1975*

[6] "Models for metasearch", by Javad A. Aslam and Mark Montague, *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, 2001*

[7] "Detector of Image Orientation Based on Borda-Count", by Loris Nanni and Alessandra Lumini, *Pattern Recognition and Image Analysis, IbPRIA, 2005*

[8] "Concurrent bilateral negotiation for open e-markets: the Conan strategy", by Alrayes, B., Kafalı, Ö., and Stathis, K., *Knowledge and Information Systems, Volume 56, Issue 2, Springer, London, 2018*

[9] "Numbers Rule: The vexing mathematics of democracy, from Plato the the present.", by George G. Szpiro, *Princeton University Press, Princeton, 2010*

[10] "The Borda Dictionary", by D.G. Saari, *Social Choice and Welfare, vol. 7, no. 4, 1990*

[11] "Chaotic Elections! A mathematician looks at voting", by Donald G. Saari, *American Mathematical Society, Providence, R.I., 2001*

[12] “Approval Voting”, by S. Brams and P. Fishburn, *Birkhauser, Boston, 1982*

[13] “Election by Majority Judgment: Experimental Evidence”, by Michel Balinski and Rida Laraki, *hal-00243076, https://hal.archives-ouvertes.fr/hal-00243076*

[14] “Behind the Ballot Box”, by Douglas J. Amy, *Praeger, Connecticut, 2000*

[15] “Complexity of and algorithms for the manipulation of Borda, Nanson’s and Baldwin’s voting rules”, by Jessica Davies, George Katsirelos, Nina Narodytska, Toby Walsh, and Lirong Xia, *Artificial Intelligence, vol. 217, December 2014*

[16] “Defining Democracy: Voting Procedures in Decision-making, Elections and Governance”, by Peter Emerson, *Springer Science and Business Media, 2011*

[17] “Declaration of Results 2019 Parliamentary Elections Anetan Constituency” *Electoral Commission of Nauru, 2019, retreived from https://www.election.com.nr/wp-content/uploads/2019/08/Anetan.pdf*

[18] “Voting schemes for which it can be difficult to tell who won the election”, by J. Vartholdi III, C.A. Tovey, M.A. Trick, *Soc Choice Welfare (1989) vol. 6, iss. 157, https://doi.org/10.1007/BF00303169*

[19] “Majority Judgement: Measuring, Ranking, and Electing”, by Michel Balinski and Rida Laraki, *MIT Press, Cambridge, 2010*

[20] “Legislative Principles: The History and Theory of Lawmaking by Representative Bovernment”, by Robert Luce, *Houghton Mifflin, New York, 1930*

[21] “Duverger’s Law of Plurality Voting” by Shaun Bowler, André Blais, and Bernard Grofman, *Springer, New York, 2009*

[22] “They Always Wanted Trump”, by Gabriel Debenedetti, *2016 Retrieved from: https://www.politico.com/magazine/story/2016/11/hillary-clinton-2016-donald-trump-214428*

[23] "Does "Bullet Voting" Really Work?", by Holly Otterbein, *Philadelphia Magazine, 2015, Retrieved from:https://www.phillymag.com/citified/2015/10/27/bullet-voting-report/*

[24] "On the Rationale of Group Decision-making", by Duncan Black, *Journal of Political Economy, University of Chicago Press, Vol. 56, No. 1, Feb. 1948*

[25] "The Borda Count and its real-worl alternatives: Comparing scoring rules in Nauru and Slovenia.", by Jon Fraenkel and Bernard Grofman, *Australian Journal of Political Science, Vol. 49 iss. 2*

[26] "The Schulze Method of Voting", by Markus Schulze, *2019 Retrieved from: https://arxiv.org/abs/1804.02973*

[27] "Amazon scraps secret AI recruiting tool that shoed bias against women.", by Jeffrey Dastin, *Reuters, 2018, Retrieved from:https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G*

[28] "Eliminating the weakest link: making manipulation intractable?", by J. Davies, N. Narodytska, T. Walsh, *Proceedings of the 26th AAAi Conference on Artificial Intelligence, AAAI-2012*

[29] "DARPA is Building a 10$Million, OpenSource, SecureVotingSystem'', byKimZetter,$

Motherboard Tech by Vice, 2019, Retrieved from: $https://www.vice.com/en_us/article/yw84q7/darpa - is - building - a - dollar10 - million - open - source - secure - voting - system$

[30] "The Borda rule and the pairwise-majority loser revisited", by Noriaki Okamoto and Toyotaka Sakai, *Review Economic Design, 2019, Vol. 23*

[31] "Varietis of failure of monotonicity and participation under five voting methods", by Dan Felsenthal and Nicolaus Tideman, *Theory and Decision, 2013, Vol. 75*