Automatic Analysis of Voting Procedures

Nadya Peek
BSc project in Artificial Intelligence
supervised by Ulle Endriss
npeek@science.uva.nl
Universiteit van Amsterdam

August 7, 2008

Since the majority of me

Philip Larkin

Since the majority of me
Rejects the majority of you,
Debating ends forwith, and we
Divide. And sure of what to do

We disinfect new blocks of days
For our majorities to rent
With unshared friends and unwalked ways.
But silence is eloquent:

A silence of minorities
That, unopposed at last, return
Each night with cancelled promises
They want renewed. They never learn.

Abstract

There are many different ways to elect a winner from a group of candidates. The best known method is the plurality rule, for which each voter selects one candidate to vote for and the candidate receiving the most votes wins. Some rules require the voters to rank the candidates, after which points are assigned to the candidates depending on their ranking position, again with the candidate with the most points winning. An example of a positional scoring rule like this is the Borda count. There are many other voting procedures in use as well, such as approval voting, where voters can select an acceptable set of candidates, or the single transferrable vote, where votes for the most unpopular candidates are redistributed until one candidate has an absolute majority. Other voting procedures considered here are Copeland and Dodgson's procedures. Each voting procedure may satisfy its own set of criteria, such as always electing the candidate who is first-ranked by an absolute majority of the votes if this candidate exists (majority criterion), or always electing the candidate who would win in a pairwise comparison to all other candidates (Condorcet criterion). Each voting procedure also may make it more or less difficult for a voter to manipulate the outcome of the election by not voting according to his true preferences. Furthermore, for some voting rules it can be very difficult to find the winner, for some rules finding the winner is even NP-hard.

For this thesis we compared a selection of voting procedures in two ways. Firstly we found the smallest election instances in which the different procedures would elect different candidates. Secondly we generated many elections with random sets of voters to check how often different procedures coincide, how often different procedures elect the Condorcet winner, or candidate who would win in a pairwise comparison with all other candidates, and how often different procedures elect the Condorcet loser, or the candidate who would lose in a pairwise comparison with all other candidates.

Contents

Chapter 1

Introduction

In the US presidential elections of 2000, Florida was a swing state. After the controversial recount, George W. Bush received 48.850% of the popular vote, and Al Gore received 48.841%, making Bush the winner by a 0.009% margin [6]. 1.633% of the voters voted for Ralph Nader, an independent. In the preferences of the voters, it seems pretty clear that the voters who would most like for Nader to be president will not have Bush as their second choice. If they had not given their vote to their most preferred candidate but instead to Gore, their likely second choice, Gore would have won Florida.

Here one might be inclined to blame the voters who made the futile decision to vote for Ralph Nader. However, those voters are merely truthfully stating their preferences, without regard to how their preferences will be taken into account in the election. One could also consider that the voting system the US presidential election uses poorly represents its voters.

Plurality voting, as is used in the US presidential elections, is the most common voting system. Each voter may cast one vote, and the candidate with the most votes wins the election. The system is so simple, it may seem strange that there is any study in the field of voting theory at all. However, plurality voting also does not take the preferences of its voters beyond their most preferred candidate into account. Many other voting systems have been developed with the aim to make a more representative aggregate of the individual preferences. Each system has its own benefits and deficiencies.

This work is about the comparison of voting systems. Chapter 2 is dedicated to summarising the different voting systems in use and explaining exactly how they elect a winner. It is examined which criteria each voting rule satisfies: do positional scoring rules always elect a majority winner? Do two small approval elections who both elect the same candidate also elect this candidate when they are combined? If a candidate is more popular than each other candidate separately, does he win— i.e. does the voting rule always elect the Condorcet winner?

Comparing only the procedures of the different voting systems is not enough to be able to understand how the rules represent different voter populations. To be able to better understand this phenomenon, we developed several experiments to rigorously illustrate the differences in the voting systems. First we exhaustively searched through all possible elections, starting from the smallest possible. This way we found the smallest election for which two or three voting rules all elect a different winner. Second, we randomly generated a large amount of elections to do some statistical analysis. With these elections, we analyse how often in an election a Cordorcet winner exists, and also how often a Condorcet loser exists. For each voting procedure we also analysed how likely it would be for the rule to elect a Condorcet winner or a Condorcet loser.

To be able to run these experiments we have developed a framework with several implementations of different voting procedures. Within this framework we can randomly generate elections and use the data to compare the voting systems. We can also read in user specified ballots in our particular balloting language.

One of the voting procedures we implemented for our comparative framework is notoriously difficult to calculate a winner for. It is the procedure that was suggested by Charles Dodgson, better known under his pseudonym Lewis Carroll. Finding the winner under his voting procedure is NP-hard, and our implementation cannot calculate the winner for anything larger than a small election. Because of these difficulties, we propose a different method for finding the Dodgson winner in chapter 3. We propose to use a heuristic guided search to find the winner.

An overview of this thesis is as follows: the background theory and explanation of the different voting systems is done in the next chapter. It is followed by our discussion of Dodgson's procedure in chapter 3. In chapter 4 we discuss the different ways we can automatically generate pools of voters and thus elections, in chapter 5 we explain how we can use these voter pools in our comparative framework. Chapter 6 is dedicated to the results from our smallest different election instance experiments, and chapter 7 is dedicated to the statistical analysis of the voting procedures. In chapter 8 we will discuss some of the interesting things we noticed through the course of this research, and finally we will conclude in chapter 9.

The framework we used, as well as this report, can be downloaded from

http://voting.infosyncratic.nl

Chapter 2

Background Theory

In this chapter, we will define different sets of voting procedures. To do this, we will first compare the different ways that we can represent individual preferences in a single voter's ballot. Then we will consider the different voting procedures which each take a particular individual preference representation. We will begin with voting procedures which use positional scoring vectors to assign points to candidates. We then will consider approval voting and some variants, followed by voting procedures which transfer votes from less popular candidates to more popular candidates during the election. Finally we will consider voting procedures which satisfy Condorcet's criterion.

Besides the comparison of the winner election algorithms, it is interesting to consider how susceptible each voting procedure is to manipulation and dishonest voters. We will consider this in section 2.3.

2.1 Individual Preference Representation

Quantifying an individual's preferences towards a set of candidates is not a trivial task. It might be possible for an individual to say that he likes candidate a better than candidate b, and also that he likes candidate b better than candidate c, and therefore we might be able to conclude that in his preferences $a > b > c$. However, it does not show *how much* the voter prefers a to b or c. Perhaps the voter agrees with candidate a's views on almost all issues, except for one issue in which she regards candidate b to have a much better approach. It is also still possible that she prefers a as a candidate for all issues. If the voter only prefers a to b for 50% of the issues, we might say that the voter is *indifferent* between a and b. In this section, we will consider different forms of rankings for the candidates, other preferences representations which can represent an individual's preferences, and finally we will discuss how feasible it is to elicit full information on the preferences of a voter.

2.1.1 Rankings

One of the most intuitive ways of comparing candidates is for a voter to list them in order of preference. This allows voters to express preferences over multiple or all candidates .

There are two different kinds of rankings: a strict ranking, and a weak ranking. In a strict ranking, the voter must always necessarily prefer one of the candidates in a pairwise comparison. Because the ranking is complete over all participating candidates, there is a full sequence denoting the preferability of a candidate, i.e. $a > b > c$. Formally speaking, strict rankings are irreflexive, asymmetric, transitive and its members are all unequal.

In a weak ranking, it is allowed for the voter to be indifferent between some of the pairs of candidates, creating ties within the ranking, i.e. $a \geq b \geq c > d$. In a weak ranking therefore some of the candidates may be considered as equal.

2.1.2 Other Preference Representations

Another way of creating an individual preference representation would be to assign points to each candidate. A voter could for instance distribute some 100 points over 10 candidates, giving the candidates he much preferred many more points that the candidates he did not like at all. This kind of structure can of course be mapped to a weak order, albeit with loss of some of the information given by how many more points a candidate was assigned than another.

A method called Approval Voting introduced in the 70s [5], which has voters approve of certain candidates and disapprove of all others, uses an approved set of candidates to represent the voters' preferences. These sets can also be translated into a weak order of the form $a_1 \geq ... \geq a_n > d_1 \geq ... \geq d_m$. This is a simpler ballot, which could be desirable if one believes voters might not care to fill out a full ranking of candidates.

2.1.3 Dealing With Incomplete Information

Especially in elections with many candidates, not all voters may have formed an opinion on all candidates. It may be useful to not always demand that all voters give a full ranking of all candidates− they might not know, and then their filling in of the ballot could become arbitrary. Being able to deal with incomplete information becomes a very desirable characteristic of a voting system. Plurality voting has very simple ballots, each listing only one candidate. Approval voting allows a voter to only fill in the candidates she approves of, possibly ignoring unknown candidates. It might however also be interesting to allow the partial ranking of candidates, i.e. if a voter knows he prefers a to b and c to d, but is unsure if he prefers d to b.

2.2 Voting Procedures

Now that we have considered the various types of individual preference representation, we will continue with the consideration of different voting systems. We will start out with the methods which take the ordinal ranking of a voter and assign points to the candidates according to where they are located on the ordinal ranking. Next we will consider approval voting and its variants. Then we will consider run-off elections, where the winner is determined after a series of competitions. Specifically we will consider (instant) run-off elections such as the Single Transferrable Vote method. Finally we will consider methods which satisfy the Condorcet criterion, i.e. that the winner of the election should also win in any election with only one of the other candidates.

2.2.1 Positional Scoring Procedures

After all voters have given a strict ordinal ranking to all of the candidates, these candidates may be assigned points, the number of which then depends on their location in the ranking. This can be done by creating a scoring vector, with at the 0^{th} position the amount of points for the most preferred candidate, at the 1^{st} position the amount of points for the second most preferred candidate, etc. Not all positions on the candidate ranking need to be assigned points, a positional scoring vector may also have the form $\langle 3, 2, 1, 0, 0, ..., 0\rangle$, where after the 3rd position no candidates receive any points.

Plurality

Plurality voting elects a single candidate and is the most simple form of a positional scoring procedure, with only the most preferred candidates receiving points. The scoring vector is a 1 followed by $n-1$ 0s, with n being the number of candidates. The candidate which then receives the most points is declared the winner. Plurality voting does not require the voters to submit a full ranking as a ballot, but merely needs one name per voter, as the remainder of the candidates will not receive points anyway. This means that balloting for plurality voting is significantly simpler than for other methods.

In plurality voting it may be the case that some candidate receives as many points as another. In this case there is a plurality tie. The resolution of ties depends heavily on the purpose of the election, and is not addressed in the procedure itself.

Antiplurality

While plurality voting assigns only one point to the most popular candidate, antiplurality assigns points to all candidates except the least popular. The scoring vector is then of the form $\langle 1, 1, ..., 1, 0\rangle$. This method can also be seen as voting for your *least* favourite candidate, with the candidate with the least amounts of votes winning [23].

Borda Count

In 1770 Jean-Charles de Borda introduced the Borda count as a single-winner election method for the members of the French Academy of Sciences [10]. When n is the number of candidates, Borda count awards n points the most preferred candidate of a ballot, $n-1$ points to the second most preferred, etc., and finally 1 point to the least preferred. This is equivalent to using the positional scoring vector $\langle n-1, n-2, n-3, ..., 0\rangle$. The points assigned to each candidate in each ballot are summed, and the candidate with the most points is declared the winner. Again it is possible for there to be multiple candidates with the same amount of points, in which case there may be a tied winner.

Other Positional Scoring Rules

There are many unnamed procedures one might also like to use which can be defined as a positional scoring rule. There are many steps between plurality $\langle 1, 0, ..., 0\rangle$, Borda $\langle n-1, n-2, ..., 0\rangle$ and antiplurality $\langle 1, 1, ..., 0\rangle$. Testing at which point the winner of an election would change depending on the voting rule would be an interesting problem to investigate [12]. We have done this for elections with three candidates, the details of this experiment are described in section 7.5.

2.2.2 Approval Voting and Variants

Approval voting was introduced fairly late, in 1976 by Guy Ottewel and Robert J. Weber [5]. Since then it has experienced some popularity as an alternative to plurality voting. One of the strongest arguments for approval voting is that voters do not need to submit a full ranking of all candidates in the election. This makes the ballot significantly easier to fill out and collect than for instance Borda count, but it remains more difficult than plurality, which only requires one candidate on a ballot.

Approval

In classic approval voting, a ballot consists of two groupings of the candidates, the candidates which are approved by the voter and the candidates which are not approved. The approved candidates all receive one point, whereas the non-approved candidates receive none [5]. Again, the candidate with the largest amount of points wins the election, and there is still the possibility of a tie.

Cumulative Voting

It may seem slightly unfair to allow a voter who approves for many candidates to be able to give each of those candidates as many points as a voter who only approves of one candidate. In a variation on classic approval voting, cumulative voting distributes the points received by the candidates according to the number of candidates that a voter approves [1]. In total, one point may be allocated to all of the approved candidates, and this is distributed according to the voter's preferences. If in total there are 3 approved candidates, each might receive $\frac{1}{3}$ point, or perhaps two will receive $\frac{1}{4}$ point, and one will receive $\frac{1}{2}$. This distribution of one point per voter helps distinguishing between voters who may approve of almost all candidates, and voters who may only approve of a few.

Size approval voting

Size approval voting is a form of cumulative voting where the weight of the vote is distributed according to how many candidates the voter has voted for. Weights are non-negative and decrease as the amount of approved candidates increases. Even and equal voting is a specific type of size approval, where the amount of points all approved candidates received is $\frac{1}{k}$ when k candidates are approved [1]. In size approval voting, one voter will only approve of a set of candidates he values more or less equally.

2.2.3 Run-off and Single Transferrable Vote

If plurality voting already provides an overwhelming majority (i.e., more than 51% of the voters vote for a certain candidate), then it might seem unnecessary to use a voting procedure more complicated than plurality. However, when there are more than 2 candidates, a majority is more often not the case. In countries where there are many candidates for president such as France, a run-off election follows the initial election, where the two most popular candidates are selected for the follow up vote. This procedure known as plurality run-off was heavily criticized in 2002 when it included Jean-Marie le Pen in the run-off presidential election in France. Although Jacques Chirac had 19.88% of the popular vote in the initial election with Le Pen trailing with 16.86%, the final election between the two resulted in a landslide victory of 82.21% for Chirac and a mere 17.79% of the vote for Le Pen [7]. This peculiar election led to questioning of the voting system applied, and whether instead a system should be used that considers more than the two most popular candidates.

If the voters submitted a full ranking of candidates, the need to submit a ballot for the second election would be made superfluous. Submitting a full ranking of candidates and using that in subsequent run-off elections is known as instant plurality run-off.

Hare's Method - Single Transferrable Vote

Instead of holding a run-off election with the two candidates that received the most points in the initial plurality election, Thomas Hare proposed an alternative system for the redistribution of votes not given to the most popular candidates [15]. In the single transferrable vote procedure, voters submit a full strict ranking of candidates. A plurality election is held between the first-ranked candidates. If one of the candidates has a majority already, she is declared the winner. Otherwise, the candidate who has received the least amount of votes in the first-ranked election is eliminated from all ballots, and the votes from the voters who ranked that candidate first are proportionally redistributed to their second-ranked candidates. If still no candidate has a majority, the process is repeated, and the candidate which now has the least votes is eliminated.

In figure 2.1 we observe and example STV election with the candidates Jagger, Dylan, Cash and Gainsbourg. In round 0 we observe the plurality scores of the candidates, where Jagger is the winner with 8 votes. However, because this is not yet a majority, the candidate with the least amount of votes, Gainsbourg, is eliminated. In round 1 we see that the votes have been reallocated to all three remaining candidates, and still no one candidate has a majority. Therefore another candidate will need to be eliminated. Now Cash has the least amount of votes. Once the Cash votes are reallocated, we see that voters who most prefer Cash have Dylan more often as a second choice than Jagger. Therefore, Dylan overtakes Jagger in the last round and becomes the STV winner.

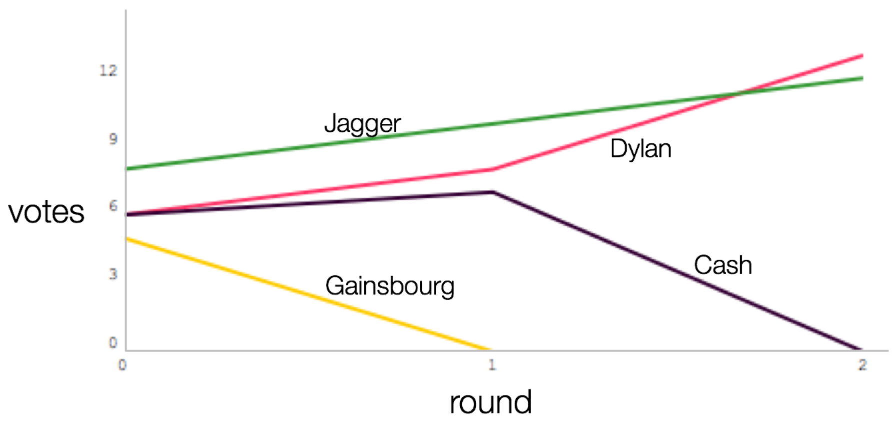

Figure 2.1:

Hare's method is also used for elections in which more that one candidate is elected, for instance when electing a board. In that case, the quota which declares a candidate a winner (in the case of one candidate this is more than half, to have a majority) is adapted to the amount of candidates who are to be elected. If a candidate receives more than the quota necessary to be elected, the surplus votes are proportionally redistributed to the candidates who are ranked after the elected candidate.

STV is used in the Republic of Ireland, New Zealand, Australia and Malta for governmental elections.

Coombs' Method

Coombs' method is very similar to Hare's method, and differs only in the selection of the candidate who is to be eliminated. Instead of the candidate who loses in the current plurality election, the candidate who is ranked last by most voters is eliminated [14]. An alternative way to see this is that in Hare's method the plurality loser is eliminated, whearas with Coombs' method the antiplurality loser is eliminated.

2.2.4 Condorcet Criterion Consistent Voting Procedures

The Marquis de Condorcet developed the concept of the Condorcet method in the 18th century. He worked simultaneously (and frequently disagreed) with Jean-Charles de Borda, whose voting procedure involved positional scoring. Instead, Condorcet proposed that the candidate who would win in all pairwise elections to all other candidates should be the winner of the election. Subsequently, this winner is known as the Condorcet winner.

Unfortunately, it is not always the case that a Condorcet winner exists. For example, in an election with three candidates and three voters, imagine the following ballots:

$$a > b > c$$
$$b > c > a$$
$$c > a > b$$

Obviously, we have a tie between a, b and c. Suppose we would then arbitrarily select one of the tied winners, for instance a. In the remaining two ballots, c is preferred to a. However, if we would then switch to c as a better winner, we would notice that in the ballots who do not have c as the most preferred candidate, b is preferred to c. The ballots which are individually transitive are aggregated to a cyclic election. This paradox is known as Condorcet's paradox.

Voting procedures which always elect the Condorcet winner when a Condorcet winner exists are said to satisfy the Condorcet criterion. Both positional scoring voting methods and approval based methods violate this criterion, but several other methods have been proposed which do adhere to this method. The differences between the methods which satisfy the Condorcet criterion lay in how they deal when a Condorcet winner does not exist.

Copeland's Method

Copeland proposed a point system where a point is assigned to a candidate each time he wins in a pairwise comparison to another candidate [8]. Wins and losses of pairwise comparisons can be shown in a matrix like below:

	a	b	c
a	0	2	2
b	3	0	4
c	3	1	0

Here each number represents the number of victories the candidate in the column has over the candidate in the rows. In total for this election, there are 5 voters. The ijth entry plus the jith entry should always sum to the amount of voters there are.

If the a candidate has more than $\frac{1}{2}m$ victories over another candidate, with m being the number of voters, the candidate wins the pairwise comparison and is awarded a Copeland point. In our example a receives 0 Copeland points, b receives 2 Copeland points and c receives 1 Copeland point.

If two candidates tie in a pairwise comparison because just as many people prefer the first candidate to the second as the second to the first, some portion of a point is allocated, most often half a point. The Condorcet winner, if one exists, would receive the $n - 1$ Copeland points, n being the number of participating candidates, winning the election. In our example, this is candidate b. The Condorcet loser would receive 0 Copeland points, always losing the election. In our example, this is candidate a.

Dodgson's Method

C.L. Dodgson, more famous under his pseudonym Lewis Carroll, proposed a voting procedure which when a Condorcet winner did not exist, would find the candidate with maximally close resemblance to the Condorcet winner [4]. He proposes to do this by examining how much the ballots submitted by the voters need to be modified to make a candidate who is not a Condorcet winner into one. The candidate needing the least amount of modification is then elected as the Dodgson winner.

More specifically, Dodgson proposes to count the number of minimal modifications in the rankings submitted by the voters, and elect the candidate with the least of these minimal modifications. The modification is then seen as the flipping of two candidates in a ballot, changing the order of the voter's ranking. It is however computationally very complex to find out which combination of flips in which combination of ballots is the minimum amount of flips to make a candidate who is not a Condorcet winner into one. Calculating the winner of a Dodgson election is even an NP-hard problem [18].

We have attempted to provide an algorithm for calculating the Dodgson winner using a heuristic guided search algorithm. More on the exact algorithm, the heuristic and the results obtained can be found in chapter 3.

Black's Method

When no Condorcet winner exists, it may be preferable to not attempt to find the candidate who comes closest to being the Condorcet winner, but choosing a candidate who then does well in the election based on other criteria. Black combined both the theories of Condorcet and Borda in his method: if a Condorcet winner exists, he elects that candidate, otherwise he elects the Borda winner [4]. Whether the Borda count winner or the Condorcet winner is better is a much disputed point in voting theory, and Black's proposal to combine the two has been well received although infrequently implemented.

2.2.5 Various Other Criteria for Voting Systems

There are many properties for which it seems intuitive that a voting system should satisfy them: if a voter abstains from voting for his favorite candidate, the candidate should not have increased chances of winning; if a candidate wins a certain election, another candidate should win the same election if he were to replace the first candidate on the ballot, and many other such criteria. We have already introduced the Condorcet winner criterion, which requires the candidate who wins in all pairwise comparisons to win the election. There are however many more criteria defined which voting procedures may or may not satisfy, which may make a certain voting procedure more or less attractive for a particular sort of election. We will explain some of these criteria here.

Condorcet Losers

If there is a candidate who loses in pairwise comparison to all other candidates, he is known as the Condorcet loser [20]. This is not the same thing as the candidate who is not the Condorcet winner, unless there are only two candidates. If a voting procedure still allows the Condorcet loser to be elected, the procedure is said to violate the Condorcet loser criterion.

Plurality voting for instance does not satisfy the Condorcet loser criterion, as can be seen in the following example:

$$\text{4 voters: } a > b > c$$
$$\text{3 voters: } b > c > a$$
$$\text{2 voters: } c > b > a$$

In this example, a is the plurality winner with the most first ranked votes. However, a loses 5 times to both b and c in the remaining votes, also making a the Condorcet loser.

Many voting procedures do not satisfy the Condorcet loser criterion, and those which do not also include all forms of approval voting. Depending on how it is decided which candidate to eliminate, STV might also elect the Condorcet loser. Borda count voting however does satisfy the Condorcet loser criterion [22], as does plurality run-off voting.

Majority

We know now that purality does not always elect the Condorcet winner. However, if more than half of the voters have a particular candidate ranked as first, there is no longer a difference between the plurality winner and the Condorcet winner. It is said that then the candidate is the *majority* winner. A voting rule is said to satisfy the majority criterion if it always selects the majority winner if it exists [22]. If a Condorcet winner exists, he is necessarily the majority winner as well. Therefore, if a voting procedure satisfies the Condorcet criterion, it also satisfies the majority criterion.

Plurality, STV, Coombs' and Plurality run-off all also satisfy the majority criterion by definition, although these do not satisfy the Condorcet criterion. Borda count however does not satisfy the majority criterion, which can be seen in the following example:

$$\text{2 voters: } a > b > c > d$$

$$1 \text{ voter: } \quad b > c > d > a$$

The majority criterion places more importance with the first ranked candidates, whereas Borda count considers the full rankings by the voters.

Examples can be constructed as well which show that approval voting does not satisfy the majority criterion. However, constructing these examples does require a ranking of the candidates, which is not inherent to the representation of the individual preferences given in approval.

Consistency

If we were to have two separate elections which both elect candidate a, then the election formed by combining the two sets of voters should also elect candidate a. If this is not the case, the voting system is said to be *inconsistent* [22]. In certain cases of run-off voting, combining two voter sets from smaller elections which both elect candidate a will not elect candidate a. Therefore STV is one of the voting rules which does not satisfy consistency.

Monotonicity

If a candidate a loses an election, then lowering her in the ballots should not cause her to win. If this is the case, then the voting procedure does not satisfy the monotonicity criterion [22]. Again, STV and now also Coombs' method do not satisfy monotonicity. The candidate with the least amount of votes plays a large role in determining the outcome, because it is by means of the second preferences of her voters that the remaining votes are redistributed. If the least popular candidate changes due to people lowering their votes, then the second preferences of the voters of the now least popular candidate matter more. It is clear that most run-off procedures will suffer from this.

Pareto Efficiency

For a voting system to satisfy Pareto efficiency, if every individual prefers candidate a to candidate b then candidate a should receive more points than candidate b for the election outcome [3]. Voting systems which do not satisfy the Condorcet criterion will also not satisfy the Pareto efficiency criterion, because then a candidate who is preferred by a majority of all voters will not win and therefore also not reach the highest number of points.

Arrow's Impossibility Theorem

One might be inclined to think that all the criteria listed above are so obviously necessary, that it would be silly to adopt any rule which does not satisfy all of them. However, it turns out that it is very difficult to find voting systems which each satisfy all of these criteria. In the 50s, Kenneth Arrow demonstrated the impossibility to satisfy universality, non-imposition, non-dictatorship, Pareto efficiency and independence of irrelevant alternatives in one single system.

The universality criterion states that all ballots submitted must be taken into account, and if the same ballots would be considered again (albeit presented in a different order) the outcome of the election should be the same. The non-imposition criterion states that any final ordering in the outcome of the election should be possible, i.e. the voters should always be able to vote in such a way that any final ranking is possible. The non-dictatorship criterion states that the voting system may not simply take one voter's preferences into account, it must be an aggregation of multiple voters when multiple voters participate. The independence of irrelevant alternatives criterion states that if a is the winner of an election, then changing the ballots with respect to losing candidates b and c but not with respect to a should not change the outcome of the election. Each of these criteria seems like they should be easy to satisfy, and yet Arrow proves that they will never all occur together [3].

2.2.6 Electing More Than One Candidate

Some of the voting procedures detailed can also be used to elect more than one candidate, for instance when electing a board or committee. A simple way to make any voting procedure into a multiple winner method is to take the as many of the top ranked candidates as is necessary to fill al the spaces available. However, often with elections which seek to elect groups of candidates, it is not only the actual candidates which are important but specifically the proportion of a certain type of candidate with respect to the others. For this reason, many multiple winner voting procedures will aim for proportional representation, or that a party who receives 20% of the vote should also have $\frac{1}{5}$ of the seats.

Our focus lays with single winner voting procedures, and therefore we will not go into further detail on how multiple winner elections work.

2.3 Eliciting Sincere Voter Behaviour

Even if a voting procedures satisfies all the criteria mentioned above, it might fail on account of being very easy to manipulate. An easily manipulatable voting rule allows voters to change the outcome to something more preferable to them by not reporting their true preferences. In plurality voting, voters are encouraged not to necessarily vote for their most preferred candidate but to vote for the candidate who is most preferred by them and has a reasonable chance at winning the popular vote. One might for instance conclude that in the US presidential elections, it was useless to vote for Nader. However, this does mean that we expect voters to vote strategically.

It is much easier to vote strategically in plurality voting that it is for instance in STV. It is however also much more difficult to calculate the STV winner than it is to calculate the the plurality winner. In the best case, we would have a voting system for which it is easy to calculate the winner but difficult to manipulate.

In the 70s Gibbard and Satterthwaite indepently both published a result concerning the manipulability of tactical voting [13, 24]. It consists of three parts, namely that either

1. a voting rule is dictatorial, i.e. the winner is appointed by a single voter,
2. at least one of the candidates will never be able to win with the voting rule,
3. a voter who has full knowledge of the preferences of the other voters will have an incentive not to vote with their true preferences.

This result is slightly unsettling, as it does shows that no voting procedure will elicit a voter's true preference. However, some voting procedures are very difficult to manipulate, for some it is even NP-hard. Manipulability is not the focus here, but can also be considered as an important characteristic to take into account when selecting a voting procedure.

Chapter 3

Dodgson's Procedure

Although Charles Dodgson himself did not necessarily acknowledge the computational complexity of determining the winner in the manner he described in his 1876 pamphlet *A Method For Taking Votes On More Than Two Issues* [4], Dodgson's Procedure has become renown for its intractability. In his pamphlet Dodgson describes a series of seemingly simple steps for determining the Dodgson winner, but behind one of the steps innocuously formulated as *When the issues to be further debated consisted of, or have been reduced to, a single cycle, the Chairman shall inform the meeting how many alterations of votes each issue requries to give it a majority over every other separately* lays a problem which Bartholdi et al. proved to be NP-hard in 1989 [18] and later Hemaspaandra et al. proved to be Θ_2^p-complete, i.e. not even in NP [17]. Despite the inconvenience caused by this computational difficulty, the Dodgson winner is very desirable to compute for it is still considered to be able to provide a very accurate approximation of the Condorcet winner should a Condorcet winner not exist. Much work has been done to find a way to approximate the Dodgson winner [25, 21] should a brute force algorithm take too long to find the actual Dodgson winner.

In this chapter, we first detail the Dodgson method as initially proposed by Dodgson and as later interpreted by other voting theorists. Then we will describe a brute force algorithm for finding the Dodgson winner, then propose a method which uses the A* search algorithm to find the Dodgson winner without having to visit the entire search space.

3.1 Finding the Dodgson Winner

The precise method as described by Dodgson in his pamphlet can be summarised as follows:

1. Single votes for issues should be cast, with abstention as a possibility.
2. If there is an absolute majority, this issue will be carried.

3. If no absolute majority is found, the voters should give a full ranking of the issues. If an issue is found to have majority over all other issues, it shall be carried.
4. If no pairwise majority is found, and there exists a cycle within which all issues in the set beat the issues outside, the cycle is retained and the other issues removed.
5. It is determined how many alterations of the votes are necessary to give an issue a majority over each other separately.
6. The electors may choose to alter their votes, thus assigning a winner.

In Dodgson's method, the Condorcet winner is the Dodgson winner if she exists. In case the Condorcet winner does not exist, we should attempt to find the candidate who requires the least amount of alterations in the voters' ballots to become the Condorcet winner. An alteration is then seen as flipping two candidates in a ranking, and the number of alterations required to make a candidate into a Condorcet winner is called her Dodgson score.

All possible combinations of all possible alterations, or flips, between any two candidates will be a very large number ($(\frac{1}{2}n(n-1))^m$ for n candidates and m voters). However, if we do not flip the candidate that we are calculating the Dodgson score for, it will not help her win more pairwise comparisons, and therefore not help her become a Condorcet winner. Therefore instead we can consider only combinations of flips where we are flipping the candidate we are calculating the Dodgson score for up.

Let us consider the complexity of this task: in the worst case scenario, for each candidate we would have to consider all possible combinations of flips in all ballots. In each ballot, in the worst case the candidate we are inspecting is all the way at the bottom and it would require $n-1$ flips to reach the top, where n is the number of candidates. It is however not necessary to flip our candidate to the top in all cases, in fact we are looking for the minimum amount of flips. Unfortunately we are unsure in which ballots our flips are more effective, for a priori we are unsure of the exact placement of the other candidates in each of the ballots. So far, per candidate that means we will have to consider $(n-1)^m$ possibilities of combinations of flips, where m is the amount of voters. This is already significantly better than $(\frac{1}{2}n(n-1))^m$.

A brute force method for finding the Dodgson winner could be done by first generating all possible combinations of flips, ordering them least amount of flips first, and iterating through them until a Condorcet winner is found. A precise specification is given in BRUTE-DODGSON.

```
Brute-Dodgson(voters)
if ∃c ∈ candidates : Condorcet-Winner(c)
    then return c

flips = Generate-flips(num-cands, num-voters)
flips = Order-flips(flips)
for c in candidates
      do
           score[c] = Minimum-Alterations(c, flips)

return candidate with the lowest score
```

It would of course be more effective to find a way to generate the flip combinations in such a way that they are already in increasing summed order. This has not been done here and would be interesting future work. The current generation is specified in Generate-Flips.

```
Generate-Flips(num-cands, num-voters)
list = [ ]

if num-voters = 1
    then
          for i ∈ [0, num-cands]
               do
                    list.append([i])
          return list

          if num-voters > 1
             then
                    sublist = Generate-Flips(num-cands, num-voters-1)
                    for i ∈ [0, num-cands]
                         do
                              for j ∈ sublist
                                   do
                                        list.append([i] + j)
                    return list
```

Another optimisation which we could consider is that not all flips are possible, for in most cases the candidate we are calculating the Dodgson score for will not be all the way at the bottom of the ballot. Only taking the possible amount of flips into account decreases the amount of combinations we can generate, but again makes the generation algorithm much more complicated and has not been done here. Instead, a flip is performed only when it is possible within a ballot, as specified in Minimum-Alterations.

```
MINIMUM-ALTERATIONS(c,flips)
for flip-combo ∈ flips
      do
         i = 0
         for ballot in votes
              do
                 ind = INDEXOF(c,ballot)
                 if ind ≥ flip-combo[i]
                    then FLIP-UP(c, flip-combo[i])
         if CONDORCET-WINNER(c)
            then return Σ_flips flip-combo
```

Using the algorithms described in this section, we can calculate the Dodgson winner for up to 10 voters and 4 candidates. This is of course not sufficient for any realistically sized election, and as long as we are not provided with endless computational resources, we will need to find a better way to calculate the Dodgson winner.

3.2 The A* Algorithm

It is clear that especially in large elections with many voters, the number of possible flips will soon grow to be prohibitively large. In Dodgson's case, there would never be many voters, for he proposed his system to be used by the governing body of the Christ Church Ordinances of 1867, which never consisted of more than 11 electors. Should we want to apply his Condorcet winner approximation to larger instances, then we will have to intelligently search through the possibilities.

A* is a search algorithm proposed in the 60s by Hart, Nilsson and Raphael [16]. Instead of visiting all nodes possible in the tree, it uses a heuristic to explore more promising nodes first. The nodes are scored by means of a cost function, which determines the cost spent so far, and a heuristic function, which estimates how much cost would be necessary to reach a goal state from the current node. A* is guaranteed to find an optimal solution provided that the heuristic gives an underestimation of the costs.

Before we specify how we should use A* in this particular case however, it is interesting to see if we can first shrink our search tree. If we are to first flip our candidate up in the first ballot, then flip her up in the second ballot, and then flip her up in the first ballot again, this will result in the same amount of alterations as flipping her up twice in the first ballot and then flipping her up in the second ballot. We can avoid these equivalent combinations by never returning to a ballot we have already flipped.

3.2.1 A* States

We can describe any node within the search tree by means of the original ballots and a vector with length m, m being the amount of voters, with at each index a number representing the number of flips for the candidate in question.

3.2.2 A* Moves

The nodes accessible from each state are those that increase the flips of the last ballot we flipped or the next. If our current state is the original ballots and a vector like $\langle 4, 5, 3, 0, 0, 0\rangle$, then our next accessible states are the original ballots and $\langle 4, 5, 4, 0, 0, 0\rangle$, $\langle 4, 5, 3, 1, 0, 0\rangle$, $\langle 4, 5, 3, 0, 1, 0\rangle$ or $\langle 4, 5, 3, 0, 0, 1\rangle$.

3.2.3 A* Heuristics

We can imagine many different heuristics which will underestimate the cost of making our current candidate into the Condorcet winner. The most simple is to assume that no further flips are necessary, and that the estimated cost is therefore 0. This will however result in another exhaustive search, so it would be good to find a better heuristic.

3.3 An Admissible A* Heuristic for Dodgson's Procedure

In our implementation, we use Copeland's method to determine whether a candidate is a Condorcet winner. Recall that a Condorcet winner will have a Copeland score of $n-1$, where n is the amount of candidates. Since we already have access to the Copeland scores of all candidates at any given state, we can use the Copeland score to determine the candidate's most fierce competitor− id est she with the highest Copeland score who is beating our candidate. If we flip our candidate over the fiercest competitor in enough ballots to make our candidate win a pairwise competition, and we select the ballots in which we are closest to the competitor, then we are at least underestimating the amount of flips we need to make, for we will need to win in a pairwise competition anyway, and while beating our fiercest competitor it is likely we might beat other candidates as well. This heuristic can then be plugged into our A* implementation.

We have not yet implemented this particular heuristic for Dodgson's procedure, but it is admissible and would be very interesting future work.

Chapter 4

Automatic Voter Generation

For most historic elections we do not have access to the voters' true preferences. Especially in the case of plurality voting, we only have their most preferred candidate. Because of this lack of data it becomes interesting to automatically generate pools of voters, which can be used in experiments and simulated elections. The parameters which could be used when generating a pool of voters are discussed in this chapter. These include how the ranking is chosen, and especially what probability any given ranking should be assigned, as well as how sets of approved candidates could be determined.

4.1 Impartial Culture Condition

For real life voters, not all rankings of candidates are equally likely. It is for instance less likely for a voter who has a progressive liberal as her first choice to list a conservative right wing candidate as her second. When automatically generating voter preferences for a candidate pool which may be as distinctive as C_1, C_2 etc, how to simulate these preferences is not so clear.

The *impartial culture condition* stipulates that any possible permutation of candidates as a ranking is as likely as any other [19]. This means that any candidate, no matter how left or right winged, is as likely to follow another candidate in a ranking as any other. It is an unbiased way of generating voter sets which has been widely adapted, and with absence of real world data one of the better ways to go [22].

The impartial culture condition has however been highly criticised as a way of representing pools of voters [26]. There are obvious disparities with how real life voters would behave, and Tsetlin, Regenwetter and Grofman assert that in a society adhering to the impartial culture condition is the most likely to create the Condorcet paradox. When a society deviates from the impartial culture condition, the probability of a Condorcet paradox will always decrease [26]. According to Tsetlin et al, this changes the way in which we should be considering voting procedures, especially ones designed to resolve the lack of a Condorcet winner, for the Condorcet winner will hardly ever be absent in an empirically observed election.

4.2 Single Peaked Preferences

If we do not want to adhere to the impartial culture condition, it would be best to have some other formal definition which restrains the preference relations that voters can express over candidates. One of the most famous ones proposed is known as the *single peaked preference* relation [4].

To be able to have a single peaked preference relation, the candidates are defined in a one-dimensional space and therefore ordered along an imaginary line. An example of how this could be considered is the manner in which politicians are situated from the political left to the political right. A voter would then have a particular most preferred point on this space, and prefer the candidate who is closest to this point the most. As other candidates are situated further away in the candidate space, they are less preferred by the voter.

More formally, a single peaked preference relation R exists with peak c_i if for a voter c_i is the most preferred candidate and for any other $c_j \neq c_i$ the voter prefers any c which is situated between c_i and c_j to c_j.

Clearly the formality of the single peaked preference relation, specifically the requirement that preferences can be ordered on a one-dimensional line, dissociates it from a real voter pool. In this way, single peaked preferences suffer the same objections that the impartial culture condition does. However, the benefit of a society with only single peaked preferences is that there can be no top cycles, and therefore the Condorcet paradox need not be resolved [4].

4.3 Other Outcome Spaces

Besides allowing the candidates (or other issues which may be put to a vote) to be distributed over a one-dimensional space, we could imagine that the candidates could be distributed over a multidimensional space, with each issue or characteristic of the candidates mapped out in their own dimension. The voters are also distributed in this multidimensional space and most prefer the candidates located closest to them. This *spatial model* was first introduced by Downs in 1957 [11], and has since been widely adopted as a general method for voter generation [19, 9, 2].

Both the voters and candidates are distributed in this multidimensional space, but it can still be difficult to see which candidate is most suitable for the voter. To find the best candidate, the voter should look for the candidate who has the best summed utility, where the utility is defined by the Euclidean distance from the voter to the candidate and decreases linearly as the candidate is placed further away. A ranking is found by listing the candidates in decreasing order of utility.

4.4 Generating Approval Sets

If we are not so much interested in rankings of the candidates by the voters, but want to generate sets of approved candidates, we might need to go about the voter generation problem in a different manner. One way we could generate approval sets is to take the rankings which exist under the impartial culture condition and randomly insert a cut-off point somewhere after the first and before the last candidate. Any candidate before the cut-off point is then approved. It is also still possible to use a spatial distribution of voters and candidates to determine the utility per candidate, and select a cut-off amount of utility that a candidate needs to capture to be approved.

The first method however is more likely to generate approval sets which are either very short or very long. Under the impartial culture condition, a candidate c will have a probabilty of $\frac{1}{n}$ to be first and there is a probabilty of $\frac{1}{n-1}$ that a cut-off point will be selected after the first candidate. Two candidates will have a probability of $(2 \times \frac{1}{n} \times \frac{1}{n-1})/(n-1)$ of being first, and still have a probability of $\frac{1}{n-1}$ of being approved. As the number of approved candidates approaches $\frac{1}{2}n$, this probability will decrease, and then again increase as the number of approved candidates approaches n. This is perhaps not ideal.

The second method depends entirely on how we select the cut-off utility amount which we should use to accept the candidates. In his 1984 paper, Merrill suggests accepting all candidates who are above a voter's average utility for all the candidates [19]. In this case, a voter would on average accept approximately half of the candidates. This seems to be an even less likely scenario.

Ideally, we should assign all proper subsets of the set of candidates the same probability. However, that requires a completely different type of voter generation than that which we will use for voters who will participate in elections with voting rules which take an ordinal ranking, making it unattractive for comparing different voting systems. For this reason, we will use the first method for our voter generation.

4.5 Voter Estimation

One of the most desirable data sources would still be actual voters' preferences regarding a past election. In the case of the US presidential election in 2000, we have assumed that the preferences of the voters who voted for Nader include Gore before Bush in their choices. If we had access to the true preferences of the voters, we could truly analyse how often governmental plurality elections have elected candidates who would not have won under other voting procedures. Perhaps we are now wrong about the preferences we have assumed the Nader voters have. Acquiring full ranking data from the voters would however be a logistically very involved task, and it is still difficult to say whether a voter could provide an informed ranking of candidates in a governmental election. There is a high chance that the rankings provided would be more or less arbitrary. If a voting system were in place that used rankings, it is very likely that the voters would exhibit a lot of strategic voting behaviour.

It could still be interesting to analyse a historic election using a voter pool that is estimated using contextual data. Many citizens opposed to the current voting procedure in their government publish analyses like these to show the inadequacy of the current system. These estimated voter pools should be considered as biased data, and therefore are not well suited to our purposes.

Chapter 5

Voting Machinery

To be able to practically compare voting procedures, we have implemented several of the procedures within a comparative framework. Besides being able to read in user-specified ballots, we have also implemented the way of automatically generating ballots using the impartial culture assumption specified in the previous chapter. This way, we can automatically generate ballots for a prespecified candidate set and use these sets of ballots to run elections.

In this chapter, we will first briefly detail which voting procedures we have implemented in section 5.1. The we will explain how a user might generate a new election in section 5.2. In section 5.3 we will explain the ballot representation language we use. Finally in section 5.4 we will explain the charts generated by the elections. An overview fo the full voting machinery can be seen in figure 5.1.

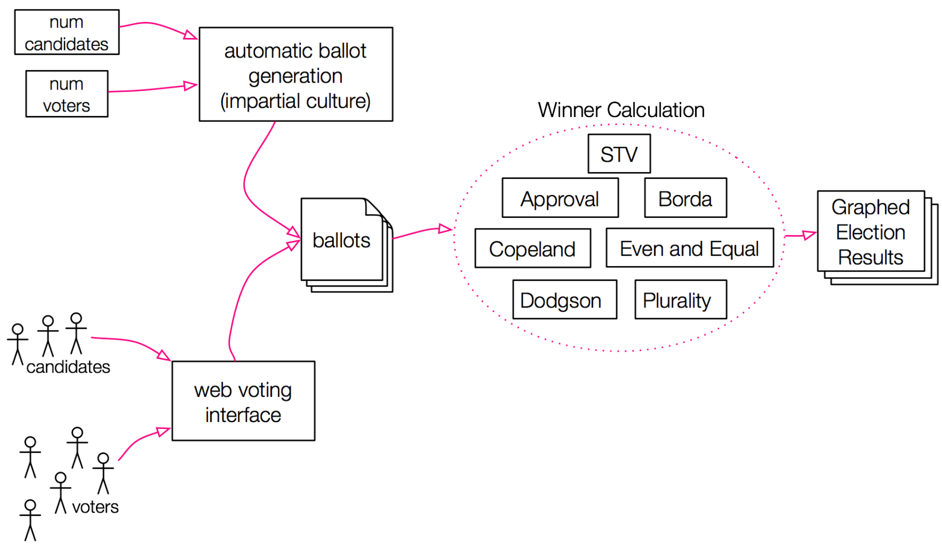

Figure 5.1:

5.1 Voting procedures implemented

Plurality

Plurality assigns one point to the first ranked candidate per voter and ignores the rest of the ballot.

Borda Count

Positional scoring rule using the scoring vector $\langle n-1, n-2, ..., 0\rangle$.

Classic Approval

Assigns one point to all candidates in the ballot before the cut-off point.

Size Approval Even and Equal

Evenly distributes one point amongst all approved candidates.

STV

Seeks a majority candidate. If none can be found, the plurality loser is removed from the ballots and a majority is sought again, this is iterated until a majority winner is found. If there is a tied plurality loser, they are removed simultaneously.

Copeland's Method

Assigns a point for each competition against another candidate won. If there is a tie against another candidate, half a point is assigned. The tie breaking amount of points can be varied between 0 and 1 as well, but the default is set to half a point.

Dodgson's Method

A Condorcet winner is sought, but if none exists the ballots are examined to find the least amount of alterations necessary to make a candidate the Condorcet winner. The candidate with the least amount of least alterations is declared the Dodgson winner.

5.2 Creating an Election

An election first and foremost specified by the participating candidates. These may be provided by the user, else the system will simply generate the number of candidates specified by the user with the names $C1, C2$ etc. The election can be run with ballots which are made through the web interface, or a specified number of ballots can be generated using the method described in the previous chapter. Part of the ballots may also be provided by the user while the other part is automatically generated.

pp

5.3 Ballot Representation

A ballot contains a ranking of the candidates and a cut-off point. It is saved as one line in a file of ballots, where the first candidate mentioned is the most popular, and all candidates before the `%` are approved. An example would be `C1,C0,%,C2`.

5.4 Representing Outcomes

Finally, when the election is run the outcomes of the different procedures are displayed on another web page. After the winners have been calculated, the user may still add more ballots and recalculate the outcomes, in which case the results will automatically adapt.

Plurality, Borda count, approval and even and equal are represented by pie charts depicting the distribution of points. Copeland's method is displayed in a bar chart showing the Copeland points. STV is a line chart with as the vertical axis the number of votes, and on the horizontal axis each round is displayed. See figure 5.2.

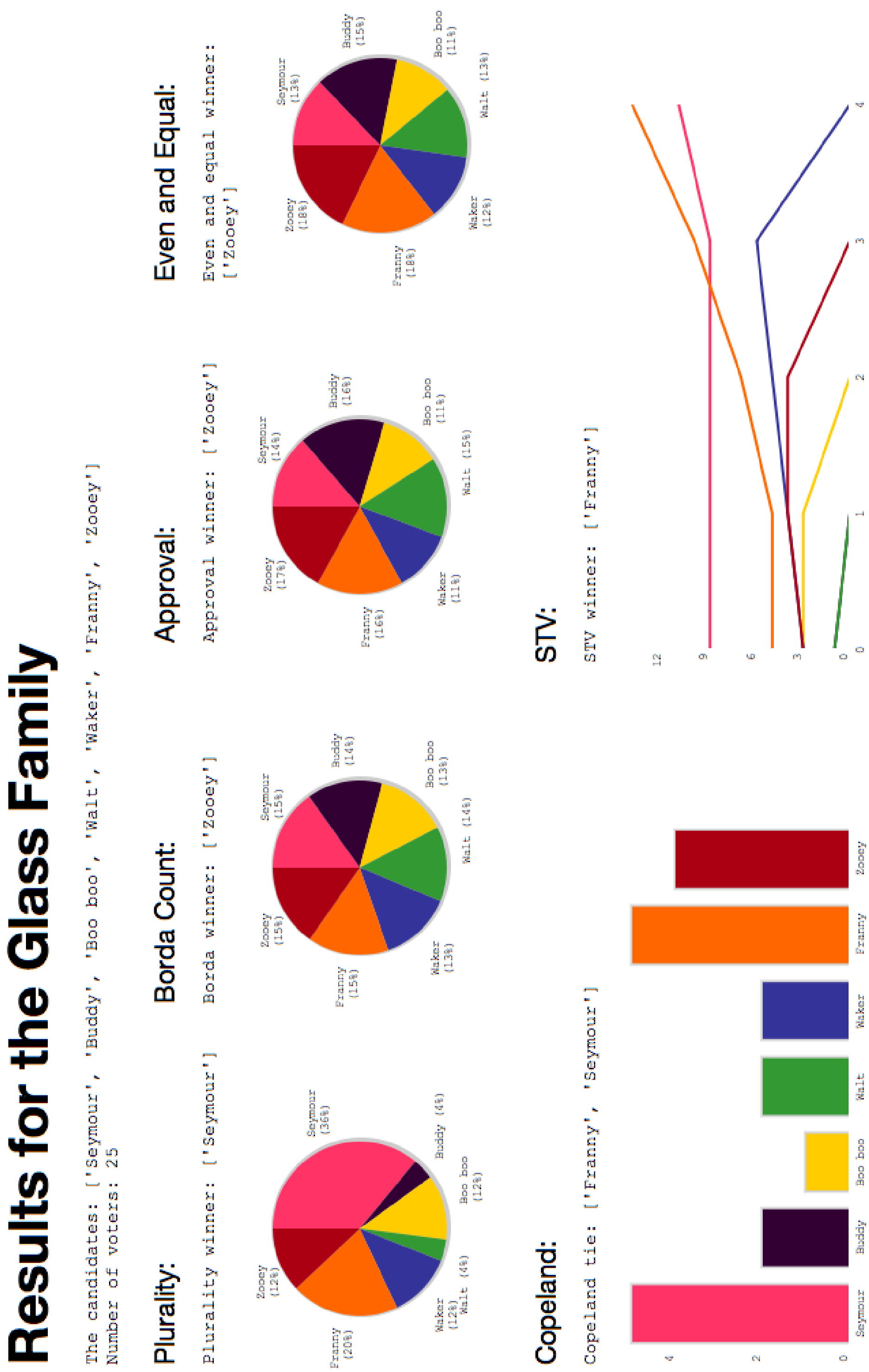

Figure 5.2: Example results web page for the most popular Glass family member

Chapter 6

Exhaustive search for smallest election instances for which voting procedures elect different candidates

It can be difficult to come up with a short example for which different voting rules will elect different winners. If we would exhaustively search through all possible elections, we could find the smallest instances for which this is the case. This provides a useful way for generating examples.

To do this, we loop through all possible distributions of a number of voters over the amount of voter types there are to find the smallest elections where a certain procedure would elect a different candidate than another voting procedure. We did this for both pairwise comparisons of voting rules and for threeway comparisons of voting rules.

6.1 Generating all possible election instances

When there are n candidates, there are $n!$ possible rankings a voters could submit of those candidates and $(n-1)n!$ possible different ballots if one would also allow the specification of a cut-off point for approval voting. We will refer to all these possible ballots as the *voter types*. For three candidates and no approval voting, this is for instance:

$$
\begin{gathered}
a > b > c \\
a > c > b \\
b > c > a \\
b > a > c \\
c > a > b \\
c > b > a
\end{gathered}
$$

As mentioned in chapter 4, it is not necessarily the case that all voter types are equally likely as all others, as this would require all candidates to be uniformly different from each other. This is irrelevant for this experiment.

The smallest possible election would start with two candidates and two voters. In this case there are only 2 possible voter types $(2-1)2!$, so there are 3 different distributions possible of the voters over these voter types, namely either both voters prefer candidate a over candidate b, or both voters prefer candidate b over candidate a or one voter prefers candidate a while the other prefers candidate b. As the number of voters and later the number of candidates increases, the amount of possible distributions over the voter types also increases. Specifically, the number of ways that one may distribute a number m of voters over a number r of voter types is equal to $\binom{m+r-1}{m}$. This can be explained as follows: imagine our voters to be marbles, and the voter types to be separated by walls. Let us have a space of slots, where we can place either marbles or walls. If we would have 10 voters and 6 voter types, we would need a space of 15 slots. We can first select the ten places that we will place our voters, and then the remaining 5 will automatically be filled by walls determining the voter types. Therefore, the combinations we can make are $\binom{15}{10}$.

To be able to exhaustively generate all possible distributions of voters over the voter types, we employ the algorithm specified in GENERATE-DISTRIBUTIONS. The algorithm will generate all distributions, first generating the vectors which most right-heavy distributions.

```
GENERATE-DISTRIBUTIONS(num-voters, num-types)
if num-types = 1:
    then return [num-voters]

if num-types > 1:
    then
        set ← [ ]
        for i ∈ [0, num-voters]
            do
                subset = GENERATE-DISTRIBUTION(num-voters−i, num-types−1)
                for k ∈ subset
                    do
                        set.APPEND([i]+k)
        return set
```

For approval voting, all voters still have a randomly generated full ranking of the candidates which is then paired with a cut-off anywhere between after the first candidate and before the last. We do not generate voters who approve of all candidates or no candidates, because in an approval election, this would be similar to abstaining.

6.1.1 Pairwise Comparison

Using the distributions as they are generated by GENERATE-DISTRIBUTIONS and a increasing number of voters and candidates, we perform an exhaustive search for the election instances with differing winners. In this section we have listed the comparisons between plurality, Borda count, classic approval, even and equal size approval, Copeland's method, STV and Dodgson's method.

The exact search algorithm is detailed in PAIRWISE-COMPARISON. The number of candidates is increased by one every time the number of voters reaches $7n!$, with n being the number of candidates. The number of possible rankings is $n!$, and 7 is an arbitrarily chosen constant which should be sufficient for most of the possible rations of distributions over the different voter types.

PAIRWISE-COMPARISON

```
num-voters = 2
while rule1.winner = rule2.winner
      do
         candidates = GENERATE-CANDIDATES(num-candidates)
         voter-types = PERMUTATE(candidates)
         distributions =
           GENERATE-DISTRIBUTIONS(num-voters, num-candidates!)

         for d in distributions
               do
                  ballots = [ ]
                  for j ∈ [0, num-candidates!]
                        do
                           for i ∈ [0, d[j]]
                                 do
                                    ballots.APPEND(voter-types[j])

                  ELECTION(ballots)
                  if rule1.winner ≠ rule2.winner
                     then break
         num-voters += 1
return ballots
```

6.2 Results

These results produce differing unique winners. We have not sought for differing results including ties, because these do not show the differences in voting procedures as well as the unique winners.

Comparing 2 voting rules

Plurality vs. STV

winner plurality: C_1, winner STV: C_2

$$2 \text{ voters: } C_0 > C_2 > C_1$$

$$\begin{gathered}
4 \text{ voters: } C_1 > C_2 > C_0 \\
3 \text{ voters: } C_2 > C_1 > C_0
\end{gathered}$$

Plurality vs. Borda

winner plurality: C_1, winner Borda: C_2

$$\begin{gathered}
1 \text{ voter: } C_0 > C_2 > C_1 \\
1 \text{ voter: } C_2 > C_0 > C_1 \\
2 \text{ voters: } C_1 > C_2 > C_0
\end{gathered}$$

Plurality vs. Copeland and Dodgson

winner plurality: C_1, Condorcet winner: C_2

This example is the smallest example where plurality does not elect the Condorcet winner.

$$\begin{gathered}
1 \text{ voter: } C_0 > C_2 > C_1 \\
1 \text{ voter: } C_2 > C_0 > C_1 \\
2 \text{ voters: } C_1 > C_2 > C_0
\end{gathered}$$

Borda vs. Copeland and Dodgson

winner Borda C_2, Condorcet winner C_1

This example is the smallest example where Borda count does not elect the Condorcet winner.

$$\begin{gathered}
2 \text{ voters: } C_2 > C_0 > C_1 \\
3 \text{ voters: } C_1 > C_2 > C_0
\end{gathered}$$

Borda vs. STV

winner Borda: C_2, winner STV C_1

$$\begin{gathered}
1 \text{ voter: } C_0 > C_2 > C_1 \\
1 \text{ voter: } C_2 > C_0 > C_1 \\
2 \text{ voters: } C_1 > C_2 > C_0
\end{gathered}$$

STV vs. Copeland and Dodgson

winner STV: C_1, Condorcet winner: C_2

This example is the smallest example where STV does not elect the Condorcet winner. This could be a disputed example however, because in our algorithm now both C_0 and C_2 will be eliminated simultaneously, because they have a tied last place.

$$\begin{gathered}
1 \text{ voter: } C_0 > C_2 > C_1 \\
1 \text{ voter: } C_2 > C_0 > C_1 \\
2 \text{ voters: } C_1 > C_2 > C_0
\end{gathered}$$

Approval vs. Plurality, Borda, STV, Dodgson and Copeland

winner approval: C_2, winner other procedures: C_1

An example of approval not satisfying the majority criterion, which is satisfied by plurality, Borda, STV and any Condorcet criterion satisfying method.

2 voters: $C_1 > C_2 \mid C_0$
1 voter: $C_2 \mid C_1 > C_0$

Approval vs. Even and Equal

3 voters: $C_2, C_0 \mid C_1$
2 voters: $C_1 \mid C_2, C_0$
1 voter: $C_2, C_1 \mid C_0$

Even and Equal Size vs. Plurality and STV

winner even and equal size approval: C_2, winner plurality and STV: C_1

2 voters: $C_1 > C_2 \mid C_0$
1 voter: $C_2 \mid C_1 > C_0$

Even and Equal Size vs. Dodgson, Borda and Copeland

winner even and equal size approval: C_1, winner other procedures: C_2

1 voter: $C_2 > C_0 \mid C_1$
1 voter: $C_1 \mid C_2 > C_0$

It would be very interesting to find the example in which Copeland and Dodgson, both being Condorcet criterion satisfying procedures, would elect a different candidate. As we are writing this, a computer has been trying to calculate this for the past 300 hours, and has of yet not come up with a solution.

Comparing 3 voting rules

The threeway comparison is identical to the pairwise comparison, only we are looking for 3 different unique winners instead of 2. Because of this, the examples will get a bit larger to encompass all different voter rule characteristics. Not all of the comparisons have been run, for we are limited to elections with at most 15 voters and 4 candidates with Dodgson. After that, calculating the winner becomes too complicated, so we have not found threeway comparisons for all of the rules and Dodgson.

Borda vs. STV/Plurality vs. Copeland

winner Borda: C_2, winner STV/Plurality: C_1, winner Copeland: C_0

6 voters: $C_0 > C_2 > C_1$
4 voters: $C_1 > C_0 > C_2$

4 voters: $C_2 > C_0 > C_1$
3 voters: $C_1 > C_2 > C_0$
2 voters: $C_2 > C_1 > C_0$

Plurality vs. STV. vs. Copeland/Borda

winner plurality: C_1, winner STV: C_2, winner Copeland/Borda: C_0

2 voters: $C_0 > C_2 > C_1$
3 voters: $C_1 > C_0 > C_2$
3 voters: $C_2 > C_0 > C_1$
1 voter: $C_1 > C_2 > C_0$

Approval vs. STV vs Dodgson/Copeland

winner approval: C_0, winner STV: C_1, Condorcet winner: C_2

2 voters: $C_0 > C_2 \mid C_1$
1 voter: $C_1 > C_0 \mid C_2$
2 voters: $C_2 > C_0 \mid C_1$
2 voters: $C_1 \mid C_2 > C_0$

Approval vs. Even and Equal vs. Copeland/Dodgson

winner approval: C_0, winner even and equal: C_1, Condorcet winner: C_2

1 voter: $C_1 > C_0 \mid C_2$
3 voters: $C_2 > C_0 \mid C_1$
2 voters: $C_1 \mid C_2 > C_0$

Plurality vs. Approval vs. Copeland/Dodgson

winner plurality: C_1, winner approval: C_0, Condorcet winner: C_2

1 voter: $C_0 > C_2 \mid C_1$
1 voter: $C_1 > C_0 \mid C_2$
2 voters: $C_2 > C_0 \mid C_1$
2 voters: $C_1 \mid C_2 > C_0$

Plurality vs. Approval vs. Even and Equal

winner plurality: C_1, winner approval: C_0, winner even and equal: C_2

1 voter: $C_0 > C_2 \mid C_1$
3 voters: $C_1 > C_0 \mid C_2$
2 voters: $C_2 \mid C_1 > C_0$

Approval vs. STV vs. Even and Equal

winner approval: C_0, winner STV: C_1, winner even and equal: C_2

2 voters: $C_0 > C_2 \mid C_1$
3 voters: $C_1 > C_0 \mid C_2$
2 voters: $C_2 \mid C_1 > C_0$

Borda count vs. Approval vs STV

winner Borda: C_0, winner approval: C_2, winner STV: C_1

2 voters: $C_0 > C_2 \mid C_1$
2 voters: $C_1 \mid C_0 > C_2$
1 voter: $C_2 \mid C_1 > C_0$

Borda vs. Approval vs. Copeland/Dodgson

winner borda: C_2, winner approval: C_0 Condorcet winner: C_1

2 voters: $C_1 > C_0 \mid C_2$
4 voters: $C_2 > C_0 \mid C_1$
2 voters: $C_1 \mid C_2 > C_0$
1 voter: $C_1 > C_2 \mid C_0$

Borda count vs. Approval vs. Even and equal

winner Borda: C_2, winner approval: C_0, winner even and equal: C_1

1 voter: $C_1 > C_0 \mid C_2$
3 voters: $C_2 > C_0 \mid C_1$
2 voters: $C_1 \mid C_2 > C_0$

Borda vs. Dodgson vs. Even and equal

winner Borda: C_2, winner Dodgson: C_1, winner even and equal: C_0

1 voter: $C_1 > C_0 \mid C_2$
5 voters: $C_2 > C_0 \mid C_1$
1 voter: $C_1 \mid C_2 > C_0$
2 voters: $C_1 > C_2 \mid C_0$

Approval vs. Copeland vs. Even and Equal

winner approval: C_0, winner Copeland: C_2, winner even and equal: C_1

1 voter: $C_1 > C_0 \mid C_2$
3 voters: $C_2 > C_0 \mid C_1$
2 voters: $C_1 \mid C_2 > C_0$

Borda count vs. Approval vs. Even and equal

winner Borda: C_2, winner approval: C_0, winner even and equal C_1

1 voter: $C_1 > C_0 \mid C_2$
3 voters: $C_2 > C_0 \mid C_1$
2 voters: $C_1 \mid C_2 > C_0$

Plurality vs Borda count vs. Approval

winner plurality: C_0, winner Borda: C_2, winner approval: C_1

1 voter: $C_0 > C_1 \mid C_2$
2 voters: $C_0 \mid C_2 > C_1$
1 voter: $C_1 > C_2 \mid C_0$
2 voters: $C_2 > C_1 \mid C_0$

Plurality vs. Approval vs. STV

winner plurality: C_1, winner approval: C_0, winner STV: C_2

2 voters: $C_0 > C_2 \mid C_1$
1 voter: $C_1 > C_0 \mid C_2$
3 voters: $C_2 > C_0 \mid C_1$
3 voters: $C_1 \mid C_2 > C_0$

STV/Plurality vs. Dodgson/Copeland vs. Even and equal

winner plurality/STV: C_2, Condorcet winner: C_0, even and equal winner: C_1

1 voter: $C_0 > C_2 \mid C_1$
2 voters: $C_1 \mid C_0 > C_2$
2 voters: $C_2 > C_0 \mid C_1$

Plurality vs. STV. vs. Even and equal

winner plurality: C_1, winner STV: C_2, winner even and equal: C_0

2 voters: $C_0 > C_2 \mid C_1$
2 voters: $C_1 > C_0 \mid C_2$
3 voters: $C_2 > C_0 \mid C_1$
1 voter: $C_1 \mid C_2 > C_0$
1 voter: $C_1 > C_2 \mid C_0$

Plurality vs. Copeland vs. Even and Equal

winner plurality: C_1, winner Copeland: C_0, winner even and equal: C_2

1 voter: $C_0 > C_2 \mid C_1$
3 voters: $C_1 > C_0 \mid C_2$
2 voters: $C_2 \mid C_0 > C_1$

Borda count vs. Even and Equal vs. Copeland

winner Borda: C_2, winner even and equal : C_0, Copeland winner: C_1

3 voters: $C_1 > C_0 \mid C_2$
5 voters: $C_2 > C_0 \mid C_1$
1 voter: $C_1 \mid C_2 > C_0$
2 voters: $C_1 > C_2 \mid C_0$

STV vs. Copeland vs. Even and equal

winner STV: C_2, winner Copeland: C_0, winner Even and equal: C_1

1 voter: $C_0 > C_2 \mid C_1$
2 voters: $C_1 \mid C_0 > C_2$
2 voters: $C_2 > C_0 \mid C_1$

It is more difficult to inspect the threeway comparison results, because more characteristics are at play. We can see that often there is somewhat of an embedding of the smaller pairwise result within the larger threeway results, but this is not surprising.

Chapter 7

Statistical analysis of characteristics of voting procedures

Although we have many theoretical results on whether voting rules satisfy different criteria, we don't know how often it is statistically the case that the rule will not be satisfied. For this reason, we would like to perform some statistical analysis on how often various criteria are satisfied and how often different voting rules agree or disagree with each other.

7.1 Probability of the Existence of a Condorcet Winner and the Existence of a Condorcet Loser

As we have shown in section 2.2.4, a Condorcet winner does not necessarily always exist. However, how often can we expect a Condorcet winner to exist given a certain amount of voters m and a certain amount of candidates n? This has already to some extent been theoretically analysed by Gehrlein [12], but we seek statistical confirmation.

First of all we will inspect how the number of voters influences the number of Condorcet winners and losers under the impartial culture condition. Next we will inspect how the number of candidates influences the number of Condorcet winners and losers under the impartial culture condition. Note that because each ranking is equally likely to occur as its exact opposite, the likelihood of the Condorcet winner is mirrored by the likelihood of the Condorcet loser.

In figure 7.1 we see how often a Condorcet winner or loser exists as we increase the number of voters. The graph shows the average existence based on 100 000 elections each for 4 candidates and between 10 and 100 voters. The graph shows how increasing the number of voters also increases the percentage of existing

Condorcet losers or winners. After a certain number of voters however, the percentage of existing Condorcet losers and winners stops increasing. For 4 candidates, this occurs at approximately 70%. We reason that this is due to there being a finite number of voter types, and after a certain amount of voters, the proportions within the pool of different voter types become steady, and increasing the number of voters only creates a multiple of an election with a smaller voter pool. A voting method which only elects the Condorcet winner is consistent, and therefore multiples will produce the same (amount of) Condorcet winners. This result has also been confirmed by results not reported here with the same experiment set up but a differing amount of candidates. A steady percentage of Condorcet winners and losers is approached as the amount of voters increases.

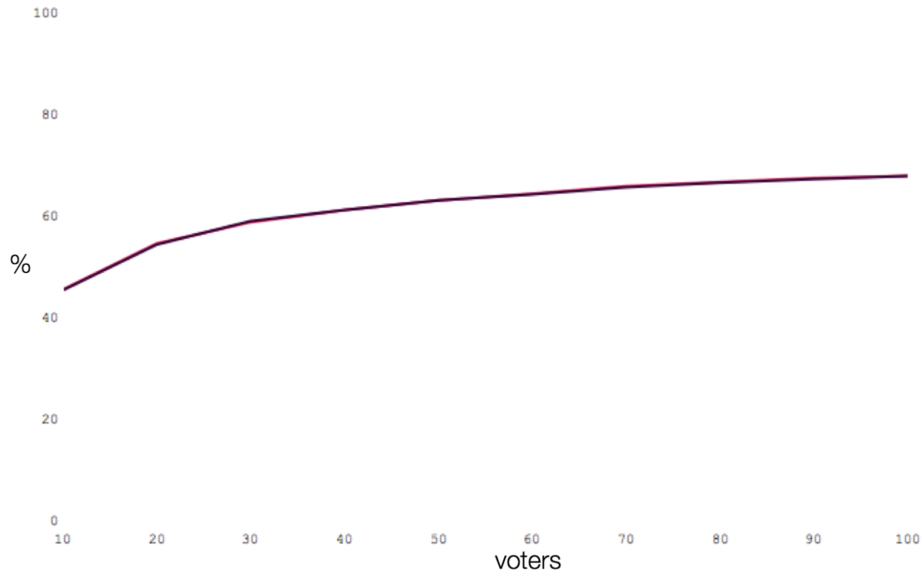

Figure 7.1: How the number of voters influences the percentage of Condorcet winners. These are averages of 100 000 elections with 4 candidates.

If we keep the amount of voters fixed at 50 but instead we vary the amount of candidates, we also observe trends in the increase and decrease of the amounts of Condorcet losers and winners that exist. Specifically, the amount elections with Condorcet winners and losers decreases as we increase the number of candidates. Unlike with increasing the number of voters, increasing the number of candidates does not cause the existence of Condorcet winners or losers to approach a limit, but continuously decreases. The rate of decrease can be observed in figure 7.2.

It is also logical that the existence of a Condorcet winner or loser should become less likely as we increase the amount of candidates under the impartial culture condition, because it becomes less likely that one candidate will beat all other candidates in pairwise competition. There are more candidates to compete with, and winning from them all becomes more difficult. This result was also found by Tsetlin et al, who used it as a critique against the impartial culture condition

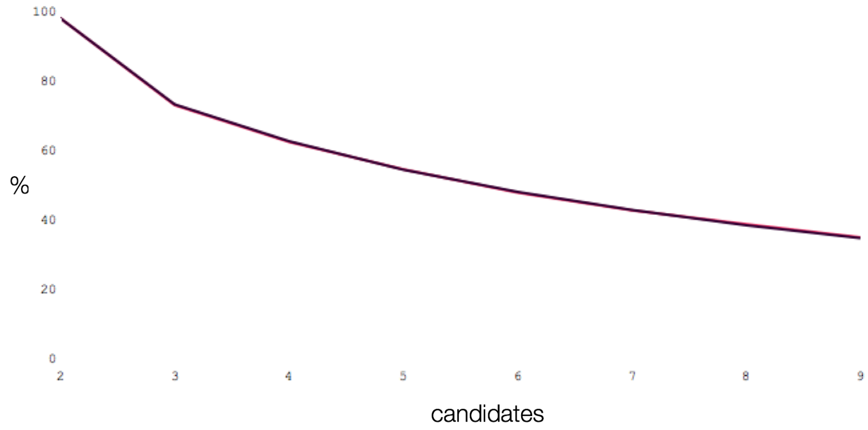

Figure 7.2: How the number of candidates influences the percentage of Condorcet winners. These are averages of 100 000 elections with 50 voters.

for voter generation [26].

7.2 Probability of Not Electing a Cordorcet Winner

The next experiment we ran involved the inspection of rules which do not satisfy the Condorcet criterion. Even though there are cases in which they will not elect the Condorcet winner, this only happens some of the time. Here we inspect exactly how often rules do not elect the Condorcet winner given the Condorcet winner exists.

In figure 7.3 we can observe the increase in percentage of not electing the Condorcet winner. This graph has been generated from table 7.1, which shows the amount of times a Condorcet winner *does* get elected.

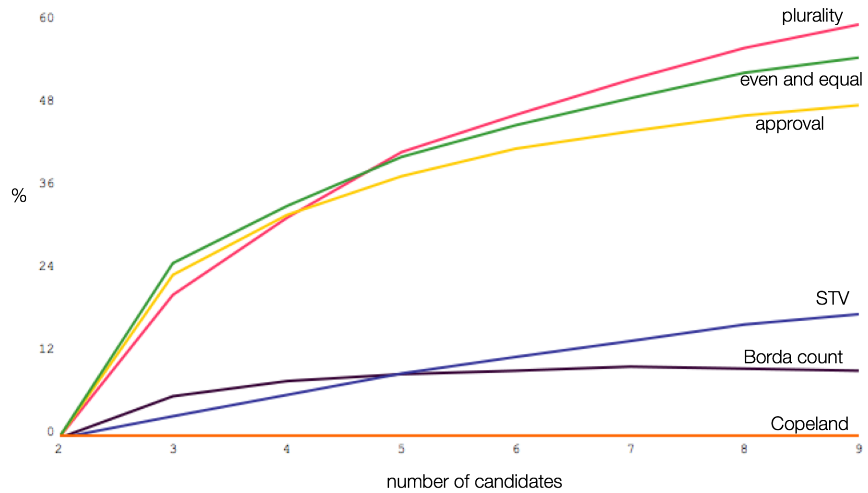

Figure 7.3: How often each rule does not elect the Condorcet winner. Average of 100 000 elections.

We already know that Copeland will always elect the Condorcet winner when one exists, and this is reflected in the graph. That approval and even and equal very often do not elect the Condorcet winner is not that surprising, as they do not take the ranking information which is crucial to the determination of the Condorcet winner into account. Similarly, although plurality does look at the best ranked candidate, it ignores all other ballot information and therefore predictably does very poorly as well.

The overall increase in the non-election of Condorcet winners can be explained by the increased amount of candidates that a voting system might elect. As there are more candidates, any candidate a priori already has less chances of being elected. If we were to assume a random voting method, which picks any candidate from the candidates as a winner, we would also observe an increase in the amount of times a Condorcet winner is not elected. Voting methods such as Copeland are specifically designed to deal with this trend, however, rules like plurality are not.

no. cands	2	σ	3	σ	4	σ	5	σ
Plurality	100.0	0.0	79.19	4.79	68.09	5.69	59.55	6.46
Borda	100.0	0.0	94.02	2.93	91.69	3.31	90.72	3.89
Approval	100.0	0.0	76.32	4.95	67.64	5.78	62.06	6.70
Even	100.0	0.0	74.57	4.90	66.31	5.72	59.23	6.56
STV	100.0	0.0	96.80	2.04	93.73	2.97	90.62	3.98
no. cands	6	σ	7	σ	8	σ	9	σ
Plurality	53.16	7.10	48.09	7.60	43.45	7.65	39.95	7.92
Borda	90.20	4.21	89.69	4.54	89.944	4.49	90.26	5.16
Approval	58.04	6.91	55.52	7.69	53.20	8.00	51.73	8.04
Even	54.62	6.95	50.67	7.82	47.03	8.06	44.78	8.18
STV	88.23	4.54	85.97	5.40	83.56	5.66	82.05	6.13

Table 7.1: How often do Condorcet winners get elected? In percent, average of 100 000 elections with 50 voters.

Similar results have been obtained by Samuel Merrill III [19]. Under the impartial culture condition he ran 10 000 elections with 201 voters and 5 candidates, and determined that under these circumstances plurality would have a Condorcet efficiency of 60%, STV one of 88% and Borda one of 85% [19]. Here we found 59.55%, 90.62% and 90.72% respectively. He also determined the Condorcet efficiency of approval voting to be 67%, but this is done with a voter generation which differs from ours, which might explain the discrepancy from our 62%.

7.3 Probability of Electing a Condorcet Loser

Another interesting criterion to analyse is the Condorcet loser criterion. Recall that the Condorcet loser criterion requires the candidate who loses in pairwise comparison to all other candidates must never be elected. Given that a rule does not satisfy the Condorcet loser criterion, how often is it the case that the Condorcet loser is elected by a given rule? Again we ran 100 000 elections with a varying amount of candidates to determine how well the different rules do. We collected the percentages of times that the rules would elect the Condorcet winner when one existed. The results can be observed in figure 7.4, a graph generated from the data in table 7.2.

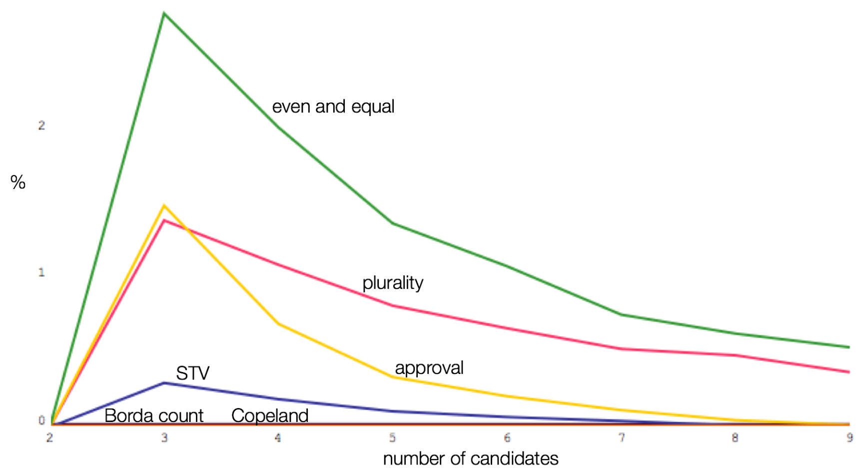

Figure 7.4: How often each rule elects the Condorcet Loser. Average of 100 000 elections.

There is a very pronounced peak for most voting procedures at 3 candidates. For two candidates it will never be the case for any voting rule that the Condorcet loser is elected, because all voting rules are then equal to plurality. Therefore it is unsurprising there is a large difference between 2 and 3 candidates. After the first 3 candidates, the amount of Condorcet losers elected decreases for all procedures. Again, there is a larger a priori chance that any candidate would be elected if there are less candidates, which explains why there is a decrease in the election of Condorcet losers.

It has already been proven that Borda count will never elect the Condorcet loser [22], and this is reflected in our graph. Approval voting decreases faster than plurality and even and equal, which can be explained by the fact that second, third, etc. ranked candidates still receive a large amount of points when they are approved in comparison to even and equal and plurality, which not only promotes them, but also helps in distinguishing from the last ranked candidates.

no. cands	2	σ	3	σ	4	σ	5	σ
Plurality	0.00	0.00	1.40	1.39	1.10	1.31	0.82	1.21
Borda	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Approval	0.00	0.00	1.53	1.42	0.72	1.06	0.34	0.80
Even	0.00	0.00	2.80	1.92	2.03	1.75	1.38	1.58
STV	0.00	0.00	0.32	0.68	0.19	0.56	0.11	0.43
no. cands	6	σ	7	σ	8	σ	9	σ
Plurality	0.67	1.11	0.53	1.12	0.49	1.11	0.37	1.03
Borda	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Approval	0.21	0.63	0.12	0.50	0.05	0.36	0.02	0.21
Even	1.09	1.47	0.76	1.31	0.63	1.26	0.54	1.21
STV	0.07	0.38	0.04	0.31	0.011	0.17	0.01	0.17

Table 7.2: How often are Condorcet losers elected? In percent, average of 100 000 elections and 50 voters.

7.4 How often do voting rules elect the same candidates?

For groups of randomly generated candidates and voters adhering to the impartial culture condition, we calculated the percentage of times that the voting rules would agree with each other. In all cases, we generated 10000 elections with 50 voters, with respectively 3, 4 and 5 candidates. With 2 candidates the voting rules all agree 100% of the time because then they all mirror plurality, but as the number of candidates increases this number diverges.

	Plurality	Borda	Approval	Even	STV	Copeland
Plurality	100.0	70.5	59.6	63.72	70.96	66.23
Borda	70.5	100.0	69.34	67.4	75.66	81.32
Approval	59.6	69.34	100.0	79.85	62.65	64.0
Even	63.72	67.4	79.85	100.0	62.56	62.52
STV	70.96	75.66	62.65	62.56	100.0	85.62
Copeland	66.23	81.32	64.0	62.52	85.62	100.0

Table 7.3: Average of 10000 elections with 3 candidates and 50 voters

	Plurality	Borda	Approval	Even	STV	Copeland
Plurality	100.0	57.61	45.58	52.06	59.13	53.57
Borda	57.61	100.0	60.22	58.54	67.7	74.99
Approval	45.58	60.22	100.0	71.35	51.05	53.53
Even	52.06	58.54	71.35	100.0	52.39	52.31
STV	59.13	67.7	51.05	52.39	100.0	74.91
Copeland	53.57	74.99	53.53	52.31	74.91	100.0

Table 7.4: Average of 10000 elections with 4 candidates and 50 voters

	Plurality	Borda	Approval	Even	STV	Copeland
Plurality	100.0	49.06	36.3	43.54	49.83	44.7
Borda	49.06	100.0	52.98	50.52	61.64	71.83
Approval	36.3	52.98	100.0	63.06	43.08	46.58
Even	43.54	50.52	63.06	100.0	43.88	44.47
STV	49.83	61.64	43.08	43.88	100.0	67.6
Copeland	44.7	71.83	46.58	44.47	67.6	100.0

Table 7.5: Average of 10000 elections with 5 candidates and 50 voters

7.5 From Plurality to Borda to Antiplurality

Recall that a positional scoring vector assigns points to candidates depending on where they are in the ranking. Plurality: $\langle 1, 0, 0 \rangle$, Borda count: $\langle 1, \frac{1}{2}, 0 \rangle$, antiplurality: $\langle 1, 1, 0 \rangle$.

For 3 candidates this could be generalized as: $\langle 1, \lambda, 0 \rangle$ for $\lambda \in [0, 1]$, where all the scoring vectors we create by varying λ represent different voting procedures. We know that Borda count does not elect the Condorcet loser [22], and we know that plurality does. When does this change? When we examine these unnamed voting rules, how often do they elect the Condorcet winner and loser?

λ	Condorcet winners not elected	σ
0.0	20.7569710672	4.69382993521
0.1	13.1364955705	3.88969320457
0.2	10.2866459258	3.50170350242
0.3	6.7884043218	2.90441992668
0.4	4.9678354044	2.50772674755
0.5	5.8773270604	2.73039370271
0.6	6.2252217648	2.82781965571
0.7	9.0160946185	3.32580327478
0.8	13.7318585388	3.99138476435
0.9	17.5088685503	4.43500579043
1.0	26.0178407320	5.11604112681

Table 7.6: Average of 1 000 000 elections with 3 candidates and 50 voters.

In both tables we can see that Borda count does the best in electing Condorcet winners and not electing Condorcet losers, and the results monotonically worsen as we increase or decrease the λ from 0.5. These results more or less match the results found by Gehrlein in his 1996 paper, where he ran the same experiment but does not mention with how many voters or how many elections he averaged [12]. He also varies λ from 0 to 0.5, and we vary λ from 0 to 1 in step sizes of 0.1.

λ	Condorcet losers elected	σ
0.0	1.4145844893	1.36843430073
0.1	1.2165588940	1.26835825673
0.2	0.5421445603	0.84599760421
0.3	0.0681647892	0.29890863742
0.4	0.0008184719	0.03344556222
0.5	0.0000000000	0.00000000000
0.6	0.0009523477	0.03599949849
0.7	0.0938891876	0.35478874761
0.8	0.5935236318	0.88103790651
0.9	1.3747336139	1.33509456372
1.0	1.7763837257	1.52177847947

Table 7.7: Average of 1 000 000 elections with 3 candidates and 50 voters.

7.6 From classic approval to even and equal

Recall size approval voting, where the amount of points is dependent on the amount of approved candidates. All size approval voting rules can be defined by a weight vector. If k candidates are approved, they all receive the amount of points specified in the kth position of the weight vector. For classic approval the weight vector is homogenous: $\langle 1, 1, 1\rangle$, and for even and equal it decreases as $\frac{1}{k}$ decreases. For three candidates, the weight vector for even and equal becomes $\langle 1, \frac{1}{2}, \frac{1}{3}\rangle$.

For 3 candidates, the weight vector could be generalized as: $\langle 1, \lambda\rangle$ for $\lambda \in [0, 1]$. Since at most $n - 1$ candidates can be approved, n being the total number of candidates, the weight vector also only needs to be $n - 1$ long.

There are now many unnamed voting rules between classic approval and even and equal size approval voting for 3 candidates. When we examine these unnamed voting rules, how often do they elect the Condorcet winner and loser?

λ	Condorcet winners not elected	σ
0.0	41.4607972575	5.66561683719
0.1	32.4272765144	5.39781829875
0.2	29.8731661125	5.29664273181
0.3	26.9514371539	5.12369231705
0.4	24.8074485792	4.94941138371
0.5	25.0430820551	4.97224795127
0.6	21.6506186499	4.73660252182
0.7	20.5394122957	4.63764501562
0.8	20.0271627413	4.58710523893
0.9	19.6000146242	4.55179231853
1.0	23.2970908415	4.86697819586

Table 7.8: Average of 1 000 000 elections with 3 candidates and 50 voters.

$\lambda = 0.0$ is of course a bit of an odd case: if more than 1 candidate is approved, no points are assigned. This translates to half of the voters really abstaining from the election, for their votes have no effect. The other half of the voters voting according to the plurality rule. Both $\lambda = 1$ and $\lambda = 0.5$ agree with previous results we obtained when running experiments with named procedures, but $\lambda = 0$ is incomparable to the effects of half the voters abstaining.

λ	Condorcet losers elected	σ
0.0	7.80094044005	3.08511452967
0.1	8.13535203584	3.16173858730
0.2	6.50226128269	2.86136649503
0.3	5.10938882109	2.54174903336
0.4	3.85881210210	2.23562767399
0.5	2.76132073023	1.88962461990
0.6	2.75264332602	1.88507580978
0.7	2.45252118512	1.78778627975
0.8	2.25656247742	1.71562194977
0.9	2.26258875927	1.72161512372
1.0	1.57445306300	1.43468100974

Table 7.9: Average of 1 000 000 elections with 3 candidates and 50 voters.

In the previous experiments we named procedures we also noticed that even and equal performs more poorly than approval voting in electing the Condorcet loser. With this experiment we observe that this is a trend that increases as we decrease λ. As λ decreases, the amount of Condorcet losers elected increases. We still think this is due 2nd and 3rd ranked candidates receiving less points, and therefore not distinguishing themselves significantly from the last ranked candidates.

Chapter 8

Discussion

Some of the experiments we ran brought up certain issues and results that we found surprising or interesting enough to bring up separately.

8.1 Approving of all candidates

In our implementation, we do not generate voters with a cut-off point which is before the first or after the last candidate. Approving of all candidates or approving of no candidates does not change the outcome of the approval or even and equal election. However, when comparing the outcomes of different election systems, approving of all candidates can cause the outcome to differ.

Our smallest election for which Borda count and approval voting differ is as follows:

$$\text{2 voters: } C_1 > C_2 \mid C_0$$
$$\text{1 voter: } C_2 \mid C_1 > C_0$$

However, one might notice that the same outcome could be achieved if we removed candidate C_0 entirely, and thus would provide us with an election instance with only two candidates.

$$\text{2 voters: } C_1 > C_2 \mid$$
$$\text{1 voter: } C_2 \mid C_1$$

If we were to allow approval of all candidates, or really abstention in our search for the smallest elections, we might find even smaller election instances. However, it is a theoretical situation which does not occur practically. For this reason, we find our original results more interesting.

8.2 Selection of candidate to eliminate in Hare's method

One of the things which was surprising in the Condorcet loser election experiments (see section 7.3) is that STV also elects the Condorcet loser at times, while STV is commonly assumed to satisfy the Condorcet loser criterion. After inspection of the elections where this was the case, we found that it was due to our implementation of STV: if there is a tied last place in plurality, we remove both last placed candidates simultaneously. However, when doing this we might also cause the Condorcet loser to be elected.

Imagine the following election between candidates A, B and C:

2 voters: $A > B > C$
2 voters: $B > A > C$
3 voters: $C > A > B$

Because A and B are both the least first ranked of the candidates, we remove them both. However, the only candidate who then remains is C, who is actually the Condorcet loser.

Coombs' method avoids this problem by selecting the antiplurality loser, and does satisfy the Condorcet loser criterion.

Hare has not formally determined what to do in case of a tie for plurality loser, and neither have other sources on how STV is used. This might be considered odd, because STV is used for many governmental elections, but the case described above would only happen for very small amounts of voters and therefore would hardly ever arise in real life.

Chapter 9

Conclusion and Future Work

We have examined many different voting procedures. First we examined their definitions and characteristics they are theoretically known to possess. Then we examined how we can make the smallest election instances where different voting procedures elect different candidates. To be able to do this, we implemented a comparative framework which can take the same ballots and run elections with many different voting procedures.

Using our comparative framework, we also ran statistical experiments which would give an indication of how often certain rules fail certain criteria if they are already theoretically known to at times do so. We checked how often different non-Condorcet criterion satisfying voting procedures would not elect the Condorcet winner when one exists. Similarly, we checked how often non-Condorcet loser criterion satisfying rules would actually elect the Condorcet loser. We found that this happens very infrequently, even for notoriously deviant rules such as approval voting.

We examined unnamed positional scoring voting procedures for 3 candidate elections, and found that as we deviate from Borda count towards plurality or antiplurality, the amount of times the Condorcet winner is not elected or the Condorcet loser is elected increases monotonically. We examined unnamed approval voting procedures for 3 candidate elections as well, and found that for size approval voting, assigning multiple approved candidates less points than single approved candidates causes it to be more likely that the Condorcet loser is elected as the amount of points decreases.

We compared how often different voting procedures would elect the same winners, and found that as we increase the number of participating candidates all voting rules diverge more in electing different candidates. This divergence is very sharp at the beginning, with very large drops from 2 to 3 candidates and from 3 to 4, but this the decrease slows and may approach some limit. It would be interesting to continue work with this experiment to determine exactly what this limit is.

For the smallest election instances, we increased the amount of voters before increasing the amount of candidates participating. It is possible that we could have found an example with 4 candidates and 2 voters where our examples have 3 candidates and 3 voters. It would be interesting to continue finding these instances and comparing them to each other.

We were unable to find the smallest election instance where Copeland would elect a different candidate than Dodgson. To be able to do this and also run other larger elections with Dodgson, we would like to implement the heuristic proposed for finding the Dodgson winner in the future.

Many of the results we found had 3 or more variables to display, we would be interested in finding a better way to display the data and perhaps be able to manipulate which parameters are being shown in real time, to make the data more easy to understand.

Finally, we would like to implement more voting procedures in our comparative framework as well as different ways of automatic voter generation, to be able to run experiments with these as well.

Acknowledgements

This work was done as my bachelor's thesis in AI at the University of Amsterdam. It was done in the context of the AI honours programme.

I would like to thank Ulle Endriss for being absolutely the best possible teacher and thesis advisor− from introducing me to the topic to poring over buggy results. I would also like to thank him for coordinating the honours programme and supervising my first research project, which is really the main reason I left art school for AI.

I would also like to thank Marco Wessel for the constant influx of consumptables without which I would have long since withered away, and allowing me to use his servers for experiment running and repository hosting. I think he might use the word *awesometastic*.

Finally I would like to thank fellow honours students Gijs Kruitbosch and Hylke Buisman for giving random advice at random hours, and also all my classmates for all their help and coffee donations.

The poem at the beginning is taken entirely without permission from Damien Hirst's latest book *Superstition*.

Bibliography

[1] Jorge Alcalde-Unzu and Marc Vorsatz, *Size approval voting*, Research Memorandum 008, Maastricht : METEOR, Maastricht Research School of Economics of Technology and Organization, 2007.

[2] John H. Aldrich, *Before the convention*, University of Chicago Press, Chicago, 1980.

[3] Kenneth Arrow, *A difficulty in the concept of social welfare*, Journal of Political Economy **58** (1950), no. 4, 328–346.

[4] Duncan Black, *The theory of committees and elections*, Cambridge University Press, 1958.

[5] Steven J. Brams and Peter C. Fishburn, *Approval voting*, The American Political Science Review **72** (1978), no. 3, 831–847.

[6] US Federal Election Commission, *2000 presidential electoral and popular vote*, 2000, http://www.fec.gov/pubrec/fe2000/elecpop.htm.

[7] Conseil Constitutionnel, *Élection présidentielle 2002*, 2002, http://www.conseil-constitutionnel.fr/dossier/presidentielles/2002.

[8] Arthur H. Copeland, *A 'reasonable' social welfare function*, Seminar on mathematics in social sciences, University of Michigan, 1951.

[9] O. Davis, M. Hinich, and P. Ordeshook, *An expository development of a mathematical model of the electoral process*, American Political Science Review **68** (1970), 426–448.

[10] Jean Charles de Borda, *Mémoire sur les élections au scrutin*, Histoire de l'Académie Royale des Sciences, 1781.

[11] Anthony Downs, *An economic theory of democracy*, Harper and Row, New York, 1957.

[12] William V. Gehrlein, *The probability of electing the condorcet loser*, Proceedings of National Decision Sciences Institute **2** (1996), no. 111.

[13] Alan Gibbard, *Manipulation of voting schemes: a general result*, Econometrica **41** (1973), no. 4, 587–601.

[14] Bernard Grofman and Scott L. Feld, *If you like the alternative vote (a.k.a. the instant runoff), then you ought to know about the Coombs rule*, Electoral Studies **23** (2004), no. 4, 641–659.

[15] Thomas Hare, *The machinery of representation*, 1857.

[16] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, *A formal basis for the heuristic determination of minimum cost paths*, IEEE Transactions on Systems Science and Cybernetics **4** (1968), no. 2, 100–107.

[17] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe, *Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP*, J. ACM **44** (1997), no. 6, 806–825.

[18] John J. Bartholdi III, Craig A. Tovey, and Michael A. Trick, *Voting schemes for which it can be difficult to tell who won the election*, Social Choice and Welfare **6** (1989), no. 2, 157–165.

[19] Samuel Merrill III, *A comparison of efficiency of multicandidate electoral systems*, American Journal of Political Science **28** (1984), no. 1, 23–48.

[20] Philip D. Straffin Jr., *Topics in the theory of voting*, Birkhäuser, Boston, 1980.

[21] John C. McCabe-Dansted, Geoffrey Pritchard, and Arkadii Slinko, *Approximability of Dodgsons rule*, Social Choice and Welfare **31** (2008), no. 2, 311–330.

[22] Hannu Nurmi, *Comparing voting systems*, D. Reidel Publishing, Dordrecht, the Netherlands, 1987.

[23] Michel Regenwetter and Ilia Tsetlin, *Approval voting and positional voting methods: Inference, relationship, examples*, Social Choice and Welfare **22** (2004), no. 3, 539–566.

[24] Mark A. Satterthwaite, *Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions*, Journal of Economic Theory **10** (1975), no. 1, 187–217.

[25] T. N. Tideman, *Independence of clones as a criterion for voting rules*, Social Choice and Welfare **4** (1987), no. 3, 185–206.

[26] Ilia Tsetlin, Michel Regenwetter, and Bernard Grofman, *The impartial culture maximizes the probability of majority cycles*, Social Choice and Welfare **21** (2003), no. 3, 387–398.