Strategic Voting Assignment
Muti-Agent Systems

Jelle Jansen, Darong de Groot, Giulio Zani,
Caio Guirado, Gilles Salem, Sergi Nogués

December 12, 2021

1 Abstract

The Tactical Voting Analyst (TVA) agent is software that computes the risk of a given voting scheme and scenario to strategic voting. In order to compute the Risk, we define the parameter Happiness representing the voter's happiness. Three risk measures (2.2) and four happiness approaches (2.1) are proposed, from which the Cubed Weighted Index Difference (CWID) happiness and Risk 2 have the best performance respectively.

All experiments use the Preference initialization (3.1) algorithm which returns the voter's preference arrays with Normal, Two-peaks, or Uniform distribution. Experiments for the Basic TVA (3.2) are: a comparison of the risk and overall happiness between the four voting schemes (3.2.1), comparison of the averaged impact each voter's best tactical options have on the overall happiness (3.2.2) and impact of tactical voting to the majority graph of the voting situation (3.2.3). A more advanced model (advanced TVA) is also proposed (3.3) for which two experiments are performed: Coalitions (3.3.1) and Runoff elections (3.3.2).

According to Risk vs Happiness experiments, Borda count and Anti-plurality suffered from instability when increasing the number of candidates. On the other hand, Anti-plurality resulted to be the scheme where more influence over the overall happiness a voter has. The majority graph experiments show that more voters result in a more stable voting situation, but more candidates in a less stable situation. Regarding the advanced TVA, the enhancements with respect to the basic TVA lead to higher stability because tactical voting strategies face more difficulties.

2 Happiness and Risk

The Tactical Voting Analyst (TVA) agent is software that computes the risk of a given voting scheme and scenario to strategic voting. In order to compute the Risk, we define the parameter Happiness, which as its name indicates, describes the happiness of a voter in a voting scheme.

Find in section 2.1 a description of the four happiness approaches (2.1.1 and 2.1.2) , their normalization (2.1.3) and a comparison of their performance (2.1.4).

Section 2.2 describes three different Risk formulas and a final discussion (2.2.1).

2.1 Happiness types

The happiness H_i of a voter i measures how similar a voting outcome O (strategic or non-strategic) is from the voter's true preference list v_i. This section contains a detailed explanation of the implemented happiness measures and a discussion about their performance.

2.1.1 Borda Happiness

The first happiness measure is Borda happiness and is defined as follows. Given a voting scheme with n candidates, the outcome O, a voter i with preferences v and the Borda count $b = [n-1, n-2, \ldots, 0]$, we assign to every candidate a Borda value based on their rank in v. That is, the candidate at index j in voter's preferences gets a value of b_j. The voter's Borda happiness H_i is calculated as

$$H_i = \sum_{j=0}^{n-1} b_j \times b_{O_j} \tag{1}$$

where b_{O_j} is the Borda value assigned to the candidate given by O_j.

For instance, let's assume $n = 4$ candidates A, B, C and D, $O = [A,C,B,D]$, $v = [A,B,C,D]$ and $b = [3,2,1,0]$. Equation 2 shows how the happiness H is obtained.

$$H = \begin{bmatrix} A = 3 \\ B = 2 \\ C = 1 \\ D = 0 \end{bmatrix} \cdot \begin{bmatrix} A = 3 \\ C = 1 \\ B = 2 \\ D = 0 \end{bmatrix} = 3 \times 3 + 2 \times 1 + 1 \times 2 + 0 \times 0 = 13 \tag{2}$$

2.1.2 Weighted Index Difference Happiness

A group of three Happiness formulas are described in this section, which is based on a weighted index difference approach.

To calculate the Weighted index difference (WID), we take the pairwise index of each candidate ranking in true preference and subtract this index with the index of that candidate in the outcome vector to obtain the index difference. This index difference is then multiplied by the corresponding weight of the positions. This is best shown via an example calculation which we will do in the next subsections with linear, squared, and cube weights. When using the WID, the index difference essentially tells us the size and direction of the displacement of the positions, and the multiplication by weights lets signal how significant the impact of these displacements has on the happiness of the voter (depending on the candidate position in the voter's true preference). Since the best possible outcome occurs when the index of the candidates in the voter true preference and the index of the candidates in the voting outcome is equal (index difference of 0), the best happiness value achievable is 0, and less ideal positions are defined by increasingly negative values. This however is dealt with in Section 2.3 where we normalize all happiness measures used.

Given a voting scheme with n candidates, the weight is defined as $w_x = [n^x, (n-1)^x, \dots, 1^x]$. For ease of reference we will now call the measure Linear Weighted Index Difference (LWID) for w_1, Squared Weighted Index Difference (SWID) for w_2, and Cubed Weighted Index Difference (CWID) for w_3. However note that you can use any value for x (which is one of the benefits of WID). In section 2.1.4 we will show that w_3 is preferred over the other 2 measures.

Assuming we have 4 candidates the weights vector is defined as $w = [4^x, 3^x, 2^x, 1^x]$. Let us have a true preference vector of voter i defined as $v_i = [A,B,C,D]$ and a voting outcome $O = [A,C,B,D]$. We can calculate the Weighted Index Difference (WID) happiness as follows:

$$WID = (1-1)4^x + (2-3)3^x + (3-2)2^x + (4-4)1^x$$

In the first term, the index difference (1-1) is a result of A which has an index of 1 in the true preference and also and an index of 1 in the outcome. This index difference is multiplied by 4^x which is the largest weight in w since voter A is the first top preferred candidate in the true preference list. However $0 \times 4^x = 0$ Therefore this term is 0. For the second term, the index difference is (2-3) This is because candidate B has an index of 2 in the true preference and an index of 3 in the outcome. This difference -1 (which indicates that the candidate has moved one unit down in the list via the negative sign) is then multiplied by 3^x which is the second-largest weight in w since voter B is the second top preferred candidate in the true preference list. The same logic can be applied to the rest of the calculations. Below shows the example of WID using w_1, w_2 and w_3

$LWID = (1-1)4 + (2-3)3 + (3-2)2 + (4-4)1 = -1$

$SWID = (1-1)4^2 + (2-3)3^2 + (3-2)2^2 + (4-4)1^2 = -5$

$CWID = (1-1)4^3 + (2-3)3^3 + (3-2)2^3 + (4-4)1^3 = -19$

2.1.3 Normalizing happiness

In this section, we briefly describe the process of normalizing the scale of the different happiness measures discussed earlier to an interval between 0 and 1. This makes the value interpretable and makes it possible to compare the performance of different measures.

Borda

For Borda happiness. The normalized value can be obtained by dividing the happiness value by the best achievable happiness value. The best happiness value acts as the normalization constant. We denote this constant as C_N, and so the normalized happiness $H_{norm} = \frac{H}{C_N}$. This constant can be obtained by computing the dot product of the Borda count by itself because the happiness obtained will be the maximum value possible. It is important to note that the only thing this normalization constant depends on is the number of candidates so we just compute this value once and use it to normalize the happiness of all voters.

Weighted index difference

The WID happiness all works in a similar manner regardless of the weights variant used. However, the normalization process of WID's is computed differently from the Borda happiness. This is because the WID has a maximum happiness at 0, and becomes increasingly negative as voters are dissatisfied. Therefore the approach here is to compute the worst happiness achievable to use as a norm constant C_N. Afterwards the normalized happiness is calculated as $H_{norm} = 1 - \frac{H}{C_N}$. Note here that we take 1 to subtract $\frac{H}{C_N}$ because this fraction is a small positive value when the voter is happy since the happiness scale approaches 0 from a negative value as happiness increase. The C_N can be obtained by computing the happiness of an outcome which is reversed true preference list. For example if the voter preference $v = [A,B,C,D]$, then we will compute the happiness that voter will get when the outcome is $O = [D,C,B,D]$. Luckily the reverse list of any list is equivalent in terms of the distance shifts in the index. For example the value computed using $v = [A,B,C,D]$ and $O = [D,C,B,D]$ is equivalent to the value computed using $v = [D,B,A,C]$ and $O = [C,A,B,D]$. This means that this normalization constant only depends on the number of candidates and the type of weights variant chosen. Therefore for a chosen weight variant, the normalization constant needed to be computed only once. Below is an example.

2.1.4 Happiness Comparison

Once Happiness measures are defined and normalized, we compare their performance in a simple example where the true preference is fixed to [0,1,2,3].

In the first case in Figure 1 where the outcome = [0,1,2,3] all measures output a 1 as expected since outcome = true preference. However, for the Borda and linear (WID) measure, we see a problem where the happiness is constant despite using different outcomes. For outcome = [0,1,3,2], [0,2,1,3] and [1,0,2,3] the Borda and linear measure outputs a fixed value of 0.93 and 0.9 respectively which is unrealistic since outcome [0,2,1,3] should have a bigger negative impact on happiness than [0,1,3,2]. Likewise outcome [1,0,2,3] should have resulted in lower happiness than [0,2,1,3]. This failure to represent the different (and representative) happiness for different voting outcomes will have a significant impact on, for example, determining tactical options. Fortunately, the cubed and squared WID's do capture this trend, with the cubed WID being more discriminated. For the last example where outcome = [3,1,2,0] (least preferred candidate wins, and first preference ends up last), all measures except Borda output very low value as expected. Borda measure on the other hand is rather generous here.

2.2 Risk Analysis

Risk measurements are defined under the following assumptions:

1. The risk is higher when the number of voters with at least one tactical option increases.
2. The risk is higher when the total number of tactical options for all voters increases.
3. The risk of a tactical vote is higher when such a tactical vote increases more the voter's own happiness. That is, the bigger the difference between the 'new own happiness' and the 'true own happiness', the higher the risk.

Three different formulas are used in order to quantify the risk. Some of these consider the risk as qualitative while others consider it as quantitative. Equation 3 shows the generic risk formula.

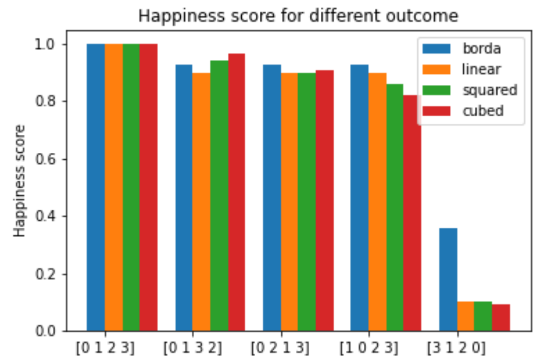


Figure 1: Happiness of different measure vs different outcomes when true preference = [0,1,2,3]

$$\text{Risk} = \frac{\sum_0^V x_i}{V} \tag{3}$$

where V is the total number of voters.

1. The first risk only takes into account assumption 1 and x_i is 1 if voter i has at least one tactical option, 0 otherwise.
$$x_i = \begin{cases} 1, & \#\,\text{tactical options}_i \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$
2. The second risk formula assumes 1 and 2 and x_i is equal to the number of tactical options that voter i can perform divided by the total number of permutations. That is:
$$x_i = \frac{\#\,\text{tactical options}_i}{\#\,\text{candidates}!} \tag{5}$$
3. The third risk formula considers assumptions 1 and 3 and x_i is defined as the maximum change in happiness. Applying the most rewarding tactical option (in terms of happiness) and the voter's own happiness.
$$x_i = \max(H_i' - H_i) \tag{6}$$
where H_i is the true voter's i own happiness and H_i' is the voter's own happiness after applying the most rewarding tactical option.

2.2.1 Risk comparison

In order to compare the three risk definitions consider figure 2, which shows the risk output given by the TVA when the input is four candidates, combinations of voting schemes, and 5 voter preferences.

See situations 3 and 5, where all three Risk approaches give high-risk values to them, but risk 1 considers both scenarios equally 'risky', while risk 2 considers scenario 5 more sensible to tactical voting and the opposite for risk 3. This is because risk 1 measures the number of voters with at least one tactical option, which is the same for both scenarios; risk 2 considers the total amount of tactical options, which is higher in 5; and finally, risk 3 sees that in scenario 3 voters' happiness increase more after each best's tactical options.

The same reasoning can be used with other scenarios such as 6, where risk 3 disagrees with risks 1 and 2 because the voter's own happiness increase is low, although two voters have 4 tactical options each.

Eventually, one risk approach is going to make better predictions than another based on your goal, so it all depends on what assumptions (see section 2.2) are a priority.

Preferences	Id	Voting Scheme	Outcome	# tactical options					Happiness change (true happiness -> new happiness)					red = high risk, orange = moderate, yellow = low		
				0	1	2	3	4	0	1	2	3	4	Risk 1	Risk 2	Risk 3
Voter 0: [2 0 1 3] Voter 1: [3 1 2 0] Voter 2: [0 2 3 1] Voter 3: [1 0 2 3] Voter 4: [1 0 3 2]	1	plurality	B: 2, A: 1, C: 1, D: 1		6					0.22 -> 0.25				0.2	0.05	0.0067
	2	vote_for_two	A: 4, B: 3, C: 2, D: 1											0	0	0
	3	borda_count	A: 9, B: 9, C: 7, D: 5		2		5	5		0.09 -> 0.22		0.82 -> 1.0	0.79 -> 0.97	0.6	0.1	0.0962
	4	anti_plurality	A: 4, B: 4, C: 4, D: 3	6					0.55 -> 0.79					0.2	0.05	0.0471
Voter 0: [1 3 2 0] Voter 1: [1 2 3 0] Voter 2: [3 1 2 0] Voter 3: [1 3 0 2] Voter 4: [0 1 2 3]	5	plurality	B: 3, A: 1, D: 1, C: 0	6	12		6		0.84 -> 0.97	0.75 -> 0.84		0.91 -> 1.0		0.6	0.2	0.0615
	6	vote_for_two	B: 5, D: 3, A: 1, C: 1	4		4			0.97 -> 1.0		0.79 -> 0.82			0.4	0.0667	0.0135
	7	borda_count	B: 13, D: 8, C: 5, A: 4				5	2				0.97 -> 1.0	0.22 -> 0.49	0.4	0.0583	0.0606
	8	anti_plurality	B: 5, C: 4, D: 4, A: 2	6		6			0.91 -> 0.97		0.55 -> 0.79			0.4	0.1	0.0587
Voter 0: [1 3 2 0] Voter 1: [3 0 1 2] Voter 2: [0 3 2 1] Voter 3: [2 0 1 3] Voter 4: [3 1 0 2]	9	plurality	D: 2, A: 1, B: 1, C: 1				3					0.09 -> 0.22		0.2	0.05	0.025
	10	vote_for_two	D: 4, A: 3, B: 2, C: 1			8					0.79 -> 0.97			0.2	0.0667	0.0356
	11	borda_count	D: 10, A: 8, B: 7, C: 5			1		1			0.79 -> 0.84		0.91 -> 1.0	0.4	0.0167	0.0288
	12	anti_plurality	A: 4, B: 4, D: 4, C: 3					6					0.46 -> 0.49	0.2	0.05	0.0048
Voter 0: [3 1 2 0] Voter 1: [2 0 1 3] Voter 2: [2 0 1 3] Voter 3: [3 0 2 1] Voter 4: [3 2 1 0]	13	plurality	D: 3, C: 2, A: 0, B: 0											0	0	0
	14	vote_for_two	A: 3, C: 3, D: 3, B: 1	8	4	4	8	8	0.18 -> 0.21	0.79 -> 0.84	0.79 -> 0.84	0.55 -> 0.64	0.36 -> 0.54	1.0	0.2667	0.0808
	15	borda_count	C: 10, D: 9, A: 6, B: 5					3					0.79 -> 0.97	0.2	0.025	0.0356
	16	anti_plurality	C: 5, B: 4, A: 3, D: 3				6					0.03 -> 0.18		0.2	0.05	0.0288

Figure 2: Risk comparison for different voting schemes and preference scenarios.

3 Experiments and Results

Two TVAs have been designed based on their assumptions: the Basic TVA (3.2) and the Advanced TVA (3.3). Different experiments are discussed during the following sections. We start by showing how voter preferences are automatically generated (3.1), then continue with the basic TVA experiments such as the Risk vs Happiness (3.2.1), the influence of voter's tactical options on the overall happiness (3.2.2) and the majority graph (3.2.3); and eventually, the advanced TVA experiments namely Coalitions (3.3.1) and Runoff Elections (3.3.2).

3.1 Preference initialization

We have created an algorithm that assigns to each permutation of preferences a number of voters by sampling from a distribution. The distributions we created are the following: Uniform, Normal with $\sigma = \frac{C!}{8}$ and Two-peak by summing up and then normalizing two normal distributions, each with $\sigma = \frac{C!}{12}$ where C is the number of candidates.

Next, we wanted to test how different preferences initialization influences happiness after voting. Our hypothesis was that the happiness would have looked like this: normal $>$ two-peak normal $>$ uniform. The reason behind this was that we assume that a bigger division among the population preference would increase the chance that the outcome will not align with the voter's expectation. On contrary, if the whole population would have the same preference, they would all vote identically and all voters would be happy with the obtained outcome. To test this repeated 100 times (since we shuffle the preference order): initialization, voting, and overall happiness measure and then averaged the obtained happiness. We used cubic happiness and the Borda voting scheme. The results are 0.562, 0.553, 0.545 for normal, two-peaks, and uniform respectively. Although the results coincide with our hypothesis, the magnitude of the difference in happiness between these preference distributions are smaller than we've expected.

3.2 Basic TVA

The TVA software takes as input: number of voters, number of candidates, voter preferences, voting scheme, happiness scheme, and risk scheme. Experiments settings will differ based on these parameters. For each experiment the following results are obtained: Comparison of the risk and overall happiness between the four voting schemes, comparison of the averaged impact each voter's best tactical options have on the overall happiness and impact of tactical voting to the majority graph of the voting situation.

3.2.1 Risk vs Happiness

This experiment aims to analyze which voting scheme is better suited for a voting situation. In order to study this, we observe the position of the voting scheme with respect to a happiness scheme and a risk metric, both described in the sections above. Figure 3 shows how to interpret the results of the experiments.

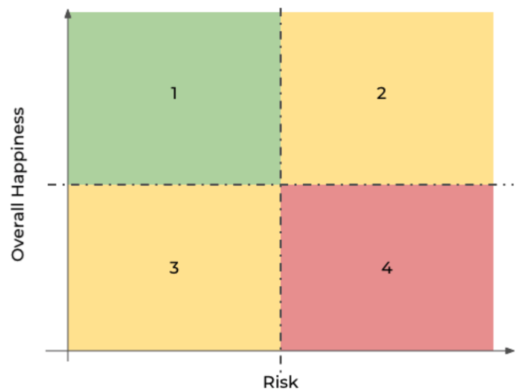


Figure 3: Risk vs. Happiness quadrants.

- Quadrant 1: Best scenario. If a voting scheme falls in this region, it provides major happiness to its voters and a lower risk of manipulation.
- Quadrant 2 and 3: Not desirable regions, but also not the worse. A trade-off between high happiness and high risk, or low happiness and low risk.
- Quadrant 4: Worst scenario. If a voting scheme falls in this region, it provides less happiness to its voters and a high risk of manipulation.

To generate an experiment, a configuration file needs to be provided, and the following parameters must be specified.

Figure 4 shows the result of the experiment that analyzes the different voting schemes over different risk types when increasing the number of candidates. The first thing that is observable is that the binary risk shows many schemes as being high risk, but that only means that the voters may have many tactical options, but don't provide information on the quality of those tactical options. By using risk types 2 and 3 in addition, it is possible to conclude that the schemes that seem risky for the binary risk are not that dangerous if we analyze the quality of that tactical options. The voting scheme that seems most risk-full is Borda Count. This is expected, since using Borda Count, the voter can affect the score of multiple candidates.

Another interesting behavior is observed by looking that a single scheme's risk increase with the number of candidates in the binary risk setup, which makes sense since it also allows more combinations of preferences. However, when looking at risk type 2, which normalizes the number of tactical options by the number of candidates, the opposite behavior is revealed, meaning that it becomes more difficult to have impactful tactical options when the number of candidates is higher. As seen before (Section 2.2.1), Risk 2 seems most fair and stable.

3.2.2 Impact Overall Happiness

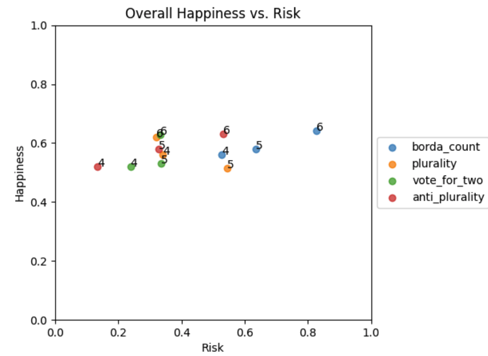
A voter's tactical option i has an impact on the final outcome O, leading to a new outcome O'_i. We measure the impact as the averaged difference between the true overall happiness H and the new overall happiness H'_i for each voter's best tactical option i (see equation 7). Happiness is measured using the cubic weighted index difference.

$$\text{Impact} = \frac{\sum_{i=1}^{n} H'_i - H}{n} \tag{7}$$

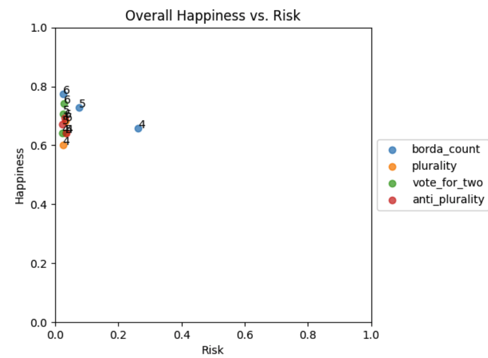
where n is the number of voters.

Figure 5 shows a comparison of the impact for different votings schemes and three types of voter preferences distribution: uniform, normal, and two-peaks. Impact values have been multiplied by 100 before plotting and averaged over 1000 episodes.

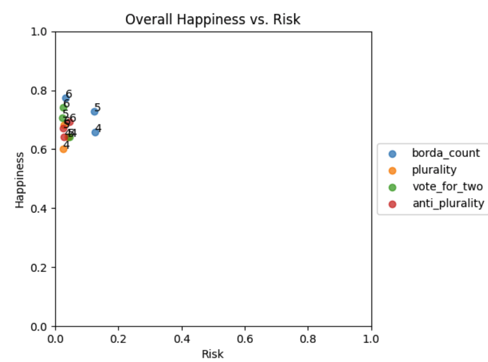
We see that in voting schemes where candidates only receive a point when they are top-ranked in a voter's preference vector, such as Plurality or Voting for two, the influence of individual tactical votes on the overall happiness is low. On the other hand, in voting schemes where $m-1$ out of m candidates receive a vote based on their rank in voter's preference, such as Borda count or Anti-plurality voting, the influence of individual tactical votes on the overall happiness is high. Voting schemes where candidates receive at most 1 point per voter, that is all voting schemes except Borda, lead to higher overall happiness after a tactical vote, which is the opposite for Borda voting. That means that in the Borda scheme all voters tend to be less happy after a tactical vote (except for the voter doing it). Overall, anti-plurality is the scheme where more influence over the overall happiness a voter has and plurality is the one with less influence per voter.



(a) Risk 1



(b) Risk Type 2



(c) Risk Type 3

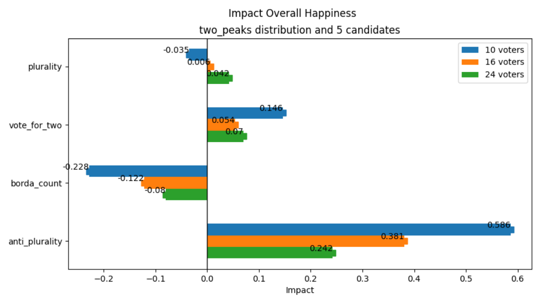
Figure 4: Increasing the number of candidates for each voting scheme. Linear weight happiness. The numbers mean the amount of candidates for the voting situation.

3.2.3 Majority graph

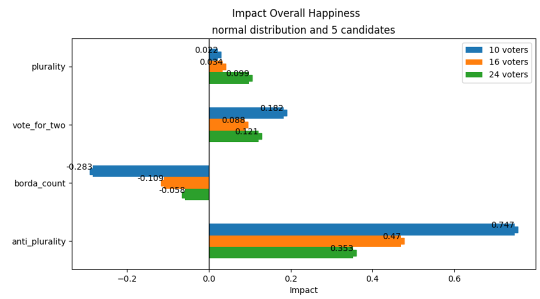
Another experiment we performed is looking at the impact of tactical options on the majority graphs using different voting schemes. The goal of this experiment is to analyze the impact of tactical options on voting situations.

To see how much impact each tactical option has, we calculated the Majority Graph for each tactical option, summed all of these matrices/tables up together, and normalized it. This allows our Majority Graph to be bi-directional, where the line width of the edge indicates the number of majority graphs having this edge in common. These normalized majority graphs have been calculated for different numbers of voters and candidates, different voting schemes to determine the tactical options and different distributions to sample preferences. The default configuration looks as follows: 35 voters, 6 candidates, Borda voting scheme, Cubic Weighted Index Difference, and normal distribution of preferences.

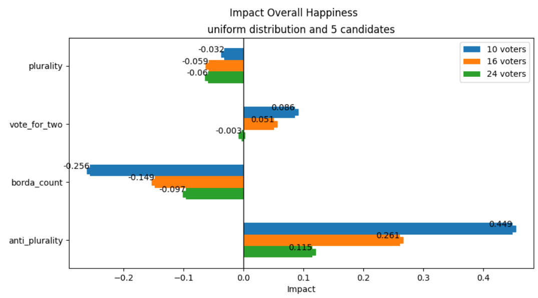
By looking at the visualizations for different configurations, we can learn more about the voting situation and risk. Please note, the results used preferences by sampling for distributions (explained in Section 3.1), this means different results are possible every time it is being run.



(a) Two-peaks distribution.



(b) Normal distribution.



(c) Uniform distribution.

Figure 5: Impact overall happiness for each voting scheme, 10, 16 and 24 voters and 5 candidates.

Different configurations have been compared by changing one of the parameters from the default configuration (number of voters, number of candidates, etc.). The majority graph of the default configuration, together with the majority graphs of the different configurations can be seen in Figure 6.

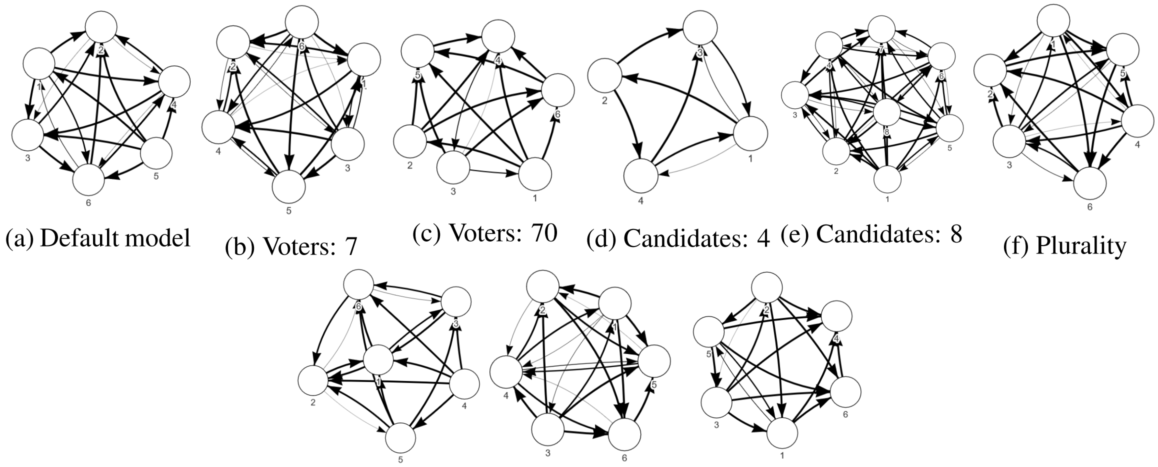
In Table 1 the results are described from the different configurations seen in Figure 6. Please note, using the middle configuration for all parameters results in the default model.

Changed	**Configuration**	**Result**
Voters	7, 35, and 70	Using more voters resulted in less bidirectional edges. We define this voting situation as being more stable. We can explain this since when there are more voters, each individual voter has relatively less influence on the majority.
Candidates	4, 6, and 8	Using more candidates resulted in a less stable voting situation. We can explain this since more candidates result in more different preferences, which makes the scores of the outcome closer to each other because of the relatively higher scores for Borda count.
Voting scheme	Plurality, Borda Count, Anti-plurality	Plurality affected the majority most. This can be explained since the tactical options from plurality only consider the majority. Borda count and Anti-plurality affected the majority less. This can be explained since for both of these to increase happiness (and so the outcome), we don't have to change the majority of voters but the score only.
Distribution	Uniform, Normal, Two-peaks	When using the uniform distribution the voting situation was less stable than when using normal and two-peaks. This can be explained since normal and two-peaks assume voters which are more same-minded. Most of these voters prefer the same candidates, which makes changing the majority harder. However, when using uniform, this is not the case. The preferences are distributed "randomly", which means it is easier to change the majority

Table 1: Experiments Majority Graph

3.3 Advanced TVA

After finishing our basic tactical voting analyst, we have removed some of the assumptions made in the Basic TVA and explored some advanced TVA features. These are summarized in this section.



(a) Default model (b) Voters: 7 (c) Voters: 70 (d) Candidates: 4 (e) Candidates: 8 (f) Plurality

(g) Anti-plurality (h) Uniform (i) Two-peaks

Figure 6: Normalized Majority Graphs different configurations

3.3.1 Coalitions

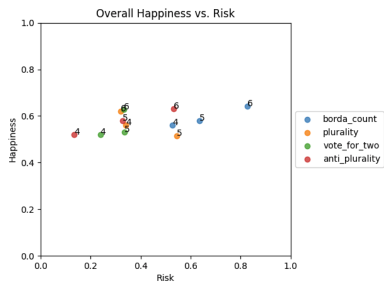
Figure 7 (a) and (b) show the results of running the same experiment applied to the basic TVA but now with the coalition setup. With this configuration, the same conclusion regarding the risk types come across, meaning that the binary risk assumes that a lot of the schemes encapsulate high risk, whereas when evaluating them with the risk type 2, they actually don't provide much risk, especially when increasing the number of candidates.

Assumption dropped: Voters can only vote alone.		
New Assumption	How coalitions are created	How it works
All voters belonging to the same coalition change their preferences to the same strategic preference.	Randomly, all of the same size.	All voters will change their preferences only when the expected outcome is favorable for all of them.
Voters having the same true preferences will also have the same strategic preferences.	Voters belong to the same coalition if they have the same preferences. Coalitions may thus have different sizes.	A coalition is treated as a unique voter. However when computing outcomes and overall happiness the size of the coalition is taken into account. For example, with respect to a single voter, a coalition of size 4 will have an impact that is 4 times larger to the overall happiness.

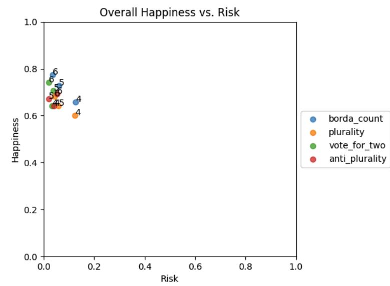
3.3.2 Runoff Election

Besides creating coalitions in our algorithm, we also looked at the run-off election instead of using a voting scheme. This resulted in tactical options for a run-off election, which look the same as the tactical options for the voting schemes. Because of this, the same experiments have been performed on these tactical options. For each of these, we performed some experiments.

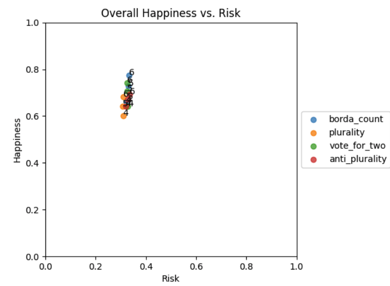
Within the run-off setup, the same experiments as the basic TVA were performed to analyze their standings with respect to overall happiness and risk. Figure 7 (c), (d) and (e) show the results of the experiments. It's possible to see that with run-off, the schemes don't present many variations in risk between themselves, and one more time it's clear that the binary risk measurement overestimates the real risk in comparison to risk type 2 and 3.



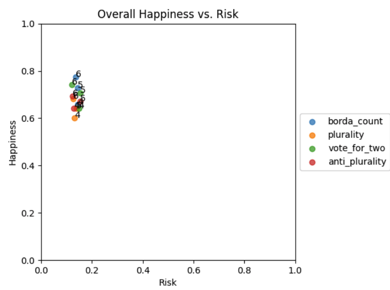
(a) Binary Risk



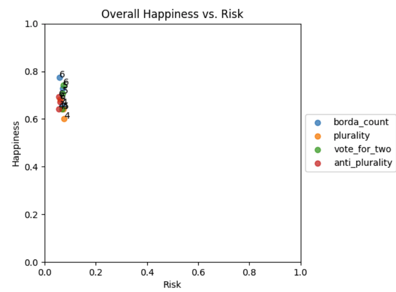
(b) Risk Type 2



(c) Binary Risk



(d) Risk Type 2



(e) Risk Type 3

Figure 7: Increasing the number of candidates for each voting scheme. Linear weight happiness. Values are the amount of candidates per voting situation. (a) and (b) correspond to coalitions, (c), (d) and (e) to Run-off configuration.

4 Discussion and Conclusion

As explained in section 2.1, Cubic Weighted Index Difference provides the highest happiness resolution. This allows for better risk measurements since all risk approaches depend on happiness values. The best risk formula is chosen based on the most relevant assumptions (section 2.2). To start with, risks 2 and 3 give better values than risk 1 since they account for more assumptions. In addition, we believe that risk 2 performs better than risk 3 since happiness should not be used as a direct risk measurement. That is, risk 3 directly depends on voter happiness while risk 2 indirectly depends on voter happiness.

According to the experiments comparing different voting schemes in regards to their happiness and risk scores, the most unstable schemes were Borda count and anti-plurality, while plurality and vote for two remained more stable even with the changes in the number of candidates. Regarding the voter's influence on the overall happiness, we see in section 7 that anti-plurality is this scheme where more influence over the overall happiness a voter has and plurality is the one with less influence per voter.

When performing the experiments using the majority graph on our best-performing happiness measure (CWID), we found out that more voters resulted in a more stable voting situation. On the opposite, more candidates resulted in a voting situation that was less stable. Plurality seemed to affect the majority most since we are only considering the majority in the voting scheme. Anti-plurality and Borda count affected the majority less. We also found out that if there are more voters which are more same-minded (normal distribution), it's harder to change the majority, than if there are only preferences that are uniformly distributed.

By comparing the basic TVA with the advanced version, it's possible to conclude that the added complexities help to make the system more stable since the agents that would be willing to manipulate the election would also face more difficulties reflected in their strategies.

Something that we should keep in mind, is that our current TVA assumes simplifications about the real world. For example, perfect information, only one voter voting, etc. We tried to make our TVA more realistic by removing assumptions (Advanced TVA). However, it is still a simplification of the real world. Besides that, all preference orderings are considered when determining the tactical options. This results in being too computationally expensive when the number of candidates increases. To improve this change have to be made.

Further experiments could be made by dropping more assumptions described in the assignment description, for instance, removing perfect knowledge in our TVA. Besides that, currently, tactical voting only consists of changing voter preferences. However, in follow-up experiments removing or adding candidates and voters could be added to the TVA to increase the number of options the voting situation can be manipulated.

5 Literature

6 Work Distribution

The tasks done by each group member are as follows:

- Jelle (Group Coordinator)
  - Code: Set-up basic structure for code
  - Code: Implement code to determine tactical options
  - Code: Numpize strategic options (optimize code)
  - Code: Implemented Majority Graphs (with visualisation)
  - Code: Advanced TVA: Voting in groups (e.g. pairs)
  - Code: Advanced TVA: Implement run-off election
  - Experiment: Majority Graphs
  - Report: Majority Graph (explanation + experiment)
  - Report: Advanced TVA (explanation)
  - Other: Determine computation time for Advanced TVA (to determine what to do next)
- Caio
  - Code: Implement Binary risk
  - Code: Implement risk analysis
  - Experiment: Visualise tactical options
  - Experiment: Try out different experiments
  - Experiment: Automated experiments
  - Experiment: Advanced TVA: Voting in Groups (e.g. pairs)
  - Report: Automated Experiments
  - Report: Advanced TVA: Voting in Groups (e.g. pairs)
  - Report: Advanced TVA: Run-off election
- Gilles
  - Code: Implement all risk functions
  - Report: Set-up report
  - Report: Abstract
  - Report: Risk (explanation + examples)
  - Report: Compare risks
  - Research: Risk

- Help: Research about happiness (Darong)

- Giulio
  - Code: Implement code to determine tactical options
  - Code: Numpize structure (optimize code)
  - Code: Optimize code by creating groups of voters
  - Code: Stochastic preference distribution and initialization
  - Experiment: Population preference initialization effect on happiness
  - Report: Implementation details
  - Report: Stochastic preference initialization
  - Help: Implementing Risk (Gilles)
- Sergi Nogues
  - Code: Implement Borda Happiness
  - Experiment: Try out different experiments
  - Experiment: Impact tactical options on overall happiness
  - Experiment: Compare risks
  - Report: Borda Happiness (explanation + examples)
  - Report: Impact overall Happiness
  - Report: Write risks
  - Report: Compare risks
  - Help: Implementing Risk (Gilles)
- Darong
  - Code: Implement WID Happiness
  - Code: Normalize Happiness
  - Code: Stochastic and Deterministic preference distribution and intialization
  - Experiment: Comparison of different happiness measure on individual voters happiness
  - Experiment: Population preference initialization effect on happiness
  - Report: WID Happiness (explanation + examples)
  - Report: Normalization Happiness
  - Report: Compare Happiness
  - Report: Output details
  - Research: Happiness and preference distributions

7 Implementation Details

We have used the Python programming language and the following libraries:

- Numpy: optimization and vectorization
- Scipy: access to specific math functions

- Matplotlib: data visualization
- Pyvis: Majority graph visualization

Python is a slow and interpreted programming language, so whenever possible we have expressed our functions using linear algebra and masking so that it could be rewritten only using Numpy arrays. Also notice that to run a few experiments and measures we had to sometimes compute all of the permutations of the candidates. For those functions, this increased the time and space complexity to $O(n!)$. Where n is the number of candidates.

Here's an example of the code that computes the cubed happiness:

```
cubed_weights = np.power(np.arange(len(ranked_candidates_id), 0, -1),3)
norm_value = np.dot(
    cubed_weights, np.arange(len(pref_norm)) - indices
)
argsorting = np.arange(len(ranked_candidates_id))
argsorting[ranked_candidates_id] = argsorting.copy()
indices = argsorting[true_preferences]
happiness = np.dot(
    cubed_weights, np.arange(len(true_preferences)) - indices
)
happiness = 1 - happiness / norm_value
```

And here an example of the code that computes risk number 3 (see the risk section):

```
    risk = sum(
        v[0][2] - v[0][1] for v in tactical_options if len(v) > 0
    ) / len(self.voting_situation.voters)
```

While this is the coding style we have used for single functions, we have used object-oriented paradime to make it easy to collaborate on the project. Here's a list of the classes we have created:

- TacticalVotingAnalyst: Main class, contains user-friendly functions that allow to compute everything seen in this project. This class can be potentially be imported and used as a deliverable.
- Voter: Contains voter-related data such as (strategic) preferences as well as happiness functions.
- VotingSituation: Container class that hosts candidates and voter's information, as well as some functions for plotting.

An example of how to use our function from within a Python script can be found in the file:

```
tactical_voting_analys/__main__.py
```

8 Output examples

In this section, we show the output of important functions that are called when the main of the TVA is executed.

The outputs will be shown for two different voting situations where,

situation 1 : preferences = [(0, 1, 2, 3), (1, 2, 3, 0), (2, 3, 0, 1), (0, 1, 2, 3), (1, 2, 3, 0), (2, 3, 0, 1), (0, 1, 2, 3),]

situation 2 : preferences = [(3, 1, 2, 0), (3, 1, 2, 0), (3, 1, 2, 0), (3, 1, 2, 0), (1, 2, 3, 0), (2, 3, 0, 1), (0, 1, 2, 3),]

get_winner() functions take an input parameter:

- voting_vector: voting scheme ndarray an array which contains a number indicating which voting scheme to use. For example: [3,2,1,0] represents borda, [1,0,0,0] represents plurality and [1,1,0,0] represent vote for two etc.

- print_winner: a boolean which indicates whether results should be printed.

This function outputs an ndarray of shape (len(candidates),) containing number of votes for each candidate. The output for the determine_tactical_options() for situation 1 and 2 is shown below in Figure 8 and 11.

determine_tactical_options() functions take an input parameter:

- voting_scheme: This is an Enum which indicates the voting scheme we want to use. for example Borda, Plurality, etc.

- happiness_scheme: This is an Enum which indicates the happiness scheme we want to use. for example Borda, Linear WID, squared WID, etc.

The output for the determine_tactical_options() for situation 1 and 2 using Borda voting scheme is shown below in Figure 9 and 12. The output displays the tactical option for each voter, their corresponding increase in happiness(for each option), and the voting outcome of the new situation.

calculate_risk() This function calculates the risk of the situation using the output of determine_tactical_options() function. The output for the 3 different types of risk measures discussed in the report for situations 1 and 2 is shown below in Figure 10 and 13.

When running our package ”tactical_voting_anaylist”, all of the outcomes above, including the used arguments will be shown. To see the optional arguments use help (-h flag). See the project’s README for more details.

```
Voting Scheme - [3. 2. 1. 0.]:
C: 13.0.
B: 12.0.
A: 11.0.
D: 6.0.
```

(a) get_winner() output for borda

```
Voting Scheme - [1. 0. 0. 0.]:
A: 3.0.
B: 2.0.
C: 2.0.
D: 0.0.
```

(b) get_winner() output for plurality.

Figure 8: get_winner() output using voter preference from situation 1

```
Current outcome: C: 13, B: 12, A: 11, D: 6

Voter 0
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: A > C > B > D, new outcome: C: 14, A: 11, B: 11, D: 6
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: A > C > D > B, new outcome: C: 14, A: 11, B: 10, D: 7
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: A > D > B > C, new outcome: C: 12, A: 11, B: 11, D: 8
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: A > D > C > B, new outcome: C: 13, A: 11, B: 10, D: 8
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: C > A > D > B, new outcome: C: 15, A: 10, B: 10, D: 7
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: D > A > C > B, new outcome: C: 13, A: 10, B: 10, D: 9
Happiness: 0.46153846153846156 -> 0.5528846153846154, tactical preference: A > B > D > C, new outcome: B: 12, C: 12, A: 11, D: 7
Happiness: 0.46153846153846156 -> 0.5528846153846154, tactical preference: B > A > C > D, new outcome: B: 13, C: 13, A: 10, D: 6
Happiness: 0.46153846153846156 -> 0.5528846153846154, tactical preference: B > A > D > C, new outcome: B: 13, C: 12, A: 10, D: 7
Happiness: 0.46153846153846156 -> 0.5528846153846154, tactical preference: B > D > A > C, new outcome: B: 13, C: 12, A: 9, D: 8
Happiness: 0.46153846153846156 -> 0.5528846153846154, tactical preference: B > D > C > A, new outcome: B: 13, C: 13, A: 8, D: 8
Happiness: 0.46153846153846156 -> 0.5528846153846154, tactical preference: D > B > A > C, new outcome: B: 12, C: 12, A: 9, D: 9

Voter 1
Happiness: 0.7884615384615384 -> 0.9663461538461539, tactical preference: B > D > C > A, new outcome: B: 12, C: 12, A: 11, D: 7

Voter 2
Happiness: 0.75 -> 0.7836538461538461, tactical preference: C > A > D > B, new outcome: C: 13, A: 12, B: 12, D: 5

Voter 3
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: A > C > B > D, new outcome: C: 14, A: 11, B: 11, D: 6
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: A > C > D > B, new outcome: C: 14, A: 11, B: 10, D: 7
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: A > D > B > C, new outcome: C: 12, A: 11, B: 11, D: 8
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: A > D > C > B, new outcome: C: 13, A: 11, B: 10, D: 8
Happiness: 0.46153846153846156 -> 0.6394230769230769, tactical preference: C > A > D > B, new outcome: C: 15, A: 10, B: 10, D: 7
```

Figure 9: The tactical options of the voters from situation 1

```
The risk is :[1.0, 0.2380952380952381, 0.13667582417582416]
```

Figure 10: The three types of risk from situation 1

```
Voting Scheme - [3. 2. 1. 0.]:
B: 13.0.
D: 13.0.
C: 12.0.
A: 4.0.
```

(a) get_winner() output for Borda

```
Voting Scheme - [1. 0. 0. 0.]:
D: 3.0.
C: 2.0.
A: 1.0.
B: 1.0.
```

(b) get_winner() output for plurality.

Figure 11: get_winner() output using voter preference from situation 2

```
Voter 0
Happiness: 0.8221153846153846 -> 1.0, tactical preference: D > A > B > C, new outcome: D: 13, B: 12,
 C: 11, A: 6
Happiness: 0.8221153846153846 -> 0.9086538461538461, tactical preference: D > A > C > B, new outcome
: D: 13, C: 12, B: 11, A: 6

Voter 1
Happiness: 0.8221153846153846 -> 1.0, tactical preference: D > A > B > C, new outcome: D: 13, B: 12,
 C: 11, A: 6
Happiness: 0.8221153846153846 -> 0.9086538461538461, tactical preference: D > A > C > B, new outcome
: D: 13, C: 12, B: 11, A: 6

Voter 2
Happiness: 0.8221153846153846 -> 1.0, tactical preference: D > A > B > C, new outcome: D: 13, B: 12,
 C: 11, A: 6
Happiness: 0.8221153846153846 -> 0.9086538461538461, tactical preference: D > A > C > B, new outcome
: D: 13, C: 12, B: 11, A: 6

Voter 3
Happiness: 0.5528846153846154 -> 0.8221153846153846, tactical preference: C > A > B > D, new outcome
: B: 12, C: 12, D: 12, A: 6
Happiness: 0.5528846153846154 -> 0.8221153846153846, tactical preference: C > B > A > D, new outcome
: B: 13, C: 12, D: 12, A: 5
Happiness: 0.5528846153846154 -> 0.6394230769230769, tactical preference: C > A > D > B, new outcome
: D: 13, C: 12, B: 11, A: 6
Happiness: 0.5528846153846154 -> 0.6394230769230769, tactical preference: C > D > A > B, new outcome
: D: 14, C: 12, B: 11, A: 5

Voter 4
```

Figure 12: The tactical options of the voters from situation 2

```
The risk is :[1.0, 0.11904761904761904, 0.157967032967033]
```

Figure 13: The three types of risk from situation 2