Final Assignment: Voting satisfaction

Door: Guy Veenhof (1764467), Ruben van Raaij (1763930), Koen van Heertum (1745530)

Inhoud

Abstract

In dit verslag wordt onderzocht welk voting systeem een hogere tevredenheid geeft met behulp van een simulatie. Er zijn 2 voting systemen: Plurality voting en Approval voting. Deze worden nog onderverdeeld in verschillende sub manieren. Ons onderzoek wordt uitgedrukt op een kwantitatieve en kwalitatieve manier. Om data te verzamelen hebben we literatuuronderzoek gedaan en daarop hebben zijn de parameters van de simulatie op gebaseerd. Na het simuleren zijn er resultaten uitgekomen die hebben beslist welk voting systeem de hoogste tevredenheid heeft. Er zijn 18 situaties gedraaid m.b.v. een simulatie met elke keer andere parameters. Elke situatie is 100 keer gedraaid en hieruit is gebleken dat uit die 18 keer er 14 keer Approval voting voor de hoogste tevredenheid heeft gezorgd en maar 4 keer door Plurality voting. Conclusie Approval voting zorgt voor een hogere tevredenheid. Maar Approval word niet echt gebruikt in de echte wereld. Er zijn

waarschijnlijk 2 redenen hiervoor. Reden 1 Approval is nog steeds niet goed uitgetest. Reden 2 Plurality is veel simpeler en makkelijker te begrijpen.

Een korte introductie over het onderwerp

Het kiezen van de beste leider van een bepaald gebied is eeuwenlang al een grote discussie. Vroeger werd deze strijd besloten met oorlog, tegenwoordig stemmen we (kapitalistisch) of kiezen we non-verbaal de leider (communistisch). Vooral het stemmen op een leider is zeer interessant om naar te kijken. Elke kandidaat kan dan op stemmen op de partij(en) die hij of zij lief is. De bekendste stemsystemen zijn: Plurarity, waarin men op de best stemt en Approval, waarin men op meerdere partijen stemmen en dus niet stemmen op de "slechtste" partij. Elk stemsysteem heeft zijn eigen voordelen en nadelen. Zo wordt in Plurality nooit de middelste partij gekozen en bij Approval wordt juist te veel op de middelste partij gestemd. Ook kan elk resultaat van een kandidaat effect hebben op de ander, want als een kandidaat weet dat zijn favoriete partij niet gekozen wordt, dan stemt hij misschien op een andere partij, wat de uitkomst kan veranderen. Kortom er is veel verschil welk systeem er wordt gebruikt en hoe de kandidaten met elkaar communiceren. Je zou bijna zeggen dat het vroeger beter was, omdat je duidelijk kan zien wie de oorlog heeft gewonnen....

De onderzoeksvraag/hypothese

Welk voting systeem heeft de hoogste tevredenheid voor de voters?

Wij denken zelf dat approval voting een betere tevredenheid geeft, omdat het de stemmers de optie geeft om meerdere partij leiders te kiezen en dus een meer een neutrale partij gekozen wordt waarmee een hoge tevredenheid kan worden verkregen.

Vooronderzoek

Simulating alternate voting systems[2] geeft een goeie demonstratie en simulatie wat de voordelen zijn van bekende votingsystemen. Zoals Plurality voting en Approval voting. Dit geeft voor ons een zeer goed voorbeeld hoe kandidaten stemmen per partij zonder gebruik van communicatie. Dit geeft voor ons een goed overzicht over hoe de simulatie eruit moet zien en dat geen van de stemsystemen perfect is. Dit zouden wij kunnen verdelen in:

Plurality:

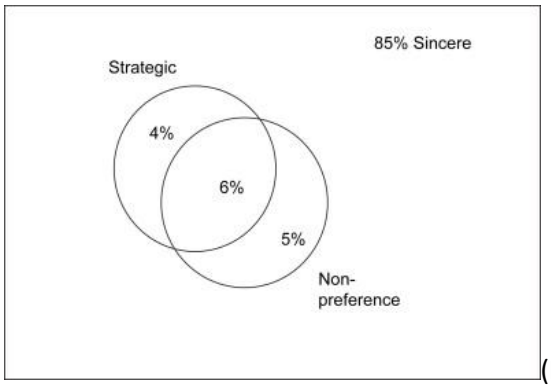
- First: Stem op de dichtstbijzijnde partij.
- Second: stem op de tweede dichtstbijzijnde partij.
- Strategic: Als je 1^{e} keus een verliezende partij is. Verander je vote dan naar je 2^{e} keus.

Approval:

- 1 rivaal: Kiest uit 1 van de top 2 partijen.
- Alleen niet de slechtste: Kiest iedereen behalve de partij waar je het minst mee eens bent.

Uit onderzoek is gebleken dat ongeveer 4 procent van alle stemmers strategisch stemmen. Dit is slechts gebaseerd op ruwe data. "Taking the percentage of self-described strategic votes at face-value suggested that approximately 10% of votes were cast strategically. However, only 6% of votes were cast both against preference and declared to be strategic, whereas 15% were one or the other. Moreover, 4% of votes were declared strategic but cast according to stated preference, indicating

some problem with the interpretation of the question" (Fieldhouse, Shryane, & Pickles, 2007, Estimating strategic vote)



(Fieldhouse, Shryane, & Pickles, 2007)

Plan van aanpak en toolkeuze

Modules

- **GUI**: De grafische interface van het programma. Gaat over de GUI van het programma waarin we onze simulatie maken en de GUI die inbegrepen is bij onze simulatie.
- **Agents:** De individuele 'nodes' van het programma. Deze vertonen allemaal individueel gedrag van elkaar of werken allemaal een soort van samen. Het doel van de simulatie is om te kijken hoe deze agents reageren op hun omstandigheden/omgeving. Ze stemmen met behulp van een afstand tot een partij(agent)
- **Environment**: De 'omgeving' van de agents. Deze kunnen we ook beïnvloeden wat ervoor zorgt dat onze agents anders gaan reageren. Bijvoorbeeld door een politicus op een andere plek in de omgeving te plaatsen zullen de agents op een andere manier gaan stemmen.
- **Staat van simulatie toonbaar:** Hoe goed is het mogelijk om te bekijken wat er precies gebeurt met de agents? Is het goed mogelijk om een staatwisseling te zien?
- **Batchrun**: Ook wel batch mode. Hoe goed is het programma in het runnen van batch mode? Batch mode is het verwerken/compilen etc. van de code in de achtergrond, terwijl de parameters steeds veranderen. Hierdoor verwerkt hij meerdere situaties snel achter elkaar waardoor je duidelijk het effect kan zien van verschillende parameters op een simulatie.

Planning

Week 1

Titel:	GUI
Omschrijving:	Als gebruiker wil ik een duidelijk overzicht waar ik ook de parameters kan veranderen zodat ik duidelijk kan zien voor welke parameters wat voor soort scenario's geven
DoD:	Agents worden duidelijk weergegeven en er is een mogelijkheid geïmplementeerd om parameters te veranderen
Prio	4 (1-5)
Tijdsduur:	8 uur

Titel:	Agents
Omschrijving:	Als agent wil ik kunnen kiezen welke partij het meest voldoet aan mijn standpunten. Dit wordt gedaan met de afstand tussen agent en party leader. Hierdoor krijgt elke agent een lijst van party leaders, gebaseerd op afstand.
DoD:	Als elke agent de mogelijkheid heeft om “toegewezen” te kunnen worden tot een partij
Prio	5
Tijdsduur:	2 uur

Titel:	Environment
Omschrijving:	Als gebruiker wil ik kunnen beïnvloeden in welke omgeving de agents geplaatst worden. Ik wil dus kunnen kiezen op welke plek mijn partijleiders terechtkomen en hoe de agents reageren op deze partijleiders. (op welke manier de agents zullen stemmen)
DoD:	De agents moeten een omgeving hebben om neergezet te worden, en deze omgeving moet beïnvloedbaar zijn door de gebruiker.
Prio	5
Tijdsduur:	1 uur.

Titel:	Staat van simulatie toonbaar
Omschrijving:	Als gebruiker wil ik kunnen zien wat de statussen zijn van agents binnen de environment zodat je kan zien wat de keuze/2^{e} keuze is van die agent.
DoD:	Wanneer je op een agent clicked dan moet er informatie over die agent worden getoond
Prio	3
Tijdsduur:	10 min

Titel:	Batchrun
Omschrijving:	Als gebruiker wil ik dat de simulatie meerdere keren kan runnen. Bovendien wil ik dat mijn computer dit op de achtergrond kan doen.
DoD:	De simulatie moet meerdere keren te runnen zijn, eventueel met andere instellingen.
Prio	2
Tijdsduur:	30 min

Titel:	Setup
Omschrijving:	Maakt aantal agents aan, agents eigenschappen, global variabele, plaats ze op oppervlakte en kiest verschillende agents als partyleader met kenmerkende eigenschappen.
Prio	4
Tijdsduur:	2 uur

Titel:	Agent keuze 1
Omschrijving:	Cleared eerst alle eerdere veranderde variabele en agents, roept de (vind de 1^{e} partij) functie aan en dan de functie countelections
Prio	4
Tijdsduur:	3 uur

Titel:	Agent keuze 2
Omschrijving:	Cleared eerst alle eerder veranderde variabele en agents, roept de (vind de 2e partij) functie aan en dan de functie countelections
Prio	4
Tijdsduur:	1 uur

Titel:	Vind de 1e partij
Omschrijving:	Elke agent gaat checken in zijn radius of er een partyleader is anders verbreed het zijn radius totdat het een partyleader vindt en dan zijn kleureigenschap veranderd naar de partyleaders en dan stopt de loop
Prio	4
Tijdsduur:	5 uur

Titel:	Vind de 2e partij
Omschrijving:	Elke agent gaat checken in zijn radius of er een partyleader is anders verbreed het zijn radius totdat het een partyleader vindt en dan zijn kleureigenschap veranderd naar de partyleaders, dit doet hij 2 keer (om de 2de partyleader te vinden)
Prio	4
Tijdsduur:	4 uur

Titel:	Clearboard (tijdelijk)
Omschrijving:	Tijdelijke functie om alle keuzes van kandidaten te resetten en daarna deze functie eigenschappen/code regels in de go functies toepassen
Prio	4
Tijdsduur:	1 uur

Titel:	CountElections
Omschrijving:	Zet de globale tellingen variabelen op nul, telt per agent hoeveel er per kleur is en zet het in een variabele. De variabelen worden uitgeprint in de output
Prio	
Tijdsduur:	3 uur

Titel:	GUI uitbreiding
Omschrijving:	Histogram van de zichzelf tellende code van het aantal kleuren en zet de in een plot met x-as de kleur code en y-as de hoeveelheid. En de output van CountElections waarin in exacte waardes te zien is welke kleuren meer votes hebben. En nog een klein notitieblokje om aan te geven welke output bij welke kleur hoort en de kleurcode nummer (white = 9.9)
Prio	4
Tijdsduur:	2 uur

Week 1 planning

Onderdeel	**Beschrijving**	**Weekdeel**	**Persoon**

PvA	User Stories opzetten	MA	Iedereen
PvA	Onderzoeksvraag en eerst inzicht uitwerken	MA	Iedereen
PvA	SF(A) modelkeuze met modules	MA	Iedereen
Simulatie	Setup + GUI + First	MA	Iedereen
Simulatie	Second + clearboard	MA-DI	Ruben
Simulatie	Vind 1e partij	DI	Guy & Koen
Simulatie	Vind 2e partij	WE-DO	Koen
Simulatie	Countelections + GUI uitbreiding	DO	Ruben
PvA	Laatste puntjes op de i	DO-FR	Guy & Koen
Simulatie	Laatste puntjes op de i	DO-FR	Ruben

Week 2

Titel:	Vooronderzoek
Omschrijving:	Onderzoek doen over vergelijkbare onderzoeken en hoe zij het hebben gedaan. Dit is ook voor het creëren van ideeën.
DoD:	Een uitleg/ bevinding stukje van meer dan 200 woorden. Over het voor onderzoek wat wij gedaan hebben
Prio:	1
Tijdsduur:	2 uur

Titel:	Onderzoeksvraag/ hypothese wordt toegelicht
Omschrijving:	Een kleine toelichting van de onderzoeksvraag en wat we ermee willen wijsmaken
DoD:	Een duidelijk stukje dat een goed inzicht geeft wat wij precies willen gaan onderzoeken.
Prio:	2
Tijdsduur:	30 min

Titel:	Voeg slider toe voor voters en partij leiders
Omschrijving:	Geeft de user de mogelijkheid om makkelijk de parameters voters en partyleaders te veranderen.
DoD:	De user kan de parameters voters en partyleaders via een slider veranderen
Prio:	2
Tijdsduur:	15 min

Titel:	Report functies toegevoegd
Omschrijving:	Meerdere Netlogo ‘report’ functies toevoegen, zoals bijvoorbeeld voor Netlogo kleuren conversatie.
DoD:	
Prio:	2
Tijdsduur:	30 min

Titel:	Loops toegevoegd voor herhalende code
Omschrijving:	Zorgt ervoor dat je extra parameters kan instellen voor veel-herhalende code, zoals create partyleader.

DoD:	Alle herhalende functies zijn uit de code gehaald, parameters toegevoegd waar nodig
Prio:	2
Tijdsduur:	1 uur

Titel:	Plurarity en Approval run code herschreven
Omschrijving:	Herschrijven van de twee run functies zodat ze veel korter en meer flexibel zijn.
DoD:	De twee functies werken als voor het herschrijven
Prio:	2
Tijdsduur:	2 uur

Week 2 planning

Onderdeel	**Beschrijving**	**Weekdeel**	**Persoon**
Simulatie	Basis van Approval toevoegen	MA	Ruben
Verslag	Planningen en sprints beschrijven	MA	Iedereen
Simulatie	Interface verbeteren	DI	Guy
Simulatie	Code overzichtelijkheid verbeteren	DI	Koen
Simulatie	Tijdelijke variabelen toevoegen	WE	Ruben
Simulatie	Comments toevoegen en verbeteren van Approval	DO-VR	Guy
Verslag	DOD afchecken	DO-VR	Iedereen

Week 3

Titel:	Elke kandidaat eigen keuze variable geven
Omschrijving:	Geef elke kandidaat een eigen lijst van voorkeur partijleiders, gebaseerd op afstand. Hierdoor kunnen we de satisfaction van een winnende partij berekenen.
DoD:	Elke turtle heeft een lijst met alle partijleiders, die op volgorde staan van dichtbij – ver weg.
Prio:	2
Tijdsduur:	1 uur

Titel:	Approval voting herschrijven met meerdere parameters
Omschrijving:	De Approval voting run functie herschrijven zodat hij meerdere parameters kan supporten, is nodig voor o.a. de hierboven genoemde lijst.
DoD:	Functie werkt hetzelfde als voorheen, maar support meerdere parameters.
Prio:	2
Tijdsduur:	1 uur

Titel:	Batchrun toevoegen
Omschrijving:	Batch run mode toevoegen aan ons programma, waardoor we meerdere runs achter elkaar kunnen doen met kleine aangepaste parameters. Hier kunnen we vervolgens onderzoek op gaan doen.
DoD:	Programma kan een batch run voltooien en de resultaten laten zien.
Prio:	2
Tijdsduur:	2 uur

Titel:	Compare voting systems met elkaar met behulp van dissatisfaction
Omschrijving:	De nieuw toegevoegde lijst gebruiken om te berekenen hoe goed elk voting system is aan de hand van de satisfaction.
DoD:	We kunnen satisfaction scores van alle voting systemen weergeven en vergelijken, o
Prio:	2
Tijdsduur:	30 min

Titel:	Comments en beschrijving toegevoegd
Omschrijving:	Comments in de code toevoegen, zodat duidelijk is wat alle functies doen. In tekstvorm.
DoD:	Als bij alle functies duidelijk vermeld staat wat het doet.
Prio:	3
Tijdsduur:	30 min

Week 3 planning

Onderdeel	**Beschrijving**	**Weekdeel**	**Persoon**
Verslag	Planningen en sprints beschrijven	MA	Iedereen
Simulatie	Hele simulatie herschrijven voor overzichtelijkheid	DI	Ruben
Simulatie	Approval scenario's toegevoegd	DI	Koen, Guy
Verslag	Laatste puntjes op de i	WO	Iedereen
Simulatie	Voting systems met elkaar vergelijken	WO-DO	Koen
Verslag	Laatste puntjes op de i	DO-VR	Iedereen

SF(A) model met modules voor tools

Scores zijn van 1-5, waarbij 1 het laagst is en 5 het hoogst haalbare is.

- Mesa: 29 punten
    - **Gui: 4**
      Je hebt een best wel moderne GUI en je kan ook het veranderen van parameters implementeren.
    - **Agents: 4**
      De agents zijn makkelijk op te zetten en makkelijk functies te maken voor de agents zelf.
    - **Environment: 4**
      Het environment is ook makkelijk op te zetten en best simpel.
    - **Staat van simulatie toonbaar: 4**
      Voor extra informatie moet je het wel zelf inprogrammeren, dus het weergeeft niet vanuit zichzelf de status van een agent als je erop klikt/erover heen hoveret.
    - **Batchrun: 4**
      Je zou makkelijk een batchrun kunnen uitvoeren in Mesa. Het is al geïmplementeerd in Mesa.
    - **Suitability: 4**
      Qua efficiëntie is Mesa redelijk. Je moet nog steeds wel aardig wat code schrijven om iets simpels te maken. Mesa heeft een aardige snelheid qua het runnen en

displaying van de code, omdat Mesa in python wordt gebruikt is de compatibility hoog
    - **Feasability: 5**
      Iedereen weet hoe ze moeten coderen in python, dus er hoeft niet echt een nieuwe taal aangeleerd te worden. Mesa is aardig goed te begrijpen. Technisch is het zeker haalbaar om in 2 week iets goed in elkaar te zetten.
- Unity: 25 punten
    - **Gui: 5**
      De GUI in Unity is heel uitgebreid en zorgt ervoor dat alle nodige info zichtbaar is, mits je weet waar je het moet vinden.
    - **Agents: 4**
      Agents zijn goed op te zetten in Unity.
    - **Environment: 4**
      Een environment opzetten in Unity is vrij makkelijk.
    - **Staat van simulatie toonbaar: 3**
      Afhankelijk van de simulatie is dit makkelijk of moeilijk. Als je een visuele indicator toevoegt wordt het al een stuk leesbaarder, maar het is moeilijker dan in de andere 2 talen
    - **Batchrun: 5**
      Batch is een zeer goed geïntegreerde command in Unity aangezien ze tijdens de development van Unity wisten dat compilen, runnen etc. lang zou kunnen duren.
    - **Suitability: 2**
      Suitability is niet erg hoog. In Unity zou onze simulatie niet efficiënt zijn, de compilatietijd zou vrij lang duren. Waarschijnlijk zou Unity beter geschikt zijn voor de snelweg-simulatie.
    - **Feasability: 2**
      Wij zijn alle 3 niet echt bekend met Unity/C#, en we zouden niet goed weten waar we zouden moeten beginnen. Technisch zou het waarschijnlijk wel haalbaar zijn, maar het zou langer duren dan nodig is en in andere talen is het vele malen makkelijker.
- Netlogo: 30 punten
    - **Gui: 5**
      De gui in Netlogo is al ingebouwd en staat als template al klaar om gebruikt te worden, het enige wat er gedaan moet worden is het aanmaken van turtles
    - **Agents: 4**
      Agents kunnen aangemaakt worden met 3 woorden, het is erg simpel om ze aan te maken en ze te inspecteren.
    - **Environment: 5**
      De basis van het environment wordt al aangemaakt als je een template aanmaakt en is daarom automatisch aangemaakt.
    - **Staat van simulatie toonbaar: 5**
      De agent informatie is erg makkelijk te zien door bij het environment op de agent te klikken en dan kan je het inspecteren en kan zien welke variabele met agent te maken hebben en welke waarde het heeft
    - **Batchrun: 3**
      Heeft BehaviorSpace i.p.v. batch mode. BS runt op een parallele core op je CPU in plaats van wanneer je PC in idle mode is. BehaviorSpace werkt wel goed maar is dus in werkelijkheid geen echte batch mode.
    - **Suitability: 4**

Heel efficient en snel, maar helaas niet compatible met externe libaries.

- **Feasability: 4**

We zijn allemaal in een korte tijd bekend geraakt met NetLogo, omdat het niet moeilijk is om te begrijpen. Hierdoor is ons project op een technisch niveau zeker haalbaar.

Conclusie tool keuze

Volgens het voting systeem wat wij boven hebben gebruikt. Is Unity de minst goeie oplossing voor een simulatie creëren. Netlogo is maar met een punt beter dan Mesa. Wij zullen dus Netlogo gaan gebruiken als tool.

Design/uitleg van het experiment (ongeveer 200 woorden)

Plurality voting systeem is het belangrijkste en meest gebruikte single vote stem systeem in de wereld (https://www.fairvote.org/research_electoralsystems_world) en het makkelijkste om te programmeren. Zo kan je simpelweg de radius van de partyleader/voters gebruiken om de juiste partij bij de juiste stemmer neer te zetten. Wat opviel is dat vooral partijen die alles prima vinden (in het midden) minder stemmen krijgt dan de extreme rechtse/linkse partijen. Wat wel vreemd is, omdat als de middelste partij gekozen komt er meer tevredenheid tussen de stemmers, maar in plaats daarvan gekozen wordt tussen extreme partijen en dus een grote kans is dat er een strijd plaatsvindt tussen extreme partijen. Approval stem systeem daarentegen geeft juist meer voorkeur naar deze middenpartijen, maar dat kan ook slecht aflopen waaruit het spreekwoord komt: als 2 honden vechten om een been loopt de derde ermee heen. Dus moet er scenario's worden gemaakt voor beide stem systeem om deze nadelen te verminderen. Zoals voor Plurality een herkenning voor een slechte (weinig gekozen) party en die stemmers laten twijfelen over hun keuze en overhalen naar een andere partij.

Het experiment bevat 5 soorten voting manieren die bestaan uit 3 manieren voor Plurality en 2 manieren voor Approval. We gaan Plurality en Approval tegen over elkaar zetten. Voor elk van die manieren gaan we ze 100 keer runnen met 4, 6 en 8 partijen en. Het aantal voters zal bij elke run altijd op 200 staan.

Resultaten van het experiment (ongeveer 200 woorden)

Om erachter te komen welke stem systeem het beste is, wordt er gekeken naar tevredenheid. In de simulatie wordt de ontevredenheid geteld en hoe minder een stem systeem ontevredenheid punten heeft hoe beter. We gaan dan one-rival-voting scenario van Approval en strategic-voting van Plurality met elkaar vergelijken, wanneer ze beide een andere partij kiezen. Wanneer dat het geval is wordt er gekeken welke gewonnen partij beter valt bij de stemmers. We testen dit meerdere malen om erachter te komen welke het beter doet.

De lijsten worden zo opgebouwd: [row points, column points, tie]. Onder de lijsten staan wie de hoogste tevredenheid heeft.

Groen = (second) best/strategy, **blauw** = one rival/only not the worst, **zwart** = gelijkspel

4 partij leiders	**One rival**	**Only not the worst**
Best	[1, 25, 74], One rival	[26, 43, 31], Only not the worst

Second best	[25, 44, 31], One rival	[7 , 27, 66], Only not the worst
Strategy	[1, 21, 78], One rival	[3, 35, 62], Only not the worst

6 partij leiders	**One rival**	**Only not the worst**
Best	[1, 32, 67], One rival	[27, 48, 25], Only not the worst
Second best	[41, 34, 25], Second best	[34, 25, 41], Second best
Strategy	[2, 24, 74], One rival	[27, 57, 16], Only not the worst

8 partij leiders	**One rival**	**Only not the worst**
Best	[3, 43, 54], One rival	[26,61,13], Only not the worst
Second best	[45, 35, 20], Second best	[51, 29, 21], Second best
Strategy	[1, 44, 55], One rival	[28, 57, 15], Only not the worst

Dissatisfaction results

4 party Plurality = 226,52

6 party Plurality = 378, 51

8 party Plurality = 566,20

4 party Approval = 212,07

6 party Approval = 369,98

8 party Approval = 526,10

Gemiddelde Plurality dissatisfaction = (226,52 + 378, 51 + 566,20) / 3 = 390,41

Gemiddelde Approval dissatisfaction = (212,07 + 369,98 + 526,10) / 3 = 369,38

Approval heeft een lagere gemiddelde dissatisfaction, dus Approval voting heeft een hogere tevredenheid.

Conclusie (ongeveer 200 woorden)

Van de 18 simulaties die er zijn gedraaid. Heeft second best er 4 van gewonnen, one rival 7 en only the worst 7. Dit geeft een duidelijke overwinning voor Approval.

Het Approval stem systeem scoort in de simulatie algemeen beter als het gaat om wie de minste gemiddelde dissatisfaction heeft, korter gezegd: de hoogste tevredenheid. Dit komt omdat er wordt gekeken welke partijen er nog meer in de buurt staan en die keuze wordt ook mee genomen. Dit geeft uiteindelijk dus een vrij neutrale partij het voordeel. Een vrij neutrale partij is zoals eerder vermeldt zeer voordelig als het gaat om tevredenheid. Qua Plurality blinkt second best voting uit, maar het is een hele kleine kans dat ook echt elke stemmer ook op zijn tweede voorkeur gaat kiezen in het echte leven. Wat het dus niet echt een goed en eerlijk scenario maakt. Dus we geven we de overwinning aan Approval, Rival one voting

Discussie (minimaal 200 woorden)

Onze hypothese was dat Approval zou winnen, omdat het kandidaten mogelijk maakt om op meerdere partijen te stemmen en dus voor een eerlijkere verdeling zorgt tussen de partijen. Uit de resultaten is dat inderdaad het geval. Waarom wordt er dan nog steeds meer Plurality gebruikt in de wereld? Dat kan wegens 2 redenen. Reden 1, omdat het nog niet goed is uitgetest en er veel verschillende resultaten eruit komen waardoor veel stem simulaties onbetrouwbaar worden. Reden 2, omdat het Plurality zo simpel is te gebruiken en makkelijk uit te leggen is. Het is simpelweg stemmen op jouw beste partij. Helaas in werkelijkheid ontstaan er veel strijden en gaan partijen om meer stemmen te vangen juist extremistischer worden. Ons onderzoek daarentegen is ook niet 100% betrouwbaar, zo hebben we namelijk gebruik gemaakt bij strategisch stemmen van een artikel waarbij 10% maar strategisch stemt, in werkelijkheid is dat een totaal ander percentage. Ook omdat de wereld zo groot is, kunnen er vele andere uitkomsten uitkomen, ondanks we het 100 keer hebben gerund, zal het misschien niet genoeg zijn. Dat is vooral voor later belangrijk, om te beseffen dat randomness erg belangrijk is om zo klein mogelijk te houden zodat je makkelijker het kan verminderen. Een goed voorbeeld van een beter onderzoek zou zijn door met een echte groep mensen te werken en dan kijken of Approval voting daadwerkelijk zorgt voor meer tevredenheid in de verkiezingen.

Referenties

1. Fieldhouse, E., Shryane, N., & Pickles, A. (2007). Strategic voting and constituency context: Modelling party preference and vote in multiparty elections. *Political Geography*, *26*(2), 159–178. https://doi.org/10.1016/j.polgeo.2006.09.005
2. Primer. (2020, 2 november). Simulating alternate voting systems [Videobestand]. Geraadpleegd van https://www.youtube.com/watch?v=yhO6jfHPFQU