

## Plan van aanpak

### Introductie:

Het gekozen onderwerp voor dit opdracht is de voting systeem en is gebaseerd op de video *Simulating alternate voting systeem*. In deze video worden 3 verschillende voting systemen gesimuleerd, namelijk de Plurality voting, de Instant runoff voting en de Approval voting. Wij gaan deze simulatie uitbreiden aan de hand van de volgende basishoofdvraag:

Wat is het effect van plurality voting op het aantal mensen dat niet op zijn eerste voorkeur kiest?

We willen met onze simulatie het verschil tussen plurality en approval voting aantonen waarbij een bepaald aantal stemmers kan stemmen op een aantal kandidaten.

We gaan hiervoor Mesa gebruiken, meer over de toolkeuze is terug te vinden in het SFA model.

### GUI:

De GUI zal twee sliders bevatten waar de gebruiker kan aangeven hoeveel kandidaten en stemmers in de scenario zal voorkomen.

### Verloop:

De gegeven stemmers worden gecreëerd met een positie in de opgeving. Een stemmer kan een eerlijke stemmer zijn of een strategische stemmer.

#### **Plurality voting**

Als een stemmer eerlijk zijn dan wordt de afstand berekend tussen die stemmer en de kandidaten. De kandidaat met de kortste afstand, zal de stemmer ervoor stemmen.

Een strategische stemmer zal, net zoals de eerlijke stemmer, stemmen voor de kandidaat met de kortste afstand. Er zal een poll uitgevoerd worden, waar de stemmers een idee heeft welke kandidaat het grootste kans heeft om te winnen. Een poll is het resultaat wanneer alle stemmers eerlijk gaat stemmen, dus stemmen voor de kandidaat met de kortste afstand

Een strategische stemmer zal stemmen voor de kandidaat met de kortste afstand een de grootste kans om te winnen.

#### **Approval voting**

Approval voting is het systeem waar kiezers elke kandidaat (of niet) kunnen stemmen, en de kandidaat die door de meeste kiezers is goedgekeurd, wint

Bij approval voting heeft de stemmer een *approval* range en alle kandidaten binnen die range, zal voor die kandidaat stemmen.

Voor onze simulatie zal elke kandidaat een range hebben en alle stemmers binnen die range, zal voor die kandidaat stemmen. Als een stemmer binnen twee of meer ranges zitten, dan zal die voor die kandidaten stemmen.

De tijd van de simulatie zal in rondes plaatsnemen, waar elke ronde de stemmer weet wie de grootste kans heeft om te winnen. Hierdoor kunnen de stemmers hun keuze veranderen en de resultaten zal veranderen.

Er zal een batch run uitgevoerd waar de simulatie verschillende keren wordt gerund. De resultaten zullen opgeslagen worden, zodat we deze data kan verwerken, analyseren en visualiseren.

Er zal een batch run

Tijd rondes- poll

Batch run- verschillende keren runnen.

## Visualisatie:

We gaan de simulatie visualiseren in Mesa, bij mesa krijg je al een gegeven GUI waarin je de simulatie kunt runnen binnen je browser. We gaan de simulatie tonen op een multigrid, hier zullen verschillende kandidaten en agents geplaatst worden, de agents brengen vervolgens hun stem uit op een van de kandidaten. Vervolgens willen we ook kunnen visualiseren wat er gebeurt binnen de simulatie, dit gaan we doen aan de hand van grafieken van python libraries zoals matplotlib en numpy.

## Dataverzameling:

Omdat tijd minder van belang is binnen onze simulatie gaan we geen grafiek visualiseren tijdens het runnen van de simulatie, maar erna. Wanneer we de simulatie gerunt hebben wordt de data verwerkt en willen we het op kunnen slaan. Wanneer de data verwerkt is willen we, waarschijnlijk in de vorm van een staafdiagram laten zien hoe de stemmen van de agents verdeeld zijn over de kandidaten.

Verwerken van de data- data verzamelen + statische analyse

Er m

Je kan de interne staat van een agent zien door er op te klikken,  
Hier zie je dan zijn voorkeur naar alle kandidaten maar ook de uiteindelijk uitgebrachte stem.

Verslag schrijven

Presentatie