PWS - Genetische Algoritmen

Julia Bertsch en Lianne Weber

5 juli 2019

1 Samenvatting

Inhoudsopgave

2 Voorwoord

3 Inleiding

4 Onderzoeksplan

4.1 Onderzoeksvraag

- Welke resultaten ontstaan er als je een populatie volgens een natuurlijk systeem laat evolueren in Javascript?
- Hoe beinvloeden *snelheid* en *sense* de organismen in een populatie die je volgens een natuurlijk systeem laat evolueren in Javascript?

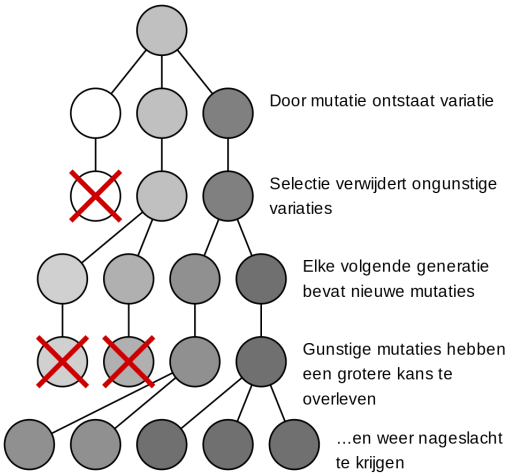
4.2 Deelvragen

- Wat is Javascript en wat is de p5.js library?
- Wat houd evolueren in?
- Wat is Cellular Automata (CA) en wat is de functie hiervan?
- Wat zijn Genetische Algoritmen (GA) en hoe pas je deze toe?
- Hoe kan je genetische algoritmen gebruiken om een populatie te laten evolueren in een computerprogramma?

5 Theoretisch kader

5.1 Evolueren

Evolutie is een geleidelijk proces waarbij populaties veranderen, ontstaan en verdwijnen. Een populatie is een groep individuen van dezelfde soort die zich onderling kunnen voortplanten. Daarom leeft een populatie in hetzelfde gebied. Een populatie evolueert door genetische veranderingen waardoor de organismen van de populatie andere eigenschappen vertonen. Een genetische verandering kan ontstaan door fouten tijdens de DNA-replicatie en door geslachtelijke voortplanting. De verandering ontstaat in een gen, stukje DNA dat bestaat uit erfelijke eigenschappen. [1]



[2]

5.2 Neo-Darwinistische evolutietheorie

De ideen van Charles Darwin en de erfelijkheidsleer van George Mendel zijn gecombineerd in de neo-darwinistische evolutietheorie. Deze theorie gaat uit van natuurlijke selectie, genetische variatie en reproductieve isolatie. [3]

5.2.1 Natuurlijke selectie

Organismen die beter zijn aangepast aan hun omgeving hebben een grotere overlevingskans. Een organisme is beter aan zijn omgeving aangepast als het gunstige eigenschappen heeft. Deze organismen kunnen hun genetische informatie doorgeven aan hun nakomelingen. Organisme met gunstige eigenschappen leven langer. Hierdoor kunnen ze meer nakomelingen maken met de gunstige eigenschappen. De organisme die niet of nauwelijks zijn aangepast

aan hun omgeving gaan eerder dood. Zij maken dan ook minder nakomelingen. Dit zorgt ervoor dat er in de volgende generatie meer organismen zijn met gunstige eigenschappen.

5.2.2 Genetische Variatie

De verscheidenheid aan eigenschappen in een populatie. Om te kunnen evalueren moeten organismen verscheidenheid aan eigenschappen hebben. Als er geen variatie is in een populatie zouden alle nakomelingen identiek zijn aan de ouders. Zo kan de populatie niet veranderen als het milieu dit wel doet. Op verschillende manieren kan er genetische variatie ontstaan. Tijdens geslachtelijke voortplanting worden de genen van de ouders met elkaar gecombineerd. Zo ontstaan er nieuwe combinaties van allelen.

5.2.3 Reproductieve isolatie

Als het milieu waarin een soort leeft veranderd in meerdere milieus, kan het zijn dat er een deel van de soort niet meer bij elkaar kan komen. Dit zorgt ervoor dat er geen voortplanting meer plaatsvindt tussen de organisme van de soort. Hierdoor kunnen er nieuwe soorten ontstaan.

5.3 Javascript

JavaScript is een veel gebruikte programmeertaal die wordt uitgevoerd op de webbrowser. De programmeertaal wordt vaak gecombineerd met HTML (HyperText Markup Language) en CSS (Cascading Style Sheets). De Syntax van JavaScript lijkt heel erg op die van java, dit komt vooral omdat beide afkomstig zijn van de programmeertaal C. Een belangrijk verschil is het soort programmeertaal. JavaScript is een functionele programmeertaal [4], waarbij informatie wordt doorgegeven door middel van prototype-gebaseerd-overerving [5]. Java is een objectgeorinteerde programmeertaal[6]. [7]

5.3.1 Libraries

Javascript bevat voorgemaakte objecten/data-elementen met methodes of functies. Tegenwoordig is veel nuttige informatie verzameld in JavaScript-libraries. Dit zijn tools die het programmeren gemakkelijker maken door objecten en data-elementen toe te voegen, die je anders zelf zou moeten maken. Voorbeelden van bekende libraries zijn tensorflow.js, node.js en react.js.

Een JavaScript-library waar wij gebruik van gaan maken zijn p5.js en chart.js. P5.js is een JavaScript-library waarbij gebruik wordt gemaakt van functionaliteit bij Processing. Processing is een op Java gebaseerde programmeertaal voor ontwikkelaar die het gemakkelijker maakt om dingen weer te geven op een scherm. Chart.js is een library die het maken van grafieken versimpeld. [8] [9]

5.4 Genetische Algoritmen

Genetische Algoritmen zijn algoritmen binnen de categorie kunstmatige intelligentie. Deze algoritmen worden gebruikt voor optimalisatie- en zoekproblemen. Er zijn veel verschillende evolutionaire algoritmen die ideen uit de evolutietheorie gebruiken, zoals *evolution strategies*.

Genetische algoritmen zijn een vorm van *reinforcement learning*. *Reinforcement learning*, de combinatie van *supervised learning* en gericht data verzamelen, is een van de meest interessante toepassingen van machine learning. Hierin wordt gewerkt met beloningen in plaats van straffen.Een genetisch algoritme heeft een populatie met objecten/elementen met oplossingen voor een bepaald probleem. Deze objecten zijn individuen of chromosomen en door het combineren van de genen van deze objecten kan je een betere oplossing vinden. Wel is het belangrijk een goede fitness-functie op te stellen die de kwaliteit van een oplossing bepaald. [10]

5.4.1 Een genetisch algoritme doorloopt de volgende stappen

1. Initializatie van de populatie
2. Evaluatie van de populatie
3. Selectie
   (a) Selecteer ouders
   (b) Genereer nakomelingen door middel van crossover van de genen van de ouders
   (c) Muteer en evolueer de nakomelingen
   (d) Bepaal de nieuwe beste populatie

5.4.2 Er zijn 2 hoofdstromen voor het evalueren van de populatie

1. *Generationeel*: De nakomelingen vervangen de ouders in een populatie
2. *Steady-state*: De populatie bestaat uit de beste nakomelingen en ouders

5.4.3 *Genetic Operator*

De belangrijkste variatie zit in de *genetic operators*:

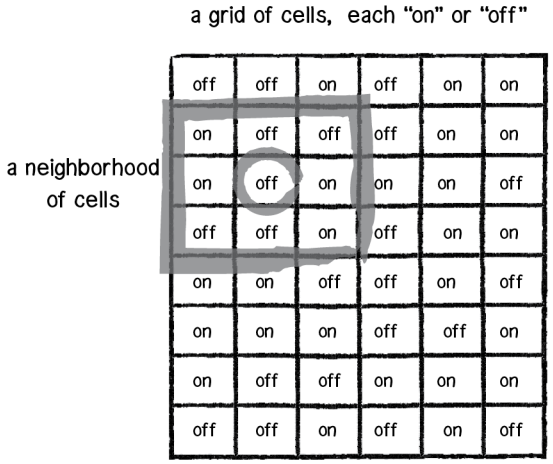
1. Selectie: Hoe kies je de populatie voor de populatie
2. Crossover: Hoe worden de ouders gecombineerd
3. Mutatie: Hoe wijzig je individuen met een *mutation rate* [11]

Selectie wordt vaak met behulp van een roulettewiel. De kans dat een individu gebruik wordt is evenredig met zijn fitness. Hierom is de fitness-functie belangrijk, zodat je de best mogelijke populatie kan vormen. [12]

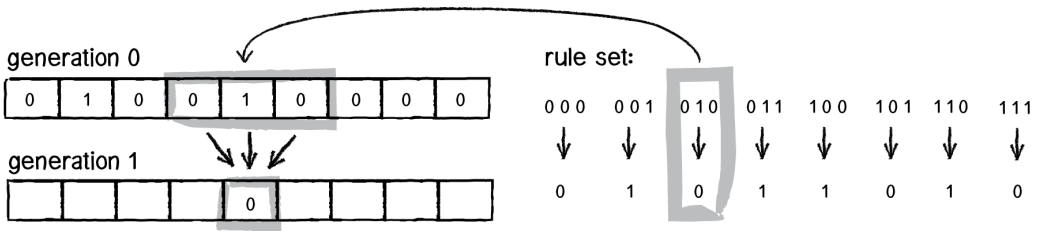
5.5 Cellular Automata

Cellular Automata is een eenvoudig model van een cel. Het wordt gebruikt voor onderzoek naar zelforganisatie in de statische mechanica. Cellular Automata wordt afgekort met de letters CA. CA bestaan uit een *grid*. De cellen leven in een hokje en elke cel heeft een eigen *state*, deze is bijvoorbeeld 0 of 1. 0 staat voor uit en 1 staat voor aan. Dit is een van de makkelijkste *grids*, omdat er twee mogelijkheden zijn. Een hokje heeft in het *grid* een aanliggende cel. [13]

De state van de cel wordt bepaald door de aanliggende cellen.



Figuur 1: Nature of Code: Chapter 7 - Cellular Automata [14]



Figuur 2: Nature of code: Chapter 7 - Cellular Automata [14]

De makkelijkste *grid* is een lijn van cellen. Deze bestaan maar uit twee mogelijkheden, namelijk aan (1) en uit (0). Elke cel heeft alleen een aanliggende cel links en rechts van zich. Deze eerste lijn is generatie 0. Bij programmeren beginnen ze met tellen bij nul. De volgende generatie wordt bepaald door de vorige generatie. Dit is ook zo met genen. De *state* kan worden vergeleken met de genen. Een CA wordt dus meegenomen naar een GA.

6 Methode

6.1 Het onderzoek

Om te onderzoeken hoe het evolutieproces wordt benvloed, gaan wij een kwantitatief onderzoek uitvoeren. Hierbij wordt het effect van verschillende factoren op de populatie bestudeerd. We gaan een computerprogramma gebruiken om deze populatie te simuleren. We willen dat de *output* van dit computerprogramma grafieken weergeeft, die het evolutieproces tonen op basis van de door ons gekozen factoren.

6.2 Materialen

- Computer of laptop
- Text editor
  - Atom
  - Visual studio code
- Javascript-libraries
  - p5.js
  - Charts.js
- Github

6.3 Programma van eisen

- Het ecosysteem moet kunnen evolueren
- Een ecosysteem moet bestaan uit een populatie
- Een populatie moet uit meerdere dots bestaan
- De dots moeten uit DNA bestaan
- Er moet selectie, cross-over en mutatie plaats kunnen vinden
- Het DNA moet de factoren snelheid en sense hebben
- Het moet data geven dat in grafieken kan worden verwerkt

6.4 Functieomschrijving

Dit onderzoek is een ontwerp onderzoek. We willen dat het computerprogramma dat we gaan ontwerpen voldoet aan de volgende eigenschappen.

De populatie gaan we uit dots laten bestaan. Een dot is een organisme. Deze populatie bevindt zich in een virtueel ecosysteem. Om de populatie te laten evolueren gaan we verschillende generaties gebruiken. De eerste generatie willen we random gaan maken. De opvolgende generaties willen we gaan maken door middel van selectie, crossover en mutatie. In dit ecosysteem gaan we random voedsel-objecten neerleggen. We willen twee verschillende soorten voedsel-objecten maken. Als een organisme geen voedsel-object vindt, gaat deze dood. Als een organisme 1 voedsel-object vindt, blijft deze leven maar doet voor de rest niks. Als een organisme 2 voedsel-objecten vindt blijft deze leven en vormt deze een nakomeling. Om een organisme een nakomeling te laten vormen, hebben we DNA nodig voor een organismen.

Elk organisme heeft een eigen DNA object. Dit DNA bevat een lijst van genen. De eigenschappen van het DNA worden tijdens reproductie doorgegeven aan de volgende generatie. De genen bevatten de factoren van een object. Wij willen de factoren snelheid, sense en voedsel als eigenschappen toevoegen aan de organismen. De snelheid van een organisme is de bewegingssnelheid. Sense is de radius vanuit het organisme waarin de organismen een voedsel-object kunnen zien.

Met de factor voedsel willen we vaststellen welk van de twee soorten voedsel het organisme eet. Als het organisme alleen het vierkante voedsel kan eten, kan hij niet het driehoekige voedsel eten. Dit is te vergelijken met planteneters en vleeseters.

De twee factoren willen we doorgeven aan de volgende generatie. Hierbij willen we dat er selectie, crossing over en mutatie optreedt. We willen het ecosysteem ongeveer 250 generaties laten evolueren.

6.5 Plan van aanpak

1. Basiskennis opdoen over het evolutieproces met computersoftware met behulp van het boek The Nature of code van Daniel Shiffman. [14]

2. Basiskennis opdoen van de software door gebruik te maken van instructievideo's op Youtube van The Coding Train. [15]

3. Een organisme maken met de eigenschappen snelheid en sense.
4. Voedsel maken.
5. Een ecosysteem maken met een organisme en voedsel.
6. Het organisme richting het voedsel laten bewegen zodra het voedsel in de sense van het organisme komt.
7. Het organisme zo programmeren dat het dood kan gaan of kan overleven.
8. Het organisme zo programmeren dat het kan reproduceren met selectie, crossover en mutatie.
9. De populatie uitbreiden met meer organisme.
10. Het ecosysteem zo programmeren dat het er grafieken als output komen.

6.6 Analyseren van de gegevens

Wij willen dat het computerprogramma de gegevens verwerkt in grafieken. In deze grafieken willen we alle factoren tegen elkaar uit kunnen zetten. Door deze grafieken af te lezen hopen we een maat te vinden van alle factoren die het meest gunstig is voor de organismen.

7 Resultaten

8 Conclusie en discussie

Referenties

[1] "Wat is evolutie?," Oct 2006.

[2] Wikipedia, "Evolutietheorie — Wikipedia, the free encyclopedia." http://nl.wikipedia.org/w/index.php?title=Evolutietheorie&oldid=54122046, 2019.

[3] "Darwins evolutietheorie," May 2019.

[4] Wikipedia, "Functioneel programmeren — Wikipedia, the free encyclopedia." http://nl.wikipedia.org/w/index.php?title=Functioneel\%20programmeren&oldid=50736465, 2019.

[5] Wikipedia, "Prototype-gebaseerd programmeren." http://nl.wikipedia.org/w/index.php?title=Prototype-gebaseerd\%20programmeren&oldid=45112996, 2019.

[6] Wikipedia, "Objectgeorinteerd — Wikipedia, the free encyclopedia." http://nl.wikipedia.org/w/index.php?title=Objectgeori\%C3\%ABnteerd&oldid=53318840, 2019.

[7] Wikipedia, "JavaScript." http://nl.wikipedia.org/w/index.php?title=JavaScript&oldid=53771654, 2019.

[8] L. McCarthy, "p5.js Reference." https://p5js.org/reference/.

[9] V. 2.8.0, "Chart.js - graphing api voor javascript."

[10] R. . [RN], "Kunstmatige Intelligentie 12: Genetische Algoritmen." 2019.

[11] Wikipedia, "Mutation rate." http://en.wikipedia.org/w/index.php?title=Mutation\%20rate&oldid=899174264, 2019.

[12] Wikipedia, "Genetic operator." http://en.wikipedia.org/w/index.php?title=Genetic\%20operator&oldid=677152013, 2019.

[13] F. Berto and J. Tagliabue, "Cellular automata," in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), Metaphysics Research Lab, Stanford University, fall 2017 ed., 2017.

[14] D. Shiffman, *The Nature of Code: Simulating Natural Systems with Processing*. Daniel Shiffman, 2012.

[15] D. Shiffman, "The coding train." https://thecodingtrain.com/.