

## Transparanten bij het Inleiding Adaptieve Systemen. Artificial Life. M. Wiering

---

### Artificial Life

Artificial life houdt zich bezig met het modelleren en simuleren van levende entiteiten en complexe systemen. Applicaties hiervan kan men vinden in:

- Biologie. Hoe interacteren levende organismen in biologische processen?
  - Economie: Hoe interacteren rationele agenten in economische omgevingen zoals onderdeel markten, e-commerce, etc.
  - Sociologie: Hoe interacteren agenten in gemeenschappen als ze competitieve/ collaboratieve doelen hebben?
  - Natuurkunde: Hoe interacteren fysische deeltjes in een bepaalde ruimte?
- 

ALife probeert de oorsprong en functionaliteit van het leven te begrijpen.

ALife probeert kunstmatige organismen te ontwerpen welke betekenisvol levend genoemd kunnen worden.

ALife probeert interactie patronen tussen organismen te begrijpen.

ALife probeert multi-agent systemen te construeren welke een bepaald multi-agent gedrag modelleren, nabootsen, en eventueel optimaliseren.

Voorbeelden:

- Markt economieën
- Sociale systemen
- Imuun systemen
- Ecosystemen

*Vraag: Wat is het verschil tussen AI en ALife?*

---

### GA en ALife

Genetische Algoritmen worden gebruikt voor diverse doeleinden:

- Optimalisatie. Bv. Numerieke optimalisatie, job-shop scheduling.
  - Robot leren.
  - Economische modellen. Zoals ontwikkeling van prijs-strategieën
  - Ecologische modellen. Zoals arm-races, co-evolutie
  - Populatie genetics modellen. Welke genen blijven in populatie?
  - Bestuderen van interacties tussen evolutie en leren
  - Modellen van sociale systemen. Zoals evolutie van cooperatie.
- 

### Interactie tussen evolutie en leren

Evolutie en leren zijn twee adaptie processen voor organismen. Evolutie is langzaam en vindt plaats over generaties van individuen. Leren is snel en vindt plaats in een individu (agent).

Er zijn 2 mogelijke effecten van de combinatie van evolutie en leren.

- Baldwin Effect. Een individu leert door zijn interactie met de omgeving. Dit leren stuurt de fitness van het individu. Hierdoor kunnen organismen die goed kunnen leren een hogere fitness behalen. Geleerde kennis wordt niet overgedragen aan de afsprong.
- Lamarckiaans leren. Een individu leert tijdens zijn leven en geeft de geleerde kennis door aan zijn afsprong.

Onderzoek toont aan dat leren en evolutie tegelijkertijd beter kan werken dan enkel 1 van beiden.

---

### Ecosystemen en evolutionaire dynamica

Ecosystemen bestaan uit een aantal individuen (agenten) welke:

- Een positie in de omgeving bezetten
- Interacteren met de omgeving en andere agenten

- Een bepaalde interne toestand (bv. energie, geld) hebben.

Door de evolutie van een ecosysteem te bestuderen kan men allerlei analogieën vinden met echte ecosystemen. Voorbeelden hiervan zijn:

- Cooperatie. Bv. handel tussen agenten.
- Competitie. Bv. gevechten tussen agenten.
- Nabootsen. Bv. een agent leert wat hij moet doen door te kijken naar een andere agent.
- Parasitair gedrag. Een agent profiteert van een andere agent.
- Evolutie van gemeenschappen. Binnen gemeenschappen kunnen agenten samenwerken en zich specialiseren.

---

## Cellulaire Automaten

Cellulaire Automaten (CA) zijn gedecentraliseerde ruimtelijke systemen met een groot aantal simpele identieke componenten met lokale verbondenheid.

Ze worden gebruikt als modellen voor biologische, sociale en fysische processen, zoals:

- Vloeistof beweging
- Galaxy formaties
- Aardbevingen
- Biologische patroon vorming
- Emergent cooperatief en collectief gedrag
- Verkeers modellen
- Bosbranden

---

## Formele beschrijving CA

CA bestaan uit 2 componenten:

- **De cellulaire ruimte.** De cellulaire ruimte is een lattice van  $N$  identieke cellen, elk met identieke patronen van lokale connectiviteit naar andere cellen.

Laat  $\Sigma$  de verzameling toestanden zijn van een cel.  $k = |\Sigma|$  is het aantal toestanden per cel.

Een cel met index  $i$  is op tijdstip  $t$  in toestand  $s_i^t$ . De toestand  $s_i^t$  tezamen met de toestanden van de cellen waarmee  $i$  verbonden is wordt de neighborhood  $n_i^t$  genoemd.

- **De transitie regel.** De transitie regel  $r(n_i^t)$  geeft de update van toestand  $s_i^{t+1}$  als een functie van zijn neighborhood.

Gewoonlijk worden alle cellen synchroon geupdate.

De regel wordt meestal geïmplementeerd als een lookup-tabel.

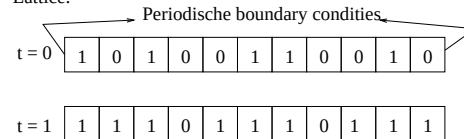
---

## Voorbeeld CA

Regel Tabel R:

|               |     |     |     |     |     |     |     |     |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Neighborhood: | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| Output bit    | 0   | 1   | 1   | 1   | 0   | 1   | 1   | 0   |

Lattice:



*Vraag: bepaal de nieuwe configuratie op tijdstap  $t=2$*

---

## Werking in tijd van CA

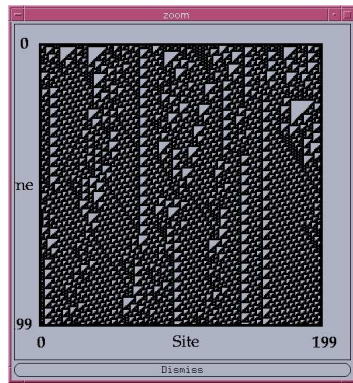
De bovenstaande CA met 1 dimensie en een neighborhood grootte van 3, is 1 van de simpelste CA.

Toch kan zo'n CA complex gedrag vertonen als we deze voortdurend door itereren:

CA zijn onderdeel van de klasse iteratieve netwerken of automata netwerken.

---

## Verschillende soorten processen in CA



Als we een gegeven CA met een begintoestand door itereren kunnen we verschillende soorten patronen verkrijgen:

- 1 stabiele eindtoestand, waarna de evolutie (verandering) ophoudt.
- 1 cyclisch patroon. Hier bevindt de CA zich achtereenvolgens in verschillende configuraties, maar deze configuraties vormen een cyclus, zodat dezelfde toestanden steeds terugkeren
- Chaotisch gedrag. De configuratie blijft veranderen en komt nooit terug in dezelfde toestand (NB. dit kan in principe enkel voor oneindig grote automaten)
- Sommige initiele patronen resulteren in complexe lokale structuren, welke soms lang doorgaan. Hierin zit vaak de meeste informatie.

We noemen de tijdsperiode totdat er een fixed point of periodieke cyclus optreedt de transient period.

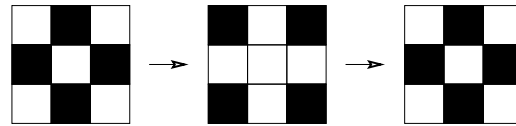
Ergens op de grens is er een fase transitie van complex geordend gedrag naar chaotisch gedrag.

### Voorbeeld van cyclische patronen

Een stabiele eindtoestand kan b.v. enkel 1'en zijn. Transitie regels met zulke eindtoestanden zijn makkelijk op te stellen.

Stel we hebben een 2-dimensionele lattice. De transitie regel is: als 2 burens actief zijn, wordt de cel geactiveerd. Anders gaat de cel uit.

Zie hieronder een lattice zonder boundary condities (we tonen eigenlijk maar een klein gedeelte van de complete lattice welke verder leeg is)



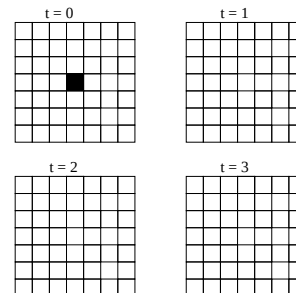
Dit leidt tot een periodieke cycle van lengte 2 voor bovenstaande initiele configuratie.

*Vraag: kan bovenstaande transitie regel volgens jou tot chaotisch gedrag leiden?*

### Opgave

Gegeven een 2-dimensionele lattice en de update-regel: als 1 buur actief is en cel was inactief maak cel actief. Anders als cel actief was op tijdstap  $t$ , maak cel ook actief op volgende tijdstap.

Evolveer de CA in onderstaande figuur:



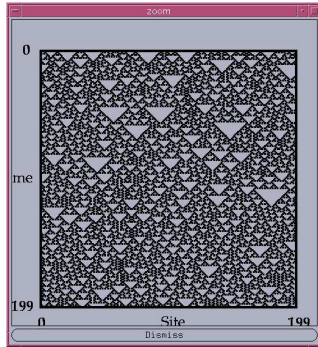
*Vraag: wat voor vorm komt eruit als we dit proces voortdurend door itereren op een grotere lattice?*

### Eliminatie van de patroon basis

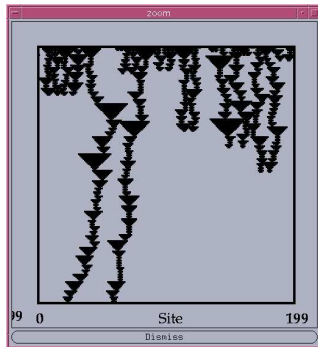
Sommige onderzoekers proberen een basis te vinden welke een achtergrond vormt (muren, singulariteiten) etc.

Door de reguliere basis weg te halen kan inzicht gegeven worden in de evolutie van chaotische of turbulente processen.

Neem het volgende chaotische proces:



Als we de reguliere achtergrond weghalen verkrijgen we:



De “embedded particles” vertonen een “random walk” gedrag.

Als de “embedded particles” elkaar kruisen, worden ze vernietigd.

### Onderzoek naar CA

CA zijn universele computatie machines. Ze zijn dus net zo krachtig als Turing Machines.

Er is veel onderzoek naar CA. Een bekend voorbeeld van een CA is Conway’s game of life.

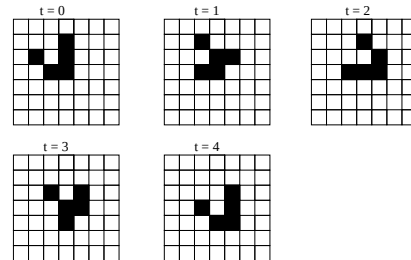
Conway’s game of life is een 2-dimensionele lattice met 8 burens voor elke cel. Verder zijn de transitie regel(s):

- Als een cel actief is en 2 of 3 van zijn andere burens zijn actief, dan wordt de cel actief.
- Anders als de cel inactief is en precies 3 burens zijn actief, wordt de cel ook actief.

- Anders wordt de cel niet actief.

### Universele rekenkracht in Game of Life

Conway’s game of life heeft ook universele rekenkracht. Deze rekenkracht berust op gliders welke informatie kunnen doorpropageren en waarmee logische and en not-gates gebouwd kunnen worden.



Deze glider beweegt zich 1 hokje diagonaal na elke 4 tijdstappen.

Door gebruik te maken van glider guns kunnen interacties tussen gliders plaats vinden.

Voorbeeld: Als gliders elkaar treffen worden ze vernietigd. Dit wordt gebruikt om een not-gate te maken.

### Ontwikkeling van CA

Von Neumann (1966) construeerde een CA welke in staat was om zichzelf te reproduceren.

Smith (1972) construeerde een CA welke context-sensitieve talen kon herkennen (bv. palindromen).

Mitchell et. al (1994) gebruikten GA om CA regels te evolueren. Ze probeerden dit uit op het meerderheidsprobleem in 1-dimensionele CA.

Dit probleem is: als de meerderheid van de cellen aan staat moeten na een periode van tijd alle cellen aangaan. Anders moeten alle cellen uitgaan.

De GA vonden hiervoor verschillende oplossingen. Sommigen maakten gebruik van embedded particles. Door de lokale verbondenheid kon echter geen oplossing voor alle initiële configuraties het juiste antwoord genereren.

## Andere CA

CA kunnen ook goed gebruikt worden voor problemen als het modelleren van verkeer. Hier is een cel b.v. actief als er een auto op een plek staat.

De CA kan complexer worden door elke auto een interne toestand mee te geven met de plaats van bestemming. (Dit kan in de cel zelf gecodeerd worden)

CA kunnen ook gebruikt worden om epidemieën mee te modelleren. Hier kan een cel een ziek, gezond, of ziek-geweest persoon zijn.

CA kunnen ook gebruikt worden voor bosbranden. Hier kan een cel zich bv. in de toestanden (Brand, Water, Bomen, Gras, Zand) bevinden.

Ook kunnen extra externe parameters zoals wind het gedrag van de CA beïnvloeden (eventueel door dynamische transitie regels)

---

## Voorbeeld Strategic Bugs

Strategic bugs is een ALife model (Bedau and Packard, 1992) waarin:

- De omgeving een 2-dimensionale lattice is.
- Een cel in de omgeving een agent of voedsel kan bevatten.
- Voedsel automatisch groeit in de omgeving: af en toe komt er voedsel in een willekeurige cel
- Bugs (agenten) overleven door het vinden en eten van voedsel.
- Bugs gebruiken energie om te bewegen en sterven als ze geen energie meer hebben.
- Bugs kunnen zichzelf klonen of met een andere bug offspring maken.

---

Een bug wordt gemodelleerd door een lookup-tabel representatie. Een voorbeeld regel is: als er meer dan 5 voedsel units ten noordwesten zitten, maak dan een stap naar het noordwesten.

Bedau en Packard bestuderen de evolutionaire activiteit: in welke mate worden nieuwe genetische vernieuwingen tot stand gebracht in de populatie?

De experimenten tonen dat golven van activiteit over tijdsperiodes worden geevolueerd. Dus de populatie vindt en exploiteert voortdurend nieuw genetisch materiaal.

Hoewel het model vrij simpel is, wordt de evolutionaire activiteit als een bruikbare variabele gezien om evolutie processen te meten.

---

## Markt modellen

Financiële markten zijn moeilijk te voorspellen. De vraag is: onder welke omstandigheden zijn voorspellingen überhaupt mogelijk?

Een uitgangspunt is de “Efficiënte Markt Hypothese” (EMH)

In een informatieve efficiënte markt zijn prijsveranderingen onvoorspelbaar als er behoorlijk rekening mee is gehouden. Dat is als ze volledig de verwachtingen en informatie van marktdeelnemers incorporeren.

Hieruit volgt: hoe efficiënter een markt is, des te willekeuriger zijn de prijsveranderingen die door de markt gegenereerd worden.

Dit komt omdat zelfs bij de kleinste informatieve voordelen van investeerders, de acties van die investeerders direct de mogelijke winst op die aandelen elimineren.

---

## Zijn markten voorspelbaar?

Maar: 1 van de neigingen van investeerders is de trade-off tussen verwachte risico's en verwachte winst.

Hierdoor kunnen bv. risk-aversieve investeerders aandelen verkopen met een hoog risico maar ook een verwachte winst.

Het gevolg is dat niet alle prijsfluctuaties willekeurig zijn.

Bepaalde studies hebben inderdaad aangetoond dat prijsfluctuaties niet volledig random zijn.

*Vraag: denk je dat computer algoritmen in staat zijn om aanzienlijke winsten te maken op de aandelenmarkt?*

---

## Niet rationele en volledig geïnformeerde agenten

De hypothese dat investeerders volledig ratio-

neel zijn en alle benodigde informatie bevatten is niet realistisch.

Dit komt omdat menselijk gedrag onvoorspelbaar is, informatie moeilijk te interpreteren valt, technologieën en instituten veranderen, en er kosten verbonden zijn aan transacties en informatie vergaring.

Een veelbelovende richting is de relatieve efficiëntie van een markt in vergelijking met andere markten. B.v. future-contracten versus opties.

Net als een koelkast met een efficiëntie van 40% behoorlijk goed lijkt en geprefereerd wordt ten opzichte van een koelkast met een efficiëntie van 35%, worden ook bepaalde markten geprefereerd.

---

### Modellen voor financiële theorieën

De wens om financiële theorieën te ontwikkelen heeft geleid tot verschillende benaderingen om de markt te verklaren:

- Psychologische benaderingen ten opzichte van risico-neem gedrag van investeerders. Hierin wordt geanalyseerd hoe de menselijke psychologie het economische beslissingsproces beïnvloedt.
- Evolutionaire speltheorie. Hierin wordt de evolutie van stabiele toestanden van populaties van competitieve marktstrategieën bestudeerd. Dit gebeurt meestal in gesimplificeerde, ideale gesimuleerde markten.
- Agent-gebaseerde modellering van markten. Hier worden investeerders gemodelleerd als agenten die bepaalde strategieën gebruiken. Dit maakt het mogelijk om complex leergedrag en dynamica in financiële markten te bestuderen.

---

### Agent-gebaseerde modellen

Een veelbelovende richting is de agent-gebaseerde modellering.

Markt-deelnemers zijn computationele entiteiten welke strategieën gebruiken op beperkte informatie.

Sommige markt-deelnemers maken verlies en anderen winst.

Verdienstelijke strategieën accumuleren geld, terwijl verliesgevende strategieën geld verliezen en tenslotte verdwijnen.

Dit kan dus gedaan worden m.b.v. genetische algoritmen.

De creatie van nieuwe strategieën kunnen andere strategieën minder/meer aantrekkelijk maken.

Een financiële markt kan dus gezien worden als een co-evoluerende ecologie van handels strategieën.

---

### Reinforcement leren voor agent-modellen

Reinforcement leren kan gebruikt worden om agenten te laten leren van hun acties. Als bepaalde strategieën winst opleveren, worden ze meer gebruikt in de toekomst.

De kern van zo'n agent is de mapping van partiële informatie naar handelingen. Deze mapping wordt aangepast door interactie met de omgeving (de markt).

Reinforcement leren stelt een agent dus bij tijdens zijn leven, terwijl GA een agent doen verdwijnen of kopiëren, zodat bepaalde winstgevendende strategieën meer in de populatie zullen komen.

Om markt-modellen realistisch te maken, kunnen ook allerlei financiële indexen, technologische innovaties, fusies etc. gemodelleerd worden in de simulatie.

---

### Artificial Art en Fractals

Het itereren van een simpele functie kan complexe, artistieke patronen genereren.

Een fractal is een patroon welke op zijn eigen subpatronen lijkt en welke vaak gemaakt kan worden door simpele vergelijkingen

Beschouw de functie:

$$x_{k+1} = x_k^2$$

Kijk naar de startwaarden voor  $x_k$  waarvoor de iteratie naar een stabiel punt convergeert

We kunnen direct zien dat voor de waarden  $-1 < x_0 < 1$ , iteratief updaten convergeert en het stabiele punt is  $x_\infty = 0$ .

De waarden waarvoor de iteratie niet naar oneindig gaat noemen we de **Julia verzameling**

---

## Fractals gemaakt m.b.v. complexe getallen

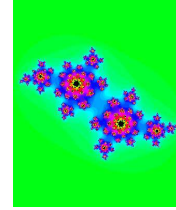
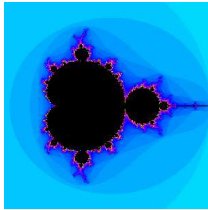
Met reële getallen kunnen er niet zo veel interessante dingen gebeuren. Dus laten we gebruik maken van complexe getallen beschreven door  $x = ai + b$

Beschouw nu functies van het type:

$$x_{k+1} = x_k^2 + C$$

Als we beginnen met  $x_0 = 0$ , voor welke complexe getallen  $C$  zal de iteratie voor deze functie dan niet oneindig worden?

Deze verzameling van complexe getallen waarvoor de iteratie niet naar oneindig gaat noemen we de **Mandelbrot verzameling**



Als meerdere agenten zich evolueren en de fitness van een agent bepaald wordt door de hele gemeenschap, wordt dit ook wel co-evolutie genoemd.

Cellulaire automaten zijn geschikt voor het modelleren van bepaalde biologische, sociale en fysische processen.

Markt economieën kunnen gemodelleerd worden als complexe wisselwerkingen tussen agenten met eigen gedragingen. De werking van zo'n systeem heeft analogieën met eco-systemen.

Fractals zijn patronen welke op hun subpatronen lijken. Mandelbrot fractals kunnen gemaakt worden met een simpel iteratief algoritme

---

## Julia verzameling

Beschouw opnieuw de geitereerde functie:

$$x_{k+1} = x_k^2 + C$$

Nu hebben we een waarde voor  $C$  welke een element is van de Mandelbrot verzameling.

Welke beginwaarden voor  $x_0$  in de complexe ruimte zorgen ervoor dat de iteratie niet naar oneindig gaat?

Deze verzameling welke behoort bij een bepaalde waarde voor  $C$  noemen we de **Julia verzameling voor  $C$** .

---

## Conclusie

ALife is geschikt om complexe processen te bestuderen, bestaande uit interacterende entiteiten.

Genetische algoritmen kunnen gebruikt worden om entiteiten (agenten) aan te passen aan hun veranderlijke omgeving.