

Transparanten bij het vak Inleiding Adaptieve Systemen: Biologische Adaptieve Systemen. M. Wierin

Natuurwetenschap

Voor de 16e eeuw geloofden nog veel wetenschappers in een deductieve benadering om kennis te vergaren

Zo dacht Aristoteles dat zware objecten sneller vallen dan lichte objecten

Dit duurde tot Galileo Galilei (1564 - 1642) dit testte waaruit bleek dat deze hypothese fout was

Hierna volgden een aantal belangrijke wetenschappelijke doorbraken welke uitdraaiden op een nieuwe natuurwetenschap

- Galilei maakte voorspellende methoden (aarde draait om de zon)
- Huygens kon betere klokken, lenzen, en telescopen maken zodat experimenten veel precieser gedaan konden worden
- Kepler benaderde banen van planeten met ellipsen i.p.v. de gebruikelijke cirkels
- Newton ontdekte de aantrekkingswet tussen 2 objecten waaruit ook volgde dat planeetbanen ellipsen waren

Omkeerbare Systemen

De nieuwe wetenschap leidde tot de gedachte dat het universum voorspelbaar was (Genius van Laplace)

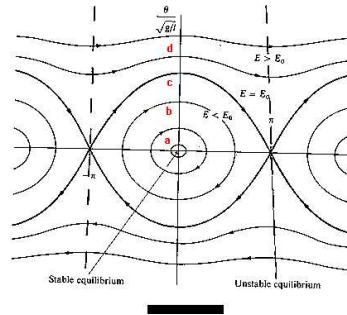
De mechanische wetten van Newton beschrijven een omkeerbaar systeem. Dit houdt in dat als we de richting van de tijd veranderen, we het verleden en de toekomst omdraaien.

Omkeerbare systemen behouden hun energie en daarom kunnen ze doorgaan met hun beweging.

Een voorbeeld hiervan is een slingerklok als we wrijving verwaarlozen.

Niet Omkeerbare Systemen

Er zijn ook veel systemen waarbij bruikbare energie verloren gaat (thermodynamische systemen)



Het is b.v. niet mogelijk om een machine te maken die altijd kan blijven doorgaan zonder dat deze extra energie krijgt

Als voorbeeld van een niet-omkeerbaar systeem nemen we een vat met twee helften waarin aanvankelijk alle gasmoleculen in 1 helft zitten (een geordende toestand)

Als we de wand die de helften scheidt weg nemen, dan zal de wanorde van het systeem enkel toenemen

Boltzmann bedacht de maat entropie om de wanorde van een systeem te beschrijven.

Stel dat er N moleculen zijn waarvan in 1 helft van het vak N_1 en in het andere helft van het vat N_2 . Dan is het aantal permutaties van die toestand:

$$P = \frac{N!}{N_1!N_2!}$$

Entropie

Het is dus logisch dat het systeem naar een evenwicht gaat met de meeste mogelijke toestanden, dus met $N_1 = N_2$.

Boltzmann definieerde de entropie van het systeem:

$$S = k \log P$$

Omdat de entropie voortdurend toeneemt en er een toestand is met maximale entropie, zal het systeem uiteindelijk in een evenwicht terecht komen, het systeem is dan dus niet omkeerbaar.

Dit leidde tot de twee wetten van de thermodynamica (Classius 1865):

- De energie van de wereld is constant
- De entropie van de wereld gaat naar een maximale waarde

Let op dat dit geldt voor gesloten systemen. Open systemen zoals levende wezens kunnen hun entropie verminderen door bruikbare energie van de omgeving op te nemen

Chaos Theorie: de Lorenz attractor

In sommige systemen veroorzaken kleine verschillen in beginvoorwaarden grote verschillen in de toekomst. Dit soort systemen volgen chaotisch gedrag

De meteoroloog Edward Lorenz vond per toeval met behulp van zijn computer een chaotisch systeem toen hij een model maakte van het weer.

Hoewel hij bijna dezelfde parameters had gebruikt en het systeem deterministisch was, volgde het systeem na een tijdje een heel ander traject

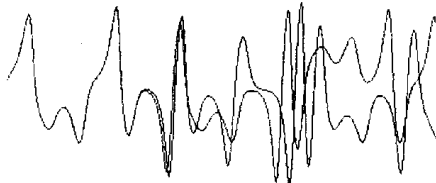
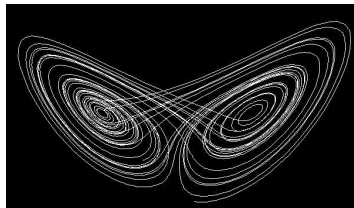


Figure 1: Lorenz's experiment: the difference between the start of these curves is only .000127. (Ian Stewart, Does God Play Dice? The Mathematics of Chaos, pg. 141)



Logistieke Map

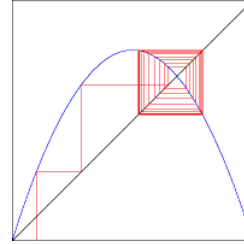
De Lorenz attractor is moeilijk om te analyseren, daarom gebruiken we de simpelere logistieke map beschreven door:

$$x(t+1) = rx(t)(1-x(t))$$

Hierin heeft $x(t)$ een waarde tussen 0 en 1. Het interessante is om te kijken wat er gebeurt als we de controle parameter r veranderen.

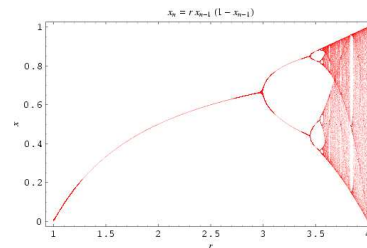
Voor $r < 1$ gaat x altijd naar 0. Als we r verhogen krijgen we eerst 1 stabiel eindpunt welke afhangt van de waarde van r .

Als we r nog meer verhogen krijgen we een periodische cyclus van lengte 2

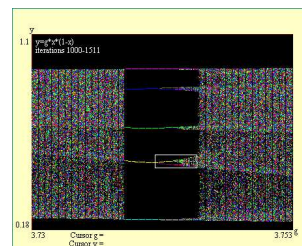


Chaos in de Logistieke Map

Maar wat gebeurt er als we r blijven verhogen?



En als we dan inzoomen:

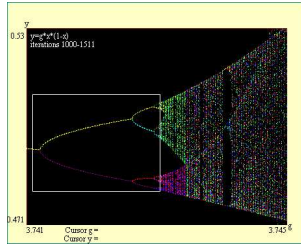


Biologische Adaptieve Systemen

Adaptieve systemen kunnen goed gebruikt worden om biologische processen mee te modelleren.

Enkele voorbeelden hiervan zijn:

- Infectie ziektes



- Bosbranden
- Overstromingen
- Vulkaan uitbarstingen
- Co-Evoluerende soorten

De eerste 4 voorbeelden hebben een overeenkomstig aspect:

Ze breiden zich uit (propageren zichzelf) over paden welke afhangen van de omgeving.

Om de propagatie tegen te gaan moeten propagatiepaden “gesloten” worden.

Ruimtelijke Modellen vs. niet-ruimtelijke modellen

We kunnen biologische processen modelleren met ruimtelijke modellen zoals cellulaire automaten, maar we kunnen deze ook direct modelleren m.b.v. vergelijkingen.

Het gebruik van ruimtelijke modellen geeft een extra vrijheidsgraad en daarmee meer mogelijke emergerende patronen.

Het gebruik van ruimtelijke modellen maken bepaalde processen makkelijker te visualiseren (v.b. bosbranden).

Het gebruik van ruimtelijke modellen is wel veel langzamer om te simuleren.

Modelleren van infectie ziektes

We zullen nu 2 manieren bekijken om infectie ziektes te simuleren.

Bij infectie ziektes bestaan er 3 soorten individuen (agenten of populaties):

- Gezonde individuen (G)
- Geïnfecteerde, zieke individuen (Z)

- Ziek geweest, immune individuen (I)

Als een gezond individu in aanraking komt met een geïnfecteerd individu, dan wordt het gezonde individu ook geïnfecteerd.

Als een geïnfecteerd individu lang genoeg ziek is geweest, wordt het een ziek geweest, immuun individu

Niet ruimtelijk model voor infectie ziektes

We kunnen een model maken m.b.v. update vergelijkingen.

We beginnen met een populatie ($Z(0)$, $I(0)$, $G(0)$), hierna gebruiken we:

$$Z(t+1) = Z(t) + aZ(t)G(t) - bZ(t)$$

$$I(t+1) = I(t) + bZ(t)$$

$$G(t+1) = G(t) - aZ(t)G(t)$$

Beginnend met een initiele populatie en gekozen parameters (a, b), kunnen we het gedrag simuleren.

Let op dat Z , I , G niet negatief mogen worden!

Uiteindelijk kunnen alle individuen ziek geweest, immune individuen worden

Vraag: begin met ($Z(0) = 10$, $I(0) = 0$, $G(0) = 90$), neem $a = 0.01$ en $b = 0.1$, itereer het model enkele keren.

Ruimtelijk model van infectie ziektes

We kunnen ook een cellulaire automaat (CA) gebruiken:

		G							
			Z						
					Z	Z			
	G								
	G		I		Z	G			
	G			I					
			Z				G		
		Z						I	
	I					Z			
				Z					

Nu moeten we regels opstellen om de toestanden in de CA te veranderen.

Regels voor het CA model

Als G een Z in een vakje naast zich heeft, wordt deze individu zelf ook een Z .

Z heeft elke tijdstap een kans p om een I te worden

Voor de navigatie kunnen we een random-walk van alle agenten gebruiken; ze maken willekeurige stapjes in alle richtingen.

We zouden ook kunnen modelleren dat gezonde mensen wegblijven van geïnfecteerden, dat veroorzaakt een heel andere dynamiek.

Vraag: Wat voor fenomenen zouden kunnen optreden als gezonde individuen uit de buurt blijven van geïnfecteerden?

Swarm Intelligence

Grote groepen van simpele organismen zoals bijen of mieren, kunnen samen intelligent gedrag vertonen.

Voorbeelden zijn:

- Foraging behavior (op zoek gaan naar voedsel)
- Bescherming van het nest
- Bouwen van het nest (v.b. termieten, waar zit de blueprint?)
- Voedsel verspreiding en opslag

Het is bekend dat mier-kolonieën bepaalde bekende problemen kunnen oplossen, zoals het vinden van het kortste pad naar een voedselplek, sorteren (clusteren en stapelen) van voedsel of mieren-lijken.

Hoewel een enkele mier of bij niet erg intelligent is, vertoont het gedrag van de gehele kolonie wel intelligent gedrag (super-intelligentie)

Sorteer gedrag van mier-kolonieën

Een enkele mier heeft maar zeer beperkte intelligente vermogens.

Toch kan het totale groepsgedrag welke emergeert uit de samenwerking tussen veel mieren in een kolonie vrij intelligent lijken.

Hoe kunnen mieren begraaftplaatsen met stapels van lijken bouwen?

We kunnen een simpel model van een mier maken:

- De mier beweegt in willekeurige richtingen
- Als een mier niets draagt en een dode mier ziet, pakt de mier het lijk op
- Als de mier een lijk draagt en een verzameling andere lijken tegenkomt, laat hij het lijk daar achter.

Deze 3 simpele regels veroorzaken de begraaftplaatsen van mierenlijken welke men kan observeren.

Vraag: bedenk hoe mieren korrels suiker en chocolade van elkaar kunnen scheiden.

Combinatorische optimalisatie

Bepaalde problemen kosten exponentieel veel tijd om optimaal op te lossen.

V.b. van exponentiele tijd problemen: Stel een probleem bestaat uit n toestanden en de tijd die het kost om het op te lossen is 2^n of $n!$

Exponentiele tijd problemen groeien veel sneller dan polynomiale tijd problemen:

$$\lim_{n \rightarrow \infty} \frac{n^p}{e^n} \rightarrow 0$$

Een aantal bekende wiskundige problemen noemen we combinatorische optimalisatie problemen (een voorbeeld zijn NP-complete problemen, welke niet in polynomiale tijd opgelost kunnen worden, tenzij $P=NP$).

Aangezien de computer kracht niet sneller dan exponentieel toe kan nemen, kunnen we bepaalde grote combinatorische optimalisatie problemen nooit optimaal oplossen.

Voorbeelden van Combinatorische optimalisatie problemen

Een aantal voorbeelden van combinatorische optimalisatie problemen zijn:

- Traveling salesman probleem: vind kortste tour tussen n steden.
- Quadratic assignment probleem: Minimaliseer de flow (totaal afgelegde afstand) als een aantal werknemers elkaar in een gebouw volgens een bepaalde frequentie opzoeken.
- 3-Satisfiability: vind waarheidswaarden van proposities die volgende vorm waarmaken:

$$\{x_1 \vee \neg x_2 \vee x_4\} \wedge \dots \wedge \{x_1 \vee \neg x_5 \vee x_7\}$$
- Job-shop scheduling: Minimaliseer de totaal benodigde tijd om een aantal jobs te laten volbrengen door een aantal machines welke in sequentie elke job moeten doorlopen.

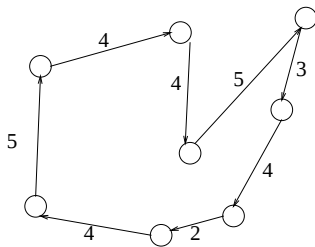
In het vak (logische) complexiteits theorie wordt hier nader op ingegaan.

Traveling salesman probleem (TSP)

Er is een verkoper die n steden aan wil doen en in zijn beginstad wil terugkomen.

Alle steden i en j zijn verbonden met een weg van lengte $l(i, j)$. Deze lengtes staan in een afstand matrix.

De agent moet nu een tour bedenken welke de totale kosten voor het aandoen van alle steden minimaliseert.



Vraag: Hoeveel mogelijke touren zijn er met N steden?

Genereren van een tour

Hoe kunnen we een tour genereren?

De constraints zijn dat alle steden precies 1 keer aangedaan worden en dat de tour terugkomt op zijn beginstad.

We kunnen een lijst bijhouden van steden die nog niet bezocht zijn: $J = \{i \mid i \text{ is nog niet bezocht}\}$

In het begin bevat J alle steden. Na het bezoeken van een stad wordt die stad uit J gehaald.

1. Kies initiele stad s_1 en haal s_1 weg uit J
2. For $t = 2$ To N :
3. Kies stad s_t uit J en haal deze weg uit J
4. Bereken lengte van deze tour:

$$L = \sum_{t=1}^{N-1} l(s_t, s_{t+1}) + l(s_N, s_1)$$

Het doel is om de tour met minimale totale lengte L te vinden

Ant Algoritmen

Een nieuw soort multi-agent adaptief systeem voor combinatorische optimalisatie is bedacht door Marco Dorigo in 1992.

Een kolonie van mieren werkt samen om b.v. voor het TSP een optimale tour te zoeken.

Foraging ants leggen een chemische substantie neer (genoemd pheromoon) wanneer ze van hun nest naar een voedsel bron gaan en vice versa.

Andere foraging ants volgen de tracks met de meeste gehalte aan pheromoon volgens een kansverdeling. Dit collectieve foraging behavior stelt de mieren in staat om het kortste pad te vinden van hun nest naar een voedsel bron.

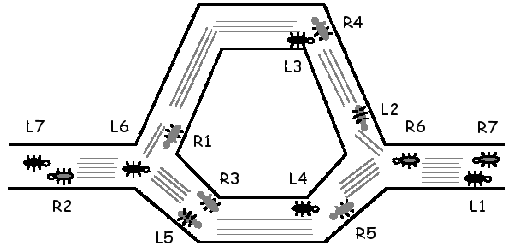
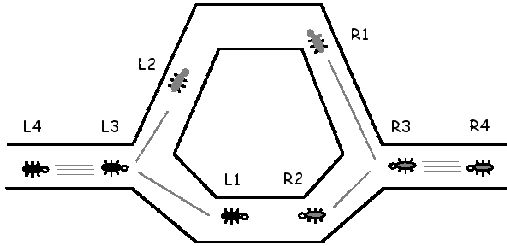
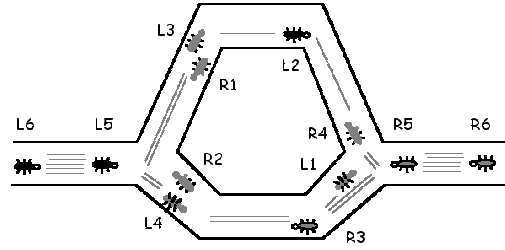
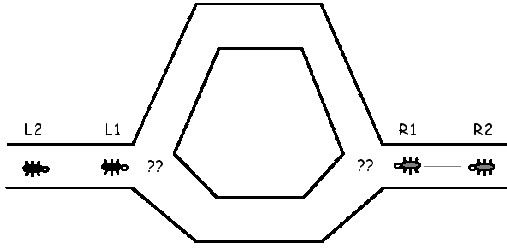
Optimalisatie algoritmen welke geïnspireerd zijn door het collectieve foraging gedrag van mieren worden Ant Algoritmen genoemd.

Voorbeeld: Foraging ants

Vervolg Foraging ants

Eigenschappen van Ant Algoritmen

Er zijn een reeks verschillende Ant Algoritmen, maar ze delen allemaal de volgende eigenschappen:



- Ze bestaan uit een kunstmatige kolonie van coopererende mieren
- Mieren maken discrete stappen.
- Mieren leggen pheromoon neer op hun gekozen paden
- Mieren gebruiken de neergelegde pheremoon tracks voor het kiezen waar ze naar toe gaan

2. Alle niet gevolgde edges (kanten) verliezen een beetje pheromoon door evaporatie
3. Alle gevolgde touren krijgen extra pheromoon waarbij de kortste tour meer pheromoon krijgt dan langere touren.

Vraag: bedenk hier een variant op en analyseer de voor- en nadelen.

Ant Algoritmen worden gebruikt voor een groot aantal Combinatorische optimalisatie algoritmen zoals TSP, QAP, network routing

Het idee dat twee individuen indirect interacteren omdat 1 van hen de omgeving verandert en de ander de veranderde omgeving gebruikt voor het nemen van besluiten wordt **stigmercy** genoemd.

Ant System

Het eerste Ant algoritme was “the Ant system” (AS). The Ant system werd initieel getest op het TSP.

The Ant System werkt als volgt:

1. Alle N ants maken een tour waarbij ze de pheromoon tracks gebruiken voor het kiezen van de volgende stad.

Formele specificatie Ant Systeem

De kolonie bestaat uit K mieren.

De mate van pheromoon tussen 2 steden i en j noteren we als $m(i, j)$

In het kiezen van een volgende stad gebruiken we een extra heuristiek: de inverse van de lengte tussen 2 steden: $v(i, j) = \frac{1}{l(i, j)}$

Nu maakt elke mier: $k = 1..K$ een tour:

1. Kies random start stad voor mier k : $i = \text{random}(1, N)$ en haal deze start stad weg uit de lijst J_k van onbezochte steden voor mier k
2. Kies steeds de volgende stad voor mier k als volgt:

$$j = \begin{cases} \arg \max_{h \in J_k} \{ [m(i, h)] \cdot [v(i, h)]^\beta \} & \text{if } q \leq q_0 \\ S & \text{anders} \end{cases} \quad (1)$$

Hierbij is q een random getal ($0 \leq q \leq 1$)
Parameter ($0 \leq q_0 \leq 1$) bepaalt de relatieve
belangrijkheid van exploitatie versus
exploratie.

We hebben allereerst gekeken hoe we modellen
voor de spreiding van infectie ziektes kunnen
maken.

Ruimtelijke modellen bieden meer simulatie vrij-
heden, maar duren langer om te draaien op een
computer.

S is een stad gekozen volgens de kansver-
deling gegeven in de volgende vergelijking:

$$p_{ij} = \begin{cases} \frac{[m(i,j)] \cdot [v(i,j)]^\beta}{\sum_{h \in J_k} [m(i,h)] \cdot [v(i,h)]^\beta} & \text{if } j \in J_k \\ 0 & \text{anders} \end{cases} \quad (2)$$

Vervolgens hebben we gekeken hoe een kolonie
van “domme” mieren toch intelligent gedrag kan
vertonen.

Zo kunnen mieren objecten clusteren m.b.v. een
aantal simpele regels

Ant algoritmen kunnen gebruikt worden om com-
plexe combinatorische problemen zoals de travel-
ing salesman probleem op te lossen.

Nu hebben alle mieren een tour gemaakt.

Andere manieren om de volgende steden te se-
lecteren zijn ook mogelijk, maar bovenstaande
werken meestal iets beter.

Als parameters moeten ingesteld worden hoe groot
de pheromoon tracks $m(i, j)$ initieel zijn, wat q_0
is, en wat β moet zijn.

Updaten van de pheromoon tracks

Er zijn meerdere mogelijke update regels. Als
keuze kan men bijvoorbeeld maken om enkel de
beste tour te updaten (en niet alle).

We noemen de best gevonden tour van alle mie-
ren in de laatste generatie S_{gb} (S globaal best).

Deze tour heeft lengte L_{gb}

De update regel komt er dan als volgt uit te zien:

$$m(i, j) = (1 - \alpha) \cdot m(i, j) + \alpha \cdot \Delta m(i, j)$$

$$\text{waarbij } \Delta m(i, j) = \begin{cases} (L_{gb})^{-1} & \text{if edge (i,j)} \in S_{gb} \\ 0 & \text{anders} \end{cases}$$

Hierbij is α de leersnelheid. Let op dat de ex-
tra gehalte aan pheromoon van de lengte van de
best gevonden tour afhangt en dat andere edges
evaporeren.

Discussie

Er zijn veel soorten biologisch adaptieve syste-
men.