

# Optimalisatie van complexiteit binnen individuen door co-evolutie

Hubert Smeitink  
*Student nummer: 3871452*  
*Universiteit Utrecht*  
*Utrecht, Nederland*  
*H.Smeitink@gmail.com*

Begeleider / Eerste beoordelaar:  
Gerard Vreeswijk

Tweede beoordelaar:  
Janneke van Lith

Datum:  
5 - 8 - 2015

ECTS: 7.5

## Abstract

Evolutionaire algoritmes (EAs) worden al met succes ingezet voor een breed scala aan problemen. EAs leiden echter niet altijd tot de globaal optimale oplossing doordat ze kunnen stranden in lokale optima. Deze paper beoogt een bijdrage te leveren aan de verbetering van de werking van EAs. Hiertoe is onderzoek verricht naar het verband tussen lokale optima en de complexiteit binnen de in een EA aan evolutie onderhevige individuen. Dit onderzoek is gedaan met als basis het werk van Stuart Kauffman. Kauffman suggereert dat de gemiddelde hoogte van lokale optima in co-evolutionaire ecosystemen hoger ligt wanneer de dynamiek van een ecosysteem op de edge of chaos balanceert, een staat tussen geordendheid en chaos. Verder suggereert hij dat co-evolutie zelf in staat is deze edge of chaos op te zoeken door de complexiteit binnen de individuen in het ecosysteem aan te passen. De genoemde complexiteit komt hierbij voort uit het aantal epistatische interacties tussen de genen van een individu.

In deze paper is de suggestie dat co-evolutionaire ecosystemen de edge of chaos opzoeken getest met behulp van een nieuw model. In dit model blijkt dat individuen met een bepaalde initiële complexiteit inderdaad het best presteren. Er blijkt echter ook dat co-evolutie in dit model slechts in beperkte mate in staat is de complexiteit van een populatie te optimaliseren.

## Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>3</b>
<b>2</b>	<b>Evolutionaire algoritmes</b>	<b>4</b>
<b>3</b>	<b>Artificiële evolutie</b>	<b>5</b>
3.1	Fitness landschap . . . . .	5
3.2	Artificiële en intrinsieke selectie . . . . .	6
3.3	Co-evolutie . . . . .	7
<b>4</b>	<b>NK model</b>	<b>7</b>
4.1	Werking NK model . . . . .	7
4.2	Bevindingen uit het NK model . . . . .	8
<b>5</b>	<b>Random boolean networks</b>	<b>9</b>
<b>6</b>	<b>Noodzaak ontwikkeling NSBN model</b>	<b>10</b>
<b>7</b>	<b>Het NSBN model</b>	<b>11</b>
7.1	Wereld . . . . .	12
7.2	Booleaans netwerk . . . . .	12
7.3	Agenten . . . . .	15
<b>8</b>	<b>Tests en resultaten</b>	<b>17</b>
8.1	Uitgevoerde tests . . . . .	17
8.2	Resultaten NSBN model . . . . .	18
<b>9</b>	<b>Conclusie</b>	<b>23</b>
	<b>Referenties</b>	<b>24</b>

## 1. Introductie

Eén van de idealen binnen het vakgebied van kunstmatige intelligentie is de toepassing van zelflerende systemen voor de ontwikkeling van intelligentie [1]. Dit doel wordt met behulp van meerdere typen algoritmes nagestreefd, waaronder evolutionaire computatie. De werking van evolutionaire computatie berust op de door Darwin gevonden principes die aan evolutie ten grondslag liggen. Binnen evolutionaire computatie bevindt zich de subcategorie evolutionaire algoritmes (EAs). Deze algoritmes ontleen hun inspiratie aan natuurlijke evolutie. Dit is terug te zien in het gebruik van mutatie, selectie, recombinitie en reproductie als de hoofdelementen van deze algoritmes.

EAs hebben reeds bewezen in staat te zijn bruikbare oplossingen te genereren voor een breed scala aan problemen. Deze brede inzetbaarheid voort uit het afwezigheid van aannames over de vorm van het fitness landschap. Dit fitness landschap wordt, in het geval van EAs, berekend met behulp van een fitness functie. Deze fitness functie geeft voor iedere oplossing (gerepresenteerd door een individu in de populatie) een waardering in de vorm van een scalar waarde. Deze waardering wordt de fitness van de oplossing genoemd. In het geval van een fitness functie gebaseerd op twee attributen van een individu kan een fitness landschap worden weergegeven in een driedimensionale grafiek. Ieder punt in het vlak  $Y = 0$  is hierbij gelijk aan een specifieke combinatie van de twee attributen. Met behulp van de fitness functie kan dan voor iedere combinatie van de twee attributen de fitness worden berekend en weergegeven in  $Y$ . Het hiermee weergegeven landschap kan (lokale) optima bezitten, in de vorm van pieken omringd door een dal. Selectie van de fittest individuen in EA's zorgt ervoor dat individuen met een hogere fitness een hogere overlevingskans hebben. Dit leidt ertoe dat de populatie per generatie fitter wordt en de optima in het landschap beklimt. Zodra de gehele populatie zich in één of meer lokale optima bevindt stopt de evolutie. Dit komt doordat elke (kleine) mutatie van een individu in een lokaal optimum een verlaging van de fitness van dit individu, en dus een kleinere overlevingskans, oplevert. Hierdoor leiden EAs als geheel niet altijd tot een globaal optimale oplossing. Omdat het vroegtijdig stranden van evolutie in lokale optima leidt tot sub optimale uitkomsten is onderzoek naar het vermijden van het stranden in lokale optima voor EAs van groot belang.

Het probleem van lokale optima komt ook naar voren in evolutie gebaseerd op intrinsieke selectie. Hierin wordt gebruik gemaakt van een directe concurrentiestrijd om ten minste één gelimiteerde bron ter vervanging van artificiële selectie (selectie met behulp van een fitness functie). Een voorbeeld hiervan is de natuurlijke evolutie op aarde, waar onder andere een concurrentiestrijd om leefruimte en voedsel kan worden geobserveerd. Deze vorm van selectie is niet bedoeld om een probleem op te lossen, maar om de werking van evolutie beter te leren begrijpen en verklaren. Modellen gebaseerd op intrinsieke selectie zijn er echter tot nog toe niet in geslaagd continue evolutie voort te brengen: de evolutie stopt na een bepaald aantal generaties [2]. Hiervoor kunnen meerdere mogelijke oorzaken worden aangewezen, waaronder het gebrek aan neutrale mutaties [3], gebrek aan genotype-fenotype distinctie [4], maar ook een ongeschikt fitness landschap.

Deze paper zal zich richten op de vorm van deze fitness landschappen. Specifiek zal worden gekeken naar de invloed van de complexiteit binnen individuen op de vorm van de bij deze individuen horende fitness landschappen, zoals onderzocht door Kauffman [5]. De genoemde complexiteit komt hierbij voort uit het aantal epistatische interacties tussen de genen van ieder individu en de genen van ditzelfde en andere individuen: een groter aantal interacties per gen leidt tot een grotere complexiteit. Met een epistatische interactie wordt bedoeld dat een gen de fitness contributie van een ander gen beïnvloed. Hierdoor is de totale fitness van een individu niet meer gelijk aan de som van de losse fitness contributies van de individuele genen. Zo kan gen A interacteren met gen B en een positieve bijdrage leveren wanneer gen B actief is, terwijl het een

negatieve fitness bijdrage levert wanneer gen B non-actief is. De door Kauffman gepubliceerde resultaten rechtvaardigen de verwachting dat langduriger ongerichte evolutie kan worden bereikt wanneer het fitness landschap tijdens het evolutieproces door evolutie zelf wordt afgestemd. Dit komt omdat het systeem dan balanceert op de zogenaamde edge of chaos. Wanneer een ecosysteem balanceert op deze rand tussen geordendheid en chaos zou het maximaal in staat zijn zich zowel aan kleine als aan grote veranderingen in het ecosysteem aan te passen. Dit leidt er dan toe dat de gemiddelde fitness van de gehele populatie tijdens het zoeken naar een optimum in het fitness landschap, en tevens na het vinden van dit optimum, hoger ligt. Een principe dat ook in EA's met behulp van expliciete selectie van nut kan zijn.

Tot slot suggereert Kauffman dat co-evolutie zelf in staat is de edge of chaos op te zoeken. Deze zelfregulerende eigenschap van co-evolutionaire ecosystemen vormt de hoofdvraag van deze paper:

*Optimaliseren co-evolutionaire ecosystemen de fitness van individuen door de complexiteit binnen individuen te optimaliseren?*

Deze paper beoogt met behulp van een nieuw model bij te dragen aan de nodige basis voor het beantwoorden van deze vraag, een volledig sluitend antwoord ligt echter niet binnen het bereik van deze paper. Alvorens hiertoe over te gaan zal een beknopte samenvatting van het reeds uitgevoerde onderzoek naar co-evolutie en complexiteit worden gegeven om de lezer zo van voldoende achtergrondinformatie te voorzien om het model en de resultaten in hun context te kunnen plaatsen.

Die hierop aansluitende verdere opbouw van deze paper is als volgt: 2. Evolutionaire algoritmes, een korte beschrijving van de werking van evolutionaire algoritmes. 3. Artificiële evolutie, een korte toelichting van artificiële co-evolutie en de interactie tussen fitness landschappen die hieruit voorkomt, inclusief het verschil tussen intrinsieke natuurlijke selectie en artificiële selectie. 4. NK model, een samenvatting van het door Kauffman ontwikkelde model dat de aanleiding gaf voor de suggestie van de edge of chaos. 5. Random boolean networks, toelichting van een veel gebruikt model voor onderzoek naar kritieke staten, dat als inspiratie dient voor het in deze paper ontwikkelde model. 6. Noodzaak ontwikkeling NSBN model, verantwoording voor de ontwikkeling van het NSBN model. 7. Het NSBN model, een gedetailleerde toelichting van de werking van het NSBN model. 8. Tests en resultaten NSBN model, een toelichting van de uitgevoerde tests en een bespreking van de resultaten. 9. Conclusie.

## **2. Evolutionaire algoritmes**

EAs vallen binnen de klasse van populatie gebaseerde stochastische zoek algoritmes gebaseerd op natuurlijke evolutie. Doordat EAs niet afhankelijk zijn van gradiënt informatie zijn ze toepasbaar op een breed scala aan problemen [22]. EAs bewijzen hun meerwaarde daarom vooral bij het oplossen van problemen waarbij deze gradiënt informatie niet beschikbaar is. Essentieel is echter wel dat een fitness functie kan worden opgesteld.

Naast het gebruik van een populatie delen alle EAs de eigenschap dat zij geïnspireerd zijn door natuurlijke evolutie. Dit is terug te zien in het gebruik van mutatie, selectie, recombinitie en reproductie. De kern van een EA bestaat uit (een variant op) de volgende stappen:

- 1) Aanmaken van een willekeurige initiële populatie
- 2) Evaluatie van ieder individu met behulp van de fitness functie
- 3) Herhaling van deze stap tot de oplossing voldoende dicht is benaderd:
  - Selecteer de meest fitte individuen voor reproductie (ouders)

- Creëer nieuwe individuen (kinderen) door recombinatie en mutatie
- Evalueer de fitness van de nieuwe individuen
- Vervang de minst fitte individuen van de populatie met de nieuwe individuen

Door de herhaalde selectie en vervanging van individuen met de laagste fitness waarden door nieuwe individuen in stap 3 stijgt de gemiddelde fitness van de populatie. Deze stijging stopt wanneer de populatie is gestrand in één of meer (lokale) optima. De gevonden oplossing is dan gelijk aan het individu met de hoogste fitness waarde. Doordat het echter niet zeker is dat ten minste een deel van de populatie het globale optimum heeft bereikt, in plaats van in een lokaal optimum te zijn gestrand, bestaat er geen garantie dat EAs altijd de optimale oplossing zullen genereren.

### 3. Artificiële evolutie

#### 3.1. Fitness landschap

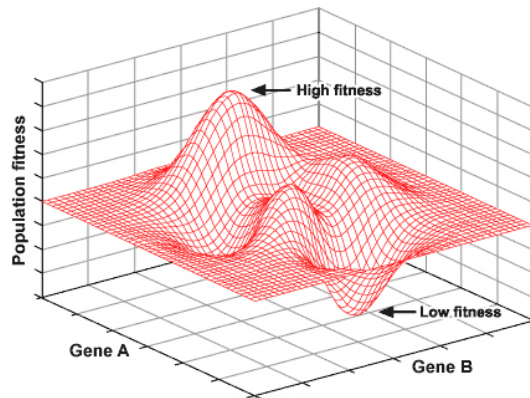
Voor inzicht in de werking van co-evolutionaire ecosystemen is het begrip fitness landschap van groot belang. Zoals in de introductie reeds is genoemd is een fitness functie een functie waarmee kan worden bepaald hoe goed een bepaalde oplossing is. Wat als goed wordt gezien hangt van de parameter af die dient te worden geoptimaliseerd om zo tot een zo optimaal mogelijke oplossing te komen.

Een voorbeeld in de praktijk is het probleem om voor een bezorger de snelste route tussen een X aantal bestemmingen te vinden. De individuen in de populatie representeren dan mogelijke routes en de fitness functie berekent de tijd die de route zou kosten. In dit voorbeeld staat een gen voor een adres en is genotype een reeks van X adressen (de route). Ieder individu beklimt in dit voorbeeld een identiek fitness landschap, waardoor effectief over één landschap kan worden gesproken. Dit komt doordat de gebruikte fitness functie enkel naar het individu zelf kijkt. Dit in tegenstelling tot co-evolutie, zoals verder besproken in 3.3, waar ieder individu een uniek fitness landschap kan hebben.

Gegeven een specifieke combinatie van genen (het genotype) kan voor iedere agent een fitness waarde worden berekend. Aan de hand van de fitness waarden van de individuen in een simulatie kan zo worden bepaald welke individuen relatief slechte oplossingen voorstellen. Dit zijn tevens de individuen die het best door nieuwe agenten kunnen worden vervangen. Het principe van een fitness functie vindt men overigens ook terug in natuurlijke evolutie in de vorm van een impliciete fitness functie. Deze impliciete functie geeft het verband aan tussen genotypen en de kans op reproductie.

Met behulp van een fitness functie kan een fitness landschap worden gemaakt. In het geval van twee genen is dit voor te stellen door een liggend vlak te nemen, waarbij de X as gelijk is aan gen A en de Z as aan gen B. Dit vlak bevat alle mogelijke combinaties van gen A en gen B. Wanneer vervolgens voor elk van deze combinaties de fitness wordt berekend en weergegeven op de Y as, en deze fitness waarden met elkaar worden verbonden, ontstaat een fitness landschap (figuur 1). Fitness landschappen met meer dan twee genen zijn ook mogelijk, maar ingewikkelder om weer te geven.

Met behulp van een grafisch weergegeven fitness landschap is goed in te zien waar lokale optima zich bevinden in de vorm van bergen in het landschap, omringd door een dal. Deze lokale optima spelen een grote rol in evolutionaire algoritmes doordat de evolutie van populaties in deze lokale optima kan stranden. De reden dat populaties niet door ontwikkelen na in een lokaal optimum terecht te zijn gekomen ligt in het feit dat elke mutatie weg van dit optimum een verlaging van de fitness oplevert. Omdat, als gevolg van selectie, individuen met een lagere fitness een kleinere overlevingskans hebben sterven deze gemuteerde individuen vaker dan de individuen hoger op de berg. Hierdoor worden de gemuteerde individuen weggefilterd uit de populatie.



*Figuur 1. Fitness landschap bij twee genen*

### 3.2. Artificiële en intrinsieke selectie

Selectie in EAs is de stap waarin wordt bepaald welke individuen voortplanten en welke individuen worden verwijderd. Deze selectie is een essentiële stap omdat zonder selectie elk individu een gelijke overlevingskans zou hebben, waardoor er geen optimalisatie plaats kan vinden.

In EAs wordt selectie over het algemeen geïmplementeerd in de vorm van artificiële selectie. Deze vorm van selectie maakt gebruik van een expliciete fitness functie en kan op meerdere manieren worden verwezenlijkt. Voorbeelden hiervan zijn "tournament selection", "stochastic universal sampling" en "fitness proportionate selection". In al deze algoritmes wordt gebruikt gemaakt van een expliciete fitness functie om te bepalen (al dan niet gebruik makend van kansen) welke individuen worden verwijderd en welke voortplanten.

In de praktijk blijkt het echter moeilijk om een geschikte fitness functie op te stellen voor problemen waar de gewenste optimalisatie (door selectie) niet geheel bekend is [6]. Een alternatief voor artificiële selectie is wat soms intrinsieke adaptatie [7] of natuurlijke selectie [2] wordt genoemd. Deze termen zijn echter beiden misleidend. De term natuurlijke selectie geeft onvoldoende de algemene aard weer van het principe van intrinsieke selectie (het is een algemeen principe dat óók in de natuur plaatsvindt). Daarnaast suggereert intrinsieke adaptatie dat de adaptatie anders verloopt, terwijl enkel hetgeen waar aan wordt geadapteerd verschilt. Om bovengenoemde redenen zal deze vorm van selectie in deze paper worden aangeduid met de term intrinsieke selectie.

Bij intrinsieke selectie wordt geen fitness functie gehanteerd. Deze vorm van selectie berust daarentegen op een vorm van concurrentie voortkomend uit interacties tussen individuen. Dit kan bijvoorbeeld worden bewerkstelligd door de genen van een individu invloed te laten hebben op de snelheid waarmee het zich kan verplaatsen. Indien dit leidt tot het kunnen verzamelen van meer voedsel, en wanneer voortplanten een vaste hoeveelheid voedsel kost, betekend dit dat agenten met bepaalde gunstige genen meer nageslacht kunnen produceren. Deze vorm van selectie is echter minder geschikt voor het vinden van een oplossing door middel van optimalisatie van een specifieke parameter. Dit is het geval omdat het moeilijk of ondoenlijk is om de evolutie op een intrinsieke wijze van de hiervoor vereiste fitness functie te voorzien. Voor het nabootsen van aspecten van natuurlijke evolutie is het daarentegen wel geschikt vanwege de gelijkenis met natuurlijke evolutie. Ook bevat intrinsieke selectie een vorm van feedback doordat de intrinsieke fitness functie mee verandert met de evolutie van de populatie. Dit aspect geldt als essentiële eigenschap voor langdurige evolutie [2].

Fitness landschappen kunnen bij intrinsieke selectie niet worden gevormd door gebruik van een fitness functie, aangezien deze niet bestaat. Er kan daarentegen wel een impliciete maat voor de

fitness worden gebruikt, zoals in het genoemde voorbeeld de hoeveelheid nageslacht als maat voor de fitness.

### 3.3. Co-evolutie

Bij zowel artificiële als intrinsieke selectie wordt gebruik gemaakt van mutaties en eventueel recombinatie. Mutaties worden toegepast op de genen van kinderen, die verder een kopie zijn van één of meerdere ouders (recombinatie). Mutaties zijn hierbij van essentieel belang omdat enkel recombineren van de genen van ouders niet toelaat dat er nieuwe genen aan de totale genenpoel worden toegevoegd. Hierdoor zou de diversiteit maximaal gelijk blijven, maar omdat individuen ook sterven (of worden vervangen) kunnen genen verloren gaan. Dit betekent dat zonder mutaties de diversiteit alleen maar kan afnemen en evolutie op den duur altijd stopt. Recombinatie is daarentegen niet van essentieel belang, zoals bijvoorbeeld kan worden teruggezien in de enkele asexueel voortplantende diersoorten op aarde.

In de standaard variant van artificiële selectie zorgen mutaties hoogstens voor een verandering in de prestatie van het betreffende individu zelf [8]. Dit komt voort uit het feit dat de fitness functie bij ieder individu uitsluitend naar de eigenschappen van dat individu zelf kijkt; de eigenschappen van de overige individuen beïnvloeden de evaluatie niet. Dit is tegengesteld aan de werking van evolutie in de natuur, waarbij sprake is van co-evolutie. Bij co-evolutie hangt de fitness van een gegeven individu niet alleen af van de genen van het individu zelf, maar ook van de genen van andere individuen. Dit zorgt voor een interactie tussen fitness landschappen waardoor een mutatie van het genotype van individu A, via zijn fenotype, de fitness van het genotype van individu B kan beïnvloeden. Een voorbeeld van co-evolutie in de natuur is de kikker die een plakkerige tong ontwikkelt, waarna het loont voor de vlieg om glibberige poten te ontwikkelen. Het in 3.1 genoemde voorbeeld van de bezorger die de snelste route zoekt kan worden veranderd in co-evolutie door een maximale snelheid aan iedere weg toe te voegen die afhankelijk is van de drukte. Hierdoor wordt de snelste route voor ieder individu afhankelijk van de routes (de genotypes) van de overige individuen.

## 4. NK model

Om de invloed van de vorm van fitness landschappen op evolutie te bestuderen is door Kauffman het NK model ontwikkeld [9][10].

### 4.1. Werking NK model

Met behulp van het NK model kan een fitness landschap worden gevormd met een instelbare grootte en grilligheid. De grootte van het landschap wordt bepaald door N, het aantal genen in een genotype, de grilligheid wordt bepaald door K, het aantal epistatische interacties binnen een genotype. K geeft hierdoor voor iedere locus aan hoeveel andere loci de fitness contributie van de betreffende locus beïnvloeden.

Het NK model vormt een zogenoemde "combinatorial phase space" bestaande uit elk mogelijk genotype (gerepresenteerd door een string) van lengte N. Voor ieder van deze genotypen kan de fitness worden berekend. De fitness van een genotype is gelijk aan de som van de individuele loci volgens formule (1). Hierin is G het genotype en  $G_i$  een locus in G.

$$F(G) = \sum_i f(G_i) \quad (1)$$

De fitness waarde van een locus hangt af van de waarde van de locus zelf en K andere loci volgens formule (2).

$$F(G_i) = \sum_i f(G_i, G_1^i, \dots, G_K^i) \quad (2)$$

Hierin is  $f(G_i, G_1^i, \dots, G_K^i)$  een mapping (de fitness functie) tussen genotypen (strings) en een scalar waarde.

Voorbeeld: Een NK model met  $K = 1$  en  $N = 4$ , met twee allelen (1 en 0). Iedere locus is afhankelijk van zichzelf en de volgende locus (4) en de laatste locus is afhankelijk van de eerste (5). De fitness functie (6) geeft voor genotype 0110 uitkomst:  $f(0,1) + f(1,1) + f(1,0) + f(0,0) = 0 + 0.4 + 0.2 + 1 = 1.6$

$$F(G_i) = f(G_i, G_{i+1}) \quad (3)$$

$$F(G_4) = f(G_5, G_1) \quad (4)$$

$$f(0,0) = 1, f(0,1) = 0, f(1,0) = 0.2, f(1,1) = 0.4 \quad (5)$$

Door  $K$  te laten oplopen van 0 tot  $N - 1$  neemt de grilligheid van het landschap toe. Er ontstaan hierbij meer en steilere bergen, die tevens gemiddeld lager zijn. Dit komt voort uit de toename van het aantal interacties tussen genen, wat leidt tot meer conflicterende belangen (de optimalisatie van het ene gen vermindert de fitness van een ander gen) en dus tot meer en lagere lokale optima (de complexity catastrophe). Bij het bepalen van de initiële fitness van de genotypen in het NK model wordt ervan uitgegaan dat de interacties tussen genen zo complex zijn dat het gerechtvaardigd is deze van willekeurig gekozen vaste fitness waardes te voorzien. Dit wordt gerealiseerd door aan iedere waarde in het bereik van de fitness functie een willekeurige waarde uit  $[0,1]$  te koppelen.

Het NK model is later door Kauffman uitgebreid met een  $C$  parameter [5], die het aantal interacties met genen van andere individuen representeert. Hierdoor is de fitness van een individu afhankelijk van de eigen genen middels  $K$  en de genen van andere individuen middels  $C$ , waardoor co-evolutie kan worden gemodelleerd.

Bij de door Kauffman uitgevoerde simulaties [5] worden de soorten waaruit de populatie is opgebouwd beschouwd als homogeen (mutaties vinden minder vaak plaats dan het tijd kost voor een gunstige mutatie om zich volledig door een soort te verspreiden). Als selectiemethode zijn meerdere vormen van selectie toegepast. Bij de eerste variant probeert de soort als geheel een willekeurige mutatie en past deze homogeen toe indien deze een hogere fitness oplevert. Bij de tweede "fitter" variant wordt gekeken naar alle mogelijke fitness verhogende mutaties, vervolgens wordt uit deze groep een willekeurige mutatie gekozen. Bij de derde "greedy" variant wordt de best mogelijke mutatie gekozen. Bij alle drie de varianten is het aantal mutaties tot één locus per keer gelimiteerd. De hieronder beschreven bevindingen gelden voor elk van deze varianten.

## 4.2. Bevindingen uit het NK model

De hieronder genoemde bevindingen zijn een beknopte samenvatting van de resultaten [5] die van invloed zijn op het voor deze paper ontwikkelde NSBN model.

Wanneer  $K$  gelijk wordt gesteld aan 0 levert dit zoals hierboven beschreven een landschap op met slechts één piek. Een piek die door elk individu (onafhankelijk van het initiële gen) middels 1 locus mutatie kan worden bereikt. Dit kan worden verklaard door de afwezigheid van interacties tussen loci, waardoor iedere locus afzonderlijk kan worden geoptimaliseerd.

Wanneer  $K$  wordt verhoogd richting  $N - 1$  ontstaat een steeds grilliger landschap. Dit heeft tot gevolg dat evolutie eerder in een lokaal optimum strandt dan bij lagere  $K$  waardes. In het extreme geval van  $K = N - 1$  leidt dit tot een verwacht aantal mutaties voor het bereiken van een lokaal optimum gelijk aan  $Ln(N)$ . In dit extreme geval vindt ook de complexity catastrophe plaats,



waarbij door het grote aantal van conflicterende interacties de optima richting de 0.5 dalen (de gemiddelde fitness in de fitness ruimte). De tijd die het kost om in een lokaal optimum te raken blijkt een kantelpunt te hebben rond  $K = C * S$ . Bij  $K < C * S$  duurt het relatief lang terwijl het bij  $K > C * S$  relatief kort duurt voordat een populatie in één of meer optima strandt. Hierin is  $S$  het totale aantal individuen, en zijn de interacties tussen individuen ( $C$ ) willekeurig gekozen binnen de gehele populatie.

Bij kleine mutatiekansen is selectie in staat om ervoor te zorgen dat een soort zich op (of zeer dicht rond) een piek (optimum) blijft bevinden. Wanneer de mutatiekans wordt vergroot, en dus het aantal mutaties toeneemt, daalt de soort aan alle kanten van de berg richting het omringende dal. Dit komt voort uit het vergrote aantal mutaties per generatie door de grotere mutatiekans, en omdat selectie bij een steeds groter wordende mutatiekans steeds minder in staat is de gemuteerde individuen weg te filteren. Selectie is hier minder toe in staat omdat een groter percentage van de populatie een fitness verlagende mutatie heeft ondervonden. Hierdoor ligt de gemiddelde fitness van de populatie lager en wijken de individuen met een fitness verlagende mutatie minder af van deze gemiddelde fitness. Daardoor is hun overlevingskans groter is dan bij kleinere mutatiekansen.

Omdat bij een hogere  $C$  waarde de invloed van een mutatie op de gehele populatie (door meer interacties tussen individuen) groter wordt blijkt dat wanneer  $C$  toeneemt de optimale mutatie kans afneemt. Als  $C$  wordt vergroot en de mutatiekans niet wordt verkleind verandert de dynamiek van het systeem als geheel in chaos, als gevolg van de ver propagerende gevolgen van de vele mutaties.

Een opvallende bevinding is dat populaties met  $K$  waardes gelijk of dicht bij  $C * S$  beter presteren, zowel voordat een optimum is bereikt als daarna, dan populaties met verder van  $C * S$  afliggende  $K$  waardes. Daarnaast blijkt uit tests met een hoge  $C$  waarde ( $C = 20$ ) dat wanneer een soort concurreert met één andere soort, met een vaste  $K$ , het loont om de  $K$  waarde te vergroten. Zo doet een  $K4$  soort het beter tegen een  $K2$  soort, dan een  $K2$  soort het tegen een  $K2$  soort doet, terwijl een  $K8$  het nog beter tegen een  $K2$  soort doet. Opvallend is hierbij dat de soort wiens  $K$  waarde vast staat, een hogere fitness behaalt wanneer het concurreert tegen een hogere  $K$  waarde. Beiden soorten behalen dus een hogere fitness (bij een hoge  $C$  waarde) wanneer één van hen de  $K$  waarde ophoogt richting de  $C$  waarde. Bij een lage  $C$  waarde ( $C = 1$ ), blijkt het omgekeerde het geval en zijn beide soorten beter af met lagere  $K$  waardes. Dit leidt tot suggestie dat, door  $K$  en  $C$  op elkaar af te stemmen, evolutie de gehele fitness van de populatie kan vergroten.

In tests waar gebruik is gemaakt van  $C$  interacties met slechts een deel van de overige populatie, waardoor een web structuur van epistatische interacties ontstaat, blijken tevens opvallende resultaten naar boven te komen. Wanneer gebruik wordt gemaakt van een dergelijk niet totaal verbonden ecosysteem is het mogelijk voor een deel van de populatie om in een (lokaal) optimum terecht te komen (als het ware bevroren te raken), terwijl de rest van populatie nog door ontwikkelt. In deze kritieke staat kunnen mutaties zowel kleine gevolgen hebben voor het ecosysteem als geheel als propageren door grote delen van het ecosysteem. De optimale  $K$  waardes voor deze ecosystemen blijken samen te vallen met deze kritieke staat, de zogenaamde edge of chaos, waarbij het bevroren van delen van het ecosysteem net begint. Dit suggereert dat co-evolutie, middels de  $K$  parameter, de gehele fitness kan vergroten door de edge of chaos op te zoeken. Een suggestie die, indien juist, van groot belang is voor zowel het begrip van natuurlijke evolutie als voor de implementatie van artificiële evolutie.

## 5. Random boolean networks

Random boolean networks (RBNs), ook wel NK networks genoemd [11], zijn door Kauffman geïntroduceerd als een abstract model van genregulatie. Genen worden in een RBN voorgesteld door knopen waarbij directionele verbindingen tussen knopen de interacties tussen

genen voorstellen. Het doel achter de ontwikkeling van RBNs als model voor genregulatie is het helpen verklaren van zelforganisatie in biologische systemen en de invloed van lokale regels op het gedrag van het netwerk als geheel. Ondanks het feit dat genen veel meer omvatten dan de in RBNs gebruikte booleaanse knopen zijn er verschillende voorbeelden waarin RBNs patronen van actieve en non-actieve genen correct weergeven [12].

In een RBN wordt het aantal knopen in het netwerk aangegeven met  $N$  en het aantal ingaande verbindingen per knoop met  $K$ . Ieder van deze knopen heeft een binaire staat, 1 of 0, en bezit een booleaanse functie die iedere combinatie van  $2^K$  binaire ingaande signalen koppelt aan een binaire output. Deze functie kan bijvoorbeeld worden geïmplementeerd in de vorm van een lookup table. De  $K$  ingaande signalen worden willekeurig gekozen en kunnen dus afkomstig zijn van alle knopen in het netwerk, inclusief de knoop zelf.

Het updaten van het netwerk gebeurt aan de hand van discrete tijdstappen waarbij de staat op tijdstip  $t + 1$  volledig afhankelijk is van het netwerk op tijdstip  $t$ . De nieuwe staat wordt bepaald door voor iedere knoop met behulp van de bij die knoop horende booleaanse functie en de ingaande signalen op tijdstip  $t$  de nieuwe output te berekenen voor  $t + 1$ . De initiële ( $t = 0$ ) output waardes van de knopen worden willekeurig gekozen.

Omdat elke knoop een binaire output geeft kan het netwerk in zijn geheel zich in  $2^N$  staten bevinden. De volledig deterministische manier van updaten zorgt er daarom samen met de finite state space voor dat het netwerk na maximaal  $2^N$  stappen periodiek gedrag vertoont. Deze periodieke herhaling vangt aan zodra het netwerk zich voor een tweede keer in een zelfde staat bevindt en wordt een "periodic attractor" genoemd. Het is mogelijk dat delen van het netwerk een kortere periodic attractor vertonen dan het netwerk als geheel.

Niet alle staten van het netwerk hoeven te behoren tot een periodic attractor. De niet tot een periodic attractor behorende reeksen eindigen één update voor een staat die tot een periodic attractor behoort. Hierdoor komen de tot deze reeksen behorende staten dus ten hoogste één keer voor tijdens het updaten. Deze staten worden aangeduid als de "basins of attraction" van de bijbehorende periodic attractor. Omdat door het deterministische karakter van het netwerk het niet mogelijk is voor dergelijke basins of attraction om te overlappen kan de "state space" worden opgedeeld in de basins of attraction en hun bijbehorende periodic attractors.

Het gedrag van een netwerk kan tot één van de drie volgende types behoren: chaotisch, bevroren en kritiek. Het type netwerk kan worden geïdentificeerd door te kijken naar de manier waarop het netwerk op kleine veranderingen reageert. Hiervoor wordt eerst de activiteit van het netwerk vanaf tijdstip  $t = 0$  vastgelegd om als referentie te dienen. Vervolgens wordt het netwerk weer gestart in tijdstip  $t = 0$ , maar ditmaal wordt de initiële waarde van één van de knopen geïnverteerd. Dit netwerk wordt net als het referentie netwerk een aantal keer ge-update waarna het wordt vergeleken met het referentie netwerk (bij dezelfde  $t$  waarde). Indien beide netwerken gelijk zijn heeft de initiële verandering geen effect gehad en is het netwerk bevroren. Als het netwerk van het originele netwerk verschilt in een toenemend aantal knopen is het netwerk chaotisch. Ten slotte kunnen de twee netwerken gelijke en afwijkende gebieden vertonen, in dit geval bevindt het netwerk zich in een kritieke staat (de edge of chaos). Uit tests is gebleken dat deze drie staten samenhangen met de  $K$  waarde. Bij  $K < 2$  bevinden netwerken zich gemiddeld in de bevroren staat, bij  $K = 2$  in de kritieke en bij  $K > 2$  in de chaotische staat. De chaotische staat gaat tevens typisch gepaard met lang durende periodic attractors.

## 6. Noodzaak ontwikkeling NSBN model

Kauffman heeft voor zijn onderzoek naar de mogelijke zelfregulerende eigenschap van co-evolutionaire ecosystemen naar de edge of chaos gebruik gemaakt van slechts één model: het NK

model. Voordat met zekerheid kan worden gesteld dat alle co-evolutie de complexiteit binnen de aan evolutie onderhevige individuen reguleert, en zo balanceert op de edge of chaos, zal deze suggestie dan ook met andere modellen moeten worden ondersteund. Dit geeft Kauffman zelf ook aan:

*"We have examined coevolution in the limited context of coupled NK landscapes and without explicit consideration either of population flow on each landscape or of the behaviour of coevolving ecosystems subject to invasions by new species. Further sophistication in these directions is certainly needed."* [8]

Om deze reden is ten behoeve van deze paper het NSBN model ontwikkeld. In tegenstelling tot het NK model bevat het NSBN model wel de verplaatsing van soorten en de verdrinking van soorten door een andere soort. Daarnaast is het NSBN model bewust dichter bij de natuurlijke evolutie gehouden door gebruik te maken van intrinsieke selectie in plaats van artificiële selectie. Dit verschil ten opzichte van het NK model vergroot de algemene aard van de resultaten van het NSBN model.

## 7. Het NSBN model

Het NSBN (Natural Selection Boolean Network) model is gebaseerd op het door Kauffman en Levin ontwikkelde NK model [9][13] en Random Boolean Networks [14][15]. Dit is terug te zien in de structuur en de parameters van het booleaanse netwerk dat elke agent bevat. Naast het gebruik van een booleaans netwerk is het NSBN model echter volledig nieuw ontwikkeld. Vanwege de gelijkenis met het NK model, inclusief de toepassing van het booleaanse netwerk, is het mogelijk om de resultaten van het NSBN model te vergelijken met de bevindingen van Kauffman met betrekking tot co-evolutie [5] en het beschikbare onderzoek [16] naar random boolean networks.

In tegenstelling tot het NK model maakt het NSBN model geen gebruik van een fitness functie. In plaats daarvan vindt een intrinsieke vorm van selectie plaats. Deze selectie komt voort uit het feit dat agenten actief voedsel moeten verzamelen en dat voortplanten voedsel kost. Zo kunnen agenten die succesvoller zijn in het verzamelen van voedsel zich vaker voortplanten en vormen hun genen (in de vorm van hun netwerk) procentueel gezien een groter aandeel in de totale genenpoel. Doordat alle agenten voor hetzelfde voedsel strijden, ontstaat een vorm van co-evolutie. Dit co-evolutionaire effect wordt versterkt wanneer agenten ook de mogelijkheid krijgen elkaar op te eten door middel van het carnivoor percentage (zie 7.3 Agenten). Voortplanten gebeurt op een aseksuele wijze door een kopie van de ouder te maken waarbij verschillende aspecten van het netwerk gemuteerd kunnen raken (zie tevens 7.3 Agenten). De wereld waarin de agenten zich bevinden is discreet en bestaat in de vorm van een toroïdaal raster. Door middel van beurten wordt er nieuw voedsel aan de wereld toegevoegd en krijgen de agenten de mogelijkheid om zich in deze wereld te verplaatsen en zich voort te planten (zie 7.1 Wereld). Hoewel ter verduidelijking verwijzingen zullen worden gedaan naar aspecten binnen de natuurlijke evolutie is het nabootsen van de natuurlijke evolutie op aarde geen doel van het NSBN model.

Bij het definiëren van het model zijn bepaalde parameters variabel gelaten. Bij de uitvoering van de verschillende tests worden deze parameters ingesteld op verschillende waardes (zie 8. Tests en resultaten). Voor de overige parameters zijn waardes gekozen die leiden tot de best waarneembare vorm van co-evolutie. Omdat de te testen hypothese, te weten het optimaliseren van de fitness door optimalisatie van de complexiteit binnen individuen, een algemeen kenmerk van co-evolutie zou zijn, is het gerechtvaardigd om deze laatste groep van parameters vast te zetten op waardes die leiden tot co-evolutie. Bij deze vaste waardes moet onder-meer worden gedacht aan de mutatie

kansen, de grootte van de wereld en de hoeveelheid voedsel die per beurt aan de wereld wordt toegevoegd. Deze en de overige vaste parameters komen in meer detail in de volgende secties aan bod.

De nu volgende gedetailleerde uitleg van het model is opgedeeld in de hoofdelementen van het model. Dit zijn de wereld, de agenten die zich in deze wereld bevinden, het booleaanse netwerk dat elke agent bevat en de manier van voorplanten, inclusief de mutaties die hierbij kunnen optreden.

## 7.1. Wereld

Het gehele ecosysteem van het NSBN model bevindt zich in een rechthoekige, door middel van een raster in cellen verdeelde, discrete wereld. Door de meest linker en rechter, evenals de bovenste en onderste rij, van dit raster aan elkaar te laten grenzen is de wereld toroïdaal van vorm. Elke cel in dit raster kan ten hoogste één agent bevatten en eventueel een hoeveelheid voedsel.

Zoals eerder genoemd werkt het model met discrete beurten. Een beurt begint met het toevoegen van voedsel aan een deel van de cellen. Dit vindt plaats door met een kans van 10% een hoeveelheid tussen de 10 en 80 voedsel eenheden aan elke cel toe te voegen. Er is gekozen voor deze waardes omdat dit in de praktijk tot een verdeling van voedsel in de wereld leidt die in staat is een voldoende grote dichtheid aan herbivoren te ondersteunen om de ontwikkeling van carnivoren mogelijk te maken (zie 7.3).

Na het toevoegen van het voedsel worden de netwerken van alle agenten ge-update. Dit kan leiden tot het zetten van een stap en/of voortplanting. De volgorde waarin agenten worden ge-update wordt bij elke beurt willekeurig bepaald. Dit voorkomt dat een bepaalde agent telkens voor dezelfde andere agent wordt ge-update, en zo voedsel weg kan nemen voor die andere agent. Indien een agent een stap zet naar een cel waar zich voedsel bevindt, wordt dit voedsel aan de agent toegevoegd. Dit is de enige manier waarop het totaal aan voedsel dat de gehele populatie bezit kan worden vergroot. Omdat elke beurt 10 voedsel eenheden kost voor elke agent, onafhankelijk van de acties van de agent, moeten agenten zich dus naar voedsel bewegen om in leven te blijven of, indien het carnivoor percentage ingeschakeld is, andere agenten aanvallen.

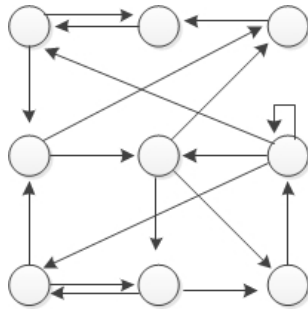
Er is gekozen voor de beschreven discrete wereld en beurten om zo de complexiteit van het model tot een minimum te beperken. Tevens is een model met discrete posities met maximaal één agent per cel relatief (in vergelijking met een niet discrete wereld) goed te vergelijken met het beschikbare onderzoek naar kritieke staten en complexiteit in cellular automata (CA) [17][18][19]. De reden dat niet voor een model in de vorm van een CA is gekozen is dat het gebrek aan verplaatsingen van agenten in een CA het hoofddoel van het model, het wisselende contact tussen subpopulaties, zou wegnemen.

## 7.2. Booleaans netwerk

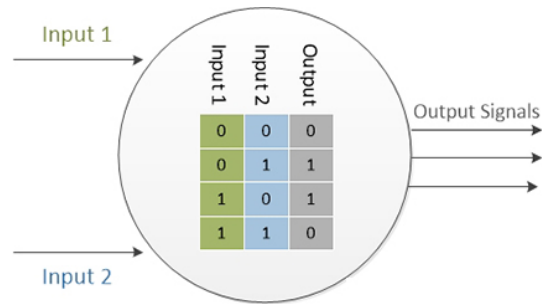
Het booleaanse netwerk dat elke agent bevat speelt een grote rol in het gekozen model. Het is dit netwerk dat kan worden gezien als het genotype en tevens het brein dat volledig verantwoordelijk is voor het gedrag van de agent. Dit netwerk is dan ook, samen met eventueel het carnivoor percentage, als enige aan evolutie onderhevig.

Elk netwerk bestaat uit  $N$  knopen. Het aantal ingaande verbindingen per knoop wordt aangeduid met de parameter  $K$ . Indien er gebruik wordt gemaakt van een strikte  $K$  waarde heeft elke knoop exact  $K$  ingaande verbindingen (zie figuur 2). Naast een strikte  $K$  waarde kan ook gebruik worden gemaakt van een maximale  $K$  waarde of een compleet vrije  $K$  waarde.

Er is reeds een grote hoeveelheid onderzoek [16][20] gedaan naar random boolean networks met betrekking tot complexiteit. Dit onderzoek verantwoordt de keuze voor het gebruik van



Figuur 2. Voorbeeld van incompleet (input en output knopen ontbreken) netwerk met strikte  $K = 2$ ,  $N = 9$



Figuur 3. Voorbeeld knoop met  $K = 2$

booleaanse netwerken als het aan evolutie onderhevige en te testen element in het model. Dit is het geval omdat de resultaten van dit model met de uitkomsten van dit eerdere onderzoek kunnen worden vergeleken. Ook is gebleken dat het ontwikkelen van netwerken in vergelijking met computerprogramma instructies leidt tot grotere kansen voor ontwikkeling door middel van evolutie. De reden hiervoor is voornamelijk de zogenaamde "graceful degradation" eigenschap van booleaanse en neurale netwerken, die computercode niet bezit. [6]

Het in dit model gebruikte booleaanse netwerk bestaat uit drie verschillende soorten knopen:

- algemene knopen
- input knopen
- output knopen

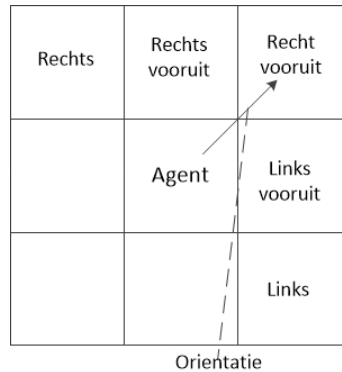
### Algemene knopen

De algemene knopen zijn gelijk aan de knopen zoals gebruikt in random boolean networks. Dit betekent dat iedere algemene knoop  $K$  ingaande verbindingen heeft, maximaal  $N$  uitgaande verbindingen en een booleaanse functie bezit (zie figuur 3). Een knoop met  $N$  uitgaande verbindingen is dus verbonden met iedere andere knoop en met zichzelf. De booleaanse functie die elke combinatie van ingaande signalen omzet in één output waarde is in de knoop opgeslagen in de vorm van een tabel. Hierdoor kan een knoop worden gezien als een binaire gate met  $K$  ingaande signalen.

Een agent kan door middel van de uit algemene knopen gevormde algemene kern (die kan worden vergeleken met een hidden layer, zie ook figuur 5), in combinatie met de output knopen, middels evolutie in verschillende situaties leren verschillend te handelen. Verwacht wordt dat door de toevoeging van deze algemene knopen een agent complexer gedrag kan vertonen dan mogelijk zou zijn met uitsluitend input en output knopen. Zo is het theoretisch mogelijk voor een agent om pas na een vertraging van meerdere beurten na het waarnemen van een bepaalde omgevingsvariabele te handelen. Dit zou bijvoorbeeld kunnen worden geïmplementeerd door het signaal eerst door een reeks algemene knopen te laten lopen (zie 7.2.3 Updaten netwerk), alvorens het bij de output knoop uitkomt.

### Input knopen

De algemene knopen alleen stellen de agent niet in staat informatie binnen te krijgen over zijn omgeving, noch stellen ze de agent in staat te bewegen of voort te planten. Om de agent van informatie over zijn directe omgeving te voorzien zijn de input knopen aan het netwerk toegevoegd. Deze knopen bevatten geen ingaande verbindingen en geen output tabel. De output van deze knopen wordt namelijk bepaald door de directe omgeving van de agent. Door middel van specifieke input



Figuur 4. Oriëntatie en zichtbare velden

knopen kan de agent zo enkele velden om zich heen zien; welke velden dit zijn hangt af van de oriëntatie van de agent. In totaal kan de agent 5 van de 8 omliggende velden zien (zie figuur 4). Voor elk van deze velden geeft één knoop (door middel van een hoge output waarde) aan of er zich daar een agent bevindt, en een andere knoop of er zich daar voedsel bevindt. Dit zorgt voor een totaal van 10 input knopen.

Er is gekozen voor een even aantal zichtbare velden rechts en links van het veld dat hoort bij de oriëntatie van de agent. Hiervoor is gekozen in de verwachting dat de eenvoud van het netwerk door middel van deze symmetrie wordt vergroot. Daarbij bleek dat 1 of 3 velden niet voldoende informatie gaven om langdurige optimalisatie door middel van evolutie mogelijk te maken, wat bij 5 velden wel het geval is.

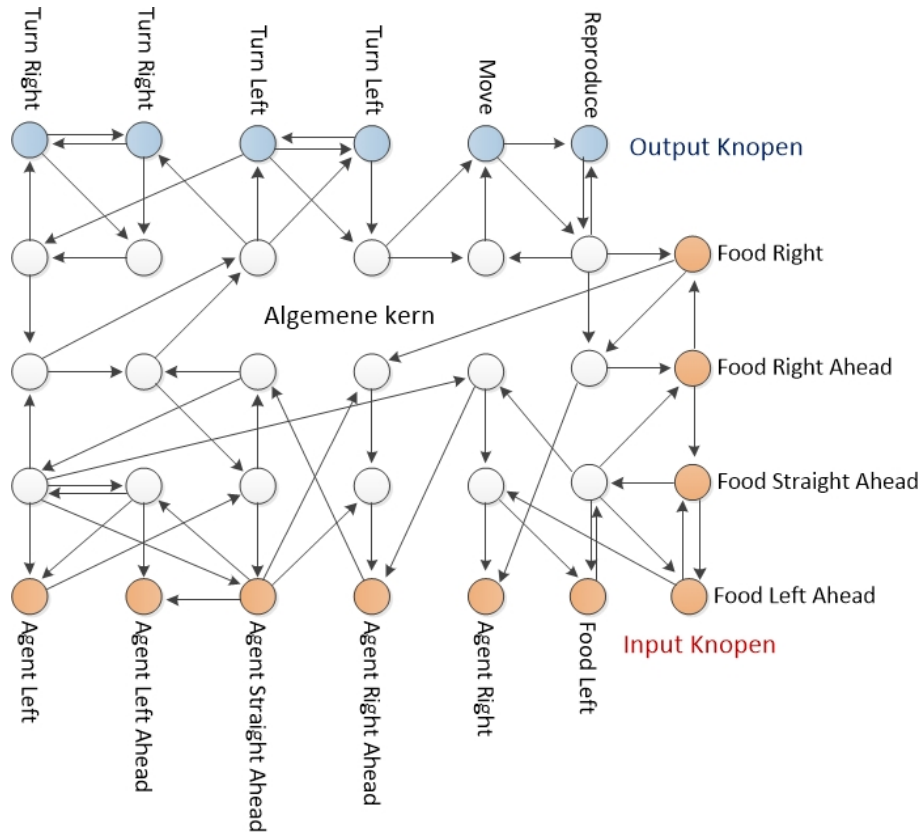
### Output knopen

Om de agent in staat te stellen te reageren op de via de input knopen verkregen informatie bevat het netwerk output knopen. Met behulp van deze knopen kan de agent zijn oriëntatie veranderen, vooruit lopen en voortplanten. Elke agent bevat de volgende 6 output knopen:

- Turn Left
- Turn Left
- Turn Right
- Turn Right
- Move
- Reproduce

Nadat het netwerk van een agent is geupdate worden deze knopen in bovenstaande volgorde uitgelezen. Dit wil zeggen dat de output van deze knopen wordt bekeken en wanneer deze 1 is wordt de staat van de agent in de wereld veranderd. Deze verandering van de wereld houdt bij de Turn Left en Turn Right knopen in dat de oriëntatie 1 of 2 velden naar links of rechts draait. Als zowel een Turn Left als een Turn Right knoop 1 teruggeven, heffen deze elkaar op. Wanneer de Move knoop hoog is, zet de agent een stap naar de cel die hoort bij zijn oriëntatie, maar enkel als het veld leeg is of als het carnivoor percentage is ingeschakeld (zie 7.3.2 Carnivoor percentage). Hierna wordt gekeken naar de output van de Reproduce knoop. Als deze knoop 1 teruggeeft en het veld achter de agent leeg is, wordt in dit veld een gemuteerde kopie van de agent geplaatst (zie 7.3.1 Voortplanting).

Deze collectie van 6 output knopen is met uitzondering van de tweede Turn Left en Turn Right knopen, het minimum aan knopen dat de agent in staat stelt om zich voort te planten en te bewegen.



Figuur 5. Compleet strikt  $K = 2$  netwerk,  $N = 34$

Deze extra 2 knopen zijn toegevoegd om de agent in staat te stellen naar alle velden te kunnen bewegen die hij kan zien, om zo meer gedrag en dus mogelijkheden voor evolutie mogelijk te maken. Het geheel aan knopen vormt een netwerk zoals het netwerk in figuur 5. Hier zijn in het rood de input knopen te zien, in het blauw de output knopen en in het grijs de algemene knopen. Zoals te zien is in dit figuur kan een signaal dus rechtstreeks of via de algemene kern een output knoop aansturen.

### Updaten netwerk

Het netwerk wordt ge-update op de in de meeste onderzoeken naar random boolean networks (RBNs) toegepaste synchrone wijze. Hierdoor staat het dichtst tot de "classical random boolean networks" (CRBNs) [21]. Dit vereenvoudigt opnieuw de vergelijking van de uitkomsten van het NSBN model met die van het onderzoek naar RBNs.

Een update van een bepaalde knoop in beurt  $t$  wordt uitgevoerd door voor alle knopen, wiens uitgaande signalen als ingaand signaal dienen voor de betreffende knoop. Hierbij wordt gekeken naar de output waarden van deze knopen op beurt  $t - 1$ . Met behulp van de interne tabel wordt vervolgens bepaald wat de nieuwe output waarde van de aan de update onderhevige knoop wordt. De input knopen vormen hierop een uitzondering, hun output waarde wordt bepaald door de omgeving van de agent (zie 7.2).

### 7.3. Agenten

Een agent bestaat uit een booleaanse netwerk dat zijn gedrag bepaalt, en heeft verder een leeftijd, oriëntatie en een hoeveelheid voedsel. Wanneer een agent de maximale leeftijd van 300

beurten bereikt sterft hij altijd. Dit voorkomt dat agenten die niet meer in staat zijn zich voort te planten, door een mutatie aan hun netwerk, eindeloos doorleven. Tevens vergroot beperken van de levensduur de diversiteit in genen. De maximale leeftijd van 300 geeft in de praktijk voldoende tijd om meerdere keren voort te planten. Naast het bereiken van de maximale leeftijd sterft een agent wanneer hij geen voedsel meer heeft, of wanneer hij wordt aangevallen en verliest.

### **Voortplanting en mutaties**

Wanneer de Reproduce knoop van een agent een hoge output geeft na het updaten van het netwerk, wordt gekeken of de cel achter de agent (de cel aan de tegenovergestelde zijde van de agent ten opzichte van zijn oriëntatie) leeg is. Indien dit het geval is, wordt in deze lege cel een gemuteerde kopie van de ouder geplaatst. Deze nieuwe agent krijgt 50 voedsel eenheden, om zo een kans te hebben zelf eten te vinden. Het maken van een kind kost 300 voedsel eenheden omdat dit, gegeven de hoeveelheid voedsel die per beurt aan de wereld wordt toegevoegd, voor een voldoende hoge dichtheid van herbivoren zorgt. Deze dichtheid is nodig om carnivoren, in het geval van een ingeschakeld carnivoor percentage, een kans te geven te ontwikkelen. Bij het maken van de kind worden verschillende vormen van mutaties toegepast op elke knoop in het netwerk, namelijk:

- Mutatie van de ingaande verbindingen:
  - Verwijdering van een ingaande verbinding
  - Toevoeging van een ingaande verbinding
- Mutatie van de interne tabel

Het toevoegen van een ingaande verbinding houdt in dat er een nieuwe verbinding wordt gemaakt tussen de betreffende knoop en een willekeurige andere nog niet met de betreffende knoop verbonden knoop. Omdat er daarna een extra ingaand signaal is, dient ook de tabel in de knoop te worden uitgebreid om alle nieuwe mogelijke input combinaties te bevatten. Dit wordt bewerkstelligd door de tabel te verdubbelen in grootte waarbij de nieuwe tweede helft een kopie is van de originele tabel. Vervolgens wordt, om te zorgen dat de nieuwe verbinding niet geheel neutraal is, 10% van de entrys in de tabel gemuteerd. De gehanteerde 10% is gekozen omdat deze in de praktijk voldoende non-neutrale combinaties oplevert om een fitness verhoging mogelijk te maken. Een hoger mutatie percentage maakt de kans op een verbetering te klein als gevolg van de grote verandering in functionaliteit van de knoop.

Het verwijderen van een ingaande verbinding houdt in dat de verbinding wordt verwijderd en de tabel gehalveerd. Hierbij wordt met een kans van 50% de helft van de tabel behouden waarin de verbinding hoog was.

Naast het verwijderen en toevoegen van verbindingen kan ook de tabel gemuteerd raken zonder dat er een verbinding verandert. Dit houdt in dat met een bepaalde kans per entry de bijbehorende output wordt geïnverteerd.

Uit tests blijkt een mutatiekans van 0.5% voor ieder van de drie vormen van mutatie een optimaal resultaat te geven. Dit wil zeggen dat een lagere of hogere mutatie kans niet tot hogere optima leidt. Om deze reden zijn de mutatie kansen in het NSBN model dan ook gelijk gesteld aan 0.5%.

Naast de genoemde mutaties aan het netwerk is er ook de mogelijke mutatie van het carnivoor percentage wanneer deze is ingeschakeld. Dit gebeurt door met een kans van 1 procent aan het carnivoor percentage een waarde tussen de -2% en 2% toe te voegen. Hierbij kan het percentage niet kleiner dan 0% of groter dan 100% worden. Deze waarden zijn gekozen omdat ze in de praktijk tot een grote reeks aan verschillende carnivoor percentages leiden.



## Carnivoor percentage

De concurrentie in het NSBN model, zoals tot dit punt beschreven, berust enkel op het verkrijgen van hetzelfde gelimiteerde voedsel. Hierdoor zou kunnen worden gesteld dat er niet voldoende interactie plaatsvindt om van co-evolutie te kunnen spreken. Om deze reden is het carnivoor percentage aan het model toegevoegd.

Het carnivoor percentage geeft een agent nog een extra eigenschap, te weten hoe zeer hij carnivoor is. Dit percentage is omgekeerd evenredig aan het percentage dat de agent herbivoor is, en bepaalt de kans dat een agent wint wanneer hij aanvalt of wordt aangevallen. Een aanval vindt plaats wanneer een agent een stap naar een bezet veld doet. Door de mogelijkheid andere agenten aan te vallen en te specialiseren tot carnivoor, herbivoor of een mix hiervan, ontstaat een directe vorm van co-evolutie.

Agenten met een ingeschakeld carnivoor percentage kunnen niet alleen meer nageslacht produceren door meer voedsel uit cellen te verzamelen, maar ook door andere agenten aan te vallen (waarbij de verliezende agent wordt verwijderd). Hierdoor kan de grootte van de populatie niet meer als impliciete fitness worden gezien. Dit is dan ook de reden dat het carnivoor percentage als optionele parameter is geïmplementeerd. Op deze manier kan het carnivoor percentage worden gebruikt om de resultaten van de tests zonder carnivoor percentage te bevestigen.

Wanneer het carnivoor percentage is ingeschakeld krijgt een agent niet de volledige hoeveelheid voedsel van een cel wanneer hij zich naar deze cel beweegt. De hoeveelheid voedsel die hij krijgt kan worden berekend met formule (6).

$$(1 - \text{carnivoor percentage agent}) * \text{voedsel in cel} \quad (6)$$

Als gebruik wordt gemaakt van het carnivoor percentage is het ook mogelijk voor agenten om andere agenten aan te vallen. Dit kan worden gedaan door een stap te zetten naar een cel waarin zich reeds een agent bevindt. De winnaar van dit gevecht is altijd de agent met het hoogste carnivoor percentage. Indien beide agenten een gelijk percentage hebben wint de aanvaller met 50% kans.

Tot slot beïnvloedt het carnivoor percentage ook de hoeveelheid eten die de winnende partij na een aanval krijgt (7).

$$\text{carnivoor percentage winner} * \text{hoeveelheid voedsel van verliezer} \quad (7)$$

Een hoger carnivoor percentage zorgt ervoor dat een agent van een groter deel van de mogelijke carnivoor percentages wint. Dit geldt zowel wanneer hij zelf aanvalt als wanneer hij wordt aangevallen. Hierdoor heeft evolutie de neiging dit percentage te vergroten. Een groter carnivoor percentage levert echt ook een verkleinde opbrengt aan voedsel van cellen op, wat de neiging het percentage te vergroten weer tegenwerkt. Deze voor en nadelen van een hoger percentage zorgen ervoor dat er in de praktijk een populatie ontstaat met groepen van verschillende carnivoor percentages. Deze vergrote diversiteit en interactie mogelijkheden tussen groepen zorgen voor een sterkere vorm van co-evolutie.

## 8. Tests en resultaten

### 8.1. Uitgevoerde tests

Om bij te dragen aan het beantwoorden van de eerder geformuleerde onderstaande onderzoeksvraag zijn meerdere tests uitgevoerd.

*Optimaliseren co-evolutionaire ecosystemen de fitness van individuen door de complexiteit*

*binnen individuen te optimaliseren?*

Hiervoor zijn drie tests ontwikkeld. In test 1 is getest of bij afwezigheid van andere K waardes bepaalde specifieke K waardes het best presteren. In test 2 is gekeken of dezelfde K waardes ook in de aanwezigheid van andere K waardes het best presteren. Tot slot is in test 3 gekeken of de co-evolutie in het NSBN model in staat is K te veranderen in de richting van de optimale K waardes, zoals gevonden in test 1 en 2. Het gebruikte programma [22] is beschikbaar op github.

### **Test 1: Individuele K waardes**

Deze test omvat runs met 1000 initiële agenten, ieder met dezelfde strikte K waarde. Per run is getest met één K waarde (variërend van K0 t/m K10) en zonder carnivoor percentage. Bij deze runs is gekeken naar het verloop van de groei van het totaal aantal agenten. Omdat het carnivoor percentage uitgeschakeld is kan de populatie grootte worden gebruikt als maat voor de fitness. Zodoende kan worden gezien welke K waardes tot de hoogste fitness waarde leiden door de populatiegrootte per beurt bij te houden.

### **Test 2: Gemixte K waardes.**

Deze test omvat runs met een gemixte populatie van 100 agenten per strikte K waarde (K0 t/m K10). In deze runs is gekeken naar het aantal agenten per K waarde gedurende de run. Zo kan worden nagegaan of de bij test 1 gevonden best presterende K waardes bij aanwezigheid van agenten met afwijkende K waardes nog steeds het best presteren. Dit is gemeten door het aantal agenten van iedere K waarde per beurt bij te houden. Door deze waardes met elkaar te vergelijken kan worden achterhaald welke K waardes de hoogste fitness (en dus de grootste populatie) behalen.

Deze test is tevens uitgevoerd met een ingeschakeld carnivoor percentage, om zo te verifiëren of dezelfde K waardes ook bij de hierdoor vergrote interactie de hoogste fitness behalen.

### **Test 3: Niet strikte K waardes**

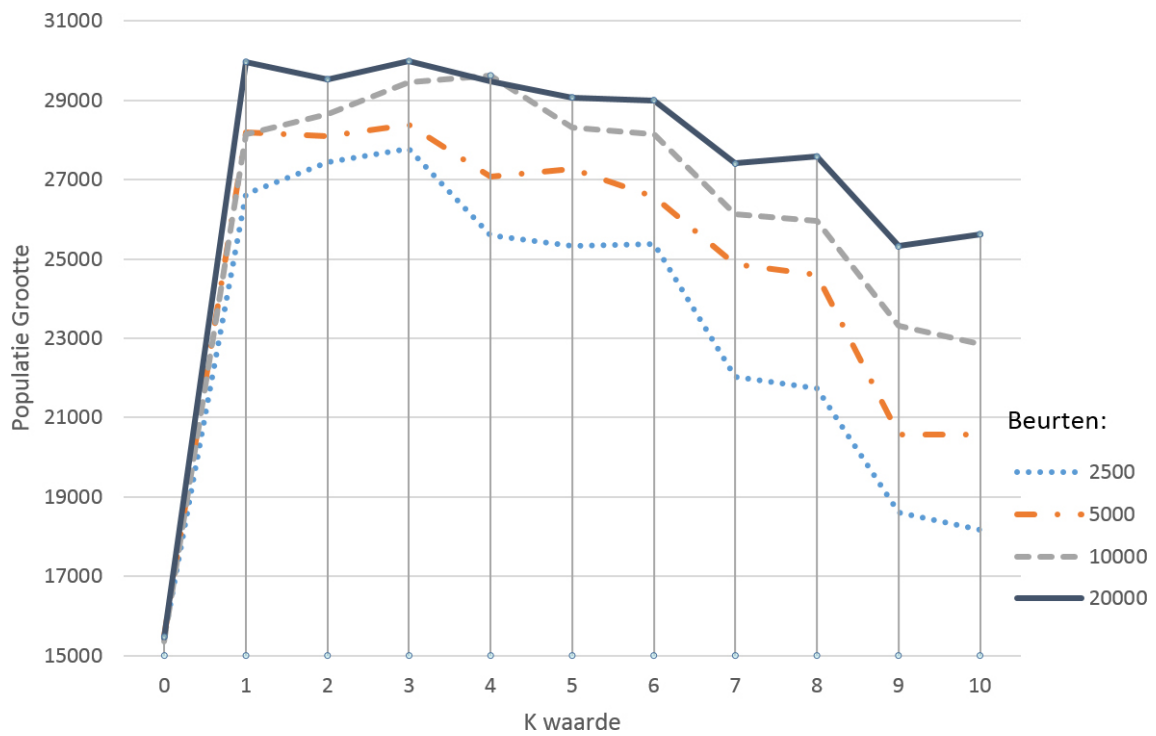
In test 3 is getest of co-evolutie zelf de K waardes aanpast naar de in test 1 en 2 gevonden optimale K waardes. Hiervoor zijn runs uitgevoerd met ieder 1000 agenten met een niet strikte K waarde van 0 (de laagst mogelijke K waarde). Daarnaast zijn runs uitgevoerd met K10 omdat K10 zowel computationeel haalbaar is, als significant lagere fitness waardes oplevert dan de in 8.2 beschreven optimale K waardes. Doordat K1 t/m K6 in test 1 het beste blijken te presteren, is getest of co-evolutie in de runs met initiële K waardes van 0 de K waarde ophoogt. Bovendien zijn tests met K10 uitgevoerd om na te gaan of co-evolutie deze K waarde verlaagt om zo tot de optimale K waardes te komen. De runs zijn herhaald met een ingeschakeld carnivoor percentage.

Alle tests zijn uitgevoerd bij N waardes van 16 (uitsluitend input en output knopen), 25 en 40 knopen

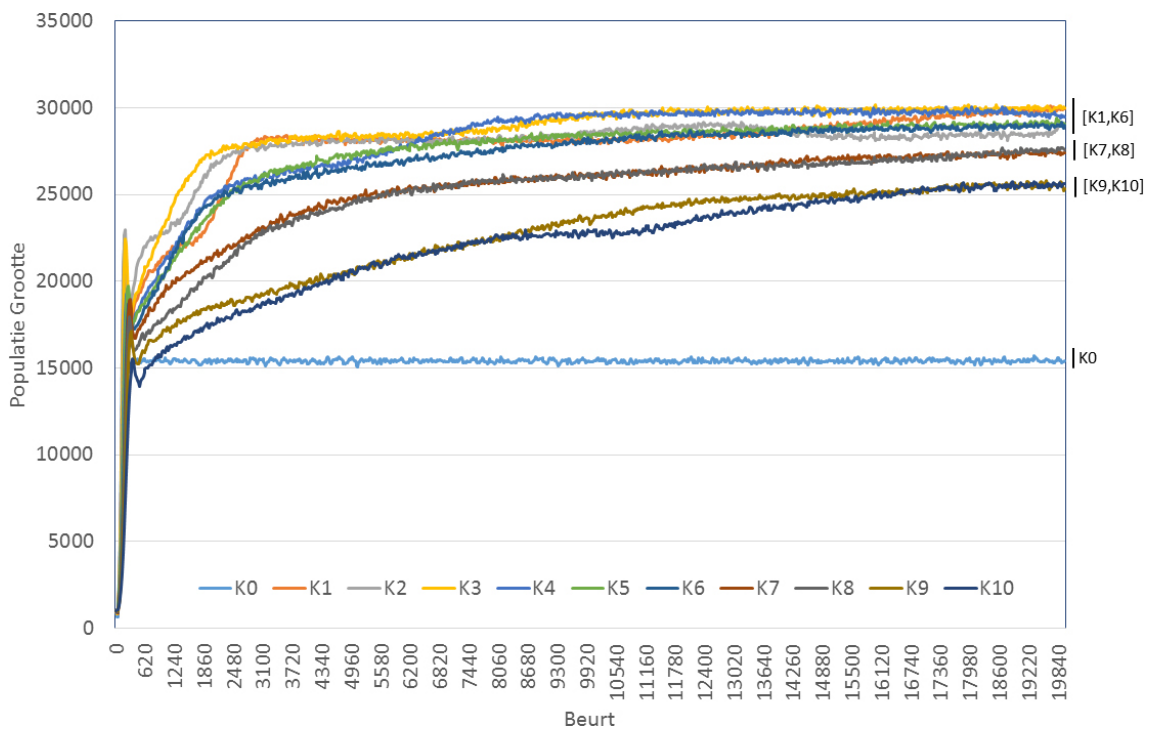
## **8.2. Resultaten NSBN model**

### **Test 1: Individuele K waardes.**

Zoals te zien is in figuur 6 is er een duidelijk verschil in het aantal agenten, voortkomend uit het verschil in K waardes. Zoals verwacht presteert K0 het slechtst (behaalt de kleinste populatiegrootte), wat te wijten valt aan het totale gebrek aan evolutie door de afwezigheid van verbindingen tussen de knopen. Opvallend is echter dat er nauwelijks verschil in populatiegrootte is binnen K1 t/m K6 wanneer het optimum is bereikt. Dit komt ook naar voren in figuur 7. Hierin is te zien dat K1 t/m K6 elkaar gedurende de evolutie afwisselen wat de grootste populatie betreft.



Figuur 6. Populatie groottes bij 2500, 5000, 10000 en 20000 beurten van typische runs van individuele K waardes bij  $N = 16$  en uitgeschakeld carnivoor percentage



Figuur 7. Populatie groottes van typische runs van individuele K waardes weergegeven per beurt bij  $N = 16$  en uitgeschakeld carnivoor percentage

Hierbij is het verschil in populatiegrootte later in het evolutieproces kleiner dan in de eerdere generaties. Dit is opvallend doordat de dynamiek van random boolean networks sterk afhangt van de  $K$  waarde, waarbij een klein verschil in  $K$  een groot verschil in de dynamiek, en dus een grote afwijking van de optimale kritieke staat, tot gevolg heeft [20]. Deze gevoeligheid lijkt afwezig in het NSBN model. Een mogelijke verklaring voor de schijnbare ongevoeligheid in het NSBN model is dat evolutie compenseert voor het verschil in  $K$ . Een manier waarop dit door evolutie zou kunnen worden bewerkstelligd is het functioneel gezien neutraal maken van verbindingen door middel van de interne tabellen van de knopen. Hierdoor bestaat de verbinding nog wel maar wordt de output van de knoop in de praktijk uitsluitend bepaald door de overige ingaande signalen. De  $K$  waarde ( $K_2$ ) waarbij random boolean networks gemiddeld in de kritieke staat verkeren ligt wel binnen de reeks optimaal presterende  $K$  waardes van het NSBN model.

Wanneer de  $K$ -waarde groter dan 6 wordt gekozen, beginnen de optima te dalen richting die van  $K_0$ . Ook wordt duidelijk uit figuur 7 dat voordat de optima zijn bereikt  $K_1$  t/m  $K_6$  altijd een grotere populatie voortbrengen (en dus een hogere fitness hebben) dan  $K_0$  en  $K > 6$ . Dit is in overeenstemming met de observatie van Kauffman in NK modellen, waarin de optimale  $K$  waardes zowel tot een hogere fitness leiden tijdens het zoeken naar een optimum als wanneer het optimum is bereikt [5]. De dalende fitness waardes bij  $K > 6$ , en dus de toenemende complexiteit, komen overeen met de door Kauffman geconstateerde complexity catastrophe.

Rond beurt 20.000 hebben de populaties van alle  $K$  waardes hun (lokale) optimum bereikt. De verschillende gebruikte  $N$  waardes (16, 25 en 40) leverden geen verschil op in de bereikte optima.

### **Test 2: Gemixte $K$ waardes**

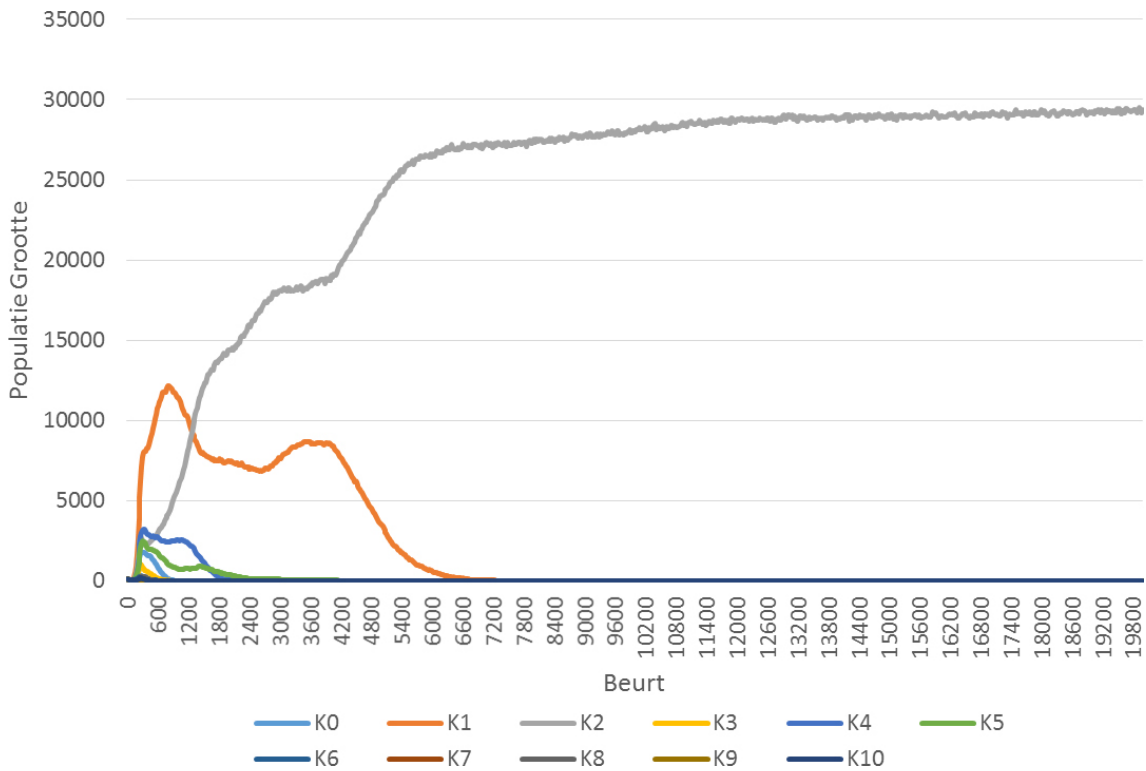
Uit de gemixte runs is gebleken dat op den duur altijd slechts een populatie van één  $K$  waarde overblijft, zie figuur 8. De tijd die het kost voordat de overige  $K$  waardes zijn uitgestorven, verschilt per run maar blijkt zonder carnivoor percentage altijd onder de 20.000 beurten, en mét carnivoor percentage altijd onder de 30.000 beurten te liggen. In de 40 uitgevoerde runs, waarvan 20 met en 20 zonder carnivoor percentage per  $N$  waarde, overleefden alleen  $K_1$  t/m  $K_4$ . Dit komt deels overeen met de best presterende  $K$  waardes uit test 1, met uitzondering van  $K_5$  en  $K_6$ , die het in de gemixte runs nooit redden. Het uitsterven van  $K_5$  en  $K_6$  kan mogelijk verklaard worden doordat in de eerste 5.000 beurten in test 1  $K_5$  en  $K_6$  slechter presteren dan later in het evolutie proces ten opzichte van  $K_1$  t/m  $K_4$ . Dit is dan ook naar alle waarschijnlijkheid de reden dat  $K_5$  en  $K_6$  al in een vroeg stadium worden overwonnen door  $K_1$  t/m  $K_4$ .

De overlap tussen de best presterende  $K$  waardes bij test 1 en 2 ondersteunt het bestaan van optimale  $K$  waardes (en de bijbehorende optimale complexiteit) binnen het in het NSBN model aanwezige co-evolutionaire ecosysteem.

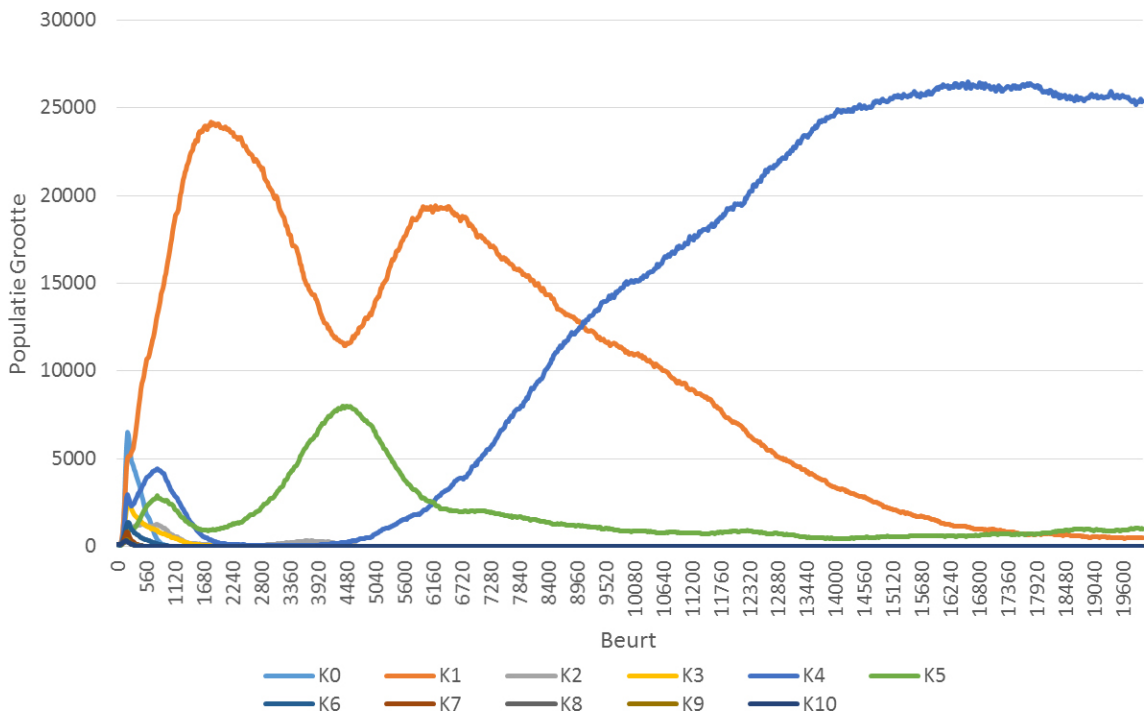
Ook in de test 2 bleek het variëren van  $N$  niet van invloed op de uitkomst. Verder bleek het inschakelen van het carnivoor percentage geen invloed te hebben op welke  $K$  waardes overleven. Wel zorgt het carnivoor percentage ervoor dat het gemiddeld langer duurt voordat de populatie uit één  $K$  waarde bestaat, zie figuur 9. Dit komt voort uit het ontstaan van 2 soorten individuen van verschillende  $K$  waardes met een roofdier-prooi relatie, waarbij soort A voornamelijk voedsel uit de cellen verzamelt, terwijl soort B voornamelijk soort A eet. Omdat er een  $x$  aantal van soort A nodig zijn voor iedere B om te kunnen overleven, houden deze soorten zich in evenwicht. Een evenwicht dat echter wankel blijkt, gegeven het uitsterven van één van de twee soorten.

### **Test 3: Niet strikte $K$ waardes**

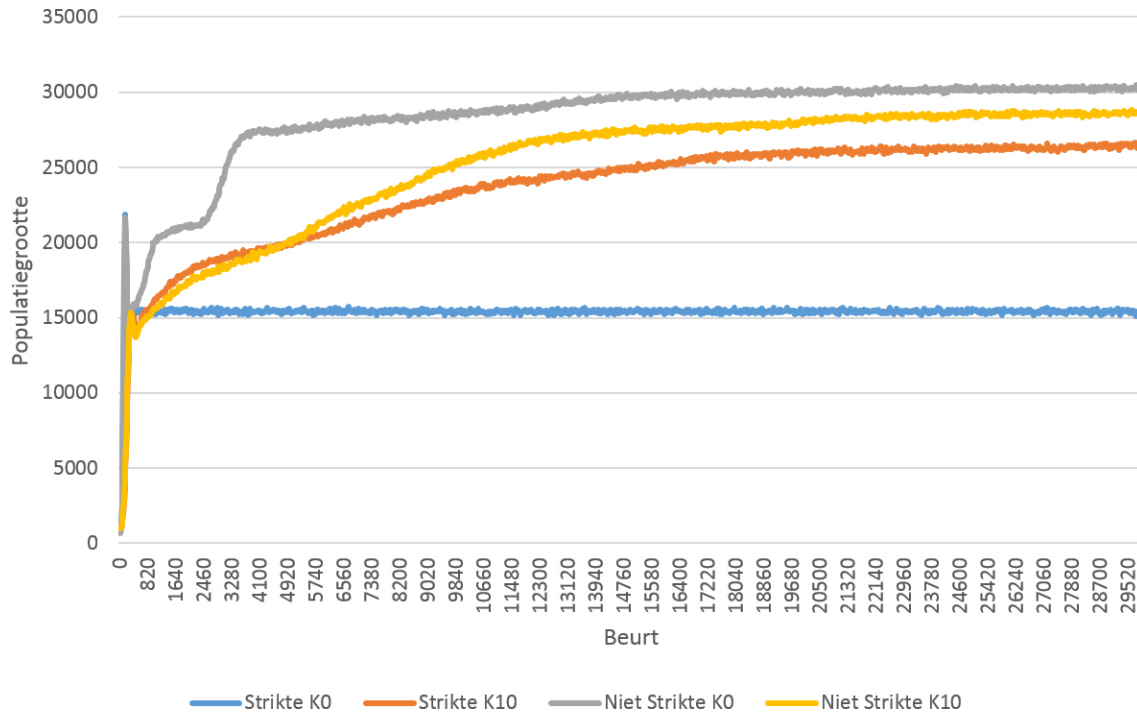
Test 3 levert opvallende resultaten op. In test 1 en test 2 is aangetoond dat het optimum voor de optimale  $K_1$  t/m  $K_4$  waardes op een populatiegrootte van 30.000 agenten ligt. Wanneer evolutie in staat zou zijn de  $K$  waarde te optimaliseren zou het, gegeven voldoende generaties, dus het



Figuur 8. Populatie aantallen van een typische gemixte run weergegeven per beurt, bij  $N = 16$  en uitgeschakeld carnivoor percentage



Figuur 9. Populatie aantallen van een typische gemixte run weergegeven per beurt, bij  $N = 16$  en ingeschakeld carnivoor percentage



Figuur 10. Vergelijking van strikte en niet strikte K0 en K10 populaties per beurt weergegeven

optimum van 30.000 agenten moeten kunnen bereiken. Dit geldt ook als wordt begonnen met een K waarde buiten  $[K1, K4]$ . Wanneer dus met een populatie van niet strikte K10 agenten wordt begonnen zou deze populatie groter moeten worden dan wanneer met een strikte K10 populatie wordt begonnen, doordat evolutie in dit geval K verlaagt. Uit figuur 10 blijkt echter dat een niet strikte K10 populatie wel beter presteert dan een strikte K10, maar nog altijd minder dan de optimale K waardes. Hierdoor wordt slechts een optimum van ca. 28.0000 agenten bereikt. De co-evolutie in het NSBN model is bij K10 dus niet in staat gebleken K voldoende verlagen om de fitness van de populatie gelijk aan de optimale K waardes te maken. Kauffmans suggestie dat co-evolutie K optimaliseert, blijkt in het NSBN model dus slechts deels het geval. Dit komt waarschijnlijk voort uit de beperkte hoeveelheid soorten in het NSBN model en het beperkte aantal generaties voordat de gehele populatie uitontwikkeld is. Test 1 en 2 tonen namelijk wel een duidelijk hogere fitness waarde voor K1 t/m K4. Dit leidt tot de suggestie dat wanneer bijvoorbeeld door middel van neutrale mutaties soorten ontstaan met optimale K waardes, deze soorten het zullen overnemen van suboptimale K waardes waardoor uiteindelijk de populatiegrootte van 30.000 agenten wordt behaald. De relatief snelle convergentie naar één soort en het relatief snelle bereiken van optima in het NSBN model ontnemen co-evolutie deze mogelijkheid doordat de gehele populatie vroegtijdig in een lokaal optimum strandt.

De runs waarbij is begonnen met niet strikte K0 populaties tonen wel een grote verbetering ten opzichte van de runs met strikte K0 populaties. In de niet strikte runs blijken de K0 populaties hetzelfde optimum als strikte K1 t/m K4 populaties te behalen. Dit komt mogelijk voort uit de het feit dat K0 veel dichterbij de optimale K waardes ligt dan K10 waardoor een kleinere optimalisatie nodig is om tot de optimale fitness te komen. Om deze suggestie nader te onderzoeken is getest met populaties van niet strikte K7, die net zo goed bleken te presteren als strikte K1 t/m K4 populaties. Dit ondersteunt de bevinding dat co-evolutie in het NSBN model slechts in staat is een kleine optimalisatie van K te bewerkstelligen.

Ook in test 3 bleken het carnivoor percentage en N niet van invloed op de uitkomst.

## 9. Conclusie

Het doel van deze paper was het leveren van een bijdrage aan verder onderzoek naar de invloed van de interacties tussen genen van individuen op de fitness landschappen van deze individuen. De uitgevoerde tests met het hiervoor specifiek ontwikkelde NSBN model tonen aan dat K in het NSBN model duidelijk invloed heeft op de fitness. Deze invloed blijkt zowel aanwezig tijdens het zoeken naar, als na het bereiken van, een lokaal optimum. Dit resultaat komt overeen met de door Kauffman genoemde eigenschappen van het NK model [5].

In tegenstelling tot Kauffmans bevindingen uit het NK model blijken in het NSBN model echter meerdere optimale K waardes aanwezig te zijn, bestaande uit de reeks K1 t/m K4. De fitness van hogere K waardes neemt af richting de fitness van K0 naarmate K verder toeneemt. Omdat met het toenemen van K het aantal interacties, en dus de complexiteit, toeneemt lijkt deze dalende fitness overeen te komen met de zogenoemde complexity catastrophe, zoals ook waargenomen in het NK model. Verder onderzoek naar deze overeenkomst is echter vereist.

De kern van de onderzoeksvraag van deze paper is of co-evolutie zelf de complexiteit optimaliseert. Deze optimalisatie blijkt in het NSBN model deels aanwezig. Wanneer wordt begonnen met een initiële populatie met bijna optimale K waardes, zoals een niet strikte K0 of K7 populatie, is een duidelijke verbetering te zien. Deze populaties behalen dan ook optima die gelijk zijn aan die van de strikte optimale K waardes. Wanneer wordt begonnen met een verder van de optimale K waardes afliggende waarde, zoals K10, vindt slechts een kleine verhoging van de fitness plaats ten opzichte van strikte populaties met dezelfde initiële K waarde. Dit geeft aan dat co-evolutie in het NSBN model maar beperkt in staat is K te optimaliseren. De waarschijnlijke verklaring hiervoor is dat het NSBN model na een beperkt aantal beurten uitontwikkeld raakt doordat het strandt in lokale optima. Wanneer langdurige co-evolutie van verschillende soorten zou plaatsvinden, geeft dit het evolutieproces meer tijd om K te optimaliseren voordat de soort in een lokaal optimum terecht komt.

Ondanks de beperkte optimalisatie in het NSBN model als gevolg van het gebrek aan langdurige evolutie zijn de resultaten van het NSBN model wel toepasbaar voor de verbetering van evolutionaire algoritmes (EAs). Net als in het NSBN model missen EAs over het algemeen langdurige evolutie [1]. Indien er sprake is van de aanwezigheid van een optimale complexiteit in combinatie met beperkte optimalisatie (een algemene eigenschap van EAs met een beperkt aantal generaties aan evolutie) zoals in het hierboven besproken NSBN model, loont het dus om vooronderzoek te doen naar de optimale complexiteit. Het starten met deze optimale complexiteit leidt namelijk tot hogere fitness waardes, en dus betere oplossingen, zowel tijdens de evolutie als wanneer een lokaal optimum is bereikt.

Met behulp van het NSBN model is het gelukt de algemene aard van bepaalde co-evolutionaire eigenschappen uit het NK model, waaronder de edge of chaos en het bestaan van optimale K waardes, verder te ondersteunen. Het NSBN model is echter slechts één van de mogelijke modellen, en nader onderzoek met behulp van andere fundamenteel verschillende co-evolutionaire modellen is dan ook vereist. Dit geldt ook voor de bevindingen omtrent de optimalisatie van de complexiteit. Deze optimalisatie bleek in het NSBN model slechts deels aanwezig, wat waarschijnlijk voortkomt uit het gebrek aan langdurige co-evolutie van verschillende soorten agenten.

Verder onderzoek met behulp van modellen waarin wel langdurige co-evolutie plaatsvindt is daarom essentieel. Dergelijk onderzoek kan de hypothese dat co-evolutie de complexiteit optimaliseert, op voorwaarde dat voldoende generaties beschikbaar zijn om voldoende optimalisatie mogelijk te maken, bevestigen.

## Referenties

- [1] M. a Bedau, J. S. McCaskill, N. H. Packard, S. Rasmussen, C. Adami, D. G. Green, T. Ikegami, K. Kaneko, and T. S. Ray, Open problems in artificial life., *Artif. Life*, vol. 6, no. 4, pp. 363376, 2000.
- [2] a. D. Channon and R. I. Damper, Perpetuating evolutionary emergence, 1998.
- [3] T. Yu and J. Miller, Neutrality and the Evolvability of Boolean Function Landscape, in *Genetic Programming SE - 16*, vol. 2038, J. Miller, M. Tomassini, P. Lanzi, C. Ryan, A. B. Tettamanzi, and W. Langdon, Eds. Springer Berlin Heidelberg, 2001, pp. 204217.
- [4] G. S. Hornby and J. B. Pollack, Creating high-level components with a generative representation for body-brain evolution., *Artif. Life*, vol. 8, no. 3, pp. 223246, 2002.
- [5] S. a Kauffman and S. Johnsen, Coevolution to the edge of chaos: coupled fitness landscapes, poised states, and coevolutionary avalanches., *J. Theor. Biol.*, vol. 149, no. 4, pp. 467505, 1991.
- [6] a. D. Channon and R. I. Damper, Towards the evolutionary emergence of increasingly complex advantageous behaviours, vol. 31, no. 7, pp. 843860, 2000.
- [7] N. H. Packard, Intrinsic adaptation in a simple model for evolution, in *Artificial life VI*, 1989, pp. 141155.
- [8] S. A. Kauffman, *The Origins of Order: Self Organization and Selection in Evolution*. Oxford University Press, 1993.
- [9] S. Kauffman and S. Levin, Towards a general theory of adaptive walks on rugged landscapes., *J. Theor. Biol.*, vol. 128, no. 1, pp. 1145, 1987.
- [10] S. A. Kauffman and E. D. Weinberger, The NK model of rugged fitness landscapes and its application to maturation of the immune response, *J. Theor. Biol.*, vol. 141, no. 2, pp. 211245, Nov. 1989.
- [11] E. Dubrova, M. Teslenko, and a. Martinelli, Kauffman networks: analysis and applications, *ICCAD-2005. IEEE/ACM Int. Conf. Comput. Des. 2005.*, pp. 478483, 2005.
- [12] R. Albert and H. G. Othmer, The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*, *J. Theor. Biol.*, vol. 223, no. 1, pp. 118, 2003.
- [13] S. A. Kauffman and E. D. Weinberger, The NK model of rugged fitness landscapes and its application to maturation of the immune response, *J. Theor. Biol.*, vol. 141, no. 2, pp. 211245, Nov. 1989.
- [14] S. a Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets., *J. Theor. Biol.*, vol. 22, no. 3, pp. 437467, 1969.
- [15] S. a Kauffman, Homeostasis and differentiation in random genetic control networks, *Nature*, vol. 224, pp. 177178, 1969.
- [16] B. Drossel, *Random Boolean Networks*, arXiv, no. June, 2007.
- [17] M. Mitchell, P. Hraber, and J. P. Crutchfield, Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations, p. 38, 1993.
- [18] W. Li, N. H. Packard, and C. G. Langton, Transition phenomena in cellular automata rule space, *Phys. D Nonlinear Phenom.*, vol. 45, no. 13, pp. 7794, Sep. 1990.
- [19] C. G. Langton, Studying artificial life with cellular automata, *Phys. D Nonlinear Phenom.*, vol. 22, no. 13, pp. 120149, Oct. 1986.
- [20] C. Gershenson, *Introduction to Random Boolean Networks*, Arxiv Prepr. nlin/0408006, p. 14, 2004.
- [21] R. K. Standish, M. A. Bedau, and H. A. Abbass, *Artificial Life 8*. MIT Press, 2003.
- [22] H.M. Smeitink, <https://github.com/hsmeitink/NSBN-Model>