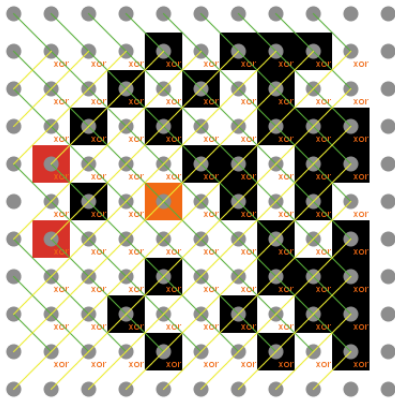
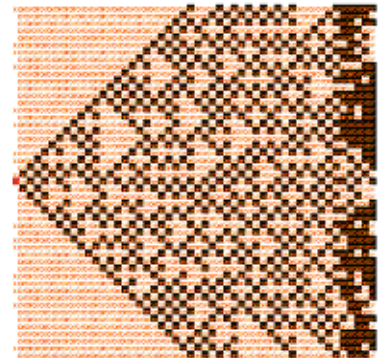


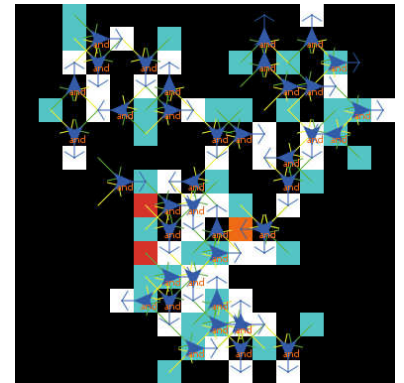
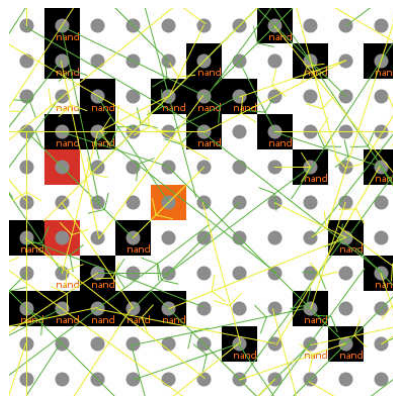
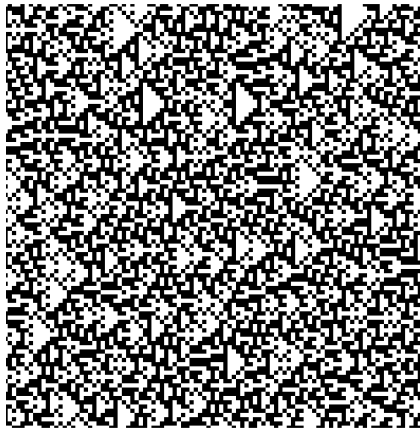
# Selbstorganisation von Informationen

**Jugend forscht 2008/2009  
Mathematik-Informatik**

von  
**Peter Schmitz**



**St. Michael-Gymnasium  
Bad Münstereifel**



## Inhaltsverzeichnis

1. Kurzfassung	2
2. Einleitung	3
3. Analyse von selbstorganisierenden Systemen	4
3.1 Analyse einer Zelle	4
3.2 Analyse von Teilchensystemen	5
3.3 Analyse von assoziativer Speicherung	6
3.4 Informationstheoretische Deutung [9]	6
3.5 Übertragung auf informationsverarbeitende Systeme	7
4. Der Genesis-Algorithmus	8
4.1 Abstraktion	8
4.2 Ansatz	9
4.3 Experiment: Fur-Optimierung	10
4.4 Erläuterung der Programmierung	10
5. Selbstorganisation von Bits in Logikstrukturen	11
5.1 Modell-Ansatz	11
5.2 Strukturbildung	12
5.3 Optimierung einer Kaskade	13
5.4 Optimierung einer freien Vernetzung	14
5.5 Züchtung selbstorganisierender Logik-Strukturen	15
6. Fazit	15
7. Ausblick	16
8. Quellenverzeichnis	16

# 1. Kurzfassung

## Thema: Selbstorganisation von Informationen

Was unterscheidet eigentlich Teilchen von Bits?

Wassermoleküle können sich bei geeigneten Systembedingungen zu Kristallen organisieren. Können auch in informationsverarbeitenden Systemen Effekte auftreten, die zur selbstständigen Strukturbildung von Informationen führen?

Diese Forschungsarbeit bearbeitet eine neue Art von Informatik, ein interdisziplinäres Konzept, das auf der Selbstorganisation der Informationen beruht. Das System liefert einen Gegenpol zu den klassischen, starren Computersystemen, und wäre ideal für Parallelrechner, die dann nichtlineare und komplexe Probleme selbstständig lösen könnten. Forschungen im Bereich selbstorganisierender Algorithmenstrukturen mit Quantensystemen zeigen, wie hochaktuell dieser Ansatz ist.

Zunächst isoliere ich die Faktoren, welche eine Selbstorganisation in physikalischen und biologischen Systemen möglich machen.

Ich übertrage diese Faktoren der Selbstorganisation auf informationsverarbeitende Systeme: Dabei entwickle ich ein Modell, das ein Pendant zur Entstehung von Feldern und Kräften aus den Grundeigenschaften der Teilchen in der Physik darstellt.

Auf Basis meiner Erkenntnisse entwerfe ich einen evolutionären Algorithmus, der eine Population von Reaktionsräumen auf ein vorgebbares Ziel hin optimiert. In jedem dieser Reaktionsräume läuft die oben genannte Modellsimulation ab.

Ein vorläufiges Ziel meiner Arbeit ist dabei die Züchtung von selbstorganisierenden Logik-Strukturen, um eine bestimmte Funktion der Signalverarbeitung zu erreichen.

Bisher konnte ich die Funktion des evolutionären Algorithmus und die Selbstorganisation von Zuständen in Logikstrukturen verifizieren. Zur Zeit arbeite ich an einer Simulation selbstorganisierender Logik-Gatter.

Um eine Anwendung in Quantensystemen möglich zu machen, plane ich ein quantenmechanisches Modell der Wechselwirkung von Informationen. Dazu beschäftige ich mich mit Analogien zwischen Informationstheorie, Quantenelektrodynamik und Thermodynamik.

## 2. Einleitung

Von Nichts kommt Nichts! – Oder?

Selbstorganisation bezeichnet das spontane Auftreten stabiler und effizienter Strukturen und Verhaltensweisen aus Systemen geringerer Ordnung. Der Effekt wird meist beobachtet, wenn sich das System nicht im thermodynamischen Gleichgewicht befindet, also Energie, Stoffe oder Informationen mit der Umgebung austauscht (offene Systeme).<sup>1</sup> Dies macht Selbstorganisation zu einem grundlegenden Phänomen bei der Generierung von Ordnung.

Der Vorteil bei der technischen Anwendung dieses Konzeptes liegt darin, dass der praktisch höchst-mögliche Grad an Effizienz bei einfachsten Komponenten erreicht wird.<sup>3</sup> Selbstorganisierende Systeme weisen, aufgrund der starken Vernetzung ihrer Komponenten, eine nichtlineare Dynamik auf. Sie reagieren mit Anpassung ihrer energetischen und materiellen Regulation auf Umwelteinflüsse (Selbstreferenz). Dabei generieren sie aus Chaos Ordnung. Sie verändern ihre grundlegende Struktur als Funktion ihrer Erfahrung und ihrer Umwelt.<sup>2</sup> Das Prinzip entstammt der interdisziplinären Systemtheorie. Es wurde in Wissenschaftszweigen wie der Quanten- und Strömungsphysik und der Thermodynamik beschrieben und findet mittlerweile breite Anwendung in der Nanotechnologie und Laserphysik.

Dort bildet dieses Konzept einen Gegenpol zu den bisher verwendeten, analytisch konzipierten Ansätzen. Will man beispielsweise im atomaren Bereich Teilchen positionieren, ist keine energie- und zeitaufwendige Verschiebung durch ein Raster-Kraft-Mikroskop nötig. Durch „Verpackung“ in Molekülstrukturen werden den Teilchen stattdessen gezielt Eigenschaften zugewiesen, die zu einem selbstständigen Strukturaufbau führen, wie beispielsweise bei der Bildung eines Kristalls.<sup>3</sup>

Heute werden Möglichkeiten diskutiert, um die Selbstorganisation von Agenten auch in der Informatik, zur Strukturierung von Daten, nutzen zu können. Hier sind Forschungsgebiete wie Schwarmintelligenz oder die von John von Neumann begründete Theorie der selbstreproduzierenden Automaten zu nennen.

Aktuell wird analysiert, ob sich die synergetischen Effekte<sup>1</sup> der Selbstorganisation auch auf atomarer Ebene, im Bereich der Quantencomputer nutzen lassen.

In dieser Arbeit befasse ich mich mit der Thematik, die Prozesse der Selbstorganisation zunächst in Logik-Netzen und später auf der Bit-Ebene von Computern gezielt zu steuern und zu optimieren. Die gebildeten Algorithmensysteme könnten zur Verarbeitung von Information verschiedener Art genutzt werden.

Zunächst analysiere ich die Faktoren, welche eine Selbstorganisation physikalisch und biologisch möglich machen. Ich interpretiere die gewonnenen Zusammenhänge mit dem Ansatz der Informationstheorie und übertrage sie dann auf informationsverarbeitende Systeme.

Auf Basis meiner Erkenntnisse entwerfe ich einen evolutionären Steuerungsalgorithmus (Genesis), der eine Population von Reaktionsräumen im Hinblick auf ihre Signalverarbeitungseigenschaften optimiert. Diesen teste ich an einem Simulations-Modell der Musterbildung bei Hautzellen.

Im Folgenden simuliere ich verschiedene Modelle der Selbstorganisation von Bits in Logiknetzen und mache mit dem Genesis-Algorithmus Experimente zur Steuerung dieser Prozesse. Darauf aufbauend entwickle ich ein System zur Züchtung selbstorganisierender Logikstrukturen.

Um Selbstorganisationseffekte auf der Atomebene der Informatik nutzen zu können, bilde ich Analogien zwischen Quantenelektrodynamik und Informationstheorie und konzipiere ein quantenmechanisches Modell der Wechselwirkung von Informationen.

Ich plane, diesen Ansatz zur Generierung selbstorganisierender Algorithmenstrukturen zu verwenden und dabei einem System nahe zu kommen, dass sich auf Quantencomputern verwenden lässt.

---

<sup>1</sup> **Synergetik** ist die Lehre vom Zusammenwirken von Elementen gleich welcher Art, die innerhalb eines komplexen dynamischen Systems miteinander in Wechselwirkung treten („Das Ganze ist mehr als die Summe seiner Teile“) Quelle: <http://www.scholarpedia.org/article/Synergetics>

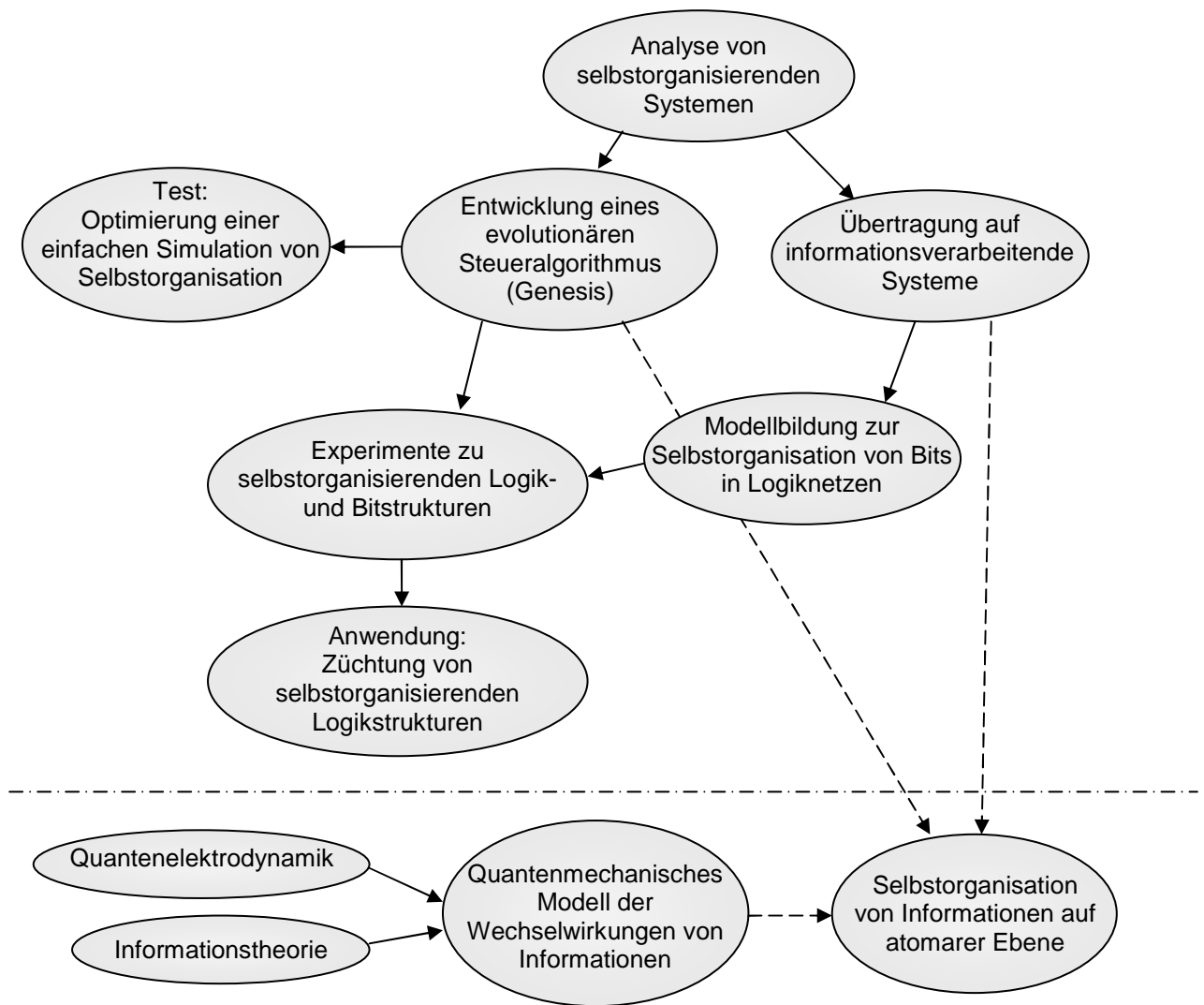


Abbildung 1: Vorgehensweise

### 3. Analyse von selbstorganisierenden Systemen

Um das allgemein anwendbare Konzept der Selbstorganisation für Fragestellungen in der Informatik nutzen zu können, musste ich zunächst die abstrakten Eigenschaften des Prozesses isolieren. Deshalb untersuchte ich biologische und physikalische Systeme auf Selbstorganisationsprozesse und bildete Analogien.

#### 3.1 Analyse einer Zelle

Die Stoffwechselprozesse in Zellen lassen sich durch Regelkreise modellieren, die über einen lokalen Reaktionsraum miteinander verbunden sind. Alle molekularen Strukturen ( $i$ ) in der Zelle haben Eigenschaften ( $\alpha \beta \gamma$ ), die sich auf die ablaufenden Reaktionen auswirken. Sie müssen in einer bestimmten Regulationsbeziehung zueinander stehen, um den Aufbau von komplexeren Molekülen oder die Erzeugung eines bestimmten Regulationsverhaltens ( $\alpha' \beta' \gamma'$ ) zu vollbringen. Diese Regulationsbeziehungen zwischen den verschiedenen Regelkreisen werden zentral durch die DNA gesteuert.

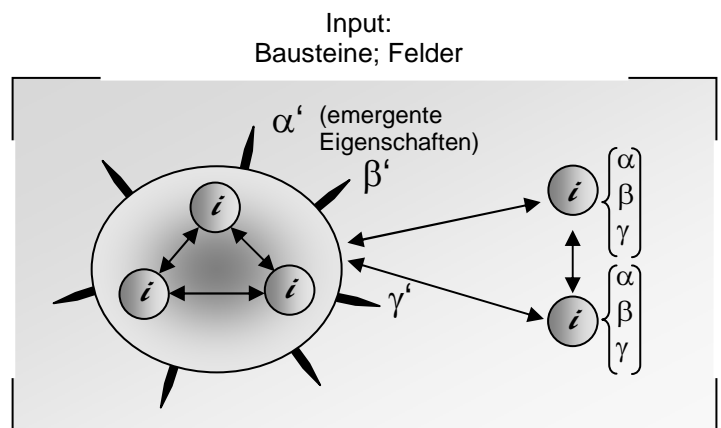


Abbildung 2: Abstrakte Darstellung der Überlagerung von Teilcheneigenschaften in einem Reaktionsraum

Auf diese Weise reagiert eine Zelle als offenes System regulativ auf Parameteränderungen in ihrer Umgebung, wie beispielsweise auf die Konzentrationsschwankung eines Stoffes oder auf ein elektromagnetisches Feld.

Im Folgenden werde ich mich auf verschiedene Systeme beziehen und dabei auf die jeweilige Definition des Informationsbegriffs und der Verarbeitung eingehen.

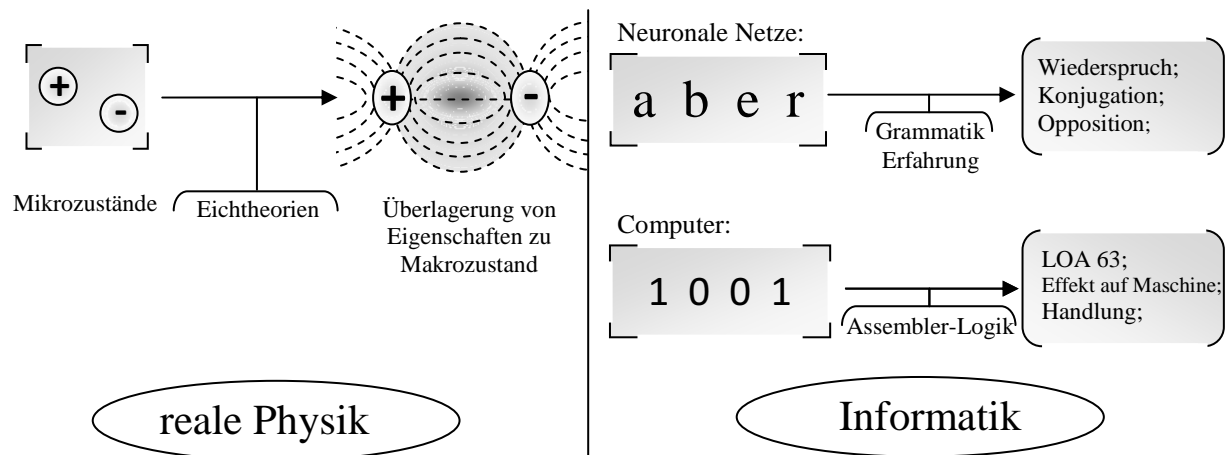


Abbildung 3: Veranschaulichung der Vermutung, dass verschiedene Regelsätze wie Physik, Grammatik oder Assemblerlogik, vergleichbare Wirkungen haben.

### 3.2 Analyse von Teilchensystemen

Da sich die Parameter des Systems in ihrer Wirkweise gegenseitig beeinflussen, liegen bei der Zelle vielfach rückgekoppelte, nichtlineare Prozesse vor. Aufgrund der Rückkopplungen können nichtlineare Systeme schon bei kleinen Änderungen der Startwerte ein völlig anderes Verhalten zeigen und weisen unter anderem chaotische Attraktoren auf. Dieses Phänomen tritt auch in komplexen Systemen auf, bei denen viele Teilchen in Wechselwirkung treten.

Im Standardmodell werden die grundlegenden Mechanismen der Wechselwirkung von Quantenobjekten durch sog. Eichtheorien, wie z.B. die Quantenelektrodynamik, beschrieben. Dabei wird angenommen, dass die Elemente eines Quantensystems Eigenschaften besitzen, die durch die Quantenzahlen charakterisiert werden können.

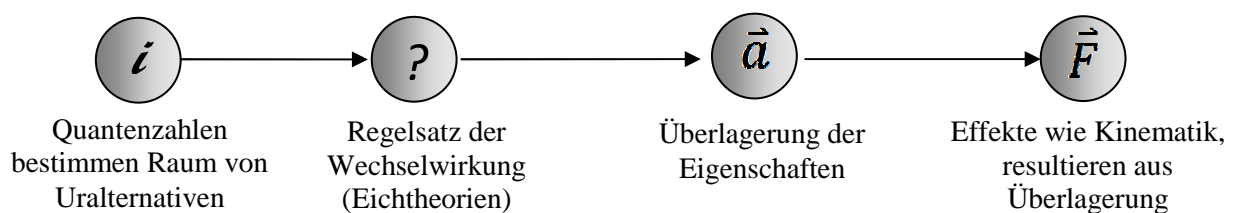


Abbildung 4: Abstraktes Schema des Standardmodells

Für ein abstraktes Modell der Selbstorganisation ist besonders der Ansatz des Physikers Carl Friedrich von Weizsäcker interessant, weil er die verschiedenen Arten der Information in Systemen beschreibt. Gemäß seinem „Postulat letzter Objekte“ besitzt jeder Mikrozustand eines Systems einen Raum von Uralternativen, dessen Dimensionen von der Anzahl der Variablen des Zustandes abhängt.<sup>4</sup> Jede Uralternative ist definiert als eine Entscheidung über eine Variable mit 2 Auswahlmöglichkeiten. Die potentielle Information eines Mikrozustandes ist gleichbedeutend mit der Anzahl von Alternativen, die durch die Existenz des Mikrozustandes entschieden werden. Sie kann daran veranschaulicht werden, wie viele Dimensionen der Alternativenraum besitzt.

Die existierende Entscheidungskonfiguration der Alternativen ist als Punkt im Raum der potenziellen Information zu verstehen und kann durch die Informationstheorie nach Shannon quantitativ beschrieben werden [9]. Sie wird strukturelle Information genannt, weil sie den Informationsgehalt des Zustandes auf materieller Grundlage kodiert.

Der Informationsgehalt sagt jedoch nichts über die Bedeutung und die Eigenschaften des Mikrozustandes in der Systemumgebung aus. Man interpretiert den Makrozustand eines Systems nicht als direkte Summe seiner Mikrozustände, sondern geht davon aus, dass diese übergeordnete Eigenschaft durch Emergenz<sup>II</sup> entsteht. Dabei nimmt man an, dass auch der Regelsatz der Wechselwirkung von der Umgebung abhängt, in welcher sich die wechselwirkenden Objekte befinden. Analog gelten auch die physikalischen Gesetze nur bei bestimmten Systembedingungen. Weizsäcker bezeichnete diesen erzeugten Regelsatz als funktionale Information.

Die Mikrozustände erlangen ihre Eigenschaften durch Wechselwirkung mit der Umgebung. Diese Eigenschaften bilden die pragmatische Information des Mikrozustandes. Ihr Einfluss ist mit der Definition eines physikalischen Feldes vergleichbar. Durch die Überlagerung der Eigenschaften von Mikrozuständen entstehen emergente Eigenschaften ( $\alpha'$   $\beta'$   $\gamma'$ ) auf der Makroebene des Systems. Das System ist „mehr als die Summe seiner Teile“.

Da die Mikrozustände selbst ihre Umgebung bestimmen, beeinflussen sie auch den Regelsatz und wirken somit auf sich selbst zurück. In rückgekoppelten Systemen entsteht so ein Phänomen, das man „Selbstreferenz“ nennt.

#### Zusammenfassung

Der Regelsatz, der durch die Umgebung gegeben wird, „decodiert“ die Eigenschaften eines Mikrozustandes im System aus dessen Raum von Uralternativen. Durch die Überlagerung von Wechselwirkungen mehrerer Mikrozustände entsteht eine Eigenschaft auf der Makro-Ebene des Systems, wie z.B. Kinematik.

### **3.3 Analyse von assoziativer Speicherung**

Die Wechselwirkung in physikalischen Systemen ist kontinuierlich, d.h. die Systeme reagieren auf jede strukturelle Information, wenn auch unterschiedlich ausgeprägt. Wenn das abstrakte Modell der Wechselwirkung von Informationen jedoch auch für diskrete Systeme wie Computer gelten soll, muss der bisher gebildete Ansatz erweitert werden. Als Bindeglied zwischen diskreter Informationsverarbeitung und kontinuierlicher physikalischer Wechselwirkung kann dabei ein künstliches neuronales Netz dienen. Dieses soll hier näher analysiert werden:

Ein künstliches neuronales Netz reagiert auf eine Folge von Symbolen. Es codiert die mit der Folge verbundenen Assoziationen jeweils in einem Muster von Signalen. Die zunächst bedeutungslosen Symbole sind die Mikrozustände des Systems. Der Informationsgehalt der Zeichenfolge ist die strukturelle Information. Die funktionale Information resultiert aus der Vernetzung der Neuronen und aller anderen Parameter, die das Assoziationsverhalten des Netzes bestimmen.

Nach der Informationstheorie hängt der Informationsgehalt einer Zeichenfolge von dem Alphabet und dem Aufbau der Quelle ab. Die Bedeutung der Symbolfolge wird jedoch erst durch das Netz selbst bestimmt, das die eingespeiste Folge dekodiert und jedem Symbol einen Raum von Assoziationen zuweist. Diese stellen die Eigenschaften der Symbole im System da (pragmatische Information). Im neuronalen Netz können sich die Assoziationsräume der Symbole überlagern, da sie jeweils in einem Signalmuster kodiert sind. Aus der Überlagerung entsteht schließlich die übergeordnete Bedeutung der Folge, also ein Verhalten des Netzes auf der Makroebene. Die Folge wird interpretiert.

### **3.4 Informationstheoretische Deutung [9]**

Wenn der Selbstorganisationsprozess auf einem informationsverarbeitenden System ablaufen soll, müssen die Wechselwirkungen, also die Umwandlungen in Kinematik, in Form von Befehlen resultieren. Die Prozesse dürfen der Thermodynamik nicht widersprechen, wenn sie real angewendet und mit der Informationstheorie quantitativ beschrieben werden sollen. An dieser Stelle sollen die gesammelten Erkenntnisse zusammengefasst und ein qualitatives Modell der Wechselwirkung von Informationen generiert werden.

#### Potenzielle Information

Die „n“ Variablen eines Zustandes spannen einen Alternativenraum mit „n“ Dimensionen auf. Jede Uralternative ist definiert als eine Entscheidung zwischen 2 Auswahlmöglichkeiten.

#### Strukturelle Information

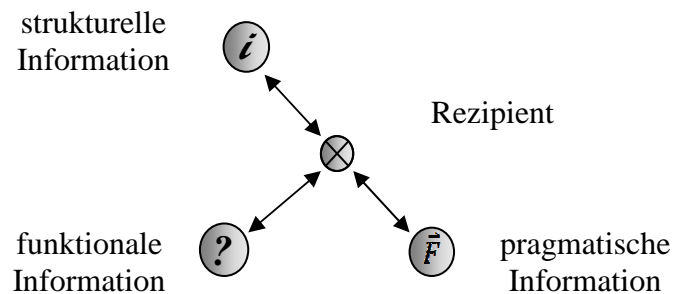
Nach Weizsäcker besitzt jeder Punkt des Alternativenraums durch seine konkrete Konstellation von Uralternativen eine strukturelle Information. Diese erfasst den Informationsgehalt eines Mikrozustandes wie er auf materieller Grundlage kodiert ist[9]. Nach Shannon besitzt somit jede physikalische Struktur, Verteilung oder Folge von Zuständen einen Informationsgehalt, der sich quantitativ über die Entropie messen lässt.

---

<sup>II</sup> **Emergenz** (lat. *emergere*: auftauchen) ist die spontane Herausbildung von Phänomenen oder Strukturen auf der Makroebene eines Systems durch das Zusammenspiel seiner Elemente

Funktionale Information

Aus der Systemumgebung eines Mikrozustands/Objektes resultiert ein Regelsatz, der die ablaufenden Wechselwirkungsprozesse bestimmt. Diese funktionale Information besitzt die Fähigkeit, strukturelle Information in pragmatische Information zu dekodieren. Der Rezipient stellt im Modell das Trägermedium der funktionalen Information der Umgebung da. Wie ich später feststellte, kann die funktionale Information durch Wechselwirkung verändert werden, weil sie auf materieller Basis im Rezipienten kodiert ist. Analog können sich die Verarbeitungseigenschaften eines neuronalen Netzwerks ändern, wenn dieses durch Aufnahme von Informationen umstrukturiert wird.



**Abbildung 5: Wechselwirkung der Informationsarten in selbstorganisierenden Systemen**

Pragmatische Information

Die pragmatische Information ist die Summe der entstandenen Eigenschaften eines Mikrozustandes. Diese können nur durch Wechselwirkung mit der Umgebung entstehen und werden somit durch die funktionale Information bestimmt. Durch die Wechselwirkung ist eine Überlagerung der Eigenschaften mehrerer Mikrozustände möglich. Das System wird verändert, d.h. es werden im abstrakten Sinne Variablen anderer Mikrozustände gesetzt. Dies ist analog zur physikalischen Kinematik zu verstehen. Auf der Makroebene des Systems entstehen somit übergeordnete Eigenschaften, welche die Selbstorganisation charakterisieren.

Selbstreferenz

Die entstandenen Eigenschaften einer Zustandsgruppe bestimmen, welcher Auswertungsprozess im verarbeitenden System abläuft, auch wenn dieses System simplen Regeln wie ein Computer folgt. Somit definieren die Symbole, in gewissen Grenzen, selbst ihre eigene regulative Wirkung in einer verarbeitenden Struktur.

Felder

Gemäß der Analogie sind Felder im Informationssinn statistische Verteilungen der Einflüsse von Symbolen in einer Systemumgebung. Sie sind Ausdruck der Eigenschaften der Mikrozustände und machen Rückkopplung und Überlagerung erst möglich.

**→Postulat**

In jeder Menge von Zuständen, die über das oben beschriebene Modell synergetisch dekodiert, also in emergente Eigenschaften umgesetzt wird, findet ein Selbstorganisationsprozess statt. Dabei bestimmen die Umgebungsbedingungen und der Ausgangszustand die spätere Organisation.

### 3.5 Übertragung auf informationsverarbeitende Systeme

**Analogien:**

- Strukturelle Information = Speichergruppen
- Funktionale Information = informationsverarbeitendes System
- Pragmatische Information = Algorithmus (Reaktion durch Befehle)
  
- Mikrozustände → Variablenmenge/Speichermenge
- Koordinate im Raum → Koordinate in informationsverarbeitender Systemstruktur
- Überlagerung der Felder von Teilchen → statistische Verteilung der Einflüsse von Nachrichten in Systemmedium

Wenn sich ein durch Selbstorganisation entstandenes System spontan an veränderte Umgebungsparameter anpassen kann, ist es autonom. Sein teils komplexes Verhalten resultiert dann aus der Organisation einfacher Bestandteile, was wenig Fremdeinwirkung erfordert. Die emergenten Eigenschaften des Systems ließen sich zur Strukturierung von Symbolgruppen verwenden. In einer konkreten Anwendung können das Speichergruppen wie Bits sein. Über den Regelsatz des verarbeitenden Systems können die entstandenen Strukturen andere Zustände im System verändern. Somit haben sie eine algorithmische Eigenschaft und können als durch Selbstorganisation gebildete Programme angesehen werden.



## 4. Der Genesis-Algorithmus

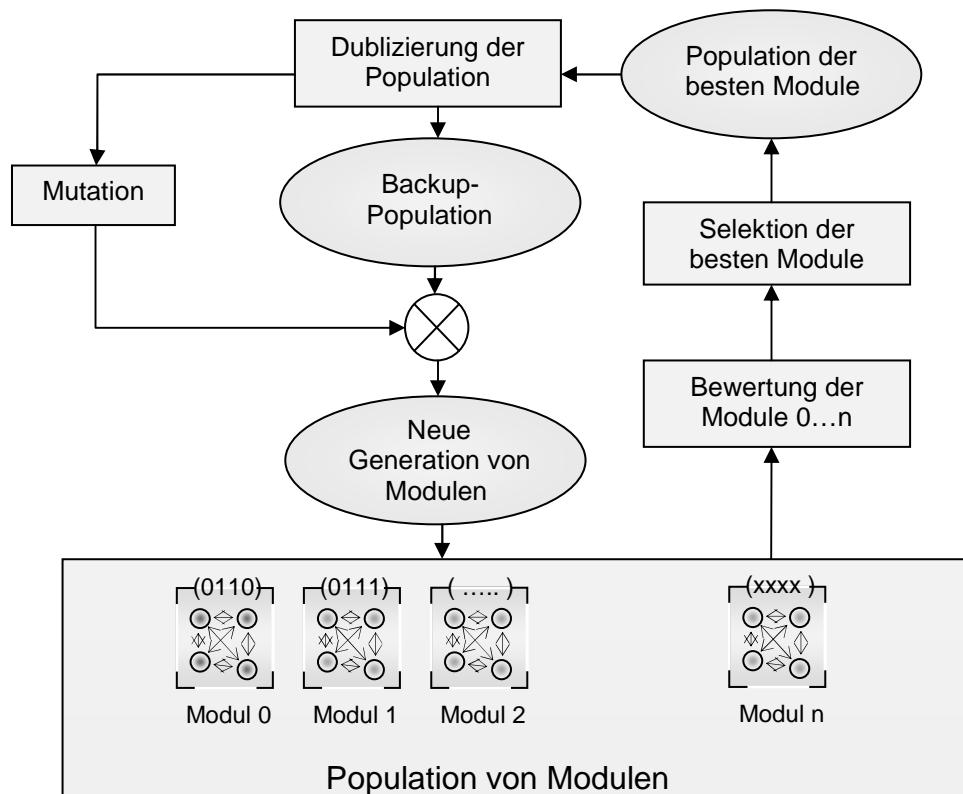


Abbildung 6: Schema des Genesis-Algorithmus

Im Rahmen meines Postulats sollte sich in einem Reaktionsraum, der den hergeleiteten Anforderungen genügt, eine aus den Systembausteinen und dem geltenden Regelsatz resultierende Struktur entwickeln, die an die Umgebung und das Eingangssignal „angepasst“ ist („angepasst“ bedeutet in diesem Sinne „aus ihr resultierend“). Der gemessene Systemabdruck ist dabei erheblich von der Position der Messfühler und der Sender im Reaktionsraum abhängig.

Damit eine Organisation entstehen kann, die das Eingangssignal wie gewünscht umwandelt, müssen sehr viele Systemparameter einen jeweils extrem unwahrscheinlichen Zustand einnehmen bzw. so gesetzt werden. Durch die Fülle von Möglichkeiten ist es zwar möglich, Organisationen zu beobachten, diese aber nicht zu „richten“, um eine bestimmte Dekodierung durch den Reaktionsraum mit seinem Satz von Regeln zu erhalten.

### 4.1 Abstraktion

Hier fällt auf, dass die erfolgreiche Entwicklung eines Steuerungsalgorithmus für eine Selbstorganisation zwei nützliche Erkenntnisse mit sich bringt.

*Der Algorithmus kann sowohl eine angepasste Struktur im Reaktionsraum liefern, als auch das „Rezept“, also eine Liste von Instruktionen, mit der temporär eine Selbstorganisation mit gewünschten Eigenschaften entsteht.*

Seit etwa 60 Jahren versucht man in einem Bereich der Biologie, wo man sich mit der Bildung von organischen Strukturen aus anorganischer Materie beschäftigt, das Prinzip der Selbstorganisation komplexer Strukturen zu verstehen. Durch das Miller-Experiment wurde der Ansatz der Forschung verifiziert:<sup>5</sup> Vor etwa 4 Mrd. Jahren existierte auf der Erde eine Vielzahl von Reaktionsräumen mit unterschiedlichen Kombinationen enthaltener Moleküle und Umgebungsbedingungen. In dieser Kombinationsvielfalt von physikalischen Effekten, einwirkenden elektromagnetischen Feldern usw. entstanden erste geordnete Strukturen wie Kristalle und letztendlich die organischen Grundbausteine des Lebens durch Selbstorganisation.

Das Modell habe ich abstrahiert und daraus den GENESIS-Algorithmus entworfen. Diesen habe ich so benannt, weil er die Bedingungen in der Frühzeit der Erde abstrakt verwendet um eine möglichst potente Umgebung für Selbstorganisationsprozesse zu schaffen:

Der Genesis-Algorithmus kann innerhalb eines vorgegebenen Baukastens Bausteine jeglicher Art vermischen und aus demselben Baukasten eine Umgebung simulieren. Diese enthält die Änderung von Feldern, Zu- und

Abflüssen von Bausteinen oder Energie-Einwirkungen über die Zeit. Bausteine und Felder können jeglicher Art sein und müssen keinen Bezug zu realen Teilchen oder physikalischen Feldern haben. (Mit einem Feld wird hier eine Verteilung der Werte einer betrachteten physikalischen oder abstrakten Variablen über die Ebene oder das Volumen des Reaktionsraumes bezeichnet. Diese kann homogen oder inhomogen sein.)

## 4.2 Ansatz

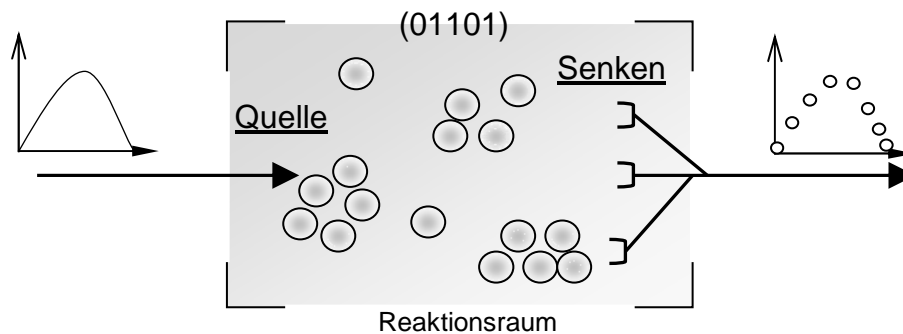
Ich habe einen evolutionären Algorithmus verwendet um die Organisationsparameter zu optimieren. Wie ich bereits in meiner letzten Arbeit<sup>6</sup> gezeigt habe sind diese geeignet, da sie eine kontinuierliche Optimierung der Kombination bei großer Parameterzahl und nichtlinearer Dynamik ermöglichen. Dabei wird eine Evolution zwischen den Kombinationen simuliert. Anstatt alle Möglichkeiten durchzugehen, werden Pfade der Verbesserung eingeschlagen und so die Auswahl sinnvoll reduziert. In der Vergangenheit wurden verschiedene Strategien für EAs entwickelt<sup>4</sup>. Bei meinem letzten Projekt habe ich einen eigenen EA entworfen und mit Simulationen getestet. Für meine aktuelle Arbeit änderte ich weite Bereiche der Software um sie möglichst vielfältig nutzbar zu machen und verschiedene Organisationsformen simulieren zu können. In den Modulen laufen 2 Phasen ab:

### Züchtungsphase

Zunächst wird aus einem Parameter-Baukasten, eine Startpopulation von Reaktionsräumen mit jeweils eigener Kombination generiert. Die Parameter stecken die Grenzen der Variabilität ab. Je größer die Population, desto schneller wird eine optimale Kombination von Eigenschaften gefunden. Der Reaktionsraum, in dem die Organisation abläuft, wird zusammen mit der Parameterkombination unter einem Modul zusammengefasst. Jedes Modul besitzt ein Erbgut, das hinzugegebene Bausteine und Felder jeglicher Art zeit- und ortsdiskret deklariert. Das Erbgut ist die in einem Rastercode kodierte Variante des „Rezeptes“ und lenkt die Selbstorganisation.

In dem jeweiligen Reaktionsraum läuft die Organisation von Komponenten nach einem festen Regelsatz ab. Dieser Regelsatz sorgt für die nötige Kinematik, aber auch für die Kodierung und Dekodierung, also für die Bedeutung der Informationen der Bausteine im Systemkontext. Alle Reaktionsräume unterliegen demselben Regelsatz, damit auch die Signalquelle und -Senke jeweils auf die gleiche Art ins System codieren bzw. bei der Messung decodieren. Ansonsten wären die Ergebnisse nicht vergleichbar.

### Testphase



**Abbildung 7:** Zur Abtastung der organisierten Struktur werden Quellen und Senken in den Reaktionsraum eingebracht und die Reaktion des Systems auf das Eingangssignal aufgezeichnet.

Zur Messung des Selektionskriteriums wird der Reaktionsraum getestet. Dafür werden Signalquellen und -Senken (Testsignale + Abtastpunkte) in das System eingebracht. Die bekannten Informationen über das gewünschte Systemverhalten werden mit den Messungen aus dem Reaktionsraum verglichen. Die Messungen erfolgen nicht kontinuierlich, sondern abtastend (diskret). Das Verhalten des Systems zwischen den Abtastpunkten wird nicht beachtet.

Einerseits benötigt man nach Shannon<sup>7</sup> eine minimale Anzahl von Messungen. Andererseits steigt der Selektionsdruck mit zunehmender Anzahl von Messungen, da die Wahrscheinlichkeit, alle Kriterien zu erfüllen, sehr klein wird. Es muss also ein Mittelweg zwischen Toleranz und Selektionsdruck gefunden werden. Aus dem Grad der Signal-Übereinstimmung, aber auch aus Parametern wie der Komplexität der nötigen Umwelt, oder der Züchtzeit bis zum Test durch das Eingangssignal wird eine Gesamtfitness des Moduls berechnet. In der Selektion wird nur eine bestimmte Anzahl der besten Module erhalten und geklont.

**Mutation + Zusammenfügen**

Der eine Satz von Kombinationen wird mutiert und der andere als Backup genutzt, was zusammen eine kontinuierliche Verbesserung der Organisation ermöglicht. Das Rezept wird in Rastercodeform durch Punktmutationen (Deletion, Addition, Substitution) verändert. Physikalische Konstanten sowie die Parameter des Signals und die einer nötigen, festen Umgebung, werden nicht mutiert.

Schließlich werden die mutierten Module und die Backup-Menge wieder zusammengefügt und bilden zusammen den Kombinationspool der nächsten Generation. Danach wiederholt sich der Prozess.

**4.3 Experiment: Fur-Optimierung**

Bevor ich eine Selbstorganisation von Informationen optimieren konnte, musste ich prüfen, ob der Algorithmus überhaupt zur Optimierung einer Selbstorganisation geeignet ist. Ich programmierte zunächst ein simples adaptives System in „Netlogo“<sup>8</sup>. In der Simulationsbibliothek der Plattform fand ich ein Beispiel der Musterbildung bei Zebras, die durch die Interaktion und Abstimmung der Pigmentzellen zustande kommt. Auf diese Selbstorganisation wendete ich meinen Optimierungsalgorithmus an:

Mein Ziel war es, eine Struktur zu erzeugen, bei der sich möglichst große Flächen homogener Färbung bilden.

Der Algorithmus erkannte relativ schnell, dass die meisten Parameter der Simulation zur Optimierung auf dieses Ziel hin irrelevant waren und ermittelte eine „initial\_density“ von 21 als ausschlaggebenden Faktor der Organisation. Das Bild wurde entweder komplett weiß, oder es zeigten sich runde Flecken mit der größt möglichen Anzahl von Zellen gleicher Färbung bei einem bestimmten Durchmesser.

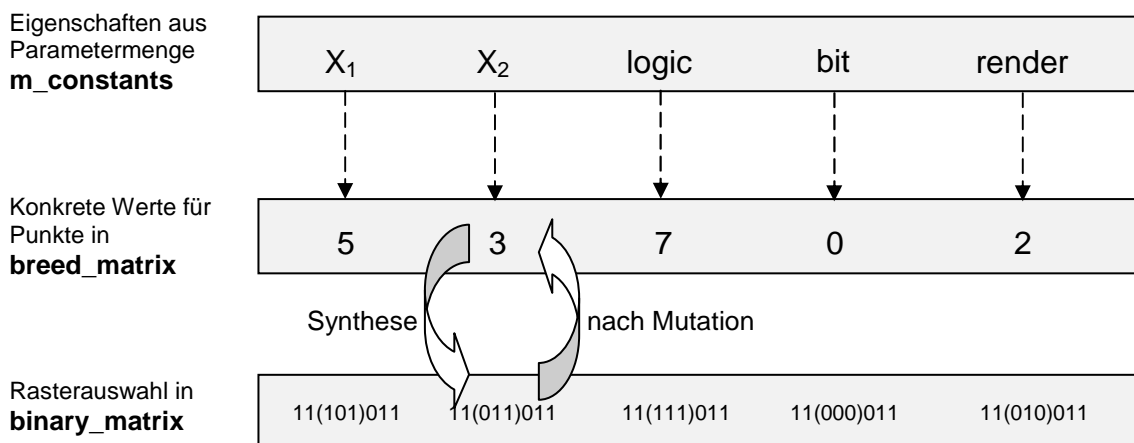
*Im Anhang befindet sich ein Screenshot der Oberfläche bei laufender Optimierung.*

**4.4 Erläuterung der Programmierung**

Im weiteren Verlauf meiner Arbeit benötigte ich eine weit potentere Organisationsprogramm als die Fur-Simulation. Meine Analyse der Selbstorganisation von Informationen erforderte den weiteren Umbau in das, was ich schlussendlich den „Genesis-Algorithmus“ nannte, da dieser jegliche Art von Parametern kombiniert. Das System behandelt drei 3 Parametermengen, die für die Organisation im Reaktionsraum benötigt werden:

- 1 Satz von Konstanten, festen Strukturen und Einstellungen der Simulation („constants“)
- 1 Auswahl von kombinierbaren Bausteinen und Umgebungsparametern (m\_constants)
- 1 Menge von Signalvektoren, bestehend aus temporären Quellen und Senken (Signal)

Jede Parametermenge der Konstanten, der Baukastenkombination und des Signals ist jeweils durch einen Vektor definiert, der ihre raumzeitlichen Koordinaten und ihre Referenzvariablen in der Simulation enthält.



**Abbildung 8: Grobe Veranschaulichung der Codehierarchie**

Jeder Vektor der Baukastenkombination eines Moduls wird vom System in 3 hierarchisch geordneten Codeschienen abgelegt. Diese ähneln sich in ihrer Syntax und werden genutzt, um bei der Variabelnsynthese und der Mutation trotz der Einhaltung eines Toleranzrahmens möglichst viele Mutationsarten zu ermöglichen. Da alle Module ihre Variablen einspeisen, sind die Schienen auch nach Modulen sortiert. Im jeweiligen Paket werden die Instruktionen zeitspezifisch gespeichert. Dieses Rezept lässt sich dann in die einzelnen Parameteränderungen auffächern.

Die Parameter der Bausteine werden zunächst in ihrer Bedeutung definiert. Dazu wird eine Schiene der Zeitpakete aller Module angelegt, die die Auswahlmöglichkeiten enthält („breed\_time\_packets“). Durch eine Reihe von Prozeduren wird eine Kombination ausgewählter Werte dezimal in der Schiene „breed\_matrix“ abgelegt. Diese wird nach jedem Zeitintervall abgelesen und gibt die Werteänderungen direkt wieder. Die dezimalen Werte der „breed\_matrix“ werden zunächst binär in der sog. „binary\_matrix“ abgelegt. Bei der Mutation ist es sinnvoll, einen Rastercode zu verwenden, weil nur ein Teilbereich des Rastercodes für den Wert eines Parameters genutzt wird und sich im Rest des Codes stumme Mutationen anlagern können. Erst dieser Mechanismus kann zu Komplexitätssprüngen führen.

### Render-system

Das System arbeitet mit verschiedenen „Render“-Typen und behandelt die Feldpunkte typspezifisch. Dies ermöglichen eine leichte Einführung von neuen Organisationsprogrammen und eine sichere Simulationsumgebung. Typen sind z.B. „Quelle“, „Senke“, „Feld“, „nicht rendern“

### Abtastsystem

Es muss möglich sein, den Reaktionsraum an beliebigen Raum- und Zeit-Koordinaten zu untersuchen. Außerdem müssen beliebig viele Signalquellen und Senken definiert werden können. Die leicht überschaubare Parametermenge wird, wie zuvor die anderen Parametermengen, in Zeitpakete geordnet („signal\_time\_packets“) und in eine „Signal\_matrix“ übersetzt. Auf diese wird nach jedem Iterationsschritt zugegriffen um nötige Änderungen an den Parametern der Quellen und Senken vorzunehmen. Wenn ein Punkt im System als Quelle bzw. Senke behandelt werden soll, wird seine entsprechende Render-Variable darauf ausgerichtet. Alle Signalquellen und Senken werden im Regelfall nach einem Iterationsschritt wieder entfernt. Manchmal erfordert die Komplexität eines Problems eine lange „Züchtzeit“. Dieser Parameter muss Modul-spezifisch sein und wird deshalb als „X“ in der Kombinationsmenge des Moduls angegeben.

## 5. Selbstorganisation von Bits in Logikstrukturen

### 5.1 Modell-Ansatz

Wenn man versucht, die Selbstorganisation zu lenken, will man diese im Prinzip Systeme bilden lassen, die bestimmte physikalische oder logische Eigenschaften haben, die also ein Signal in bestimmter Weise verarbeiten. Tut man dies auf der Ebene der Informatik, sucht man nach Algorithmen oder Logikstrukturen, die bestimmte Eingangssignale an der Quelle in bestimmte Ausgangssignale an der Senke umsetzen. Alle Regeln, nach denen sich die Selbstorganisation im Reaktionsraum entwickelt, müssen systemextern prozessiert werden. Diese Rechenleistung geht also ohnehin bei der Organisation verloren. Da die Selbstorganisation jedoch aus einfachen Regeln und einfachen Bausteinen komplexes Verhalten generieren kann, ist dieser Prozess optimal geeignet, solange viele Agenten parallel angesteuert werden können. *Wie bereits im ersten Abschnitt erwähnt, liegen die Stärken dieses Systems eher in der Lösung nichtlinearer und komplexer Probleme.*

Wenn man z.B. ein Logik-System benötigt, das aus einem Signal von Eingangsbits ein spezielles Signal von Ausgangsbits erzeugt, kann man dieses analytisch errechnen (Boolesche Algebra). Man kann aber auch, wie im folgenden gezeigt, einen evolutionären Algorithmus mit einer Selbstorganisation kombinieren:

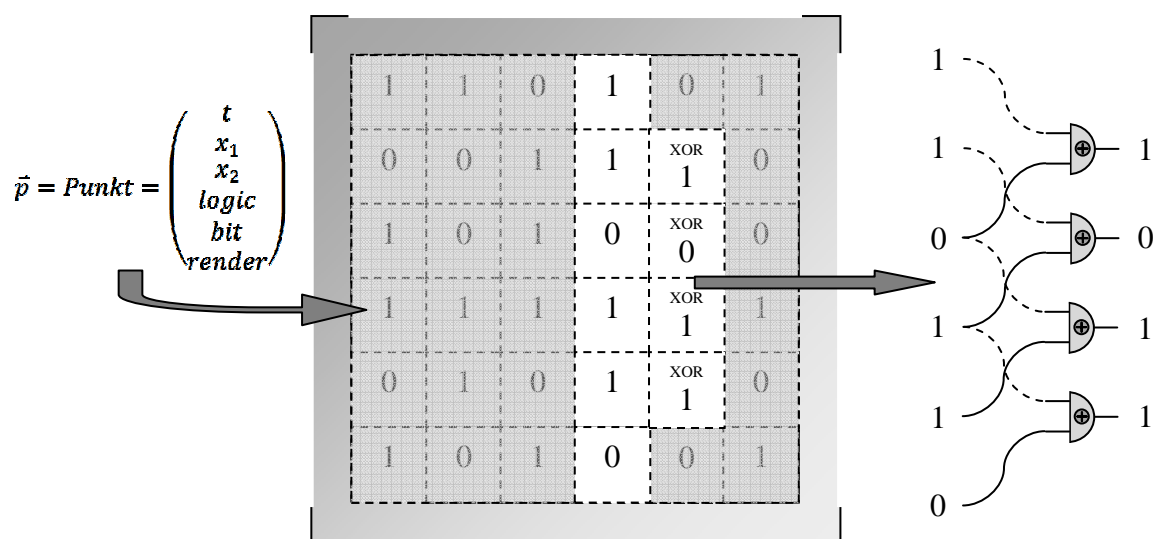


Abbildung 9: Schematisches Logik-Feld im Reaktionsraum. Im Anhang ist ein entsprechender Screenshot zu finden

Gemäß der Analogie sind Felder im Informationssinn statistische Verteilungen der Einflüsse von Symbolen in einer Systemumgebung.

Im hier behandelten Modell wird eine Verteilung von Zuständen in einer Logikstruktur zur Organisation gebracht. Die Verteilung ist die strukturelle Information. Sie wird durch die funktionale Information in den Rezipienten (den Logikgattern) strukturiert. Jeder Punkt der Ebene besitzt eine festgelegte Funktion aus der Booleschen Logik. Das Organisationsverhalten hängt von der Wirkweise der Gatterpunkte ab. Aus Gründen der Einfachheit setzte ich zunächst fest, dass ein Rasterpunkt des Feldes gleichzeitig Ausgang und Träger eines Gatters ist, dessen Eingänge links darüber und links darunter liegen. Da das Logiknetz konstant bleibt, bestimmt die gebildete Verteilung von Ausgangszuständen ein logisches Reaktionsverhalten des Gesamtsystems. Dieses ist temporär und nur in einem Zeitfenster vorhanden. In der Simulation steht Weiß für 0 und Schwarz für 1.

Bei allen Anwendungen ist zu beachten, dass das evolutionär entwickelte System das gewünschte Reaktionsverhalten nur im vorgegebenen Zeitpunkt aufweisen muss. Dabei regeneriert es seine Verteilung und schwingt nicht auf das Signal, sondern auf seine Startparameter ein. Die Reaktionen auf zeitlich gestaffelte Eingangskombinationen können sich in y-Richtung überlagern und sind nicht unabhängig, wie bei einer normalen Logikfunktion. Dies ist sinnvoll, weil bei dem angedachten Modell eines selbstorganisierenden Computers eine Art gegenseitiger Regulation von Modulen möglich sein muss.

## 5.2 Strukturbildung

Zunächst analysierte ich den Entstehungsprozess von Strukturen bei unterschiedlicher Ebenengröße, Bitverteilung und Logikumgebung. Da in der Realität die Startzustände eines Gatters unbestimmt sind, habe ich zunächst eine Zufallsverteilung von Bitwerten über die Ebene gelegt. Die Musterbildung verläuft dann stets von links nach rechts, was auf die Ausrichtung der Gatter zurückzuführen ist.

Die entstehenden Muster waren vor allem von der Art und Verteilung der Logik-Gatter abhängig. Bei einem homogenen „Xor“-System bildeten sich beständige, pyramidenartige Muster, die in ihrer Größe variierten. Dabei wiederholten sich die Mustergruppen und ihre Verknüpfungen in einem bestimmten Schema. *Im Anhang ist eine weitere Musterbildung bei einem Inhomogenen Logikfeld dokumentiert*

Im Folgenden testete ich den Einfluss von homogenen Systemen aus „Or“-Gattern“ bzw. „And“-Gattern“. Es bildeten sich kristallartige Strukturen an den Rändern, da diese aufgrund einer Begrenzung konstant geblieben waren. Abseits dieser Strukturen hatten die Gatterausgänge jedoch meist den gleichen Wert und noch vorhandene Inselstrukturen lösten sich schnell auf. Bei einer homogenen „And“-Logik wurden die Ausgänge dieser Bereiche 0, bei einer homogenen Or-Logik 1.

Die unterschiedlichen gebildeten Strukturen lassen sich durch eine Analyse der Ausgabewahrscheinlichkeiten erklären: Jedes Gatter besitzt zwei Eingänge, von denen jeder den Zustand 1 oder 0 annehmen kann. Bei einer Zufallsverteilung ist die Wahrscheinlichkeit, dass ein Bit den Wert 0 oder 1 annimmt 0,5 und die einer Kombination in der Wahrheitstabelle jeweils 0,25. Bei „And“-Gattern führen alle Kombinationen außer (1|1) zur Ausgabe einer 0.  $P(X=0) = 0,75$ . Statistisch werden somit die Bits mit dem Wert 1 benachteiligt und verschwinden auf Dauer. Beim homogenen „OR“-Netz ist dagegen  $P(X=1) = 0,75$ , da alle Kombinationen außer (0|0) zur Ausgabe einer 1 führen. Somit verschwinden die Bits mit dem Wert 0

→ Diese erste Analyse ergab, dass bestimmte Logiksysteme Zustandsattraktoren anstreben, die von den Ausgangsparametern des Selbstorganisationsprozesses abhängen.



Abbildung 10: Homogenes Xor-Logik-System nach der Organisation

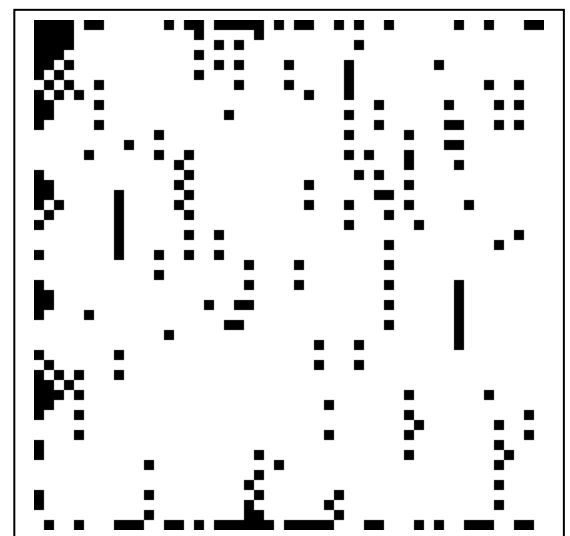
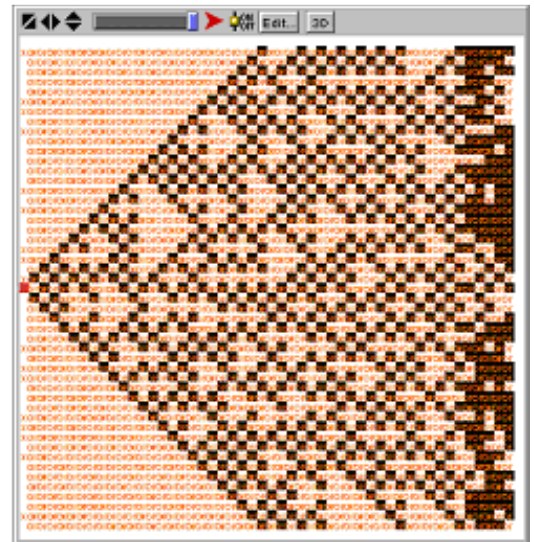


Abbildung 11: Struktur- und Inselbildung bei einer homogenen And-Logik

Quelleneinfluss:

Als ich die Reaktionsräume in den EA einbaute, konnte ich durch Zufall beobachten, wie sich, ausgehend von meiner Signalquelle, ein symmetrisch erscheinendes, konstantes Muster bildete. Der Algorithmus hatte ein inhomogenes Logik-System erzeugt und die Signalquelle nach dem Iterationsschritt nicht gelöscht.

Dies ist, nach meiner Argumentation zur Selbstorganisation, mit einem kontinuierlichen, gleichbleibenden Signal- oder Energieinput auf ein materiell geschlossenes System mit definierter Umgebungsanordnung zu vergleichen. Es könnte durch Zufall ein bildlicher Ausdruck der Einflüsse einer Quelle in diesem System sein. In meiner vorigen Argumentation zum Aufbau einer Informationsphysik hatte ich angenommen, dass sich solche Feldeffekte in informationsverarbeitenden Systemen bilden, nun wurden sie bestätigt. Das System reagierte mit Umstrukturierung auf geänderte Signale der Quelle und regenerierte sich bei einer neuerlichen Induzierung der ursprünglichen Folge.



**Abbildung 12: Auswirkung des konstanten Wertes einer Quelle in einer inhomogenen Logik-Umgebung (orange)**

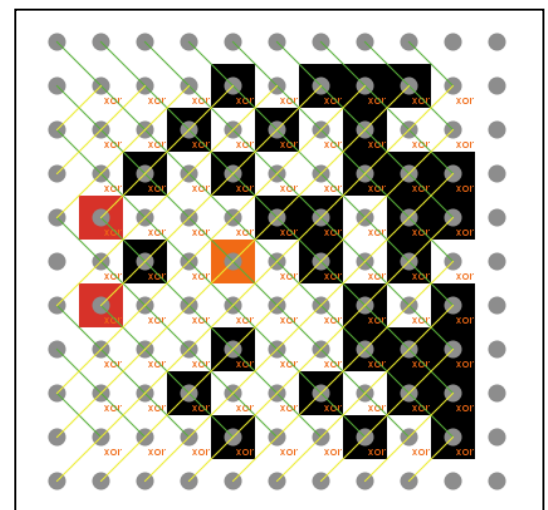
### 5.3 Optimierung einer Kaskade

Die Bildung eines Attraktors und des Feldeffektes bestätigten mein Modell der Wechselwirkung von Informationen. Die Attraktor-Konfiguration der Ausgänge kann, ähnlich der Dotierung von Halbleitern durch den konstanten Rand auf der linken Seite und durch zusätzlich eingebrachte konstante Werte gesteuert werden. Somit ist eine Anwendung des evolutionären Algorithmus möglich.

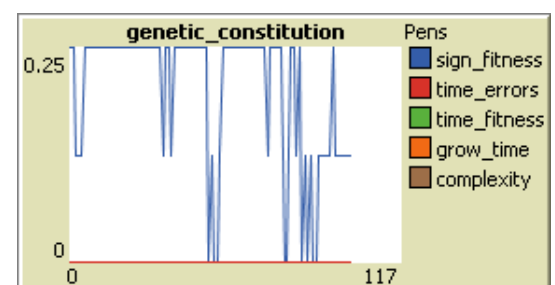
Ich platzierte Signalquellen(rot) und -Senken(orange) an verschiedenen Punkten der Ebene und gab dem evolutionären Algorithmus zunächst das Ziel einer „AND“-Funktion vor. Quellen und Senken werden zusätzlich zeitversetzt abgetastet, um die Gatterlaufzeit zu beachten, die für den Übertragungsweg mindestens benötigt wird. Die aufgezeichnete Wertefolge an der Senke wird mit den gewünschten zeit- und ortsspezifischen Logik-Werten verglichen und der Grad der Übereinstimmung bestimmt. Dieser geht in die Kalkulation der Fitness des Moduls ein.

Bei jeder Signalinduktion wurde im System eine Kaskade ausgelöst, die durch vorpositionierte Bitwerte beeinflusst werden konnte. Es ergaben sich stark schwankende Fitness- und Komplexitätskurven, die ich aufgrund der Backup-Funktion des EA zunächst nicht erwartet hatte. Die Analyse mehrerer Prozesse ergab außerdem, dass zwischen Signalfitness und Komplexität eine Koevolution wirkte.

Als Ursache der Schwankungen konnte ich die zufällige Startverteilung der Bits identifizieren. Weitere Tests mit festgelegten Startparametern blieben jedoch erfolglos, weil es im Rahmen der entstandenen Kaskade nicht möglich war, das von mir gewünschte Verhalten zu erzeugen. Außerdem zeigte sich, dass durch die homogene Ausrichtung und den relativ kleinen Bezugsraum der Gatter keine Rückkopplungen möglich waren. Wenn auch dieses Logiksystem zunächst meine Theorie bestätigte, fehlte ihm damit die Eigenschaft, die in selbstorganisierenden Systemen Urheber der Nichtlinearität ist.



**Abbildung 13: Quellen und Senken in einem homogenen XOR-Feld**



**Abbildung 14: Chaotische Schwankung der Signalfitness durch zufällige Startparameter**

## 5.4 Optimierung einer freien Vernetzung

Um Rückkopplungen und nichtlineare Prozesse wirken zu lassen, fügte ich einen Parametercode in den Bausteinpool des EA ein, der eine freie Auswahl der Eingangspunkte der Gatter in der Ebene ermöglicht. Somit wurde im Code jeden Moduls eine zunächst zufällige Variante einer freien Vernetzung gespeichert. Über den Parametercode konnte ich jeweils den Wertebereich für die Mutation definieren.

Die Analyse der Organisation im Modul bestätigte meine Vermutungen: Neben auskristallisierten Bereichen bildeten sich zyklische Strukturen, also rückgekoppelte Systeme. Darin waren nach einer Startphase periodische, aber auch chaotische Grenzzyklen zu beobachten. Das anfänglich stark wechselnde Bitmuster näherte sich einer Grenzverteilung oder, bei starker Vernetzung, einem Grenzzyklus.

Als ich diesmal versuchte, die Selbstorganisation mit dem evolutionären Algorithmus zu optimieren, ergaben sich zunächst konstante Fitness- und Komplexitätskurven. Dies führte ich auf die Einstellung der Mutationskoeffizienten zurück:

Da das Erbgut aus ineinander verschachtelten Listen aufgebaut ist, nimmt mit zunehmender Listentiefe die Wahrscheinlichkeit für eine Mutation ab. Um dies zu korrigieren analysierte ich zunächst die Wahrscheinlichkeitsverteilung bei verschiedenen Codes und Mutationseinstellungen. Danach entwarf ich eine komplementäre Verteilung, die es ermöglichte, die Wahrscheinlichkeit der Mutationssorten und -Tiefen zentral zu steuern. Ich ermittelte ein Optimum der Anpassung bei 10% neu hinzukommendem Listen-Erbgut, 20% Listenentfernung und 70% Punktmutationen im Binärastercode

Trotz der Rekalibrierung konnte der Algorithmus weiterhin keine Fitnessoptimierung durchführen. Jedoch gelang eine Anpassung der nötigen Komplexität. Um den problematischen Faktor zu isolieren, gab ich dem EA die Aufgabe, einen einzelnen Punkt zu treffen, der die Senke darstellte. Die Auswertung zeigte, dass das Ziel zwar erreicht wurde, jedoch vorher lange Phasen ohne Fitnessverbesserung zu beobachten waren.

### Erklärung:

Im EA bestimmen Selektion und Mutation den Grat der Anpassung. Im Fall der Punktoptimierung konnte der EA die Module nicht richtig selektieren, da nur ein einziger Wert richtig war. Die Wahrscheinlichkeit, mit einem Punkt die Quelle zu treffen, ist antiproportional zur Fläche der Ebene. Wenn der Parametercode noch weitere Variablen besitzt, wie z.B. zeitliche Ausdehnung oder eine Einflussart, benötigt der EA genügend Resonanz vom System um die richtige Lösung in diesem mehrdimensionalen Raum zu ermitteln. Diese Resonanz war beim Punktbeispiel nicht genügend gegeben, weil nur zwischen wenigen Punkten eine Wechselwirkung bestand und nur eine Senke existierte.

Diese Erkenntnisse lassen sich auch auf die Logiknetzoptimierung beziehen: Bei einer zu geringen Abtastung durch Signalsenken wird auch die Trefferwahrscheinlichkeit klein. Ohne eine ausreichende Wechselwirkung, d.h. ohne eine genügende Anzahl von Rückkopplungen ist keine Differenzierung durch den EA möglich.

➔Aktuell analysiere ich quantitativ den Einfluss von Nichtlinearität und Abtastung auf die Funktion des EA

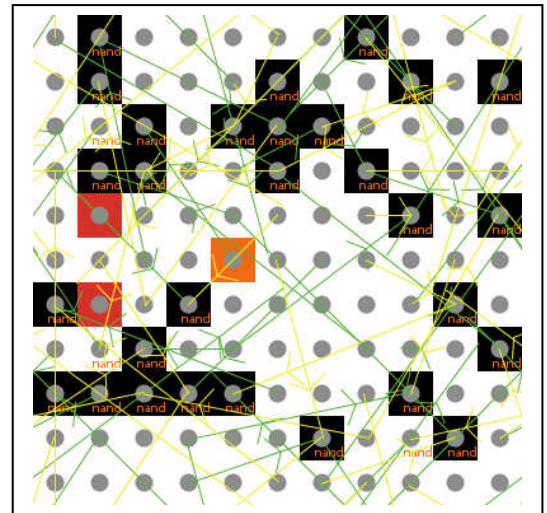


Abbildung 15: Test einer freien Vernetzung von Nand-Gattern

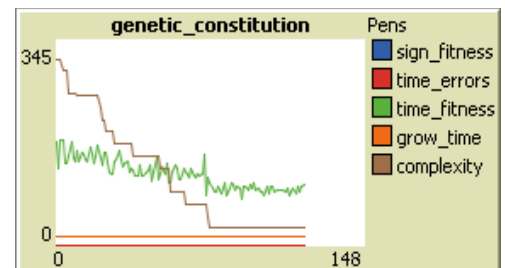


Abbildung 16: Optimierte Komplexität des Erbguts einer freien Vernetzung

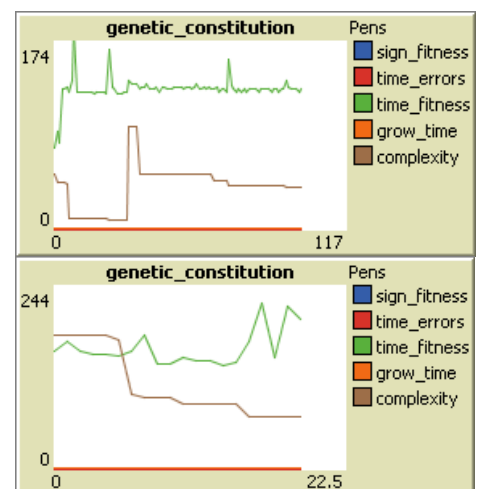


Abbildung 17: Die überflüssigen Punkte wurden bei der Punktoptimierung nach und nach entfernt. Komplexitätssprünge (wie oben) blieben jedoch selten

## 5.5 Züchtung selbstorganisierender Logik-Strukturen

Ich analysierte einige in der Literatur bekannte Agentensimulationen zur Selbstorganisation und stellte fest, dass auch die funktionale Information über die Rezipienten durch den Organisationsprozess beeinflusst werden muss.<sup>9</sup> Bei der freien Vernetzung von Logikbausteinen war dies nicht der Fall, weil die Gatter in ihrer Funktion unbeeinflusst blieben. Um eine funktionsfähige Selbstorganisation aufzubauen und diese zur Konstruktion von Logikstrukturen zu nutzen entwickelte ich ein 2-Phasen-Modell, das innerhalb des Reaktionsraumes des EA abläuft. In der Züchtungsphase wird eine Selbstorganisation von Agenten simuliert, deren Regelsatz besonders zur Bildung von Netzwerken geeignet ist:

### Regelsatz der Züchtungsphase:

1. Jeder Agent ist Träger eines Gatters und hat 2 Eingänge(Beine) und einen Ausgang (Kopf)
2. Der Agent bewegt sich mit zufälliger Richtung und einstellbarer Schrittweite über das Feld. Dabei vermeidet er, dass sein Ausgang auf einem anderen Ausgang oder einer Quelle stoppt.
3. Der Agent bleibt stehen und dockt an, wenn:
  - sein Ausgang auf einer Signal-Senke liegt, oder
  - einer seiner Eingänge auf einer Signal-Quelle liegt, oder
  - sein Ausgang auf dem Eingang eines angedockten Agenten liegt, oder
  - einer seiner Eingänge auf dem Ausgang eines angedockten Agenten liegt und die entstehende Verbindung keine direkte Rückkopplung zweier Agenten ist.
4. Für den Agenten läuft ein Countdown ab, solange wie beide Eingänge noch frei sind oder der Ausgang noch frei ist.
5. Nach Ablauf des Countdowns dockt der Agent ab und bewegt sich wieder wie unter 1.  
Der Countdown stoppt, wenn der Ausgang und mindestens ein Eingang verbunden ist.

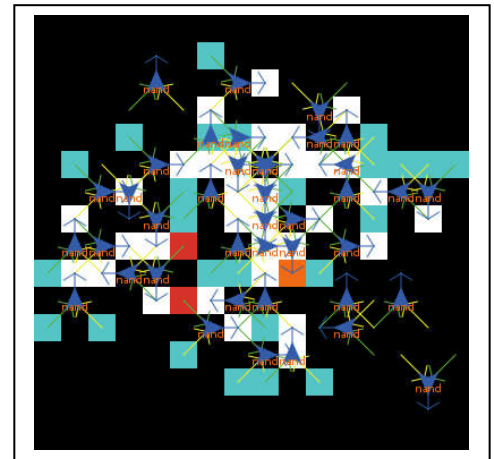
### Testphase:

Nach der Züchtungsphase transferieren die Agenten ihre Logikfunktion und ihre Vernetzung in das Ebenenelement unter ihnen. Nun kann über die Quelle eine Eingangsfolge in die entstandene Logik-Struktur induziert werden. Die Reaktion des Systems wird an der Senke aufgezeichnet und geht, wie bereits beschrieben, in die Kalkulation der Fitness des Moduls ein.

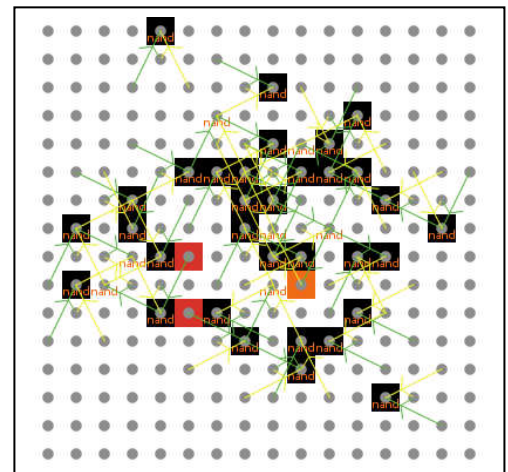
In ersten Experimenten konnte ich die Bildung von Netzwerken zwischen den Quellen und Senken beobachten. Der EA benötigt eine Parametermenge, mit der diese Form der Selbstorganisation gezielt steuerbar ist: Da die Agenten ein hohes Maß von Mobilität besitzen, kann das Systemverhalten nur über die Schrittweite, die Dauer der Verbindung und den Aufbau der Agenten beeinflusst werden. Zusätzlich ist es möglich, weitere Kristallisationspunkte in die Reaktionsebene einzubringen.

## 6. Fazit

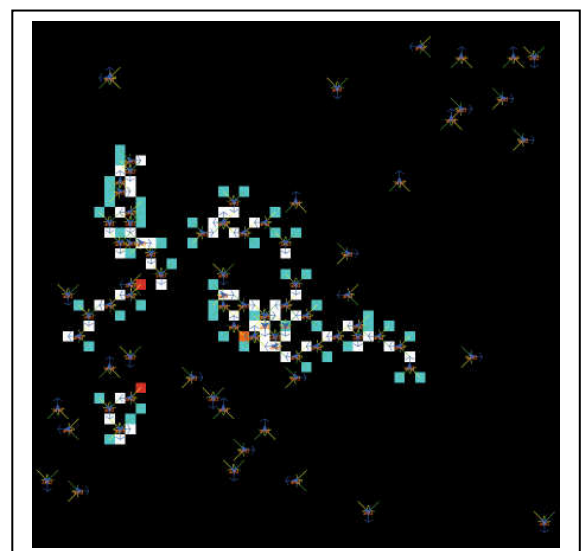
Durch die verschiedenen Simulationen konnte ich mein Modell zur Selbstorganisation von Informationen bestätigen und erweitern. Dabei ist vor allem die Entdeckung von Attraktoren in Logikstrukturen und die Analysen zur Wechselwirkung und Rückkopplung in Bezug auf Selbstorganisation und Evolution zu nennen. Die 3 existierenden Simulationsansätze können nun so modifiziert werden, dass eine Steuerung der Selbstorganisation durch den EA möglich wird.



*Abbildung 18: Selbstorganisation eines Logiknetzes durch Reaktion der Agenten auf Markierungen*



*Abbildung 19: Test einer in die Ebene transferierten Logik-Struktur mit auskristallisierten Bereichen*



*Abbildung 20: Suche nach Steuer-Parametern bei Netzwerkbildung mit vielen Agenten*



## 7. Ausblick

- Quantitative Analyse der Informationsverteilung und der Phasendiagramme der Attraktoren
- Optimierung einer Form der Selbstorganisation durch den EA

### Selbstorganisation von Bits zu Strukturen mit algorithmischen Eigenschaften

Im nächsten Schritt könnte erprobt werden, ob die Selbstorganisation auch auf der untersten Ebene der Informationsverarbeitung genutzt werden kann. Bisher basierten meine Ansätze für die Organisation von Symbolen auf einer Logik-Umgebung. Um den Prozess jedoch auch für komplexere Aufgaben nutzen zu können, müssen die entstehenden Bit-Systeme wie Algorithmen wirken. Dazu wird eine verarbeitende Umgebung benötigt, in der solche Organisationsprozesse ablaufen können. Die Wechselwirkung der selbstorganisierenden Speichergruppen müssten über den Regelsatz der verarbeitenden Struktur definiert werden. Es wird also eine Art Prozessschaltung oder ein anderes informationsverarbeitendes System benötigt. Eine Anwendung wäre prinzipiell in der Chemie (chemische Algorithmen) oder auf Quantencomputern möglich. Um einem quantenmechanischen Regelsatz zur Wechselwirkung von Informationen näher zu kommen, beschäftige ich mich zur Zeit mit Quantenelektrodynamik, Quanteninformation und Informationstheorie und bilde Analogien. Ich plane, diesen Regelsatz später zu einer Simulation der Selbstorganisation von Q-Bits nutzen zu können.

→ *Meine bisherigen Überlegungen zu diesem Thema sowie weitere Nutzungsansätze sind im Anhang zu finden (Quantenmechanisches Modell der Wechselwirkung)*

## 8. Quellenverzeichnis

---

<sup>1</sup> **Hermann Haken** <http://www.scholarpedia.org/article/Self-organization>

<sup>2</sup> <http://de.wikipedia.org/wiki/Selbstorganisation>

<sup>3</sup> **Nanotechnology-** Physics, Chemistry, and Biology of Functional Nanostructures; Landesstiftung Badenwürttemberg

<sup>4</sup> **Carl Friedrich von Weizsäcker**, Die Einheit der Natur, München: dtv, 1974

<sup>5</sup> 1953 stellte **Miller**, ein Student aus Chicago, die Bedingungen der anorganischen Uratmosphäre nach und konnte die Selbstorganisation zu organischen Strukturen wie Aminosäuren oder Desoxiribosen zeigen.

<sup>6</sup> **Evolutionäre Programmierung-** Peter Schmitz; Jugend-Forscht Mathematik/Informatik 2007/08;

<sup>7</sup> <http://de.wikipedia.org/wiki/Nyquist-Shannon-Abtasttheorem>

<sup>8</sup> „**Netlogo**“ ist ein am MIT entwickeltes Multiagentensystem, das die Sprache „Logo“ verwendet

<sup>9</sup> **Frank Schweitzer**: Selbstorganisation und Information, In: Komplexität und Selbstorganisation – "Chaos" in Natur- und Kulturwissenschaften (Hrsg. H. Krapp, Th. Wägenbaur), Wilhelm Fink Verlag, München 1997, S. 99-129