

9. Aktivität, Lernen und Kooperation

Erinnerung an Kapitel 1:

Merkmale lebender Systeme:

- ein lebender Organismus existiert in Raum und Zeit
- er wächst oder dehnt sich aus
- er hat einen Stoffwechsel (ist fähig, chemische Energie aus aufgenommener Materie in andere Energieformen zu verwandeln)
- er wird aktiv in seiner Umgebung, insbesondere ist er reaktionsfähig auf Reize aus der Umgebung

Aktivität:

- Reaktivität (Reaktionen auf Reize; "Stimulus-Response-Modell" – B. F. Skinner)
- Eigenaktivität (Proaktivität)

einfache Aktivitätsmuster lassen sich schon in die bisher behandelten formalen Modelle einbauen:

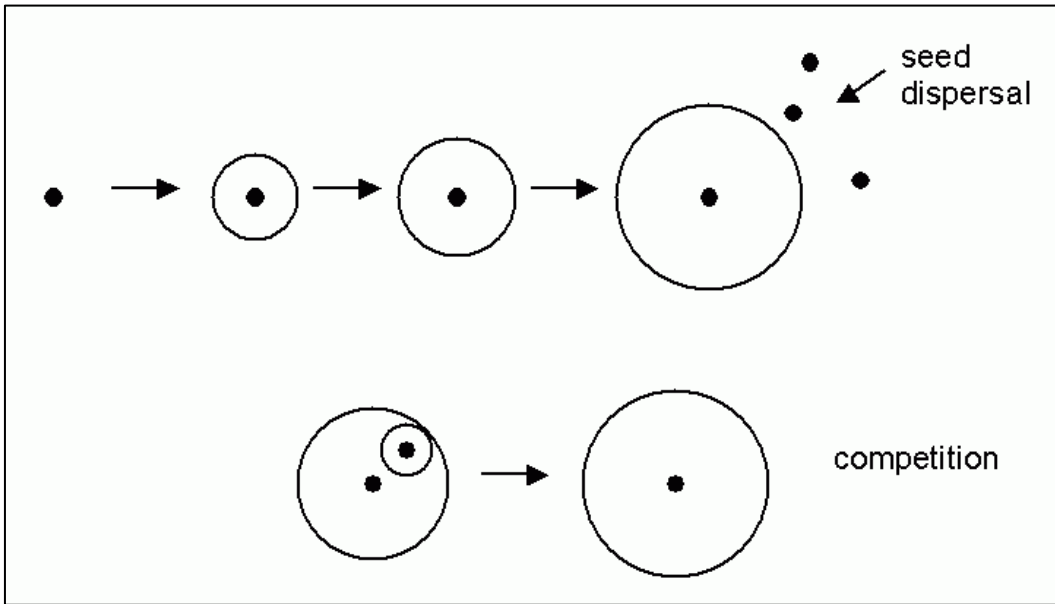
- Reaktivität durch Abfrage von Bedingungen (Verzweigungen; case-Anweisungen)
- Eigenaktivität durch Zufallskomponenten

Beispiel:

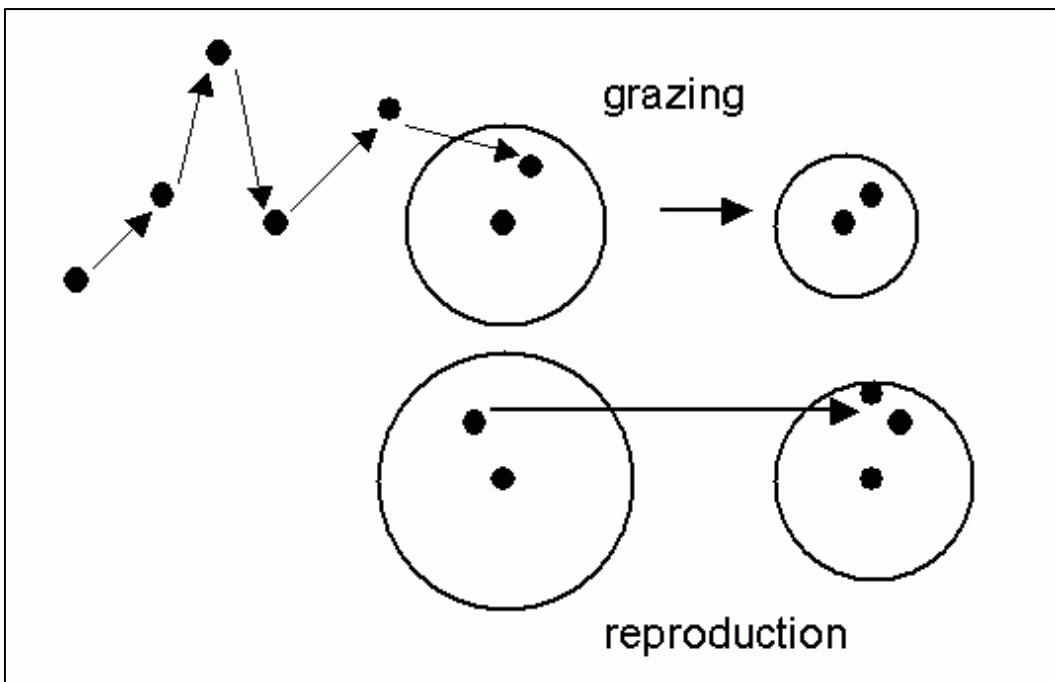
L-System-Modell eines einfachen "Ökosystems" aus Pflanzen und Tieren

Aktivitäten der Tiere:

- Nahrungssuche (Suche nach Pflanzen; *random walk*)
- Fressen (dabei Bewegung vermindert)



Verhalten der Pflanzen



Verhalten der Tiere

Regelsystem (in XL):

```
/* phytophag.rgg: specification of a grazing and competition model
with circular-shaped plants and animals */

module Plant(int t, super.radius) extends Cylinder(1, radius)
  {{setColor(0x00aa00);}}
module Animal(int t, super.radius) extends Cylinder(2, radius)
  {{setColor(0xff0000); setBaseOpen(true); setTopOpen(true);}};

double pgrow = 0.9; /* regular growth increment per timestep */
double seed_rad = 0.1; /* initial radius of a plant */
int pmaxage = 30; /* maximal age of a plant */
int pgenage1 = 10; /* first reproductive age level */
int pgenage2 = 18; /* second reproductive age level */
double distmin = 15; /* minimal seed distance */
double distmax = 40; /* maximal seed distance */
double pminrad = 9; /* necessary plant radius for reproduction */
double pgenfac = 0.5; /* ratio #seeds/radius */

int lag = 15; /* sleeping time for animal at start */
double shortstep = 0.4; /* movement of animals inside plant canopy */
double longstep = 15; /* movement of animals outside */
double f_e = 0.2; /* ratio radius / energy of animals */
double init_e = 4; /* initial energy amount of animals */
double respi = 0.25; /* energy consumed by animals' respiration */
double thr = 7.6; /* energy threshold for reproduction
of animals */
double eat = 1.1; /* energy transferred during grazing */

protected void init()
[
  Axiom ==> Plant(0, seed_rad) [ RH(random(0, 360))
    RU(90) M(10) RU(-90) Animal(-lag, f_e*init_e) ];
]

public void make()
{ growAnimals(); derive(); growPlants(); }

public void growAnimals()
[
  Animal(t, e), (t < 0) ==> Animal(t+1, e); /* start lag */
  Animal(t, e), (e <= 0) ==> ;
  Animal(t, e), (e > f_e*thr) ==>
  [ RH(random(0, 360)) RU(90) M(shortstep) RU(-90)
    Animal(0, e/2 - f_e*respi) ]
  RH(random(0, 360)) RU(90) M(shortstep) RU(-90)
  Animal(0, e/2 - f_e*respi);

  a:Animal(t, e), (* p:Plant(u, r) *), (distance(a, p) < p[radius]) ==>
  RH(random(0, 360)) RU(90) M(shortstep) RU(-90)
  Animal(t+1, e + f_e*eat - f_e*respi) { p[radius] -= eat; };

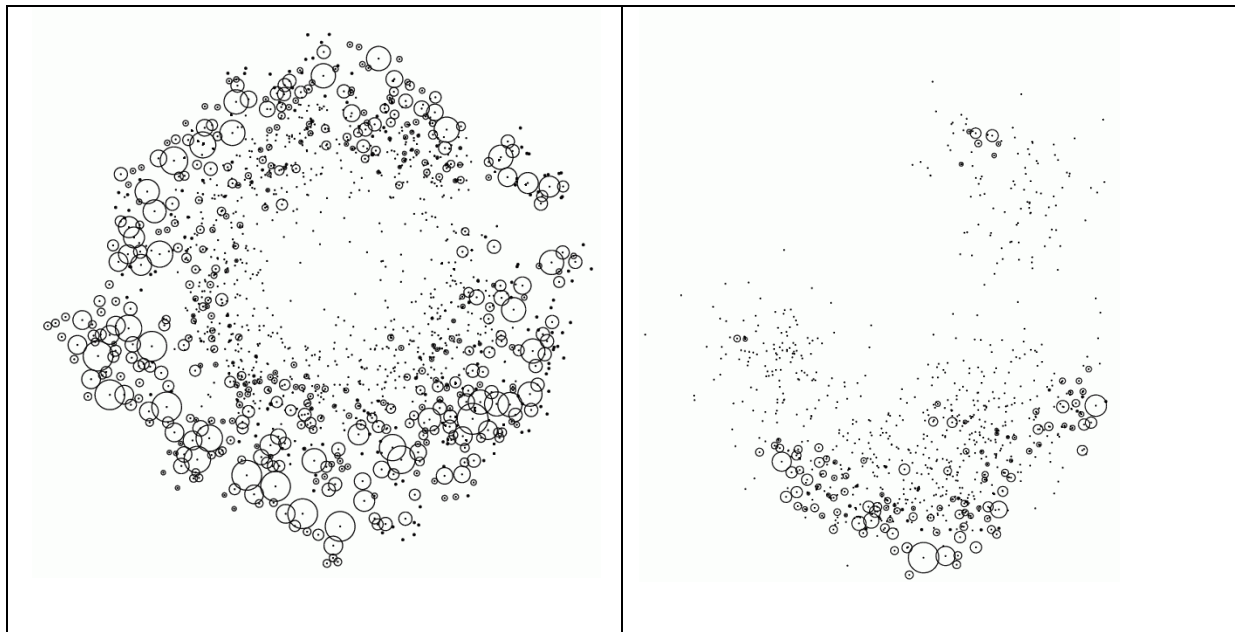
  Animal(t, e) ==>
  RH(random(0, 360)) RU(90) M(longstep) RU(-90)
  Animal(t+1, e - f_e*respi);
]
```

```

public void growPlants()
[
  Plant(t, r), (t > pmaxage) ==> ;
  Plant(t, r), (r < 0) ==> ;
  p:Plant, (* q:Plant *), (distance(p, q) < q[radius]
    && p[radius] <= q[radius]) ==> ;
  Plant(t, r), ((t == pgenage1 || t == pgenage2) && r >= pminrad)
    ==> for ((1 : (int) (pgenfac*r)))
      ( [ RH(random(0, 360)) RU(90) M(random(distmin, distmax))
        RU(-90) Plant(0, seed_rad) ] )
      Plant(t+1, r);
  Plant(t, r) ==> Plant(t+1, r+pgrow);
]

```

2 Simulationsergebnisse mit verschiedenen Parametersätzen:



Typen einfacher Verhaltensweisen (gewonnen am Beispiel der Bewegung von Insekten – Beer et al. nach Thro 1994):

Reflex	sofortige Reaktion auf plötzlichen Reiz (beim Menschen unbewusst)
Taxis	Orientierung aufgrund von Gradienten in der Umgebung (Licht, Schall, chem. Konzentration, Schwerkraft...) – zur Quelle hin oder von ihr weg
Reak. auf angeborenen Stimulus	Auslösung von Verhaltensweisen durch festes Schema (Flucht vor Raubtier-Silhouette; Kindchenschema...)
Appetenzverhalten	ausgelöst durch Kombination von Reiz und einem inneren Zustand (z.B. Fressen – bei Vorhandensein von Futter und Hunger)
bedingter Reflex	Verhaltensänderung als Folge von Reizwiederholung

Einfache Typen von Bewegungsverhalten für die Robotersteuerung (nach Anderson & Donath; Thro 1994):

Anziehung nach vorn

bewegt den Roboter auf seinem aktuellen Kurs

Ortsanziehung

Roboter geht zu einem bestimmten Standort im Raum (vgl. Zugverhalten der Vögel)

Anziehung durch ein Objekt

Roboter bewegt sich auf ein von ihm ermitteltes Objekt zu. Wenn kein Objekt ermittelt, keine Bewegung.

Folge Objekt gegen den Uhrzeigersinn [im Uhrzeigersinn]

Roboter dreht sich um das Objekt

Anziehung durch begrenzte Räume

bewegt den Roboter in das Gebiet, das sich im kleinsten Winkel öffnet. Dies ermöglicht Schutz vor "Feinden", die zu groß sind, um in diesen Raum hineinzupassen

Anziehung durch offene Räume

bewegt den Roboter in das Gebiet, das sich im größten Winkel öffnet. Nutzen: bestmöglicher Überblick über die Umgebung

Ortsgebundene Anziehung durch offene Räume

Roboter bewegt sich durch offene Gebiete, die (ungefähr) in der gleichen Richtung liegen wie der Zielort

Passive Vermeidung

Erstarren auf der Stelle, wenn eine Kollision mit einem Objekt droht

Aktive Vermeidung

Roboter weicht dem Objekt aus

Durch Kombination einiger dieser Verhaltensweisen entstehen neue Verhaltensweisen:



Anziehung nach vorn + Aktive Vermeidung + Passive Vermeidung bewirkt *generelles Herumwandern*, bei dem der Roboter einen großen Teil seiner Umgebung durchstreift.



Aktive Vermeidung + Passive Vermeidung + Ortsanziehung bewirkt *einfache Navigation*, bei der sich der Roboter ohne Kollision mit irgendeinem Objekt auf sein Ziel zubewegt.



Aktive Vermeidung + Ortsanziehung bewirkt *Stagnation*, die ähnlich wie einfache Navigation ist, ihre einfachsten Verhaltensweisen neutralisieren sich jedoch gegenseitig. Der Roboter hält an, kurz bevor er die angepeilte Position erreicht hat.



Passive Vermeidung + Aktive Vermeidung + Anziehung durch ein Objekt + Folge Objekt (im und gegen den Uhrzeigersinn) bewirkt das *Folgen an der Peripherie*, bei dem der Roboter sich dicht an umgrenzenden Mauern und anderen Objekten der Umgebung entlangbewegt.



Anziehung durch offene Räume + Aktive Vermeidung + Passive Vermeidung + Anziehung nach vorn bewirkt das *Durchwandern weit offener Räume*, bei dem der Roboter durch das Zentrum dieser Räume marschiert.



Anziehung durch begrenzte Räume + Aktive Vermeidung + Passive Vermeidung + Anziehung nach vorn bewirkt das *Durchwandern eng begrenzter Räume*, bei dem sich der Roboter in der Nähe der Öffnung bewegt.



Ortsgebundene Anziehung durch offene Räume + Aktive Vermeidung + Passive Vermeidung + Anziehung nach vorn bewirkt die *ortsgerichtete Durchwanderung offener Räume*, die ein zyklisches Muster darstellt. Der Roboter bewegt sich auf seinen Zielort zu, wobei er in einem offenen Raum anhält und dann weitermarschiert. Dieses Muster wiederholt sich mehrere Male, bevor er den angestrebten Ort erreicht.

Wie können Verhaltensregeln verknüpft und repräsentiert werden?

Klassifizierungssystem (John Holland):

- Menge von Regeln (*Classifyern*)
jede codiert durch 2 Bitstrings: z.B. 01# : 110 (# = "don't care")
oder durch je 2 Symbolfolgen
- Detektoren, um Fakten aus der Umwelt wahrzunehmen
- diese werden als Symbolfolgen codiert und auf eine "Anzeigetafel" gesetzt
- wenn die Nachrichten auf der Tafel die Bedingungen der Klassifizierungsregeln erfüllen, erfolgt eine Aktion: neue Nachricht wird auf die Tafel gesetzt; ggf. werden zusätzlich "Effektoren" für Außenwirkung aktiviert
- die Regeln werden durch Stärke-Indices gewichtet

Beispiel "Frosch":

WENN kleines fliegendes Objekt von links, DANN sende @.
WENN kleines fliegendes Objekt von rechts, DANN sende %.
WENN kleines fliegendes Objekt in der Mitte, DANN sende ž .
WENN großes näherrückendes Objekt, DANN sende !.
WENN kein großes näherrückendes Objekt, DANN sende *.
WENN * und @, DANN bewege den Kopf 15° nach links.
WENN * und %, DANN bewege den Kopf 15° nach rechts.
WENN * und ž , DANN bewege dich in die Richtung, in die der Kopf zeigt.
WENN !, DANN bewege dich schnell in die entgegengesetzte Richtung, in die der Kopf zeigt.

Anzeigenbrett

(Enthält die jüngsten Nachrichten)

* @ .

Individuenbasierte Tiermodelle (Etho-Modelling)

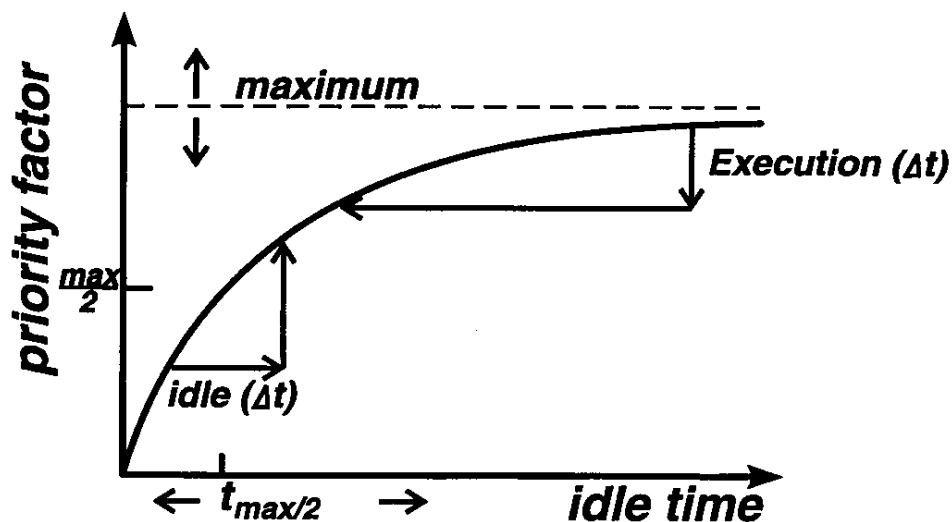
Beispiel: Rotkehlchenmodell von Reuter & Breckling (Breckling et al. 1997)

Tiere als Objekte i. Sinne der OOP

Variablen für jedes Individuum: Position, Gewicht, Energievorrat, Territorialbesitz ...

Methoden (Tasks): Singen, Jagen, Ruhen, Junge füttern...

- jede Aktion kostet Energie
- Verhaltensregeln verknüpfen Bedingungen mit den Methoden
- Tasks sind mit veränderlichen Prioritätsfaktoren versehen



- Auswahl einer Task zu einem gegebenen Zeitpunkt resultiert aus: Umwelteinflüssen, innerer Zustand (z.B. Energiebedarf; aktuelle Phase im Brutzyklus), Interaktion mit anderen Rotkehlchen (Revierverhalten; Paarung; Brutpflege) und mit Raubtieren
- durch *time scheduling* (kontinuierliche Zeit; in SIMULA inhärent) wird für alle Individuen simultan der Aktivitätszustand verwaltet
- die Parameter beruhen auf gemessenen Daten!

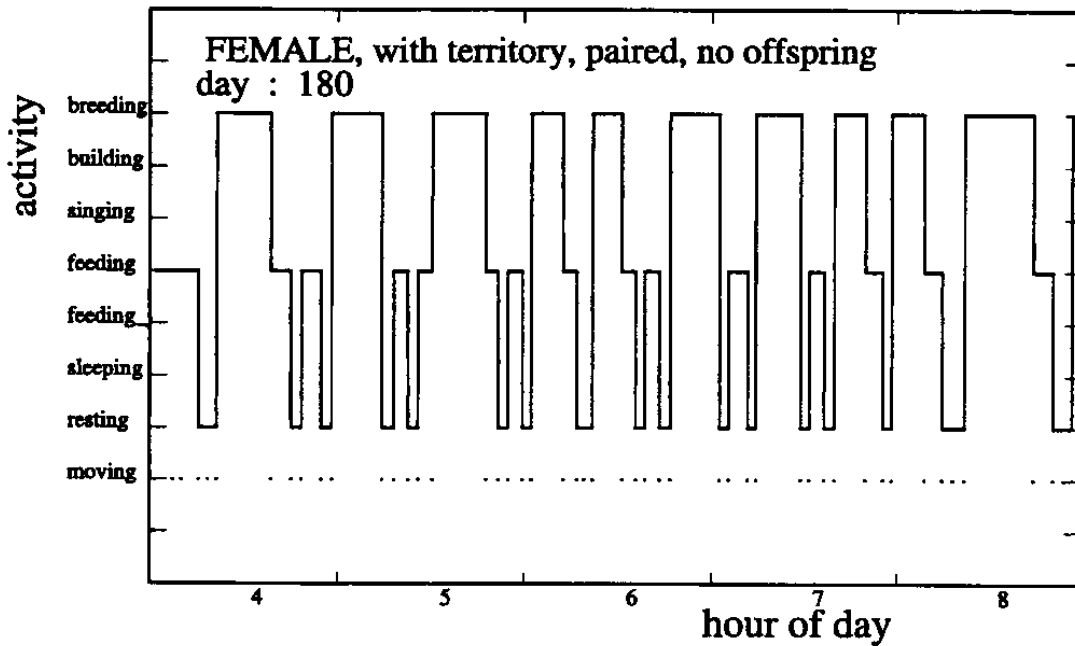


Figure 7a. Activity sequence for a breeding female. Results from the simulation program.

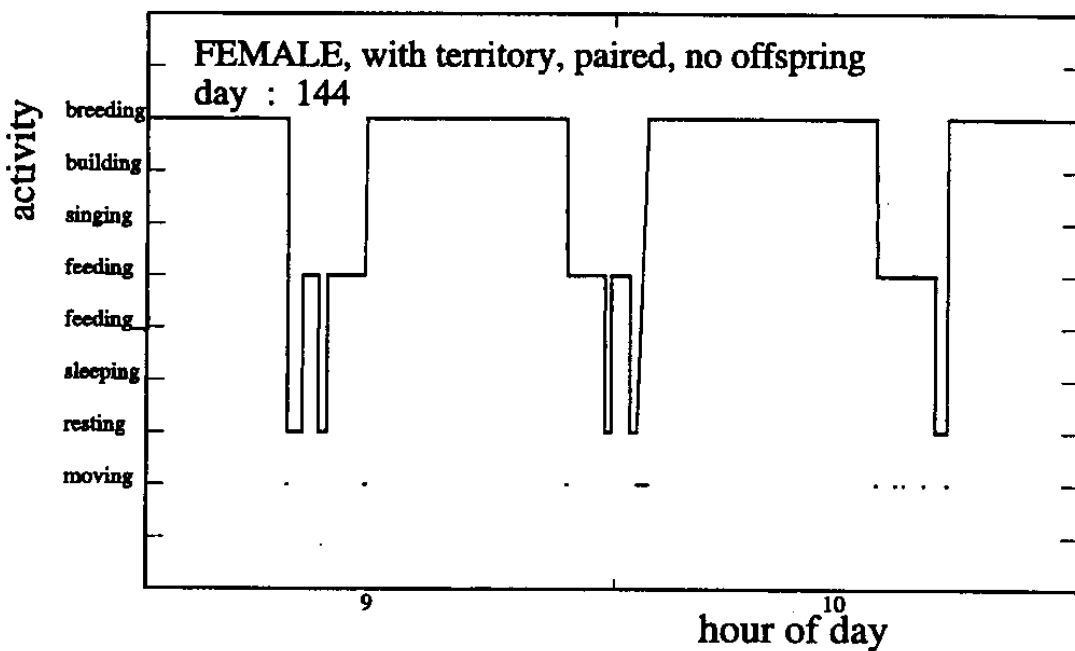


Figure 7b. Activity sequence for a breeding female. Field data (personal communication, B. Grajetzky, Project Center for Ecosystems Research, unpublished data).

Simulationsergebnis (oben) und gemessene Daten (unten)

(aus Breckling et al. 1997)

Von höherentwickelten Tieren erwarten wir, dass sie nicht nur nach einem festen Regelsystem aktiv sind, sondern frühere Erfahrungen einbeziehen in die Steuerung ihres Verhaltens
– dass sie *lernen*

Was ist Lernen?

8 Lerntypen nach Gagné:

Typ 1: *Signallernen* (bedingter Reflex; Pawlowsche Konditionierung)

Futter → Speichelfluss

Futter + Glockenton → Speichelfluss (wiederholt)

Glockenton → Speichelfluss

Typ 2: *Reiz-Reaktions-Lernen* (Stimulus-Response; Skinner'sches Lernen)

Lernen durch Verstärkung (positiv: Belohnung, negativ: Bestrafung); Verhalten wird häufiger, wenn eine Verstärkung damit verknüpft wird

- funktioniert auch (sogar besser!) bei "intermittierender" (unzuverlässiger) Verstärkung

Typ 3: *Kettenbildung / motorische Ketten*

Verbindung einer Abfolge motorischer Reiz-Reaktions-Verhaltensweisen

Kompetenzlernen, z.B. Radfahren, Schwimmen, eine Suppe zubereiten...

Typ 4: *Kettenbildung / sprachliche Assoziation*

Verbindung einer Abfolge verbaler Reiz-Reaktions-Verhaltensweisen

Beispiele: Zählen, Gedicht aufsagen

Typ 5: *Lernen multipler Diskriminationen*

Lernen, zwischen hochgradig ähnlichen Reizinputs zu unterscheiden

z.B. gleich klingende Wörter in verschiedenen Sprachen

Typ 6: *Begriffslernen*

Ordnen von Dingen zu Klassen, Reagieren auf Klassen als Ganze

z.B. Begriffe "Hund", "Mensch"

Typ 7: *Regellernen*

"Regel" hier als eine erschlossene Fähigkeit, die das Individuum befähigt, auf eine Klasse von Reizsituationen mit einer Klasse von Leistungen zu reagieren

Typ 8: *Problemlösen*

Anwendung mehrerer Regeln bringt Regeln höherer Ordnung hervor

z.B. Strategien beim Schachspiel

nicht zu diesem Schema passend (z.T. quer liegend):
weitere Lerntypen

Prägung: Erlernen eines komplexen Musters in einer speziellen sensitiven Phase (Konrad Lorenz – Graugänse-Küken lernen, Mutter zu erkennen)

Imitationslernen (oft bei Kettenbildung (Typ 3, 4) beteiligt, aber evtl. auch bei höheren Typen): Lernen durch Imitation (Nachahmung) eines Vorbilds

protokollarisches Lernen: Aufnahme von Ereignissen ins Gedächtnis

man unterscheidet:

- sensorisches Gedächtnis (hohe Kapaz., ca. 1 Sek.)
- Kurzzeitgedächtnis (nur ca. 7 Items, 10 Sek.)
- Langzeitgedächtnis (unbegrenzt)

Priming-These: Alle Gedächtnisinhalte sind verbunden mit den Umweltbedingungen während der Informationsaufnahme (z.B. Gerüche, Körperhaltung, Räumlichkeiten etc.)

Vergessen: Gedächtnisinhalt zerstört oder nicht auffindbar

Unbewusstes (Unterbewusstsein); *Sigmund Freud* als Pionier – beim Menschen ist sehr vieles unbewusst gespeichert

soziales Lernen: besondere Sensibilität für Gesichter, Stimmen, Gestik, Mimik, soziale Beziehungen und Hierarchien

Die drei wichtigsten Lerntheorien und ihre Grundauffassungen:

Behaviorismus, Kognitivismus und Konstruktivismus

Kategorie	<u>Behaviourismus</u>	<u>Kognitivismus</u>	<u>Konstruktivismus</u>
Hirn ist ein	Passiver Behälter	Informationsverarbeitendes "Gerät"	Informationell geschlossenes System
Wissen wird	Abgelagert	Verarbeitet	Konstruiert
Wissen ist	Eine korrekte Input-Output-Relation	Ein adäquater interner Verarbeitungsprozess	Mit einer Situation operieren zu können
Lernziele	Richtige Antworten	Richtige Methoden zur Antwortfindung	Komplexe Situationen bewältigen
Paradigma	Stimulus-Response	Problemlösung	Konstruktion
Strategie	Lehren	Beobachten und helfen	Kooperieren
Lehrer ist	Autorität	Tutor	Coach, (Spieler) Trainer
Feedback	Extern vorgegeben	Extern modelliert	Intern modelliert
Interaktion	Starr vorgegeben	Dynamisch in Abhängigkeit des externen Lernmodells	Selbstreferentiell, zirkulär, struktur determiniert (autonom)
Programmerkmale	Starrer Ablauf, quantitative Zeit- und Antwortstatistik	Dynamisch gesteuerter Ablauf, vorgegebene Problemstellung, Antwortanalyse	Dynamisch, komplex vernetzte Systeme, keine vorgegebene Problemstellung
Software-Paradigma	Lernmaschine	Künstliche Intelligenz	Sozio-technische Umgebungen
"idealer" Softwaretypus	Tutorielle Systeme, Drill & Practice	Adaptive Systeme, ITS	Simulationen, Mikrowelten, Hypermedia

(nach Baumgartner & Payr 1994; WWW)

Erlernen von Regeln in Klassifizierungssystemen

Grundidee: Erfolgreiche Regeln "belohnen" – Stärke der Regel heraufsetzen

Platz in der Regelliste ist begrenzt, Regeln müssen um ihren Platz "kämpfen"

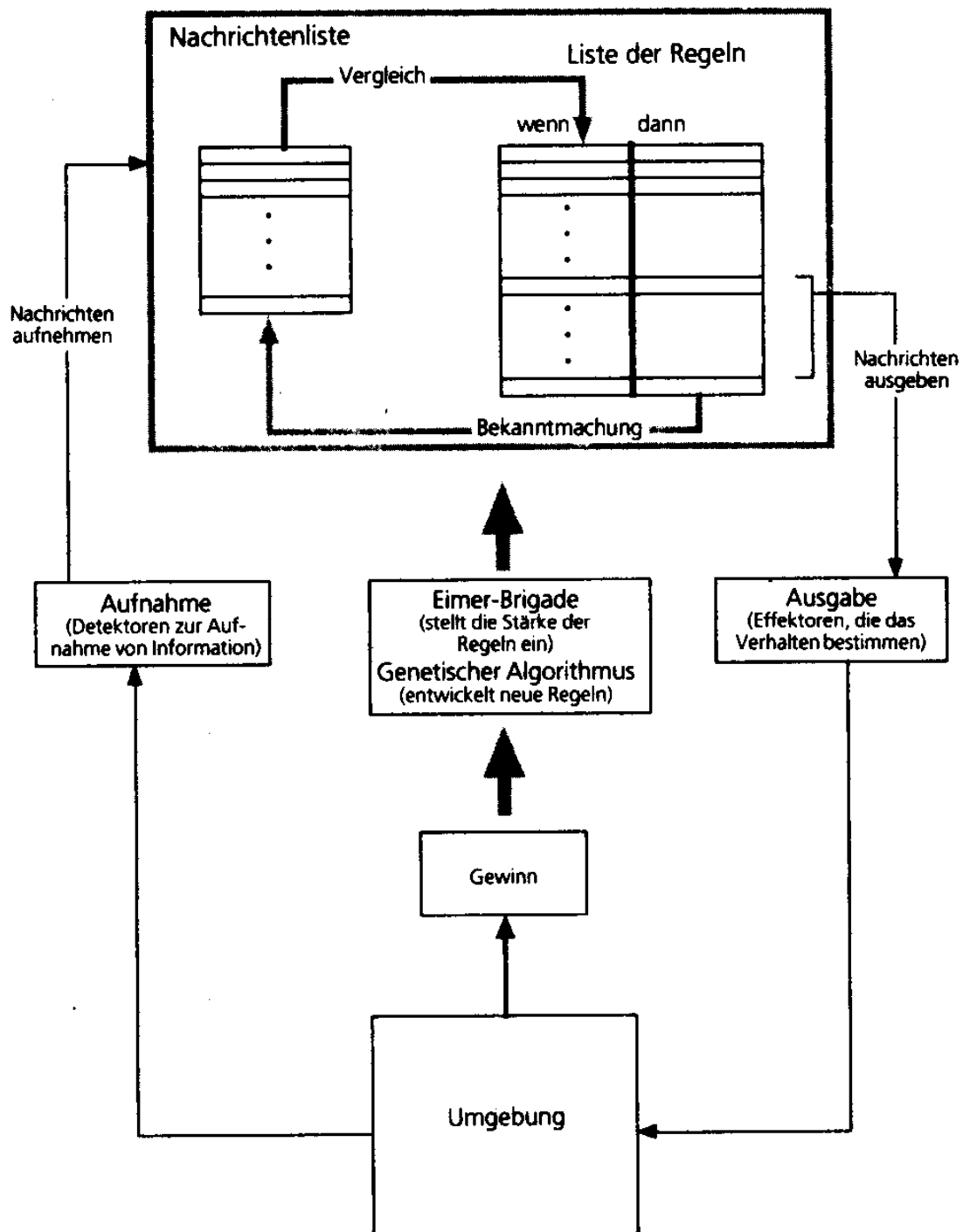
Probleme:

- Regelketten; es würde nur die letzte belohnt
- Erfahrungen treten in keiner festen Reihenfolge auf
- Erfahrungen in der Realität erfordern unterschiedliche (disjunkte oder auch überlappende) Mengen von Regeln

Lösung des Kettenproblems durch "Eimerbrigaden-Algorithmus" (*bucket brigade*):

- jede Regel, die in der Lage war, ihre Nachricht abzugeben, gibt eine "Belohnung" weiter an diejenigen Regeln, die dafür gesorgt haben, dass die Liste in dem passenden Zustand war

→ durch das System geht ein "Stärkefluss", der von Regeln ausgeht, die externe Belohnungen erhalten haben (erfolgreich waren)



Einsatz von lernfähigem Klassifizierungssystem (zusätzlich mit genetischem Algorithmus zum Entwickeln neuer Regeln ausgestattet) in einem künstlichen Tier:

"*Animat*" (animal automat, Begriff von Stewart Wilson)

erster Animat: "*"

einziges Bedürfnis: Nahrung

Umgebung lieferte 92 verschiedene sensorische Eingaben

* wusste anfangs nicht, woran Nahrung zu erkennen war,

es gab Hindernisse ("Bäume" und "Felsen") und eingeschränkte Sehfähigkeit

Wilson sagte nach den ersten Modell-Läufen über *:

*Er lernte tatsächlich. Er schuf Klassifizierungssysteme. Wenn er gegen einen Felsen gelaufen war, erhielt er eine schmerzhaft Rückmeldung, aber da sich häufig Futter in der Nähe der Felsen befand, lernte er nicht nur, nicht gegen den Felsen zu stoßen, sondern auch, nicht einfach wegzulaufen, da sich auf der anderen Seite möglicherweise Futter befand. Und während * sich in dieser Landschaft aufhielt, entwickelt er interessante Verhaltensweisen. Er bewegte sich nicht nach dem Zufallsprinzip, sondern ließ sich normalerweise in eine Richtung treiben, was übrigens auch die beste Taktik für jemanden ist, der sich im Nebel verlaufen hat. Sobald er etwas sah, reagierte er augenblicklich. Wenn es Futter war, fraß er das Futter auf der Stelle, und wenn er auf einen Felsen stieß, bewegte er sich um ihn herum.*

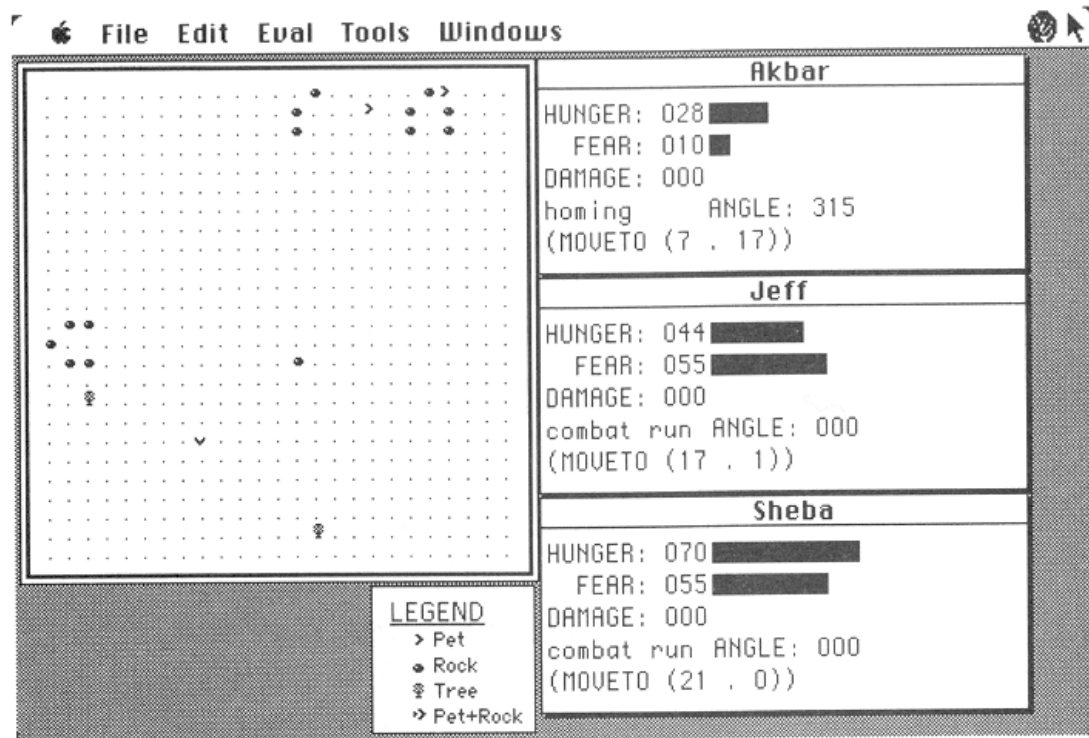
Variante: hierarchische Organisation von ganzen "Verhaltens-Modulen" (nicht bloß Tabelle einzelner Regeln)

– Beispiel: *Petworld* (Bill Coderre, MIT)

Petworld:

- 2-dim. Welt mit Tierchen, Bäumen (die die Tiere nach Nahrung absuchen), Steinen
- Tierchen mit Blickfeld von 90°
- können einzelne Steine tragen
- Tierchen begegnen sich als Feinde (keine Fortpflanzung)
- interne Zustände (Hunger, Verletzung, Angst, Nestlokalisierung, Nutzlast) mit Wertebereichen zwischen 0 und 100
- Gehirn besteht aus Hierarchie von Modulen ("Experten"), jedes ist auf eine Verhaltensweise spezialisiert (z.B. Nahrungssuche, Nestbau)
- Hierarchiestufe hängt von der Bedeutung des Verhaltens ab
- jedes Modul bekommt Eingaben von untergeordneten Modulen und von der Umwelt
- als Reaktion wird eine Rangliste möglicher Verhaltensweisen zur Verfügung gestellt und nach oben weitergereicht
- Ranglisten werden fortschreitend modifiziert (Ergebnis sind Kompromisse, z.B. zwischen Kämpfen und Nahrungssuche, situationsabhängig)
- die vom höchstrangigen Modul empfohlene Aktion wird ausgeführt – und gespeichert (Erinnerung)
- Handlungsempfehlungen können nach dem Erfolg bewertet werden, dadurch Lernfähigkeit (optional)

Screenshot von Petworld:

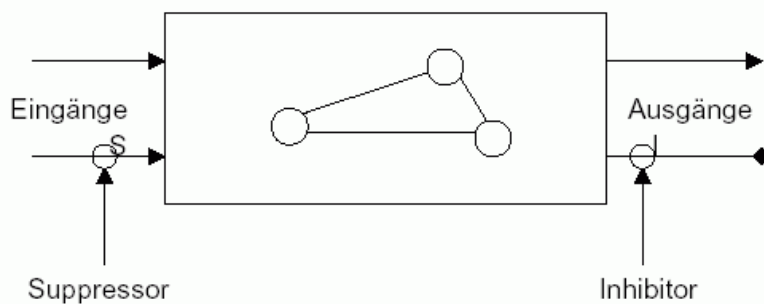


Beobachtung: Neubildung von Verhaltensweisen, z.B. lagerte ein Tierchen zusätzliche Steine in der Nähe seines Nestes

Prinzip der Verhaltens-Module ähnelt der *subsumierenden Architektur* bei Robotern (Rodney Brooks)

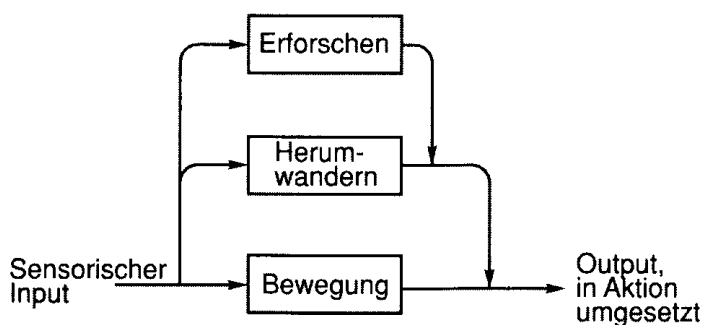
Subsumptionsarchitektur

- häufig in autonomen Robotern eingesetzt
- Idee der verhaltensbasierten Aufteilung
- Konkrete Architektur
- Agent besteht aus einzelnen Modulen mit endlichen Automaten
- Verbindungen zwischen Modulen können verändert werden (Suppressor und Inhibitor)



- Hierarchische Struktur der unabhängigen Module (über Suppressor und Inhibitor)
 - Kollisionsvermeidung über Herumwandern
 - Kollisionsvermeidung über Wegplanung
 - Wegplanung über Herumwandern

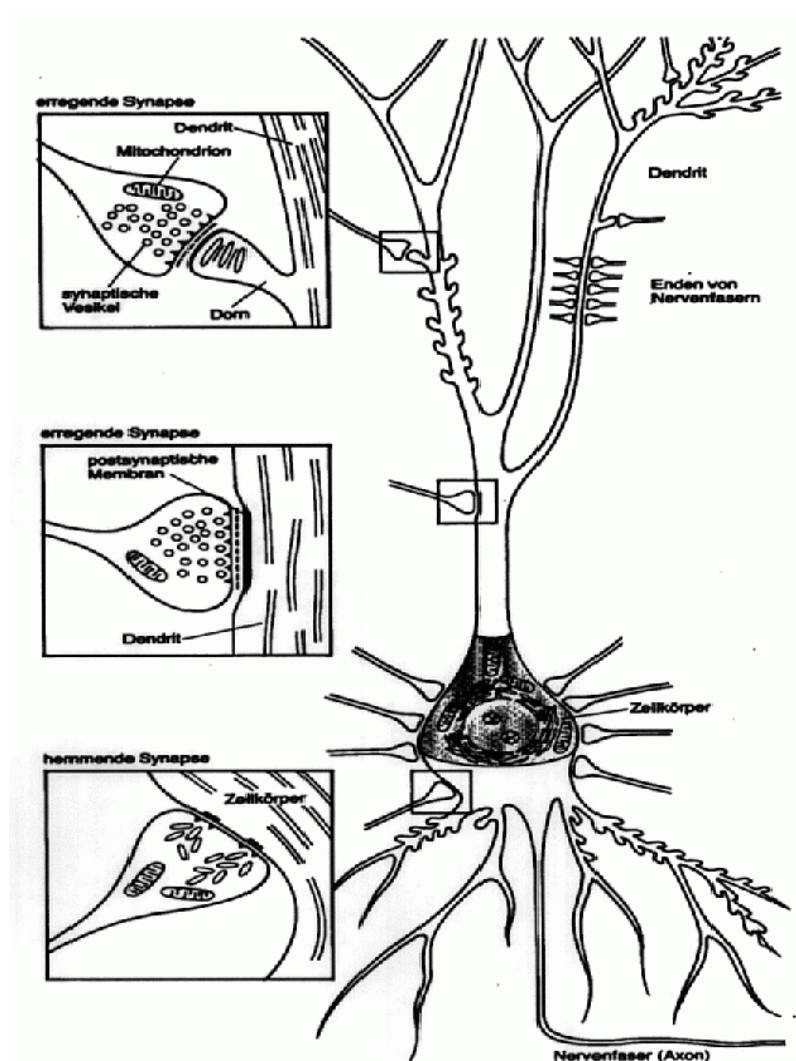
(Klügl 1998)



(Thro 1994)

wie funktioniert Lernen im "natürlichen" Leben?

- nicht vollständig verstanden
- Neubildung, Inhibition und Verstärkung von Verschaltungen zwischen Neuronen spielt wichtige Rolle
- auch chemische Prozesse beteiligt
- extremer Stress in frühen Phasen des Gehirnwachstums kann langfristig hormonale Konzentrationen verändern und die kognitiven Leistungen einschränken



Nervenzelle (Neuron)

(aus Levi 2002)

Ansatz der Nachbildung des zentralen Nervensystems der höheren Tiere, bzw. des Gehirns:
"Konnektionismus" (schon früh ein Ansatz der AI-Forschung, dann zeitweise "aus der Mode" gewesen) –
künstliche neuronale Netze (kurz knN oder NN)

Modellneuron:

Dendriten und Synapsen → gewichtete Verbindungen w_{ij}

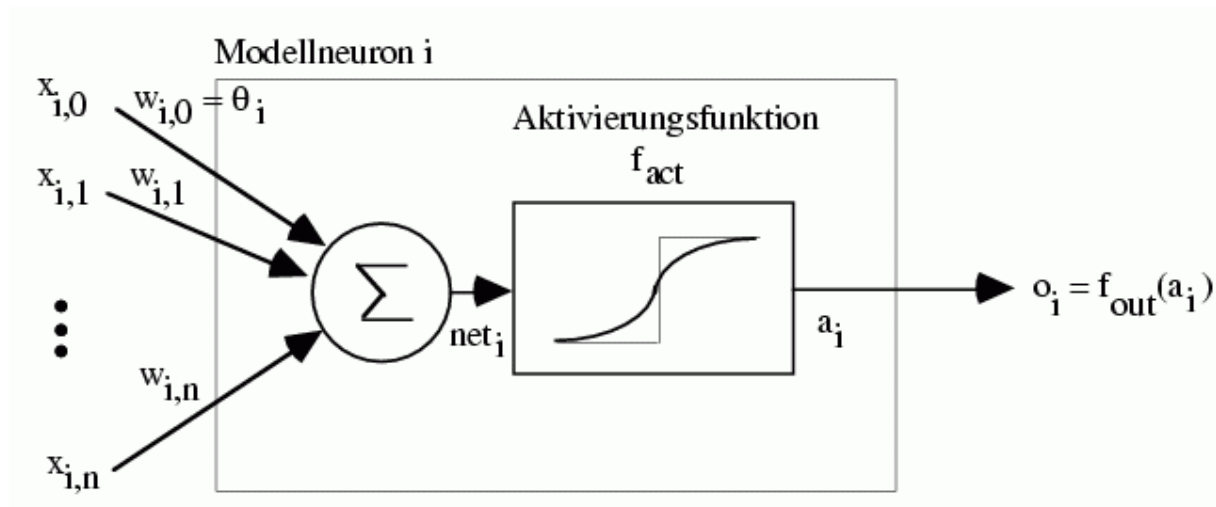
Hemmung (Inhibition) → $w_{ij} < 0$

Erregung (Exzitation) → $w_{ij} > 0$

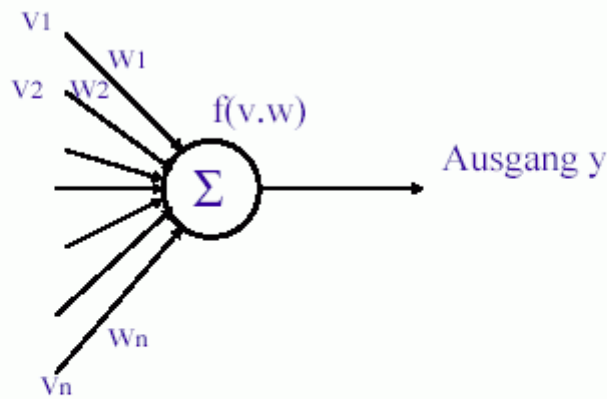
keine Verknüpfung → $w_{ij} = 0$

Zellkörper → Aktivierungszustand a_i , Aktivierungsfunktion f_{act}

Axon → Ausgabefunktion f_{out}



vereinfacht:



Gesamteingang:

$$x = \sum_{i=1}^n v_i w_i$$

$$x = \sum_{i=1}^n v_i w_i - \theta$$

Transferfunktion:

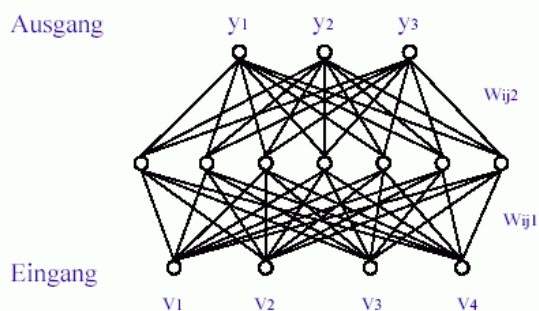
$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (\text{Beispiele})$$

Neuronales Netz:

- Zusammenschaltung **mehrerer Neuronen**
- Ausgang eines Neurons wird Eingang eines anderen Neurons
- **imitiert** die hohe Anzahl von Verbindungen einfacher Neuronen im **menschlichen Hirn**
- Netzwerk besteht aus mehreren Eingängen und mehreren Ausgängen

Beispiel: Feed-Forward-Network mit 1 hidden layer:



Einteilung der Schichten in

- Eingabeschicht
- verdeckte Schichten (*hidden layers*) (optional)
- Ausgabeschicht.

Achtung: in der Literatur wird bei der Zählung der Schichten die Eingabeschicht oft nicht mitgezählt.

Lernverfahren für knN

(a) überwachtes Lernen (*supervised learning*), "Lernen mit Lehrer":

- Trainingsmenge von Eingabe- und Ausgabemustern
- zu jedem Eingabemuster existiert ein eindeutiges korrektes (bestes) Ausgabemuster

Lernen:

Die Gewichte und evtl. Schwellenwerte werden solange durch nochmaliges Anlegen der Eingabemuster verändert, bis die Paarung (Eingabemuster, Ausgabemuster) für die Trainingsmenge stimmt.

Generalisierung:

Ähnliche Eingabemuster, die nicht zur Trainingsmenge gehören, werden nach der Trainingsphase entweder in bereits trainierte Ausgabemuster (Perzeptron) oder in ähnliche Ausgabemuster (Backpropagation-Netzwerke) überführt.

(b) unüberwachtes Lernen (*unsupervised / self-organized learning*):

Lernen erfolgt durch Selbstorganisation. Ähnliche Eingabemuster werden assoziativ als ähnlich klassifiziert. Beisp.: Kohonen-Netze.

– siehe auch Lehrveranstaltungen der Neuro-Informatik –

Anwendungsbeispiel für knN im "Artificial Life"-Bereich: David Ackley & Michael Littman: "AL"

- 100 × 100 - Gitter
- *Agenten*
- Pflanzen (Futter)
- Bäume
- Felsen (Wände)
- Fleischfresser (Feinde)
- jeder Agent kennt den Inhalt von 4 Nachbarzellen in jeder Richtung
- jeder Fleischfresser 6 Nachbarzellen in jeder Richtung

Gehirn der Agenten: ein Auswertungs- und ein Aktionsnetzwerk

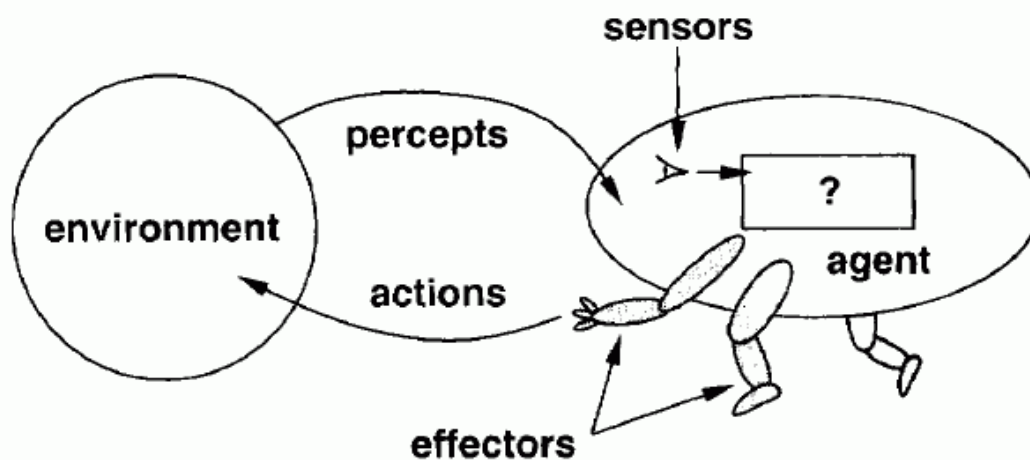
- *Auswertungs-Netzwerk*: übersetzt sensorische Informationen in eine numerische Rangordnung
- *Aktions-Netzwerk*: setzt auf der Basis der Ziele aus dem Auswertungs-Netzwerk Informationen in Verhaltensweisen um, misst den momentanen Erfolg eines Agenten, vergleicht ihn mit früheren Auswertungen und ermöglicht Lernen durch Verstärkung!
- sexuelle Fortpflanzung
- Vererbung (siehe später)



Das Konzept der "Agenten" findet jedoch in den letzten Jahren immer mehr Interesse – auch zum praktischen Problemlösen

„Agent“

„Asking the question of what an agent is to a DAI researcher is as embarrassing as the question of what intelligence means is to an AI researcher“
(C. Hewitt, DAI-Workshop, 1994)



(Klügl 1998; DAI = distributed artificial intelligence)

Russell & Norvig 1995:

"An agent is anything that can be viewed as perceiving its environment through its sensors and acting upon that environment through effectors"

Eigenschaften eines Agenten:

- er existiert (mit gewisser Dauer) in seiner Umgebung (*Situatedness* und *Permanenz*)
- *autonom*: Verhalten wird durch den Agenten selbst bestimmt, ohne Kontrolle von außen
- *reaktiv*: Agent kann Ereignisse wahrnehmen und auf dieser Grundlage seine Aktionen abstimmen (*Responsiveness*)
- *proaktiv*: Agent reagiert nicht nur auf Reize, sondern kann von sich aus die Initiative ergreifen
- *sozial*: strukturierte Kommunikation mit anderen Agenten ist möglich ("social abilities")

weitere Eigenschaften:

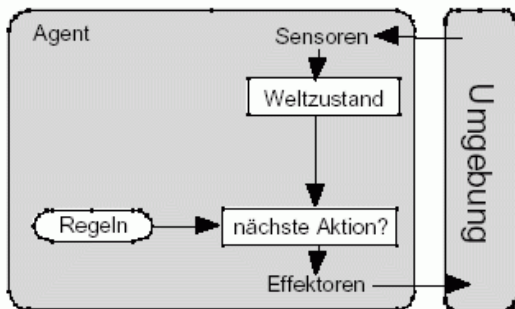
- mentale Konzepte (Wissensverarbeitung, Affekte, Ziele)
- Ziel-Orientiertheit
- "Rationalität"
- Mobilität
- Adaptivität
- "Aufrichtigkeit", "Gutwilligkeit" (?)

Agent versus Objekt

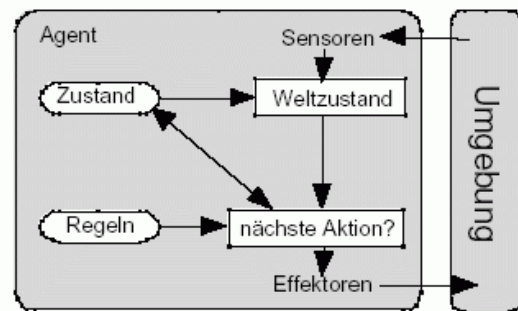
- Objekt kapselt seinen Zustand und stellt mit Methoden Schnittstellen nach außen zur Verfügung, mit denen Zustand verändert werden kann. Ein Objekt hat aber keine Kontrolle, wann diese Methoden aufgerufen werden.
- Agent kapselt Verhalten: Über die Schnittstellen nach außen (Sensoren) kommen Anfragen. Der Agent entscheidet, ob und wie er diese Anfragen bearbeitet.
- „Objects do it for free, agents for money“

Architektur-Taxonomie nach Russell/Norvig

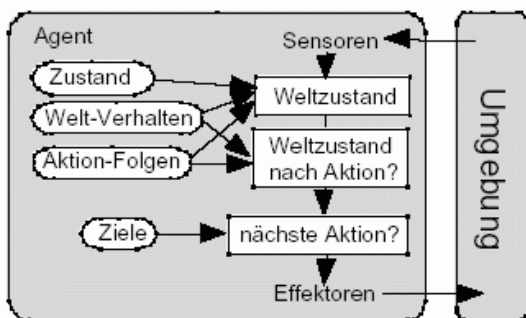
Einfacher Reflex-Agent



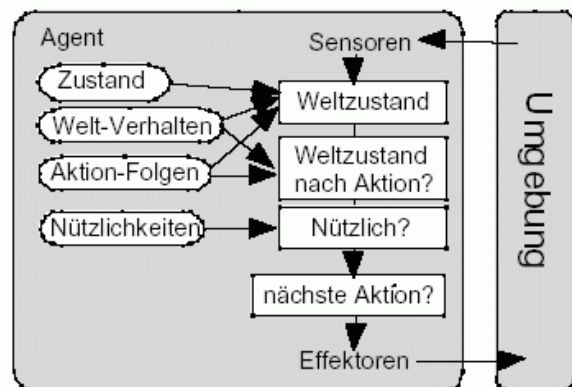
Reflex-Agent mit internem Zustand



Agent mit expliziten Zielen



Utilitaristischer Agent



Literatur:

S. Russell & P. Norvig: Artificial Intelligence - a Modern Approach, Prentice Hall, 1995

Kooperation, Netzwerke

- Wie entsteht koordiniertes Verhalten mehrerer Organismen?
- Vorbilder: Schwarmtiere, Insektenstaaten, menschliche Gesellschaft
- "*Sozionik*": Erkundung hypothetischer Gesetzmäßigkeiten, nach denen menschliche Gesellschaften funktionieren, durch agentenbasierte Modelle ("kollektive" Erweiterung von ALife)
- Anwendungen in Ökologie, Soziologie, Ökonomie
- "distributed problem solving" – Anwendung auf konkrete AI-Probleme (z.B. kooperierende Roboter)

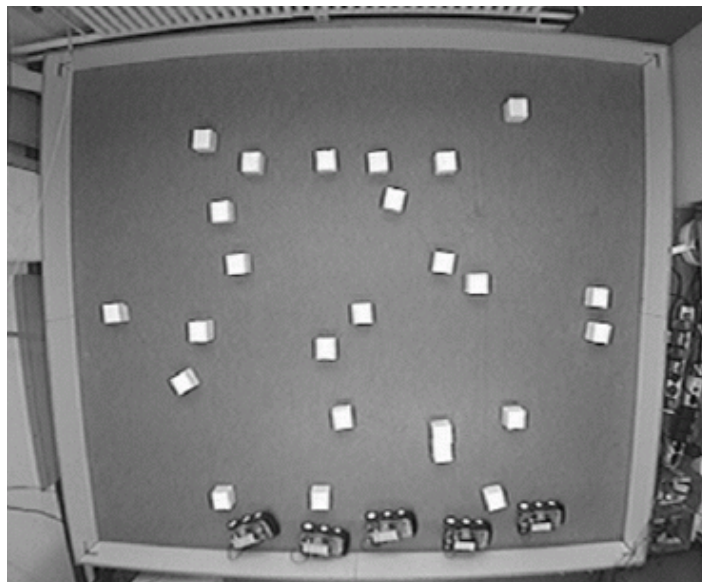
einfachste Form der Entstehung kollektiven Verhaltens:

- Anwendung fester, lokaler Regeln auf die einzelnen Agenten
- Gruppen-Verhalten als emergentes Ergebnis

Beispiel:

"Didabots" (Maris & te Boekhorst 1996, nach Pfeifer et al. 2002)

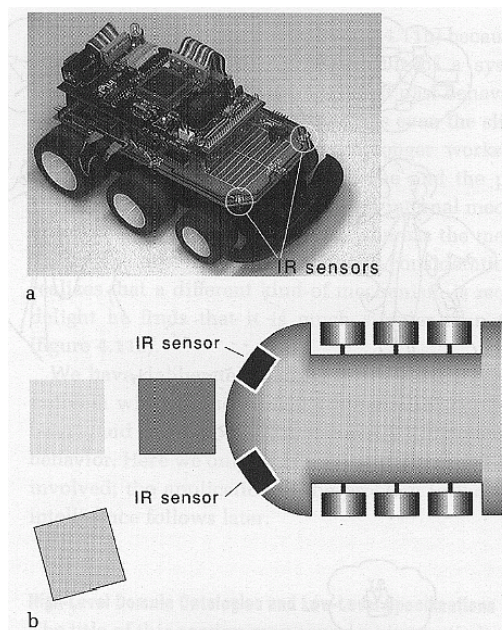
mehrere einfache Roboter in einer Arena mit Klötzchen



die Roboter sind mit 2 Infrarotsensoren ausgestattet ("Sehen" im Nahbereich).

Regeln:

- konstante Vorwärtsbewegung
- wenn Stimulus links, nach rechts drehen
- wenn Stimulus rechts, nach links drehen
- ein (kleines) Klötzchen genau in Bewegungsrichtung wird nicht wahrgenommen und weiter mitgeschoben



Ergebnis:



die Didabots "räumen auf"!

scheinbar komplexes Muster (Haufenbildung) aus einfachen lokalen Regeln

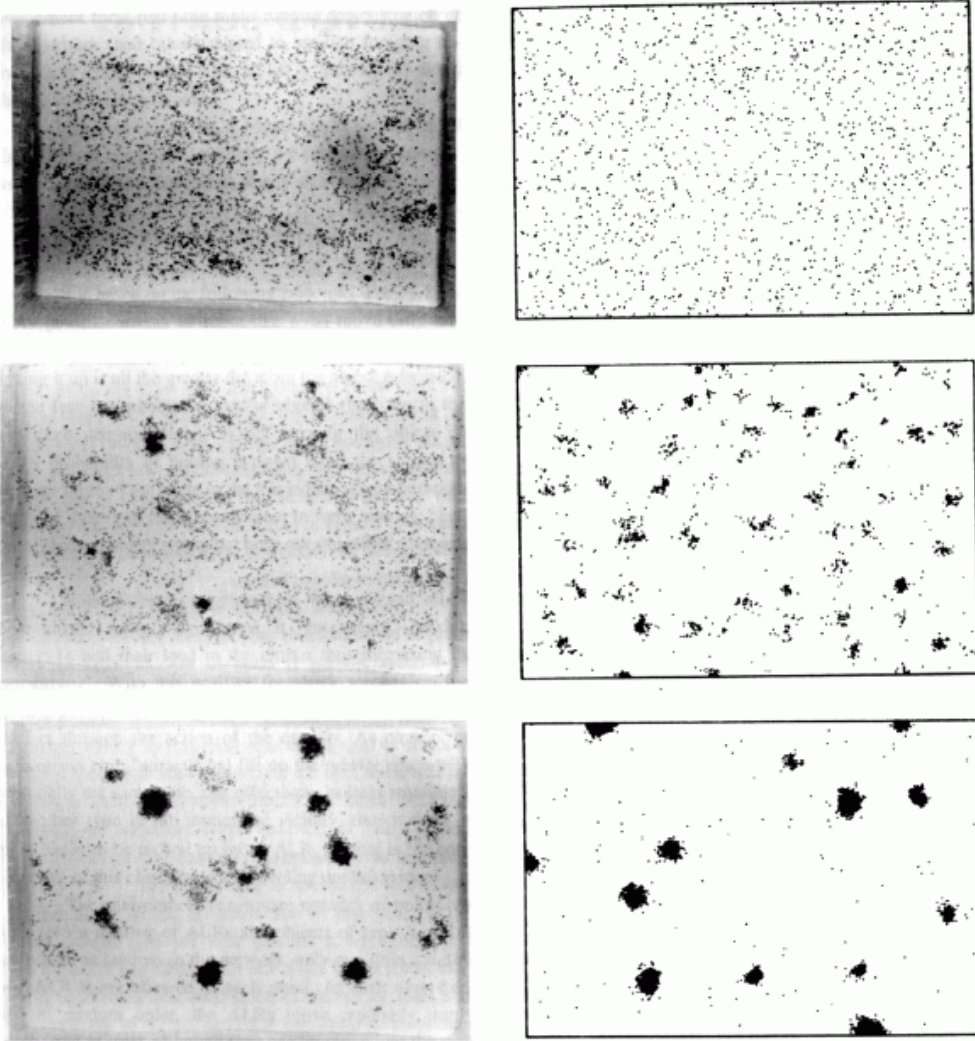
ähnlich bei Ameisen

Virtuelle Ameisen

Steuerung des Verhaltens über Wahrscheinlichkeiten des Aufnehmens und Ablegens von Objekten

- Ameise nimmt Objekt nur wahr, wenn unmittelbar davor
- Wahrscheinlichkeit des Aufnehmens am größten, wenn kein weiteres Objekt derselben Sorte in der Nähe
- für Ablegen umgekehrt

Ergebnisse:

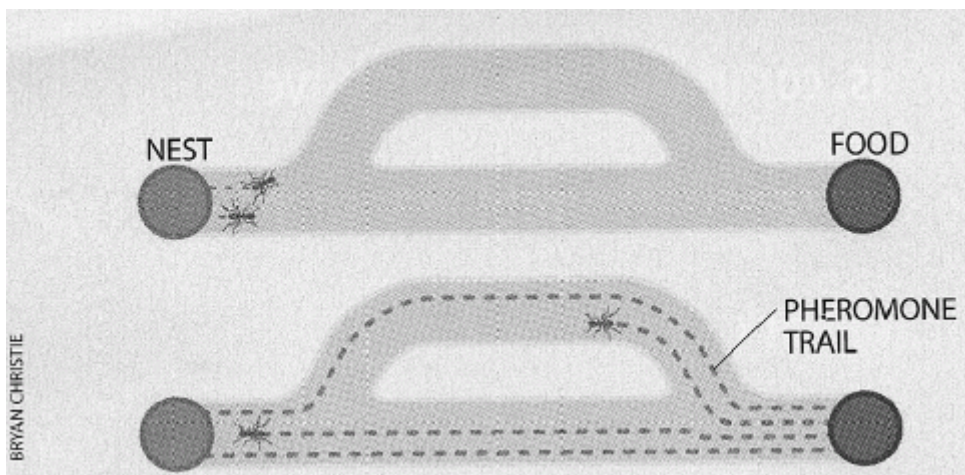


(links: echte Ameisen, rechts: Simulation)

Kommunikation von Ameisen durch "stigmergische Interaktion" (Informationsübertragung vermittelt durch die Umgebung):

Legen von Pheromonspuren

- je intensiver die Pheromonspur, desto größer die Wahrscheinlichkeit, dass weitere Ameisen diesem Pfad folgen
- dadurch Optimierung von Wegen möglich



Anwendung dieses Prinzips in Optimierungsproblemen: "Ant Algorithms", "Ant based control" (u.a. bei Hewlett-Packard und British Telecom für optimale Auslastung von Telefonnetzen)

Updaten einer "Pheromon-Tabelle" für jede Start-Ziel-Kombination in jedem Knoten durch virtuelle Ameisen, die das Netz durchwandern

Update-Formeln:
$$p_{new} = \frac{p_{old} + \Delta p}{1 + \Delta p}$$
 für den Knoten von dem die Ameise gerade kommt,
$$p_{new} = \frac{p_{old}}{1 + \Delta p}$$
 für die übrigen

Anwendungen: Traveling Salesman Problem, Job scheduling, shortest common supersequence, Graphenfärbung...

vgl. auch "ant"-Beispiel in GroIMP

anderes Beispiel: Schwarmverhalten (Vögel, Fische...),
virtuell: "Boids" (bird-like organisms)

Um komplexere "künstliche Gesellschaften" zu generieren:

- Verwendung komplexerer Agenten (mehr Regeln, interne Zustände)
- strukturierte Umwelt (die ebenfalls durch Regeln beschrieben wird)
- Lernfähigkeit
- experimentelle Simulationsumgebung, die das Testen diverser Regelsysteme ermöglicht

Beispiel: "**Sugarscape**" (Epstein & Axtell 1996)

Anspruch: Virtuelles Labor für die Sozialwissenschaften

- Modell-Universum, dessen Regeln nach Bedarf verfeinert und ergänzt werden können

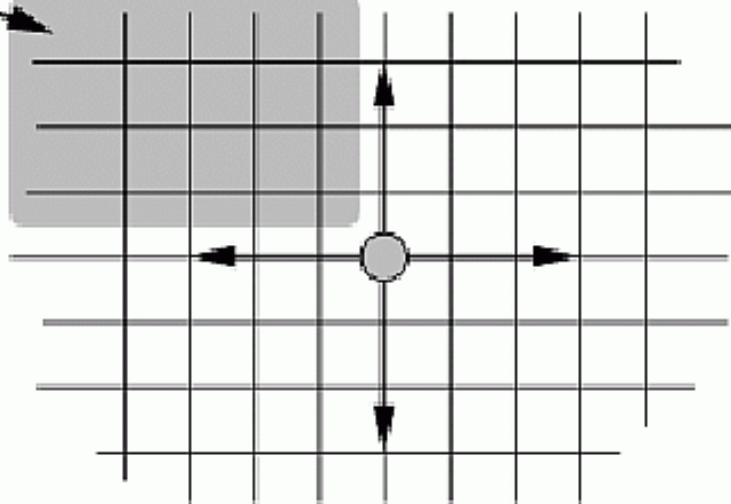
Die Umwelt:

- 2-dimensionales Gitter (50×50), Torus-Topologie
- dort wächst "Zucker" (in diskreten Mengen: 0 bis 4)
- in jeder Gitterzelle festgelegt: aktuelle Zucker-Menge und maximale Kapazität
- Zucker-Regenerationsregel: Zucker-Menge nimmt in jeder Gitterzelle linear zu mit Rate α pro Zeitschritt, bis max. Kapazität erreicht

grundlegende Regeln für die Agenten:

- Metabolismus: pro Zeiteinheit wird feste Menge Zucker verbraucht
- Agent stirbt, wenn kein Zucker mehr verfügbar
- Zuckervorrat (potenziell unbegrenzt) wird mitgeführt (Startvorrat > 0)
- endliches Blickfeld, Blick nur in die 4 Haupt-Himmelsrichtungen, Weite: für jeden Agenten zufällig bestimmt (zwischen 1 und 6 Felder)

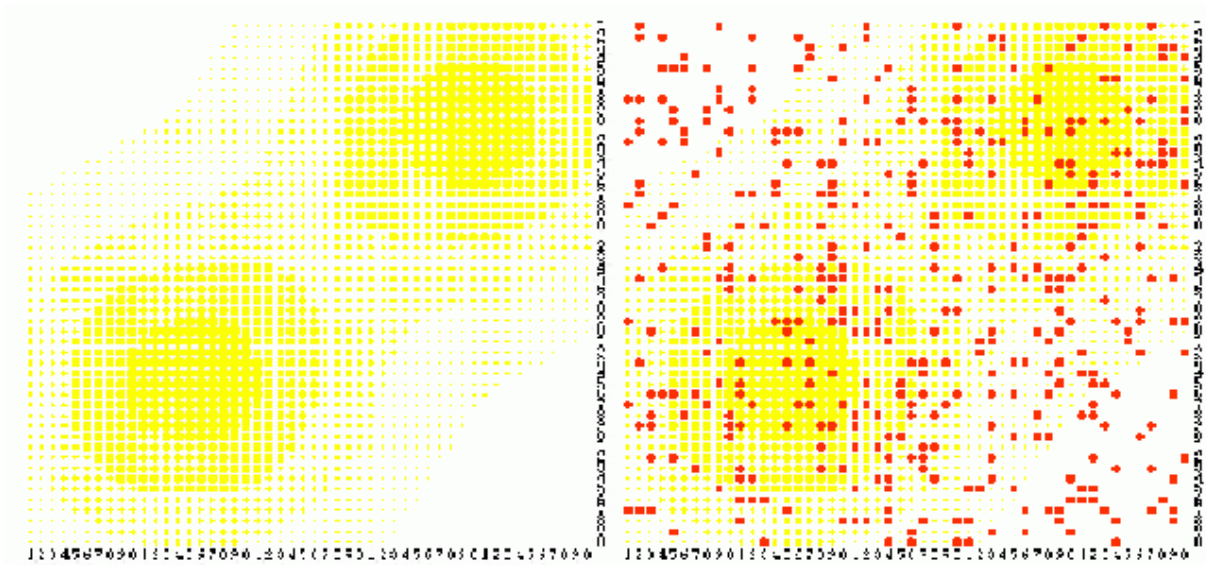
Agent cannot "see"
in diagonal directions



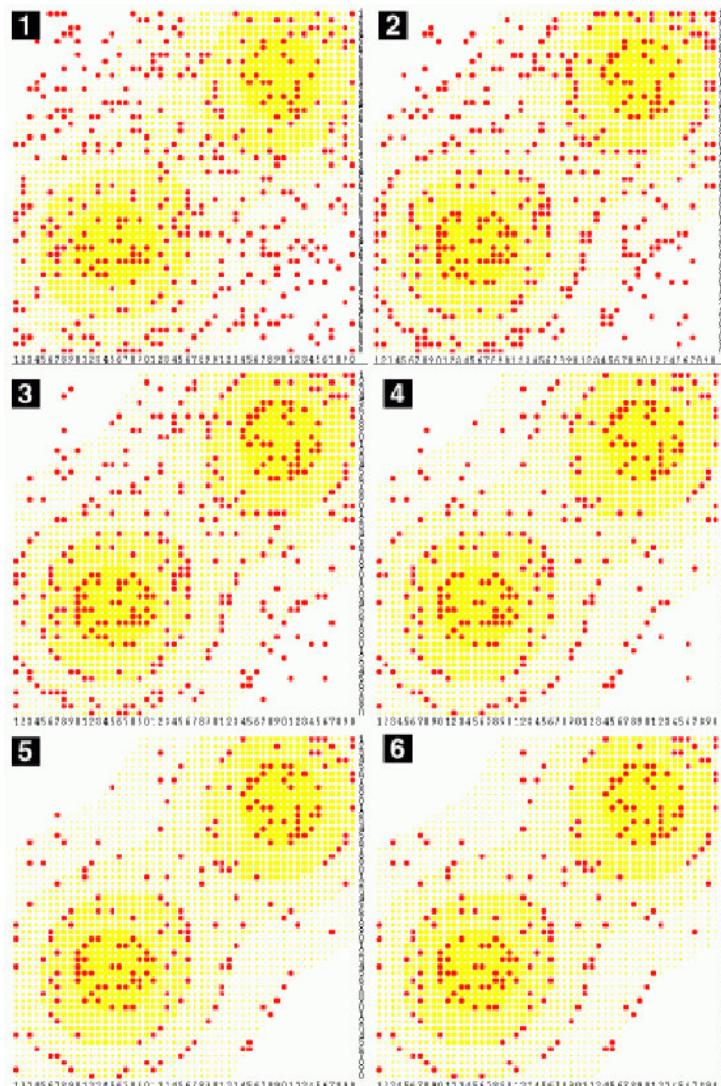
Bewegungsregel für Agenten:

- blicke soweit möglich in die 4 Blickrichtungen und suche die unbesetzten Gitterzellen mit dem meisten Zucker
- suche davon die nächste
- bewege dich dorthin
- "ernte" allen Zucker auf diesem neuen Gitterfeld

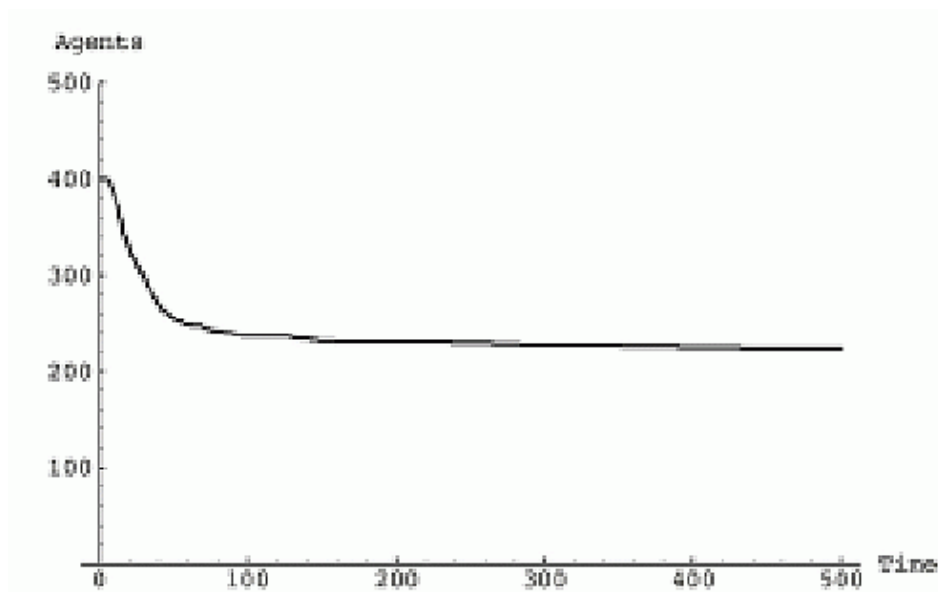
initiale Zuckerverteilung (links); initiale Verteilung der Agenten (rechts):



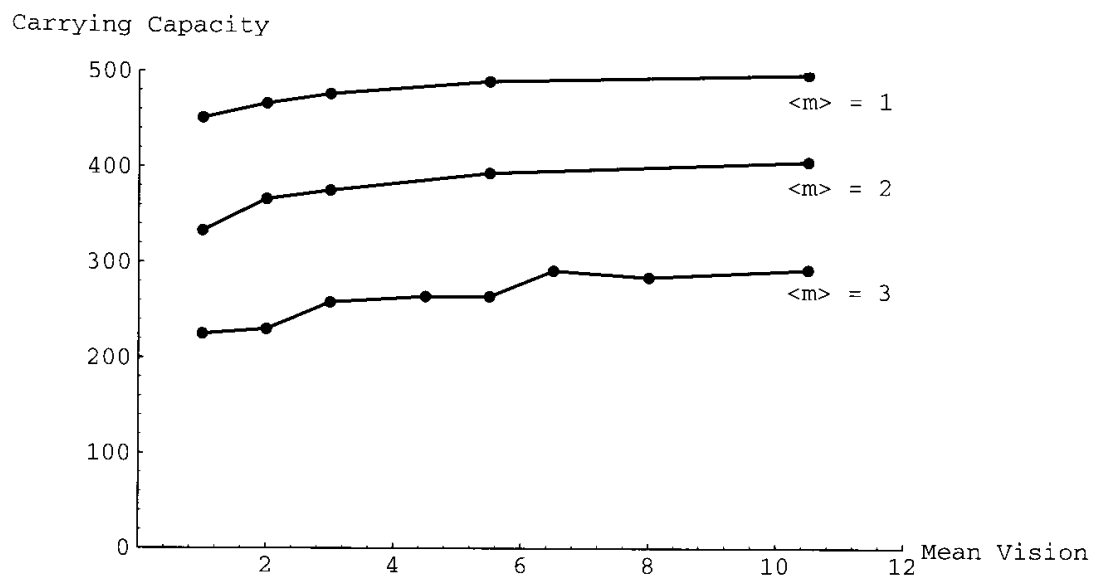
Entwicklung des Sugarscape aufgrund der obigen Regeln:



Entwicklung der Populationsgröße:



⇒ Konvergenz gegen eine "carrying capacity", die u.a. von der Wachstumsrate α des Zuckers und von der metabolischen Umsatzrate der Agenten abhängt – aber auch z.B. von der mittleren Sichtweite der Agenten:



Um Verteilung der Zuckervorräte ("Reichtum") zu studieren:
unendliches Akkumulieren von Zucker ist nicht realistisch

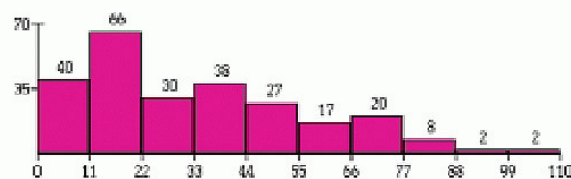
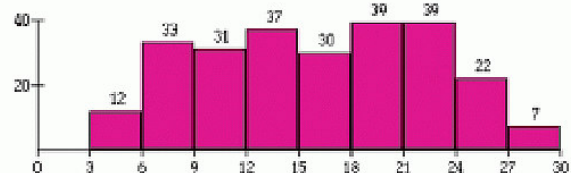
Modifikation der Regeln:

Einbau eines maximalen Lebensalters (zufällig gewählt aus Intervall $[a, b]$ für jeden Agenten); bei Tod eines Agenten wird ein neuer geboren mit zufälliger Position u. zufälligem Zuckervorrat

es entwickelt sich "ökonomische Ungleichheit" als emergente Struktur

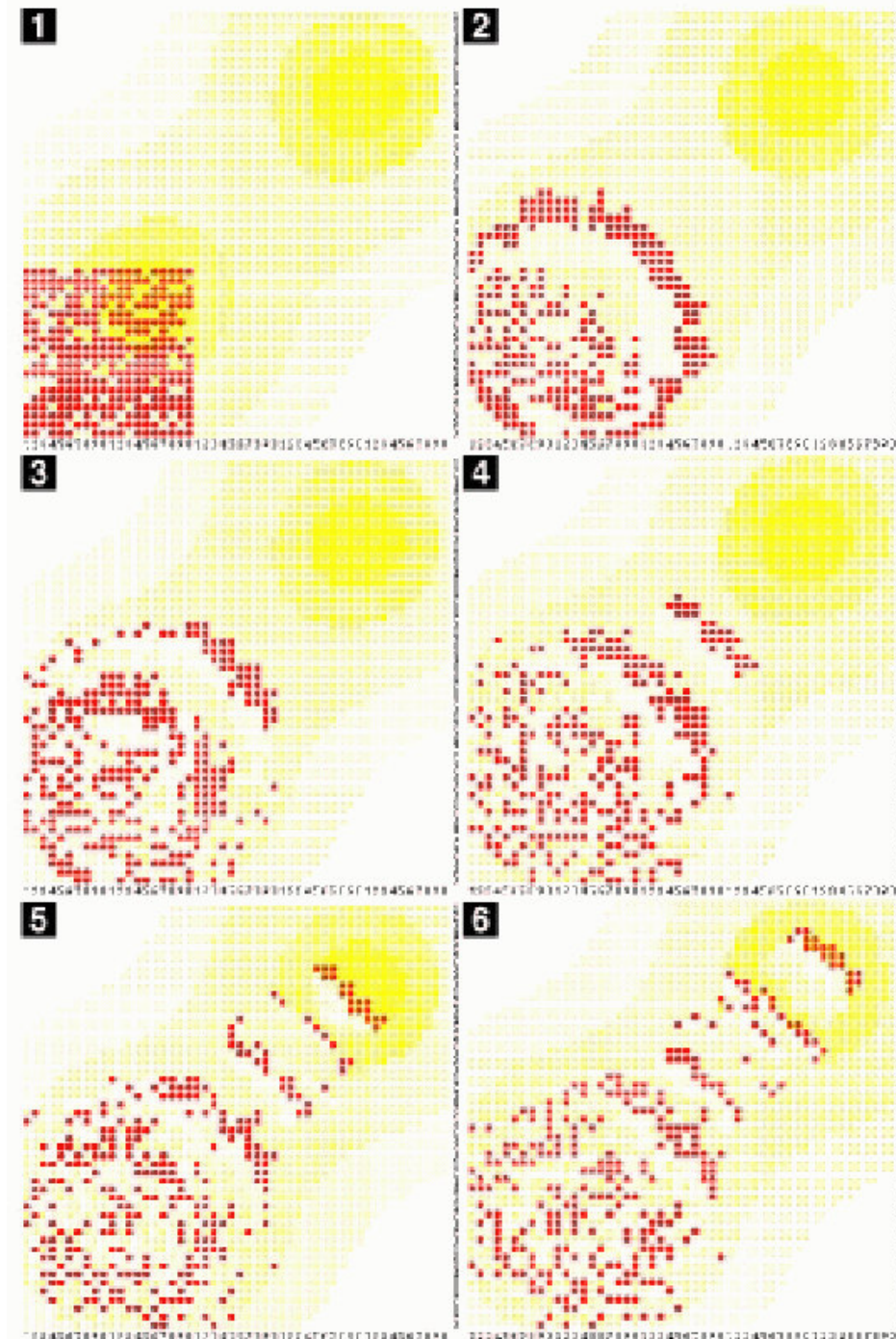
(im Beispiel ist $[a, b] = [60; 100]$; Anzahl Agenten = 250)

wealth histograms

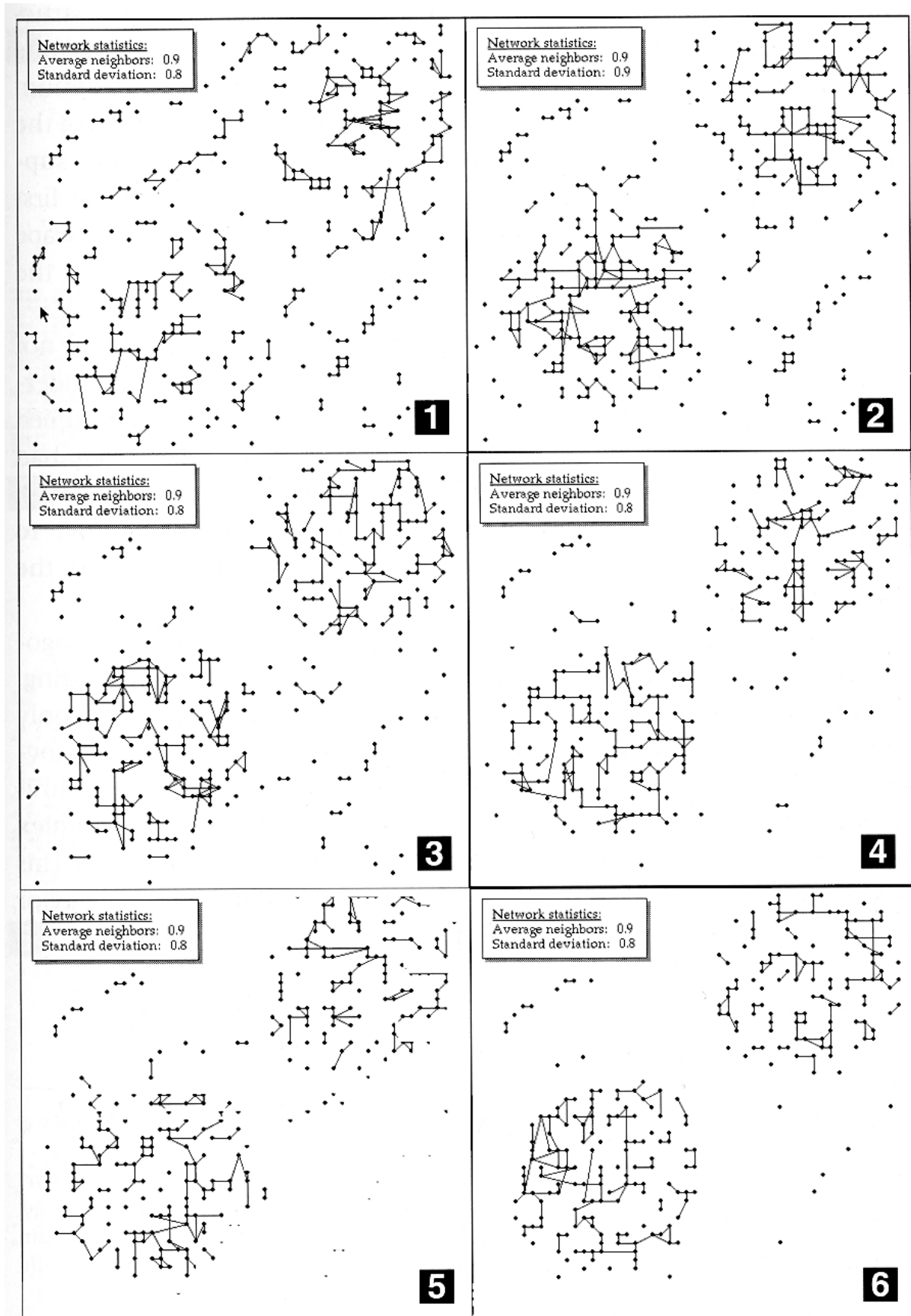


weitere emergente Strukturen:

- Migrationswellen
- diagonale Bewegungsrichtung der Population (einzelne Agenten können sich nicht diagonal bewegen!)



Aufbau von "Bekanntschafts-Strukturen" (Netzwerke früherer Nachbarn):



Einführung von sexueller Vermehrung und Vererbung:

- 2 Geschlechter von Agenten (zufällig verteilt)
- "Fertilität": Agent muss Altersgrenze überschritten und genug Zucker gesammelt haben
- Fortpflanzungsregel ("Agent sex rule"):
 - wenn fruchtbar, suche zufällig einen Nachbarn aus
 - wenn dieser auch fruchtbar *und* vom anderen Geschlecht
und einer von beiden hat eine leere Nachbarzelle, dann
mache ein Kind und positioniere dies in der leeren Nachbarzelle
 - dem Kind wird von beiden Eltern ein Zuckerbetrag mitgegeben
 - iteriere dies für alle Nachbarn.

Weitergabe genetischer Attribute durch Mendelsche Vererbung:

z.B. Sichtvermögen v und Metabolismus-Rate m

(m, v) kreuzt sich mit (M, V) :

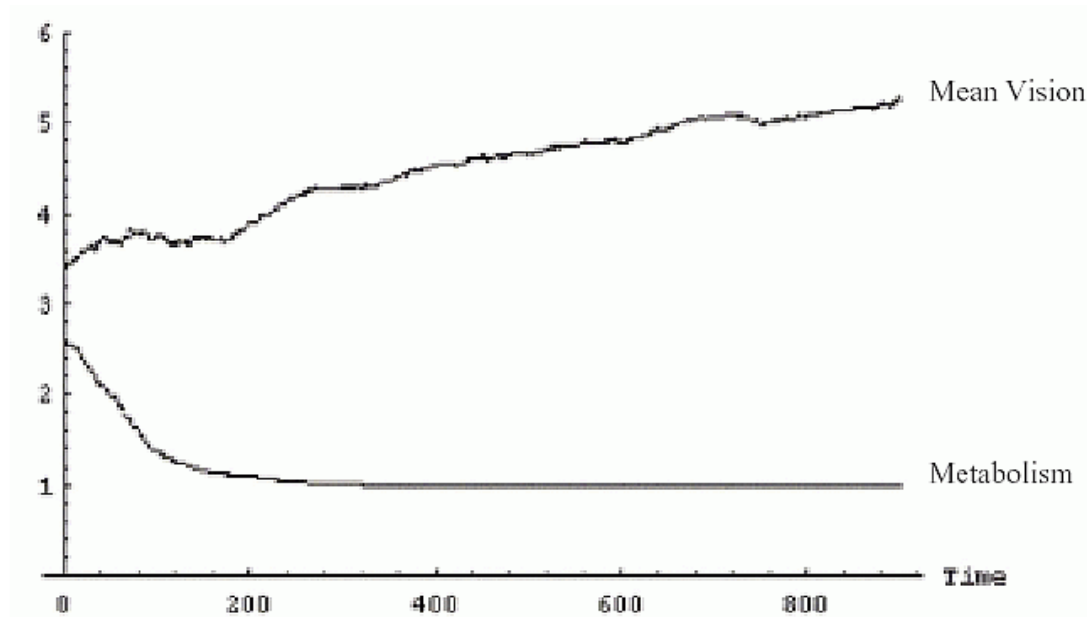
	<i>Metabolism</i>	
<i>Vision</i>	m	M
v	(m,v)	(M,v)
V	(m,V)	(M,V)

Zufallsauswahl aus den 4 Alternativen

⇒ genetischer Algorithmus (dazu später mehr)

Selektion führt zu Veränderung der Häufigkeit der genetischen Merkmale in der Population

Entwicklung von mittl. Sichtweite und metabolischer Rate unter dieser Regel:



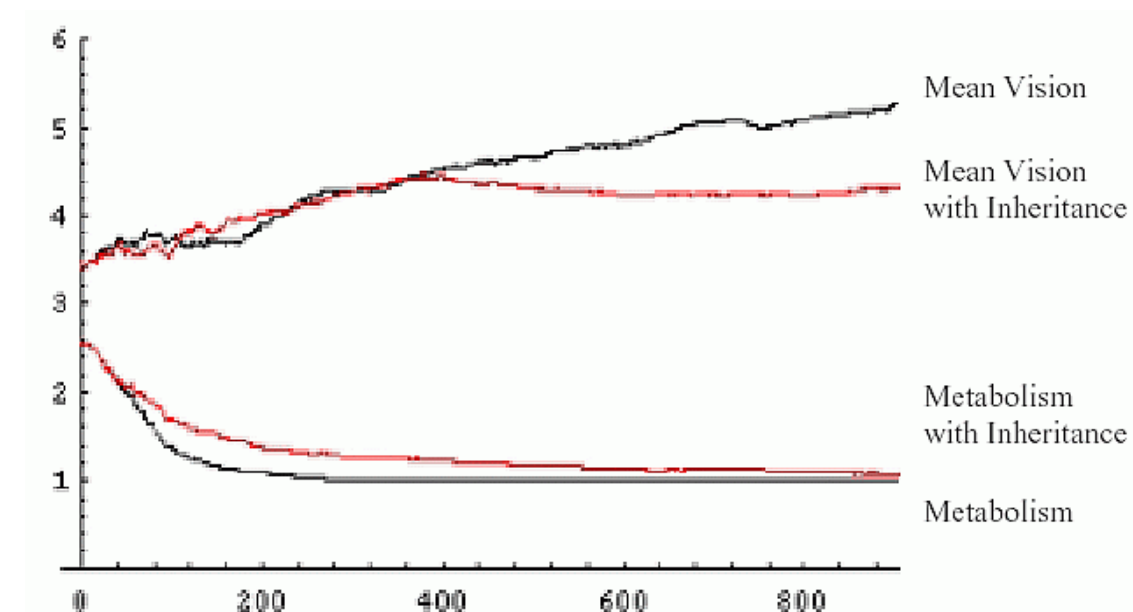
weitere Modifikation:

auch "Güter" (Zuckervorräte) können "vererbt" werden
(keine biologische, sondern kulturelle Regel!)

Vererbungsregel (*inheritance rule*):

Wenn ein Agent stirbt, wird sein Zuckervorrat zu gleichen Teilen unter all seinen lebenden Kindern aufgeteilt

Auswirkungen:



⇒ Selektionsdruck auf die Sichtweite wird offenbar durch die kulturelle Regel vermindert!

Einführung weiterer kultureller Merkmale:
codiert durch Bitstring s fester Länge
jedes Bit entspricht einem Merkmal

kulturelle Transmissionsregel ("tag-flipping"):

für jeden Nachbarn:

- eine Bitposition i wird zufällig ausgesucht
- wenn $\text{Nachbar.s}[i] == \text{selbst.s}[i]$, ändere nichts, sonst: $\text{Nachbar.s}[i]$ wird auf $\text{selbst.s}[i]$ gesetzt

(sexuelle Vermehrung bei diesem Versuch ausgeschaltet)

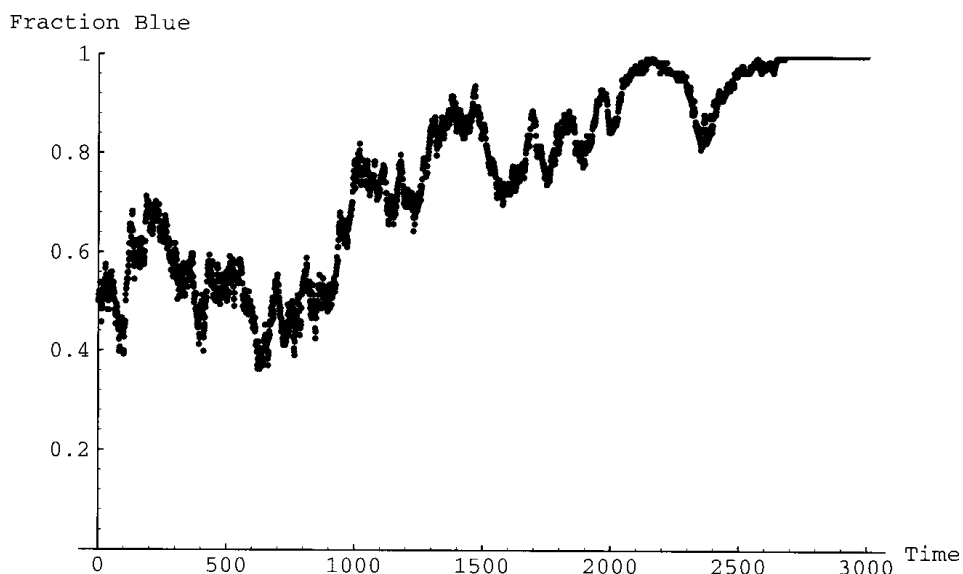
Folge:

mit der Zeit gleichen sich die Bitstrings immer mehr an

Gruppenmitgliedschafts-Definition: Agenten gehören "zu den Blauen", wenn sie mehr Nullen als Einsen in ihrem Bitstring haben, sonst "zu den Roten" (willkürl. Setzung)

Entwicklung des Anteils der "Blauen":

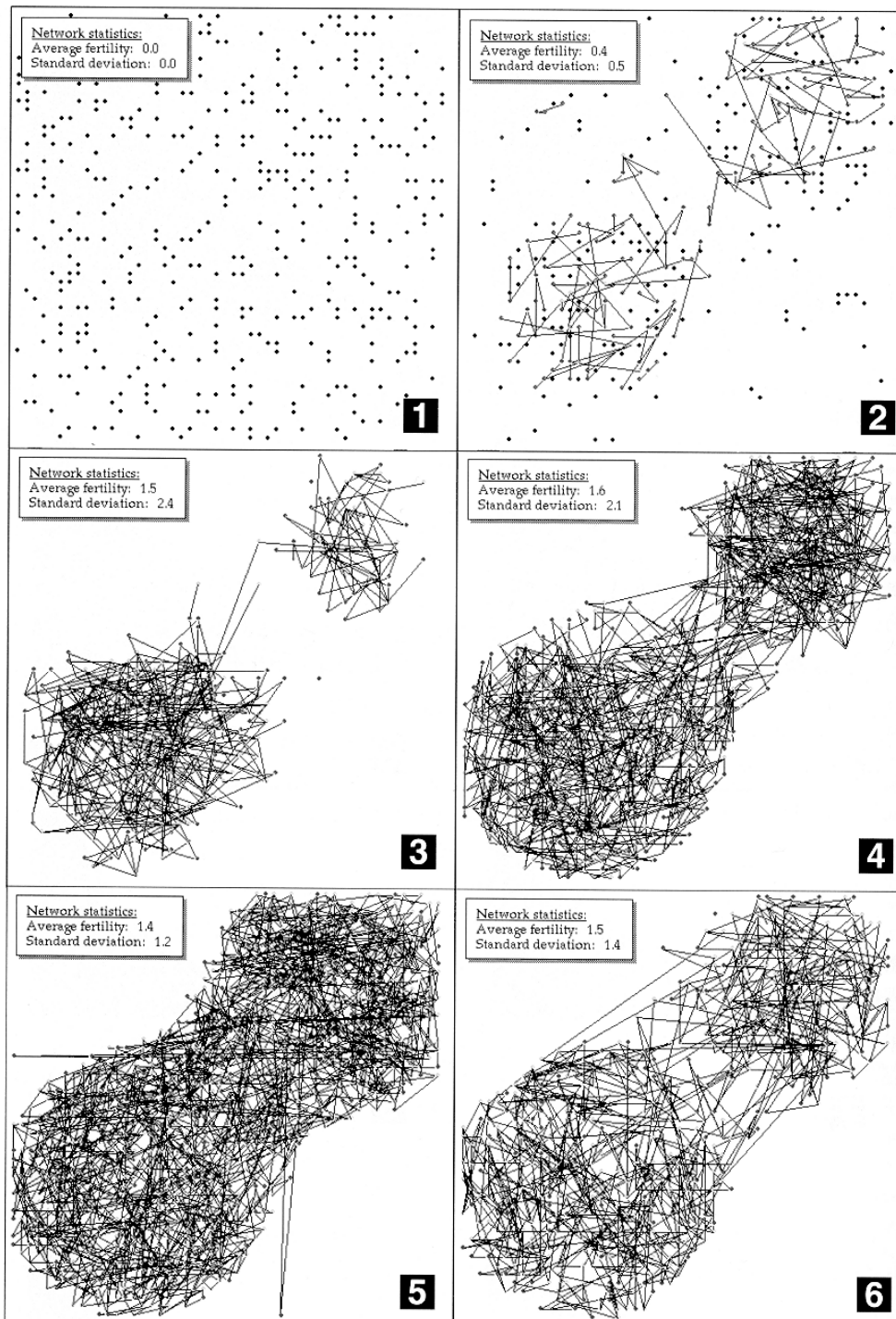
(zufälliger Ausgang – Symmetriebrechung)



Kombination mit sexueller Vermehrung erlaubt Unterscheidung von "horizontaler" und "vertikaler" (d.h. Eltern an Kinder) kultureller Transmission

weitere Studiengebiete im Sugarscape (näheres bei Epstein & Axtell 1996):

- Formierung genealogischer Netzwerke



- Freundschaftsnetzwerke ("Freunde": Nachbarn mit ähnlichen Bitstrings kultureller Merkmale)
- Kampf (um Zucker) gegen Mitglieder anderer "Stämme"
- Handel (jetzt 2 essenzielle, nachwachsende Güter: "sugar and spice", die getauscht werden können)
- Marktentwicklung, Kreditverteilung
- Auswirkung wechselnder kultureller Präferenzen für Güter
- Handelsnetzwerke
- Auswirkung von Umweltverschmutzung ("pollution" entsteht durch Ernten und/oder durch Konsumption von Zucker)
- Seuchenausbreitung
- Immunsysteme

USW. USW. ...