

Evolution kooperativen Verhaltens

Ein Software-Projekt – Gesamtdokumentation

Inhaltverzeichnis

Vorwort	2
Pflichtenblatt	3
<i>Ziel</i>	3
Das zu lösende Problem	3
Grundlegende Entwurfsregel	4
Vorgehen	4
<i>Produktfunktionen</i>	4
Allgemeine Beschreibung	4
Eingabe	6
Verarbeitung	6
Ausgabe	7
<i>Bedienoberfläche</i>	7
Bildschirmlayout	7
Darstellung der Welt	7
Interaktive Steuerung der Simulation	7
Protokollierung der Ergebnisse	8
<i>Testszenarioszenarien</i>	8
Entwurf	8
<i>UML-Diagramm</i>	8
<i>Entwurf der grafischen Oberfläche</i>	9
Das Steuerungsfenster	9
Syntax der Textdatei für die Eingabeparameter	9
Farbkodierung der Strategien	9
Realisierung	10
<i>Klassendiagramm im BlueJ-Format</i>	10
<i>Das Programm</i>	10
Programmtext	10
Algorithmus der Nachbarnauswahl	10
Algorithmus der zufälligen Mutationen	11
<i>Ergänzende Dokumentation zur Version 3</i>	11
Problem und Ziel	11
Lösungsvorschlag	12
Umsetzung	12
Abnahme	13
<i>Deterministische makroskopische Simulation</i>	13
<i>Stochastische mikroskopische Simulation</i>	13
Experimente	14
<i>Perfekte Kommunikation ($f=0$), kleine Umgebung ($k=1$)</i>	14
<i>Wirkung der Globalisierung (Vergrößerung der Umgebung)</i>	14
<i>Fehlerhafte Interaktionen</i>	15
<i>Entstehung von Kooperation in feindseliger Umgebung</i>	15
Schlussfolgerungen	17
<i>Biologie</i>	17
<i>Wirtschafts- und Gesellschaftswissenschaften</i>	18
Literaturverzeichnis	19
<i>Ergänzende Dokumentationen</i>	19
<i>Quellen</i>	19

Vorwort

Ziel ist die Entwicklung eines *Leitprojekts* für die Lehrveranstaltung *Problemlösen* (ET540) im Master-Studiengang *Systems Design & Production Management (SDPM)*.

Grundsätzlich geht es bei dem Projekt um die Frage (von Pierer, Oetinger, 1997): „Wie kommt das Neue in die Welt?“ Wir fragen uns, wie sich die Kreativität mit Computerunterstützung steigern lässt. Die Antwort: Mit kreativen Algorithmen. Wir kreieren Kreativität. Demnach sind zwei Ebenen der Kreativität zu unterscheiden:

1. Das Programm realisiert einen Evolutionsprozess. Dieser Sachverhalt wird durch das Oxymoron „*automatisierte Kreativität*“ treffend beschrieben.
2. Die Software-Konstruktion legt die Evolutionsstrategie des Prozesses fest. Hier ist menschliche Kreativität gefragt.

Frühe Versuche auf dem Gebiet starteten damit, zwei Evolutionsmechanismen auf technisch-wirtschaftliche *Optimierungsprobleme* zu übertragen: *Mutation* und *Selektion* (Rechenberg, 1973).

Und so funktionieren die Mutations-Selektions-Verfahren: Man beginnt mit einem Startpunkt für die Lösung. Aus der jeweils gültigen Lösung, dem *Aufhaltungspunkt*, wird durch zufällige Variation ein *Mutationspunkt* ermittelt. Ergibt sich eine Verbesserung der *Zielgröße*, wird der Mutationspunkt zum neuen Aufenthaltspunkt, ansonsten wird der Mutationspunkt verworfen (Selektion). Nach mehreren vergeblichen Verbesserungsversuchen wird der Prozess abgebrochen¹.

Neuere Evolutionsverfahren gehen einen Schritt weiter: Anstatt immer nur einen Lösungspunkt zu betrachten, geht man auf eine ganze *Population* von Lösungen über. Dadurch lassen sich weitere Mutationstechniken der Natur übernehmen, beispielsweise das *Crossing-Over* (Holland, 1992; Michalewicz/Fogel, 2000; Ayala, 1979).

Außerdem muss nun nicht mehr eine starre Zielfunktion zur Bewertung der Lösungen vorausgesetzt werden. Die *Bewertungsfunktion* darf von der Umgebung und von der Population selbst abhängig sein. Da mehrere und weit voneinander abweichende Lösungen in der Population auftreten können, lässt sich die Gefahr vermindern, auf ein nur lokales Optimum zuzulaufer und dabei das globale Optimum zu übersehen.

Wegen der Unbestimmtheit des Ergebnisses gewinnen die *Visualisierung des Evolutionsprozesses* und die *interaktive Steuerung* desselben an Bedeutung.

Die Vorgehensweise ist auf alle möglichen technischen und wirtschaftlichen Fragestellungen anwendbar: Immer geht es darum, dem Programm einen Freiraum einzuräumen, so dass es die (nahezu) optimale Lösung eines Problems in einem interaktiv gesteuerten *Evolutionsprozess* bestimmen kann².

¹ Auf der CeBit 1991 wurde das Projekt „Computersimulation und Optimierung von Fahrzeugbewegungen in landwirtschaftlichen Betrieben“ vorgestellt. Die Ermittlung der (nahezu) optimalen Fahrzeugbewegungen unter der Maßgabe möglichst geringer Lenkbewegungen geschieht bei dem Simulationsprogramm mit einer solchen Evolutionsstrategie (Mutations-Selektionsverfahren).

² Beispiele sind der Filterentwurf, die Gestaltung von widerstandsarmen Rohrumlenkungen, die optimale Regelung, die Optimierung von Motoren, die Fabrikplanung, die Routenplanung und viele andere.

Wie kommt man zu einem guten Evolutionsverfahren? Drauflosprogrammieren ohne Plan ist der direkte Weg in die Sackgasse. Kreativitätstechniken sollen frühzeitige Blickverengung vermeiden helfen. In diesem Sinne rechne ich die *Vorgehensmodelle der Software-Konstruktion* zu den modernen Heuristiken des Problemlösens. Das hier beschriebene Projekt habe ich nach einem einfachen vierstufigen Vorgehensmodell abgewickelt.

1. *Analyse*. Anforderungserfassung. Das ist der Moment der anfänglichen Ideenfindung. Ergebnis ist das *Pflichtenblatt*.
2. *Entwurf*. Erstellen einer grafischen Darstellung der Programmstruktur in der *Unified Modeling Language* (UML). In diesem Schritt werden die Mutationsstrategien, die Bewertungsfunktion, die Selektionsmechanismen sowie die Steuerung und Visualisierung festgelegt, soweit das nicht bereits im *Pflichtenblatt* geschehen ist. Die zentralen Abläufe und Algorithmen werden gegebenenfalls im Pseudocode unter Benutzung von Elementen der ausgewählten Programmiersprache (hier: Java) beschrieben.
3. *Realisierung*. Erstellen des lauffähigen Programms, Codierung. Die Kreativität richtet sich nun auf die Gestaltung der zentralen Algorithmen unter der Zielsetzung Lesbarkeit, Korrektheit und Effizienz.
4. *Abnahme*. Durchführung der Tests. Prüfung der Dokumentation.

Projekt-Historie: Die Dokumentation des Projekts gilt für die erste Programmversion. Eingearbeitet sind die geringfügigen Änderungen der Version 2. Die Änderungen hin zur vorliegenden dritten Version werden in einem eigenen Unterabschnitt zur Realisierung behandelt.

Pflichtenblatt

Ziel

Das zu lösende Problem

Ein zentrales Problem der Verhaltensforschung (Biologie, Soziologie, Soziobiologie) ist die Erklärung der Evolution kooperativen Verhaltens. An diesem Beispiel lassen sich sehr gut die Notwendigkeit und die Vorgehensweise der Simulation demonstrieren. Außerdem liefert das Beispiel nebenbei grundlegende Erkenntnisse darüber, was erfolgreiches soziales Handeln auszeichnet.

Relativ einfach lässt sich der Selektionsprozess simulieren. Er beantwortet die Frage, welche Strategien unter einer Reihe von vorgegebenen Strategien erfolgreich sind und sich durchsetzen werden. Ein solcher Selektionsprozess ist im berühmten Computerturnier von Robert Axelrod (1987) auf der Grundlage des Gefangenendilemmas inszeniert worden.

Im Web-Kurs „Umweltsimulation mit Tabellenkalkulation“ ist eine ökologische (also makroskopisch und deterministisch angelegte) Simulation des Turniers realisiert.

Weder Axelrods Turnier noch die ökologische Simulation können erklären, wie kooperatives Verhalten *entsteht*. Zwar können kooperative Strategien, haben sie sich erst einmal etabliert, die selbstsüchtigen Strategien verdrängen. Aber es gelingt ihnen unter den gegebenen Randbedingungen nicht, sich zu etablieren.

Frühe Versuche auf der Grundlage einer mikroskopischen statistischen Simulation der Evolution wurden abgebrochen mit der Erkenntnis, dass zum Entstehen kooperativen Verhaltens weitere Evolutionsmechanismen nötig sind. Einer dieser Mechanismen ist Ortsgebundenheit der Individuen und Strategien, die *Territorialität*.

Grundlegende Entwurfsregel

Das Modell soll zeigen, wie unter extrem einfachen Modellannahmen und minimalen Voraussetzungen interessantes Neues entsteht: Allein der Zufall im Verein mit selektionswirksamer Notwendigkeit generiert nützliche Information.

Robert Axelrod (1997, S. 4 f.) formuliert das Wesen der hier anzuwendenden Simulation so: “Although agent-based modeling employs simulation, it does not aim to provide an accurate representation of a particular empirical application. Instead, the goal of agent-based modeling is to enrich our understanding of fundamental processes that may appear in a variety of applications. This requires adhering to the KISS principle, which stands for the army slogan ‘keep it simple, stupid.’”

Vorgehen

Die mikroskopische statistische Simulation der Evolution kooperativen Verhaltens wird erneut aufgegriffen. Dabei soll jetzt die Ortsgebundenheit berücksichtigt werden. Das geschieht mit dem Ansatz der *Kugelspiele* (Eigen/Winkler, 1975). Durch eine Kombination der Grundideen des Computerturniers mit denen der Kugelspiele sollte es möglich sein, die Evolution kooperativen Verhaltens zu modellieren (Nowak, May und Sigmund, 1995).

Produktfunktionen

Allgemeine Beschreibung

Die Simulation der Evolution findet auf einer Matrix `world` mit $n \times n$ Plätzen statt. Jeder dieser Plätze kann leer sein oder eine Referenz auf ein Objekt vom Typ `Strategy` enthalten. Diese Wesen haben eine *Strategie* (einen *Charakter*) `strategy`, ein Gedächtnis `memory` und eine Lebenskraft `strength`. Die Strategie legt fest, wie beim Zusammentreffen mit einem anderen Wesen in Abhängigkeit von der Erfahrung mit jenem Wesen zu reagieren ist: Kooperieren oder Betrügen (Defektion). Der Gewinn je Zusammentreffen ergibt sich aus der Spielmatrix des Gefangenendilemmas. Die *Lebenskraft* ist gleich dem gewichteten Mittelwert der in der Vergangenheit erlangten Punkte je Zusammentreffen (gleitende Mittelwertbildung).

Für jeden Platz (x, y) der Matrix wird – unter Zugrundelegung der Summennorm – die k -Umgebung $U_k(x, y)$ definiert durch

$$(x', y') \in U_k(x, y) \text{ genau dann, wenn } |x'-x|+|y'-y| \leq k \tag{1}$$

Alternativ dazu wird die Maximumnorm zu Grunde gelegt:

$$(x', y') \in U_k(x, y) \text{ genau dann, wenn } \max\{|x'-x|, |y'-y|\} \leq k \tag{2}$$

In der nebenstehenden Matrix sind die 1-Umgebungen der Felder $(2, 2)$ und $(6, 6)$ grau markiert. Bei $U_1(2, 2)$ wurde die Summennorm zu Grunde gelegt, bei $U_1(6, 6)$ die Maximumnorm. (Merkregel zur Indizierung: **Z**eile zuerst, **S**palte später.)

	0	1	2	3	4	5	6	7	8	9
0										
1										
2			■	■	■					
3				■						
4										
5										
6							■	■	■	
7								■	■	■
8										
9										

Für die Simulation wird die Maximumnorm festgelegt. Der Wert k ist ein Eingabeparameter. Unter Umgebung eines Feldes (x, y) ist die durch diese Festlegungen definierte Umgebung $U_k(x, y)$ zu verstehen.

Die Welt wird rund gemacht: Die linken Nachbarn von linken Randpunkten sind die rechten Randpunkte und umgekehrt. Dasselbe gilt für die oberen und unteren Randpunkte. Die Welt ist nun schlauchförmig.

Die Simulation besteht aus einer Folge von *Spielzügen*. In jedem Spielzug wird ein Feld (x, y) der Matrix nach dem Zufallsprinzip ausgewählt.

Kampf. Ist das Feld nicht leer, wird aus der Umgebung nach dem Zufallsprinzip als Kontrahent ein weiteres Wesen ausgewählt. Für jedes Wesen wird die Reaktion, Kooperieren (kurz: K) oder Defektion (kurz: D), gemäß seiner Strategie erzeugt. Die Gewinnmatrix legt den Gewinn bzw. den Verlust für jeden der Kontrahenten fest. Dieser Gewinn wird zur (jeweils um den Dämpfungsfaktor q verminderten) vorhandenen Lebenskraft addiert. Das ist eine *gleitende Mittelwertbildung* des Gewinns mit dem Dämpfungsfaktor q .

Voreingestellt ist die (leicht modifizierte) Gewinnmatrix von Axelrods Computerturnier:

		Aktion des Gegenübers	
		K	D
Nutzen der Aktion	K	2	-1
	D	4	0

Die Werte der Matrix können durch die Eingabe verändert werden. Die Werte der Matrix werden mit KK, KD, DK und DD bezeichnet. Bei der Gewinnmatrix des *Gefangenendilemmas* sind grundsätzlich die folgenden Relationen erfüllt:

$$KD < DD < KK < DK,$$

$$(KD + DK) / 2 < KK.$$

Geburt. Ist das ausgewählte Feld leer, kann es in einem Erzeugungsprozess durch ein neues Wesen besetzt werden. Der Erzeugungsprozess wird mit der Wahrscheinlichkeit b gestartet. Im Erzeugungsprozess wird ein mögliches Elter aus der Umgebung des leeren Feldes zufällig ausgewählt. Überschreitet dessen Lebenskraft einen gewissen Schwellwert, so wird ein neues Wesen erzeugt, andernfalls nicht. Das neue Wesen erhält – bis auf Punktmutationen, die mit der Wahrscheinlichkeit p eintreten – dieselbe Strategie wie das Wesen aus seiner Umgebung. Das Elter gibt seine halbe Lebenskraft an das Kind ab.

Tod. Unterschreitet die Lebenskraft des (ursprünglich) ausgewählten Wesens den Schwellwert, wird es mit der Wahrscheinlichkeit d gelöscht.

Der *Schwellwert für Geburt und Tod*: In den Betriebsmodi 0 und 1 wird als Existenzminimum der arithmetische Mittelwert der Lebenskraft aller Individuen der Umgebung des Wesens gebildet. In den Betriebsmodi 2 und 3 wird das Existenzminimum auf $(KD + DK) / 2$ festgelegt.

Anfangsbelegung: In den Betriebsmodi 0 und 2 werden die Felder zufällig mit Individuen belegt, deren Strategien und Gedächtnis ebenfalls zufällig sind. Die statistischen Eigenschaften dieses Prozess der zufälligen Erzeugung werden durch die Eingabeparameter festgelegt. In den anderen beiden Betriebsmodi (1 und 3) werden die Basischaraktere jeweils mit gleichen Wahrscheinlichkeiten ausgewählt.

Eingabe

Auflistung der Eingabeparameter (nicht kursiv geschrieben, da Programmvariable):

mode	Betriebsmodus 0, 1, 2 oder 3
n	Größe der „Welt“ (integer): ein Spielfeld aus $n \times n$ Feldern.
a	Wahrscheinlichkeit für Startbelegung eines Feldes
as	Vorbelegungswahrscheinlichkeit je Bit der Strategie (Wahrscheinlichkeit, dass der Wert des Bits 1 ist)
k	Umgebungsparameter. Legt die k-Umgebung fest
h	Anzahl der Spielzüge bis zum Halt
b	Geburtswahrscheinlichkeit
d	Sterbewahrscheinlichkeit
p	Mutationswahrscheinlichkeit je Bit (Bitfehlerwahrscheinlichkeit)
f	Wahrscheinlichkeit für Interaktionsfehler
q	Dämpfungsfaktor für Lebenskraft
j	Länge des Gedächtnisses
r	Wahrscheinlichkeit für eine 1 bei Vorbelegung der Historie (s. Verarbeitung)
kk, kd, dk und dd	sind die (vorgebbaren) Parameter der Gewinnmatrix.

Verarbeitung

Gedächtnis und zur Strategie eines Wesens werden als Ganzzahlvariablen angelegt und als Bitmuster interpretiert. Im Programm werden diese Variablen grundsätzlich mit Bitoperationen manipuliert. Bei den Bits steht 0 für Kooperation und 1 für Defektion.

Jedes neu geschaffene Wesen bekommt eine eindeutige Identifikationsnummer mit. Trifft ein Wesen auf ein anderes, ihm noch unbekanntes Wesen, so legt es für dieses neue Wesen ein Gedächtnis an. Dieses History-Byte wird in der Dokumentation mit h bezeichnet.

Historie. Historien sind zu einer linearen Liste verkettbar. Durch j wird die Gedächtnistiefe festgelegt. Die j niedrigstwertigen Stellen der History-Bytes halten die Aktionen des betreffenden Gegners aus den letzten j Treffen fest. Offenbar ist $0 \leq h < 2^j$. Für die hier durchgängig auf 3 beschränkte Gedächtnistiefe ist $0 \leq h < 8$.

Die Auswahl der Bits einer Ganzzahlvariablen wird in dieser Dokumentation mittels Array-Notation geschrieben: $h[0]$ ist das niedrigstwertige Bit und gibt die Reaktion des Gegners im letzten Treffen an. Ist $h[0]$ gleich 0, so heißt das, dass der Gegner beim letzten Mal kooperativ war, und $h[0]$ gleich 1 heißt, dass er betrogen hat. In derselben Weise bezeichnet $h[1]$ die Aktion des Gegners im vorletzten und $h[2]$ die Aktion im vorvorletzten Treffen.

Als Beispiel nehmen wir ein h mit dem Dezimalwert 6. Diese Zahl hat die Dualzahldarstellung $(110)_2$, was dem Bitmuster 00000110 entspricht. Also ist $h[0]$ gleich 0; $h[1]$ und $h[2]$ sind jeweils gleich 1. Das bedeutet, dass der entsprechende Gegner beim letzten Treffen kooperiert und bei den beiden Treffen davor betrogen hat.

Eine neue Historie für einen noch unbekanntes Gegner wird folgendermaßen erzeugt: Jedes relevante Bit erhält mit der Wahrscheinlichkeit r den Wert 1, ansonsten den Wert 0. Nach jedem Treffen werden die Bits der Historie h unter Wahrung der Gedächtnisbegrenzung nach links verschoben und die Reaktion des Gegenübers wird auf $h[0]$ abgespeichert.

Strategie. Die Strategie wird durch den Wert einer Ganzzahlvariablen s festgelegt, die wiederum als Bitmuster zu interpretieren ist. Ist h die Historie des momentanen Gegenspielers, dann wird die Aktion $s[h]$ gewählt. Beispiel: Sei die Strategie s gegeben durch $s = 170 = (10101010)_2$ und die Historie h für das Gegenüber durch $h = 5 = (101)_2$, dann ist die sich daraus ergebende Aktion $s[h]$ gleich $s[5]$, also gleich 1 (eine Defektion).

Interaktionsfehler. Sei z eine Zufallszahl, die mit der Wahrscheinlichkeit f den Wert 1 annimmt, dann ist die Aktion bestimmt durch $s[h]*(1-z)+(1-s[h])*z$. In Worten: Das Aktionsbit wird invertiert, falls $z=1$, ansonsten bleibt es unverändert.

Wie sehen die bekannten Strategien (Basischaraktere) aus? Betrachtet wird hier durchgängig der Fall $j=3$, es werden also nur die drei letzten Begegnungen erinnert. Es gibt 256 mögliche Charaktere bzw. Strategien. Neue Historien sind gleich null, es wird also $r = 0$ gesetzt. Die Strategien der Basischaraktere sind in der folgenden Tabelle definiert.

Die Strategien der Basischaraktere

h	S _{Tit for Tat}	S _{Vergelter}	S _{Immer K}	S _{Pawlow}	S _{Immer D}	S _{Hinterlist}
000	0	0	0	0	1	1
001	1	1	0	1	1	0
010	0	1	0	1	1	1
011	1	1	0	0	1	0
100	0	1	0	1	1	1
101	1	1	0	0	1	0
110	0	1	0	0	1	1
111	1	1	0	1	1	0

Näheres über die Strategien ist im der Lektion „Kooperation unter Egoisten: Axelrods Computerturnier“ des Web-Kurses „Umweltsimulation mit Tabellenkalkulation“ zu finden.

Die Strategien *Vergelter* und *Pawlow* werden nicht korrekt dargestellt. Das sind Strategien, die eigentlich zusätzlich ein Gedächtnis für ihr eigenes Verhalten benötigen. Der Vergelter ist hier mit Verzögerung doch versöhnlich. Die Strategie *Zufall* lässt sich gar nicht darstellen, es sei denn, die Interaktionsfehlerwahrscheinlichkeit f wird characterspezifisch variabel gemacht. Auch darauf wird zunächst verzichtet.

Ausgabe

Nach einer vorgebbaren Anzahl von Spielzügen wird die Simulation angehalten und die Darstellung der Welt aktualisiert. Das Spiel kann danach fortgesetzt oder abgebrochen werden. Für die Ausgabe wird die jeweilige Stärke der Individuen mit dem Faktor $1-q$ multipliziert. Bei dieser Normierung gilt, dass bei jeweils gleichem Gewinn g je Spielzug auch der Mittelwert gleich g ist.

Bedienoberfläche

Bildschirmlayout

In der Entwurfsphase festzulegen: Parametereingabe. Interaktive Steuerung der Simulation. Bildschirmdarstellung des rechteckigen Spielfelds („Welt“). Abspeicherung des Ergebnisses in einer Hintergrunddatei zur Weiterverarbeitung mit Excel.

Darstellung der Welt

Auf dem rechteckigen Spielfeld wird jedes Wesen durch einen Farbpunkt (eigentlich ein kleines Quadrat) dargestellt. Wird ein solcher Farbpunkt auf dem Spielfeld angeklickt, erscheint eine vollständige Darstellung aller Attribute des entsprechenden Wesens.

Interaktive Steuerung der Simulation

Die Aktualisierung des Spielfelds und das Neuzeichnen geschehen jeweils nach einer festen Zahl von Spielzügen. Dazu wird das Spiel angehalten. Über Buttons kann das Spiel fortgesetzt oder abgebrochen werden. In den Spielpausen ist das Spielfeld aktiv und das Anklicken

eines Farbpunktes öffnet ein Informationsfenster, das Auskunft über das zugehörige Wesen gibt.

Weitergehende Einflussnahme und Abänderung des Evolutionsprozesses ist hier nicht angemessen: Es geht darum, Gesetzmäßigkeiten der Evolution zu erkunden. Also müssen die Randbedingungen, unter denen die Evolution stattfindet, zu Beginn festgelegt und danach für den gesamten Evolutionsprozess beibehalten werden.

Protokollierung der Ergebnisse

Die Entwicklung der Populationsgrößen wird in der Datei LogFile.txt protokolliert. Beispiel für ein solches mit Excel bearbeitetes LogFile ist in der Projektdokumentation enthalten.

Testszenarien

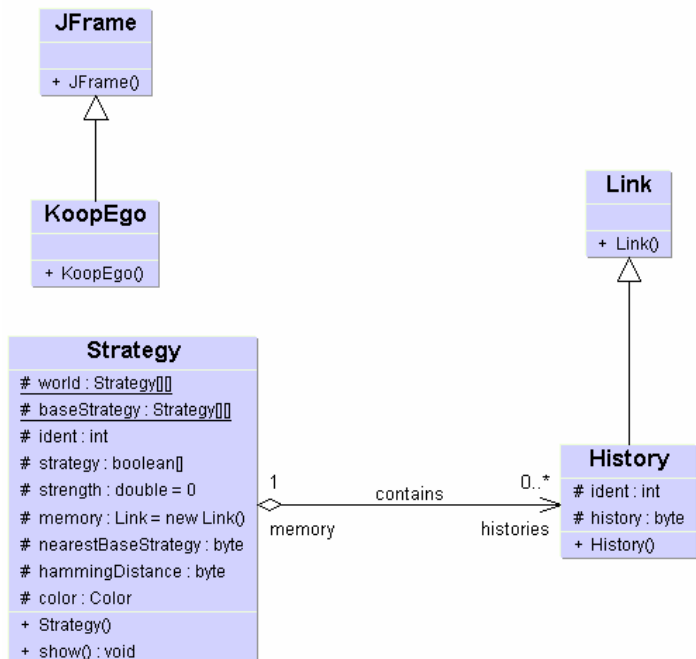
Mit dem Simulationsprogramm lassen sich näherungsweise Evolutionsbedingungen darstellen, die bereits mittels ökologischer Simulation untersucht worden sind (Umweltsimulation mit Tabellenkalkulation). Für den Fall, dass die Basischaraktere anfangs gleich stark vertreten sind und Mutationen sowie Interaktionsfehler ausgeschlossen sind, ist ein Vergleich der Simulationsergebnisse durchzuführen.

Das vorliegende Programm, geht von einer mikroskopischen Weltansicht aus, die ökologische Simulation arbeitet mit aggregierten Größen und ist makroskopisch angelegt. Der zeitliche Verlauf einer Populationsgröße ist im ersten Fall ein stochastischer Prozess und im zweiten eine deterministische Funktion. Eine exakte Übereinstimmung lässt sich allein deswegen nicht herstellen. Der Ergebnisvergleich ist auf qualitative Merkmale beschränkt, beispielsweise auf die Rangfolgen der Strategien.

Entwurf

UML-Diagramm

Im nebenstehenden UML-Diagramm sind die Eingabevariablen nicht enthalten. Diese Variablen werden im Laufe der Programmierung den passenden Klassen zugeordnet. Der Grafik-Entwurf wird separat abgehandelt.



Das statische Attribut world stellt die Welt dar, in der jedes Wesen seinen Platz hat. Der Platz ist gegeben durch die Indizes (Koordinaten). Die grafische Repräsentation der Welt ist ein Quadrat, in dem jedes Wesen an seinem Platz durch einen „Farbpunkt“ dargestellt wird. Der Farbwert wird aus der Strategie des Charakters abgeleitet. Bei Klick auf einen Bildpunkt wird show aufgerufen. Die Repräsentation des Bildpunkts wird beim nächsten Klick wieder gelöscht.

Das Array baseStrategy decodiert die Zahlen von 0 bis 5 in die Basisstrategien Tit for Tat, Vergelter, immer Kooperation, Pawlow, immer Defektion und Hinterlist.

Entwurf der grafischen Oberfläche

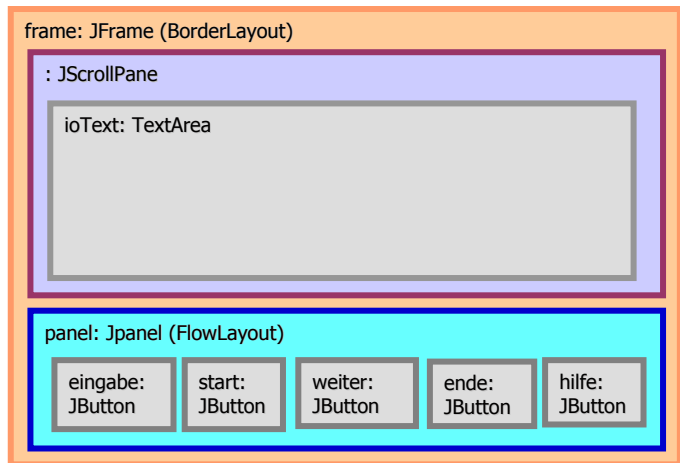
Das Steuerungsfenster

Die Eingabeparameter werden nach Drücken des Eingabe-Buttons aus einer Textdatei in ein Textfenster geladen. Der Text ist editierbar.

Mit einem Start-Button wird die aktualisierte Textdatei abgespeichert und die Simulation des Evolutionsprozesses initialisiert.

Mit dem Weiter-Button wird die Simulation gestartet bzw. fortgesetzt. Nach einer vorgebbaren Anzahl von Spielzügen hält die Simulation jeweils an.

Die Bewohner der Welt können nun inspiziert werden. Drücken des Ende-Buttons beendet die Simulation. (Änderungen Version 2: Anstelle des Weiter-Buttons gibt es einen Stop-and-Go-Button. Der Weiter-Button wird nun für ununterbrochene Simulation verwendet. Ein Halt-Button unterbricht die Simulation.)



Syntax der Textdatei für die Eingabeparameter

Die Parametereingabe geschieht nach folgendem Schema

```

<Parametername> = <Wert>
<Parametername> = <Wert>
<Parametername> = <Wert>
...







```

Das Gleichheitszeichen ist durch Leerzeichen vom Parameternamen und vom Wert getrennt.

Farbkodierung der Strategien

Die Farbe gibt den Charakter (die Strategie) des Wesens in komprimierter Form wieder. Der dreidimensionale Farbwert (Rot, Grün, Blau) wird für jede Farbkomponente in ganzen Zahlen im Bereich 0..255 angegeben. Den sechs Basischarakteren werden die drei Grundfarben der additiven Farbmischung und die drei Grundfarben der subtraktiven Farbmischung zugeordnet.

Die Farbkodierung der Basischaraktere

Tit for Tat	(255, 0, 0)	rot	
Vergelter	(255, 0, 255)	magenta	
Immer K	(255, 255, 0)	gelb	
Pawlow	(0, 0, 255)	blau	
Immer D	(0, 255, 0)	grün	
Hinterlist	(0, 255, 255)	cyan	

Alle anderen Charaktere werden folgendermaßen codiert: Zuerst wird der nächstliegende Grundcharakter im Sinne des Hamming-Abstands aufgesucht. Der Abstand, also die Anzahl der unterschiedlichen Bits in der Strategie, und der Farbwert des Basischarakters bestimmen die Farbe: Der Grundfarbe des Basischarakters werden beim Abstand d die Nebenfalten mit dem Wert $d \times 64 - 1$ beigemischt. Das heißt: Die Farbe wird umso stärker in Richtung Pastell (Ungesättigt) verändert, je mehr der Charakter vom jeweiligen Grundcharakter abweicht. Je Heller ein Feld, desto ungewisser ist die Zuordnung zu einem Basischarakter.

Die Farbkodierung lässt nur in begrenztem Maße Rückschlüsse auf den Charakter zu. Die gleiche Grundfarbe besagt beispielsweise noch nicht, dass zwei Charakter in Punkto Freund-

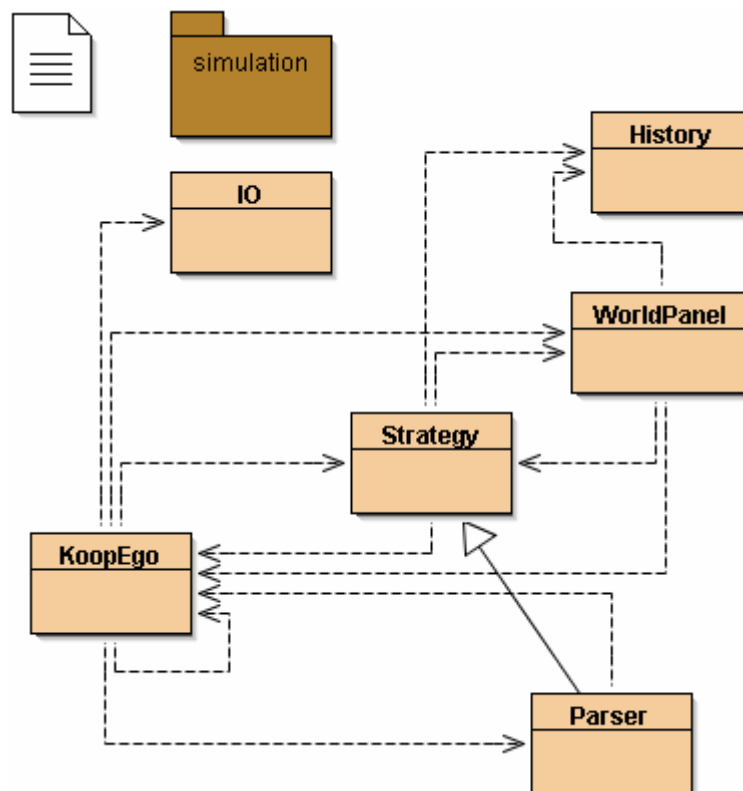
lichkeit übereinstimmen. Meist wird es so sein. Im Zweifelsfall liefert das Informationsfenster Aufschluss über das Wesen. Dieser Dialog wird durch Klick auf den Farbpunkt des Wesens geöffnet.

Per Default werden Felder ohne Wesen schwarz – also mit dem Farbwert (0, 0, 0) – eingefärbt.

Offene Fragen: Anfänglich wurden Farbcodierungen ohne Bezug zu den Basischarakteren betrachtet. Die Interpretation der Farbwerte machte Schwierigkeiten. Die hier gewählten Codierung („Je blasser, desto weniger lässt sich sagen“) mit einer farblich prägnanten Darstellung und guten Unterscheidbarkeit der Basischaraktere scheint zunächst einmal plausibel und aussagekräftig zu sein. Im Zuge der Realisierung ist über die Farbcodierung erneut zu befinden.

Realisierung

Klassendiagramm im BlueJ-Format



Das Programm

Programmtext

Der Programmtext ist – zusammen mit den BlueJ-Projektdateien – im Java-Archiv KoopEgo.jar enthalten. Hier werden nur noch einige Hinweise zu den Algorithmen gegeben. Insbesondere wird erläutert, wie die Entwurfsregel „einfach aber flott“ umgesetzt wird.

Algorithmus der Nachbarnauswahl

Die Auswahl eines Nachbarn für ein vorab bestimmtes Individuum geschieht mit nur einem einzigen Aufruf des Zufallszahlengenerators. Am Beispiel einer 2-Umgebung sei das Prinzip erläutert: Die Felder der Umgebung eines Elements werden oben links beginnend Zeile für

Zeile durchnummeriert. Nur das letzte Feld wird weggelassen. Allerdings erhält auch das zentrale Feld, das Feld des vorab bestimmten Individuums, eine Nummer (hier ist es die Nummer 12). Dieses Feld spielt den Stellvertreter für das Feld rechts unten. Nun wird eine der Nummern aus der Menge $\{0, 1, 2, \dots, 23\}$ durch Aufruf des Zufallszahlengenerators ausgewählt. Damit ist der Nachbar bestimmt. Nur im Fall, dass das Zentrum (die 12) getroffen wird, ist der rechte untere Nachbar dran. Links ist die fortlaufende und rechts ist die tatsächlich wirksame Nummerierung der Felder dargestellt.

	0	1	2	3	4				
	5	6	7	8	9				
	10	11	12	13	14				
	15	16	17	18	19				
	20	21	22	23					

	0	1	2	3	4				
	5	6	7	8	9				
	10	11		13	14				
	15	16	17	18	19				
	20	21	22	23	12				

Algorithmus der zufälligen Mutationen

Bei Erzeugung eines neuen Wesens ist jedes Bit seiner Strategie mit der Wahrscheinlichkeit p von einer Mutation betroffen. Zur Realisierung solcher Mutationen könnte man für jedes Bit der Strategie getrennt entscheiden, ob die Bitinvertierung stattfindet oder nicht. Jedes Mal wäre der Aufruf des Zufallszahlengenerators fällig.

Man halte sich vor Augen, dass die Mutationswahrscheinlichkeit p im Allgemeinen ziemlich klein ist ($1/100$ und kleiner). Der Zufallszahlengenerator-Aufruf bleibt folglich meist ohne Konsequenzen. Deshalb ist es effizienter, zunächst mit Hilfe der Binomialverteilung festzustellen, wie viele Bits der Strategie überhaupt zu invertieren sind. Dazu verwendet man ein Standardverfahren der Zufallszahlenerzeugung bei vorgegebener Verteilungsfunktion. Dann invertiert man die nötige Anzahl an Bits. Die Bits werden wieder durch Zufallsauswahl der Stelle bestimmt, wie bei der Nachbarnauswahl.

In der Arbeit von Schaake (2008) ist das Verfahren im Detail beschrieben.

Ergänzende Dokumentation zur Version 3

Problem und Ziel

Die Neuankömmlinge in der Welt erzeugen in den Kontrahenten eine mehr oder weniger zufällige „Scheinerinnerung“. Der Parameter r legt fest, mit welcher Wahrscheinlichkeit ein Betrug (eine 1) erinnert wird. Für $r = 0$ werden nur gute Erinnerungen für den Neuling angelegt ($history=0$) und für $r = 1$ erscheint er als der vollkommene Bösewicht ($history=7$). Diese Scheinerinnerungen haben Einfluss auf den Evolutionsprozess und erschweren die Interpretation der Abläufe.

Abhilfe: Es sollen nur die tatsächlichen Begegnungen erinnert werden. Das heißt: Wird ein Neuling angetroffen, so ist der Erinnerungsspeicher, das History-Byte, leer. Nach einer Begegnung enthält er nur die Erinnerung an diese eine Begegnung und so weiter. Der Parameter r kann entfallen.

Lösungsvorschlag

Das History-Byte enthält auf der niedrigstwertigen Stelle die Erinnerung an die gegnerische Aktion der jüngsten Vergangenheit. Auf der nächsthöheren Bitposition steht die Erinnerung an die vorletzte Begegnung, und auf der Bitposition der Wertigkeit 2^2 steht die Erinnerung an die Aktion der drittletzten Begegnung. Die Bitkombination 00000101 heißt also: Bei der letzten und der drittletzten Begegnung hat der Gegner betrogen, beim vorletzten Mal nicht. Die Werte für die History-Bytes reichen von 0 bis 7.

Die Erinnerung an die momentane Aktion wird für die an der Interaktion beteiligten Wesen durch die folgenden einfachen Java-Anweisungen in die History-Bytes eingetragen:

```
//Erinnerung auffrischen
h0.history= (byte)((h0.history<<1)& mask | action0);
h1.history= (byte)((h1.history<<1)& mask | action1);
```

Die Konstante mask hat den Wert 7, als Bitkombination geschrieben also 00000111.

Im Zuge der Realisierung verschiedener Gedächtnistiefen soll diese Technik im Grunde beibehalten und nur geringfügig modifiziert werden. Entscheidend ist eine geeignete Codierung des History-Bytes.

Das History-Byte wird um eine Stelle erweitert. Das nun höchstwertige Bit mit dem Wert 2^3 hat bei voller Historie (Gedächtnistiefe 3) den Wert 1. Damit liegen die Werte des History-Bytes für die volle Erinnerung im Bereich von 8 bis 15. Für die Erinnerungstiefe 2 wählen wir die Zahlen von 4 bis 7, für die Erinnerungstiefe die Zahlen 2 und 3 und für die fehlende Erinnerung die Zahl 1.

Mit den Bits x für die letzte, y für die vorletzte und z für die drittletzte Erinnerung sieht das History-Byte für die verschiedenen Gedächtnistiefen so aus:

Gedächtnistiefe	history
3	00001zyx
2	000001yx
1	0000001x
0	00000001

Das History-Byte für Neuankommlinge ist folglich mit dem Wert 1 zu initialisieren. Zur Fortschreibung des Gedächtnisses genügt nun ebenfalls eine einfache Schiebeoperation mit anschließender Gedächtnisbegrenzung. Allerdings reicht eine einfache Maskierung zur Gedächtnisbegrenzung nicht mehr aus. Es ist dafür zu sorgen, dass bei maximalem Gedächtnis die führende Eins an ihrer Stelle bleibt.

Umsetzung

Die vorgeschlagene Modifikation löst das Problem der Scheinerinnerung an Neuankommlinge. Aber sie hat auch Nachteile: Das Programm wird komplizierter. Vor allem lassen sich die Strategien nicht mehr einfach darstellen und es gibt sehr viele davon, nämlich 2^{15} . Für die Repräsentation der Strategien sind nun zwei Bytes nötig. Das höherwertige der Bytes entspricht den bisherigen Strategien. Für die Klassifizierung der Strategien wird nur dieses höherwertige Byte herangezogen.

Die Detaildarstellung eines Wesens ist im nebenstehenden Kasten wiedergegeben. Die Strategie ist in vier Gruppen aufgeteilt. Links ist die Reaktion bei noch leerem Gedächtnis. Dann kommen die Reaktionsmuster für die Gedächtnistiefen 1, 2 und 3.

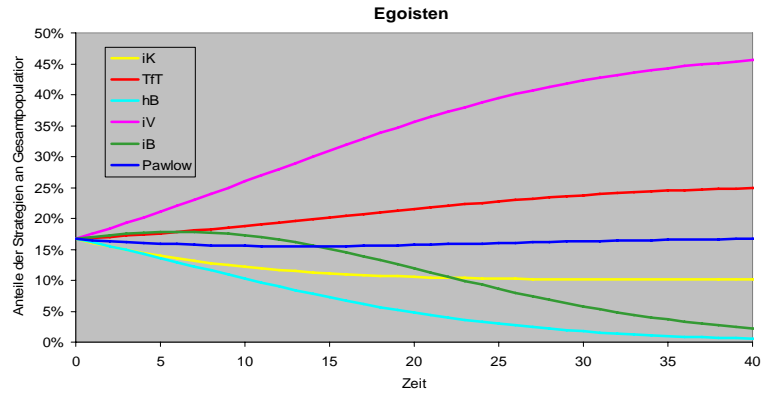
```
Identität = 442 (11, 2)
Lebenskraft = 1.5776306
Basistyp = Vergelter
Entfernung = 0
Strategie = 0 10 1110 11111110
Erfahrungen <Name (x, y) Historie>:
756 (18, 36) 00
1642 (7, 8) 0
722 (18, 2) 0
688 (17, 8) 110
...
641 (16, 1) 110
521 (13, 1) 111
```

Abnahme

Das Testszenario: Es geht um den Selektionsprozess (ökologischen Simulation). Und nur die reinen Strategien treten gegeneinander an. Die Prognose wird mittels makroskopischer Simulation erstellt. Ergeben sich bei der mikroskopischen Simulation mit KoopEgo im Rahmen der Vergleichbarkeit, also vor allem qualitativ gesehen, dieselben Ergebnisse, dann ist der Abnahmetest bestanden.

Deterministische makroskopische Simulation

Die makroskopische Simulation wird mit einer Variante des Tabellenkalkulationsblatts Ego.xls aus dem Internet-Kurs „Umweltsimulation mit Tabellenkalkulation“ durchgeführt. Und das ist das Ergebnis: Die freundlichen Strategien bleiben übrig. Sieger ist der konsequente Vergelter iV mit einem Anteil an der Bevölkerung von 50%. Danach folgen Tit for Tat mit 24%, Pawlow mit 16% und der gutmütige Trottel iK mit 9%.

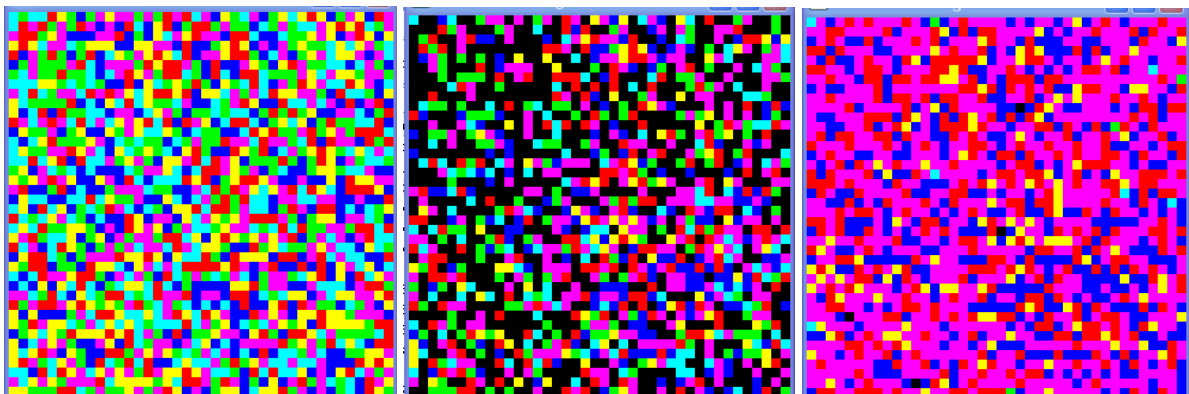


Stochastische mikroskopische Simulation

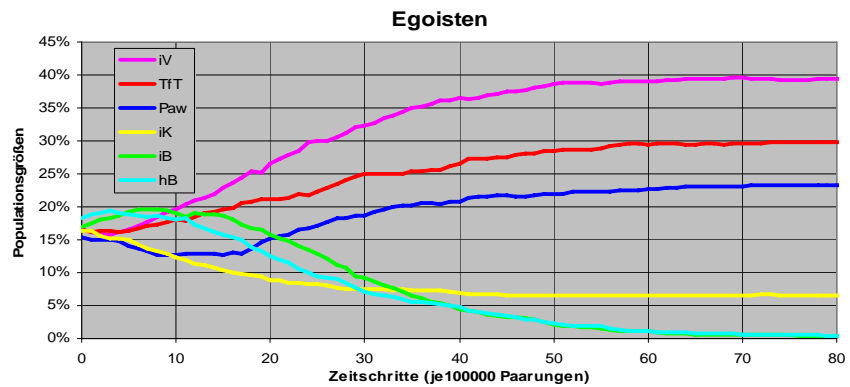
Die Randbedingungen der makroskopischen Simulation werden durch die nebenstehenden Parameterfestlegungen für KoopEgo angenähert.

Die folgenden Schnappschüsse von der Entwicklung der Populationen zeigen links die Anfangsverteilung, in der Mitte die Verteilung nach 20 Schritten und rechts das Schlussbild. Die Entvölkerung der Welt durch die rücksichtslosen Egoisten kommt in einem Rückgang der Gesamtpopulation zum Ausdruck. Die mikroskopische Simulation zeigt diesen Einbruch, der ungefähr nach 20 Zeitschritten maximal ist. Am Ende ist die Welt wieder nahezu vollständig bevölkert.

Anfangsbelegung: Standardstrategien
Lebensminimum: $(KD+DK)/2$
$n = 40$ (Spielfeldgröße)
$k = 7$ (Umgebungsparameter)
$h = 100000$ (Anzahl der Paarungen je Lauf)
$a = 1.0$ (Wahrscheinlichkeit fuer Vorbelegung einer Position der Welt)
$b = 0.0020$ (Geburtswahrscheinlichkeit)
$d = 0.0010$ (Sterbewahrscheinlichkeit)
$p = 0.0$ (Mutationswahrscheinlichkeit)
$f = 0.0$ (Interaktions-Fehlerwahrscheinlichkeit)
$q = 0.9$ (Daempfungsfaktor)



Die xy-Grafik gibt den zeitlichen Verlauf der Populationsentwicklung wieder. Für eine bessere Vergleichbarkeit sind auch bei der mikroskopischen Simulation die prozentualen Populationsanteile angegeben.



Abgesehen von der prinzipbedingt unterschiedlichen zeitlichen Skalierung erhält man für die Populationsentwicklungen der makroskopischen und der mikroskopischen Simulation eine sehr gute Übereinstimmung. Die Endzustände stimmen hinsichtlich der Rangordnung der Strategien überein. Der Test ist bestanden.

Experimente

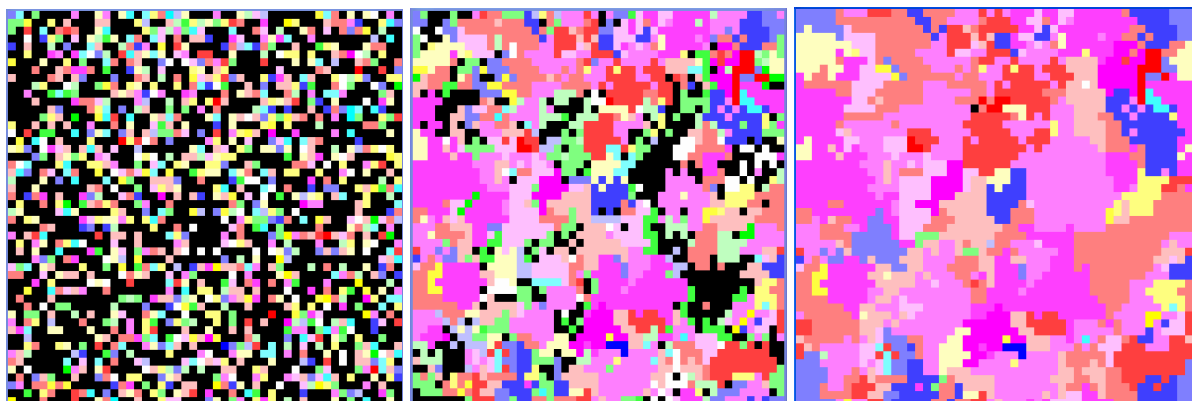
Perfekte Kommunikation ($f=0$), kleine Umgebung ($k=1$)

Die Parameterfestlegungen für der Simulation sind rechts tabellarisch zusammengefasst.

Simulationsverlauf: Das linke Bild zeigt den Anfangszustand der Welt. Nach einer Million Spielzügen haben sich die freundlichen Strategien (xxxxxxx0) weitgehend durchgesetzt (mittleres Bild). Die unfreundlichen (xxxxxxx1) sind vor allem unter den grünen Elementen zu finden. Betrug kommt in den Regionen der freundlichen Vergelter (magenta) nicht vor. Nur an den Grenzen zu den unfreundlichen Strategien (grün) kommt es zu Betrug und geringer Lebenskraft.

Anfangsbelegung: zufaellige Strategien
Lebensminimum: $(KD+DK)/2$
$n = 50$ (Spielfeldgrosesse)
$k = 1$ (Umgebungsparameter)
$h = 100000$ (Anzahl der Paarungen je Lauf)
$a = 0.5$ (Wahrscheinlichkeit fuer Vorbelegung einer Position der Welt)
$as = 0.5$ (Wahrscheinlichkeit fuer eine initiale 1 je Genposition)
$b = 0.1$ (Geburtswahrscheinlichkeit)
$d = 0.01$ (Sterbewahrscheinlichkeit)
$p = 0.01$ (Mutationswahrscheinlichkeit)
$f = 0.0$ (Interaktions-Fehlerwahrscheinlichkeit)
$q = 0.9$ (Daempfungsfaktor)

Typisch für die Regionen der unfreundlichen Strategen ist ihre Löchrigkeit: Die unfreundlichen können nur die freundlichen Nachbarn ausnutzen. Nach fünf Millionen Spielzügen haben die freundlichen Strategen die Welt unter sich aufgeteilt. Betrug kommt praktisch nicht mehr vor (rechtes Bild).



Wirkung der Globalisierung (Vergrößerung der Umgebung)

Die 2-Umgebung bewirkt kein grundsätzlich anderes Verhalten. Bei der 3-Umgebung kommt es oft zu einer Verödung der Welt: Alle Strategien sterben aus. Es kann aber auch zur Ent-

wicklung einer Monokultur aus einer freundlichen Strategie kommen. Manchmal können sich aber auch ein paar wenige freundliche Strategien etablieren. Bei noch größeren Umgebungen entvölkert sich die Welt rasch. Kurz: Globalisierung ist innovationsfeindlich. (Aber auch das regelt sich in der „wirklichen Wirklichkeit“ von selbst: Die Innovation wird dann von denen kommen, die sich – zumindest vorübergehend – vom Globalisierungstrend abgekoppelt haben.)

Fehlerhafte Interaktionen

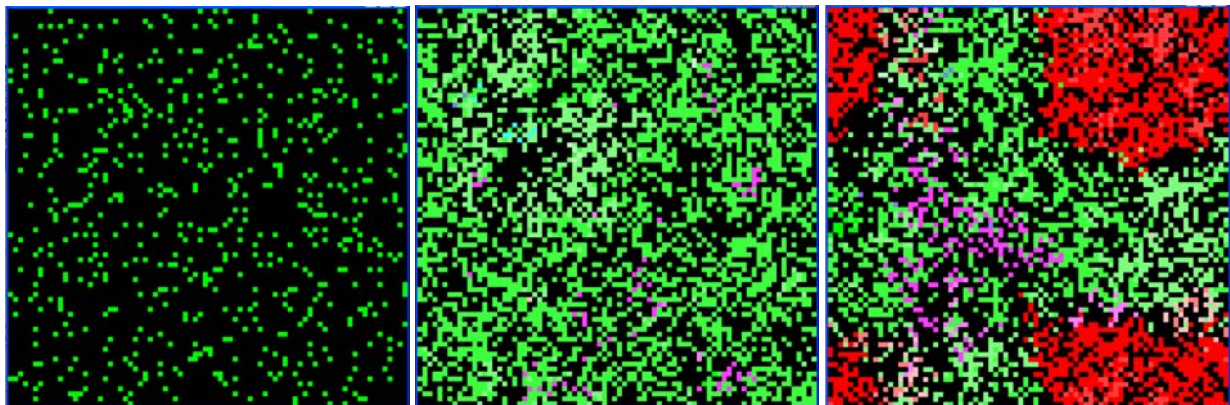
Erste Versuche zeigen, dass auch bei fehlerhaften Interaktionen ($f=1\%$) die freundlichen Strategien Oberhand gewinnen. Aber reine Vergelter haben auf Dauer keine Chancen. Typische Sieger sind die freundlichen Strategien (xxxxxxx0) mit höchstens vier Einsen.

Entstehung von Kooperation in feindseliger Umgebung

Wie sieht es aus, wenn die Welt anfangs nur dünn besiedelt ist, und zwar mit lupenreinen Betrügern, ohne jeglichen Anflug von Kooperationsbereitschaft? Kooperative Strategen können dann nur durch Mutationen entstehen. Außerdem werden Interaktionsfehler zugelassen: Jemand wollte eigentlich kooperieren, aber es kommt unabsichtlich ein Betrug heraus, oder ein vorsätzlicher Betrug entpuppt sich als Kooperation. Der Versuch wird mit der nebenstehenden Parameterbelegung (Grundeinstellung) durchgeführt:

Anfangsbelegung: zufällige Strategien
Lebensminimum: mittlere Lebenskraft der Umgebung
$n = 80$ (Spielfeldgröße)
$k = 2$ (Umgebungsparameter)
$h = 1000000$ (Anzahl der Paarungen je Lauf)
$a = 0.02$ (Wahrscheinlichkeit fuer Vorbelegung einer Position der Welt)
$as = 1.0$ (Wahrscheinlichkeit fuer eine initiale 1 je Genposition)
$b = 0.02$ (Geburtswahrscheinlichkeit)
$d = 0.01$ (Sterbewahrscheinlichkeit)
$p = 0.01$ (Mutationswahrscheinlichkeit)
$f = 0.01$ (Interaktions-Fehlerwahrscheinlichkeit)
$q = 0.9$ (Daempfungsfaktor)

Auf dem linken Schnappschuss ist die Anfangssituation dargestellt. Das Universum ist dünn besiedelt mit Wesen, die nur Betrug kennen (grün).



Bei dieser Simulation lassen sich grob vier Phasen der Entwicklung unterscheiden.

1. Phase: *Zufall – Ohne Fehler läuft gar nichts.* Da hier nur derjenige Nachkommen haben kann, dessen Lebenskraft sich über den Durchschnitt seiner Nachbarn erhebt, ist Trostlosigkeit und Stagnation angesagt. Aber es gibt ja die Irrtümer. Manch einer tut seinem Nachbarn aus Dussligkeit einen Gefallen, und schon hat dieser die Kraft für Nachkommen. So kommt Bewegung in das Spiel.

2. Phase: *Zufall – Selektionsneutrale Mutationen.* Bei der Entstehung neuer Wesen kann es zu zufälligen Mutationen kommen. Die meisten dieser Mutationen sind weder von Vorteil noch von Nachteil. Die neuen Strategien mischen sich unauffällig unters Volk und machen die

Welt bunter (mittlerer Schnappschuss). Mit zunehmender Variabilität baut sich Entwicklungspotential auf.

3. Phase: *Zufall – Glückliches Zusammentreffen*. Nun vergrößert sich die Chance, dass ein Wesen mit bislang schlummernden kooperativen Neigungen (hellgrün, magenta) einen Nachbarn erhält, der diese Ansätze der Kooperation hervorlockt und erwidert. Es kommt zu einzelnen Fällen von Nachbarschaftshilfe, zu Inseln der Kooperation. „Créer, c’est unir“ meint Teilhard de Chardin. Und für Konrad Lorenz (1973) entsteht das Neue bei der schöpferischen Vereinigung von Verschiedenem zur funktionellen Ganzheit als blitzartige Erleuchtung (Fulguration). Unter den Wesen habe ich zwei Nachbarn gefunden, die diesen Effekt sehr gut repräsentieren. Ich nenne sie A und B. Die Strategien der beiden sind

A: 1 01 1111 11011101
B: 1 11 1100 11010011

Augenscheinlich ist der Verwandtschaftsgrad zwischen beiden sehr gering.

A lebt inmitten von Seinesgleichen ganz passabel: Die Folge der Aktionen ist gleich **101010...** (Farblich und fett hervorgehoben ist die Periode.) Da diese Aktionen gleich lautend erwidert werden, ist der mittlere Nutzen eines jeden Individuums gleich $(DD+KK)/2 = 2/2 = 1$. Schlechter dran ist B. Unter Seinesgleichen herrscht konsequenter Betrug. Der mittlere Gewinn ist gleich 0.

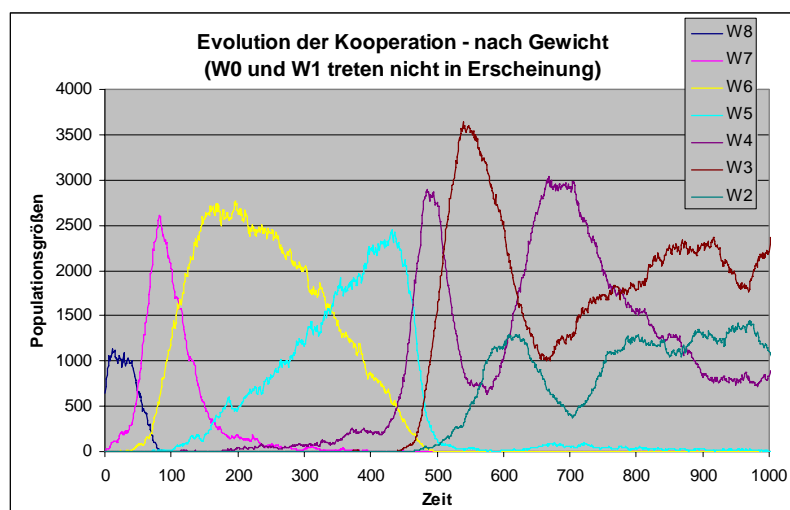
Treffen jedoch A und B wiederholt aufeinander, so zahlt sich das für beide aus. Die Folgen der jeweiligen Aktionen beim wiederholten Aufeinandertreffen sind gegeben durch

A: **10111**1011110111101...
B: **11100**11110011100111...

Der mittlere Nutzen für A ist gleich $(DD+KD+DD+DK+DK)/5 = 7/5$. Das ist eine Verbesserung gegenüber den Treffen mit Seinesgleichen. Der mittlere Nutzen für B ist gleich $(DD+DK+DD+KD+KD) = 2/5$. Das ist nicht viel, aber doch besser als das von Seinesgleichen zu gewinnende Nichts. Dennoch sind die Prognosen für B schlecht: Gegen A kann sein Wohltäter auf Dauer nicht bestehen.

4. Phase: *Notwendigkeit*. Auf den Inseln der Kooperation steigt die Lebenskraft der Individuen. Nachdem die eigene kleine Hütte in Ordnung gebracht worden ist, kann die Eroberung starten. Das kooperative Verhalten (rot) breitet sich aus und verdrängt das betrügerische Verhalten (rechter Schnappschuss). Der Selektionsprozess mündet in einen Zustand dynamischer Stabilität. Das Gute siegt.

Die Grafik zeigt den Simulationsablauf für eine Milliarde Paarungen (eine Mio. Paarungen je Zeitschritt). Sie macht deutlich, wie die Wesen mit einer Maximalzahl betrügerischer „Gene“ (W8) durch weniger feindselige verdrängt werden. Schließlich teilen die kooperativen Strategien mit einer geringern Anzahl an betrügerischen „Genen“ (W4, W3 und W2) die Welt unter sich auf. Die genauere Analyse zeigt, dass nur wenige Strategien übrig bleiben, die allesamt freundlicher Natur sind: Es kommt zu dauerhafter und weit verbreiteter Kooperation.



Die genauere Analyse zeigt, dass nur wenige Strategien übrig bleiben, die allesamt freundlicher Natur sind: Es kommt zu dauerhafter und weit verbreiteter Kooperation.

Etwas unerwartetes Neues ist bei der Simulation auch entstanden, ein freundlicher Charakter, der erst ab dem zweiten Schlag zurückschlägt, ein neutestamentarischer Tit-for-Tat-Strategie sozusagen. Für die Gedächtnistiefe 3 sieht das Bitmuster seiner Strategie so aus: 10101000.

Er ist unter vielen Bedingungen sehr erfolgreich. Wenig Chancen haben die allzu Gutmütigen: Wer kein oder zu wenige betrügerische Gene für Vergeltungszwecke hat (W0 und W1), der kommt nicht zum Zuge.

Schlussfolgerungen

Die Ergebnisse der Simulation stehen in gutem Einklang mit dem Stand der biologischen Forschung und aktuellen Erkenntnissen der Wirtschafts- und Gesellschaftswissenschaften. Das soll mit einer kleinen Zitatsammlung belegt werden.

Biologie

„Human beings appear to be sufficiently selfish and calculating to be capable of indefinitely greater harmony and social homeostasis. This statement is not self-contradictory. True selfishness, if obedient to the other constraints of mammalian biology, is the key to a more nearly perfect social contract.” (Wilson, 1978, S. 157)

„Erst wenn sich die abweichenden Varianten lange Zeit nur noch untereinander kreuzen, werden sich ihre Eigenschaften so verstärken können, dass sich eine neue Varietät, dann eine Rasse und schließlich eine eigenständige Art bildet. Der Schlüssel hierfür ist die reproduktive Abgrenzung. Diese wird – davon geht man heute aus – wesentlich durch geographische Isolation erreicht.“ (Hösle/Illies, 2005, S. 55)

„Beim großen Ostafrikanischen Grabensystem, das sich vom Roten Meer nach Süden längs durch den Kontinent zieht, hört [das Verbreitungsgebiet der Schimpansen und Gorillas] ziemlich abrupt auf... Tektonische Geschehnisse haben ökologische Veränderungen zur Folge, die oftmals mit Evolutionsschüben einhergehen... Bereits in den sechziger Jahren hatte allerdings der Verhaltensforscher Adriaan Kortlandt von der Universität Amsterdam auf der Grundlage der damals verfügbaren Befunde ein ähnliches Szenario entworfen, das er ‚Western-Rift-Hypothese‘ nannte; im Jahre 1972 legte er dann in seinem Buch über die Evolution von Menschen und afrikanischen Menschenaffen ausführlich dar, wie das Grabensystem beide Linien getrennt haben könnte – unter anderem dadurch, dass die in der Störungszone entstandenen Flüsse und Seen eine zusätzliche Barriere bildeten, weil beide Primatengruppen von ihrer Anatomie her schlechte Schwimmer sind.“ (Coppens, 1994)

„Betreibt also ein Affe oder sonst ein soziales Tier eine ‚innere Buchführung‘, sodass er – wie ein Mensch – stets weiß, wie viel er jedem seiner Genossen schuldig ist und umgekehrt? Wohl kaum. Es genügt, wenn er seine soziale Welt in bevorzugte ‚Freunde‘ und wenig beachtete ‚Nichtfreunde‘ aufteilt... Wer einzig Profitmaximierung im Sinn hat, nimmt, was er kriegen kann, ohne sich von Zorn oder Neid beeinflussen zu lassen... Dagegen geht die ökonomische Verhaltensforschung davon aus, dass die Evolution Emotionen hervorgebracht hat, die den Gemeinschaftssinn fördern.“ (Waal, 2006)

„Each culture develops its own particular rules to safeguard personal property and space... Our brains do appear to be programmed to the following extent: we are inclined to partition other people into friends and aliens, in the same sense that birds are inclined to learn territorial songs and to navigate by the polar constellations. We tend to fear deeply the actions of strangers and to solve conflict by aggression. These learning rules are most likely to have evolved during the past hundreds of thousands of years of human evolution and, thus, to have conferred a biological advantage on those who conformed to them with the greatest fidelity.” (Wilson, 1978, S. 109, 119)

Wirtschafts- und Gesellschaftswissenschaften

Die Simulations-Ergebnisse liefern starke Argumente für die *Freiheitsrechte*, das *Subsidiaritätsprinzip*, den *Föderalismus* und den *Minderheitenschutz*. Effizienzdruck und Evaluierungen kurzfristigen Erfolgs im Bereich von Forschung und Lehre sehen im Licht der Ergebnisse eher nach Misserfolgsrezepten aus. Fragezeichen werden hinter die Forderung nach übertriebener möglichst europa- oder weltweiter Standardisierung gesetzt. Die allgemeinen Globalisierungstendenzen und – im Kleinen – der Bologna-Prozess wirken möglicherweise verschlankend und effizienzsteigernd. Aber den schöpferischen Prozessen sind sie nicht so förderlich.

In der Wirtschaft gibt es zwei gegenläufige Bewegungen oder Trends: Die Verschlinkung und globale Orientierung einerseits und das Schaffen von Freiräumen für Kreativität andererseits. „Die Isolation kann ein wichtiger Bestandteil der Kreativität sein... Man macht die Sachen anders als andere... Es ist ein Ausbalancieren, der alte Konflikt: Geborgenheit oder Abenteuer. Es ist ein Ausbalancieren zwischen einem individuellen und einem gemeinschaftlichen Weg.“ (v. Pierer/v. Oetinger, 1997, S. 231)

Verschlinkung. Sie setzt auf Effizienzsteigerung und vermehrte Wettbewerbsfähigkeit durch Konzentration auf das Kerngeschäft: „Die Weltklasse spielt nach neuen Regeln... Fokus macht fit.“ (McKinsey, 1994) Es geht um den kurzfristigen weltweiten Erfolg. „Mut zum Verzicht haben nicht nur Unternehmen, die ihre Variantenzahl beschränken, sondern auch diejenigen, die Altprodukte vorausschauend ausphasen und sich nicht scheuen, mit neuen Produktreihen ihre eigene Produktportfolios zu kannibalisieren.“ (McKinsey, 1994, S. 55) Innovationswirksam sind vor allem neue Zielmärkte: „Innovative Kunden sichern dem Hersteller eine hohe Anregungsdichte. Präsenz in innovativen Leitmärkten schützt vor dem Risiko, wesentliche Innovationen zu verschlafen.“ (McKinsey, 1994, S. 115).

Freiräume für Neues. „Der Freiraum sollte einen Sog ausüben, durch den der ‚Schüler zu seiner Frage‘ kommt. ‚Ein Lehrer, der mit Antworten droht, tötet den Schüler.‘“ (v. Pierer/v. Oetinger, 1997, S. 17f.) „Kaos ‚Tugend der Unordnung‘ führt uns in eine neue Wirtschaft, deren Wettbewerbsfähigkeit weniger durch Effizienz und Kontrolle als durch ‚Geschwindigkeit, Originalität, Intelligenz und Beweglichkeit‘ gekennzeichnet ist. Das verlangt Kreativität... Mitarbeiter benötigen den virtuellen Holzschuppen, in dem die Vorurteile zurück- und die besten Ideen vortreten können, bis das Neue Gestalt annimmt.“ (v. Pierer/v. Oetinger, 1997, S. 24)

Pluralistische Kulturtheorie. “Countries are constituted of competing political cultures, not a single political culture... The tendency to attach culture to nations persists despite strong evidence suggesting that variations in political attitudes and values within countries are often greater than those between countries.” (Thompson/Ellis/Wildavsky, S. 216, 219)

Toleranz. “In contrast to the moral arguments for toleration which we have just considered, *epistemic* arguments for toleration emphasize the contribution that tolerance makes to *knowledge*. Such arguments find their most systematic articulation in the work of John Stuart Mill. According to Mill [“On Liberty”], toleration is necessary because (1) discovering the truth (or believing what is true *in the right kind of way*) contributes to overall utility; and (2) we can only discover the truth (or believe what is true *in the right way*) in circumstances where different beliefs and practices are permitted to flourish.” (Leiter, 2006)

Meine Kurzfassung von „On Liberty“ (Mill, 1859) geht so:

Alles wurde ein erstes Mal gemacht. Dazu braucht es Freiheit, Individualität, Originalität.

Literaturverzeichnis

Ergänzende Dokumentationen

- Grams, Timm: Ist das Gute göttlich oder Ergebnis der Evolution? Kooperatives Verhalten in einer Welt voller Egoisten. Vortrag zur XVIII. GWUP-Konferenz „Kreationismus“ am 3. Mai 2008 in Darmstadt – [pdf-Datei](#)
- Schaake, Henrik: Simulierte Kreativität am Beispiel der Entstehung kooperativen Verhaltens. Diplomarbeit im Fachbereich Elektrotechnik und Informationstechnik an der Hochschule Fulda, 19.6.2008. – [pdf-Datei](#)

Quellen

- Axelrod, Robert: Die Evolution der Kooperation. Oldenbourg, München, Wien 1987
- Axelrod, Robert: The Complexity of Cooperation. Princeton University Press, Princeton, New Jersey 1997
- Ayala, Francisco J.: Mechanismen der Evolution. Spektrum der Wissenschaft (1979) 5, 8-18
- Coppens, Y.: Geotektonik, Klima und der Ursprung des Menschen. Spektrum der Wissenschaft (1994) 12, 64-71
- Culwin, Fintan: A JAVA GUI Programmer's Primer. Prentice Hall, Upper Saddle Rier, New Jersey 1998
- Doberenz, Walter; Druckenmüller, Uwe: Java. Programmierung interaktiver WWW-Seiten. Hanser, München, Wien 1998
- Eigen, Manfred; Winkler, Ruthild: Das Spiel. Naturgesetze steuern den Zufall. Piper, München 1975
- Fischer, Paul: An Introduction to Graphical User Interfaces with Java Swing. Addison Wesley, Pearson Education, Harlow 2005
- Grams, Timm: Umweltsimulation mit Tabellenkalkulation. <http://www.fh-fulda.de/~grams/oekosim.htm>
- Holland, John H.: Genetische Algorithmen. Spektrum der Wissenschaft (1992) 9, 44-51
- Hösle, V.; Illies, C.: DARWIN. Buchners Verlag, Bamberg 2005
- Leiter, Brian: Why Tolerate Religion. University of Texas, 25.5.2006. Auf der Website "Social Science Research Network", <http://ssrn.com>
- Lorenz, K.: Die Rückseite des Spiegels. Piper, München 1973
- McKinsey & Company, Inc.: Wachstum durch Verzicht. Schneller Wandel zur Weltklasse: Vorbild Elektronikindustrie. Schäffer-Pöschel-Verlag, 1994
- Michalewicz, Zbigniew; Fogel, David, B.: How to Solve It: Modern Heuristics. Springer-Verlag, Berlin, Heidelberg 2000
- Mill, John Stuart: On Liberty. 1859 („John Stuart Mill: On Liberty and Utilitarianism“. Oxford University Press 1969. Everyman's Library 1992.)
- Nowak, Martin A.; May, Robert M.; Sigmund, Karl: Das Einmaleins des Miteinander. Spektrum d. Wiss. (1995) 8, 46-53
- Pierer, Heinrich von; Oetinger, Bolko von: Wie kommt das Neue in die Welt? Carl Hanser Verlag, Wien 1997
- Rechenberg, Ingo: Evolutionsstrategie. Optimierung technischer System nach den Prinzipien der biologischen Evolution. Friedrich Frommann Verlag, Stuttgart-Bad Cannstatt 1973
- Thompson, M.; Ellis, R.; Wildavsky, A.: Cultural Theory. Westview Press, Boulder, Colorado 1990
- Waal, Frans B. M. de: Tierische Geschäfte. Spektrum d. Wiss. (2006) 6, 50-58
- Wilson, E. O.: On Human Nature. Harvard University Press, Cambridge, Mass. 1978
- Wilson, Edward Osborne: Sociobiology. The New Synthesis. Cambridge, Massachusetts 1975, 2000