

# Evolutionäre Algorithmen

## Evolutionstrategien, Verhaltenssimulation

**Prof. Dr. Rudolf Kruse**     **Pascal Held**

`{kruse,pheld}@iws.cs.uni-magdeburg.de`

Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik

Institut für Wissens- und Sprachverarbeitung

# Übersicht

## 1. Evolutionsstrategien

Selektion

Varianzanpassung

Kovarianz und Korrelation

Crossover/Rekombination

Plus- versus Komma-Strategien

## 2. Evolutionäre Algorithmen zur Verhaltenssimulation

# Evolutionstrategien (ES)

- Beschränkung auf **numerische Optimierung**

gegeben: Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$

gesucht: Minimum oder Maximum von  $f$

Chromosomen: **Vektoren reeller Zahlen**

**Mutation:** Addition eines normalverteilten **Zufallsvektors**  $\mathbf{r}$   
jedes Element  $r_i \in \mathbf{r}$  ist Realisierung einer normalverteilten  
Zufallsvariable mit

- Erwartungswert 0 (unabhängig vom Elementindex  $i$ ) und
  - Varianz  $\sigma_i^2$  bzw. Standardabweichung  $\sigma_i$
  - $\sigma_i^2$  abhängig/unabhängig vom Elementindex  $i$  und Generation  $t$
- **Crossover:** hierauf wird meistens verzichtet.

# Selektion

## Strenges Eliteprinzip:

- nur besten Individuen kommen in nächste Generation
- Bezeichnungen:
  - $\mu$  – Anzahl der Individuen in der Elterngeneration
  - $\lambda$  – Anzahl der (durch Mutation) erzeugten Nachkommen

## Zwei prinzipielle Selektionsstrategien:

- **+ -Strategie** (Plus-Strategie,  $(\mu + \lambda)$ -Strategie)  
aus den  $(\mu + \lambda)$  Individuen der Elterngeneration und der erzeugten Nachkommen werden die besten  $\mu$  Chromosomen ausgewählt  
(hier gilt meist  $\lambda < \mu$ )
- **, -Strategie** (Komma-Strategie,  $(\mu, \lambda)$ -Strategie)  
erzeugt  $\lambda > \mu$  Nachkommen, aus denen die besten  $\mu$  Chromosomen ausgewählt werden  
(Chromosomen der Elterngeneration gehen verloren)

# Selektion

**Beispiel:** Spezialfall der (1+1)-Strategie

„Anfangspopulation“:  $\mathbf{x}_0$  (zufällig erzeugter Vektor reeller Zahlen)

Erzeugen der nächsten Generation:

1. erzeuge reellen Zufallsvektor  $\mathbf{r}_t$  und berechne  $\mathbf{x}_t^* = \mathbf{x}_t + \mathbf{r}_t$
2. setze dann

$$\mathbf{x}_{t+1} = \begin{cases} \mathbf{x}_t^*, & \text{falls } f(\mathbf{x}_t^*) \geq f(\mathbf{x}), \\ \mathbf{x}_t, & \text{sonst.} \end{cases}$$

Erzeuge weitere Generationen bis Abbruchkriterium erfüllt ist

- entspricht dem **Zufallsaufstieg**
- ⇒ allgemeine +-Strategie = paralleler Zufallsaufstieg, der gleichzeitig an mehreren Orten von  $\Omega$  durchgeführt wird (wobei stets die erfolversprechendsten  $\mu$  Wege verfolgt werden)

# Evolutionsstrategien

Optimiert werden nicht nur Organismen, sondern Mechanismen der Evolution: Vermehrungs- und Sterberaten, Lebensdauern, Anfälligkeit gegenüber Mutationen, Mutationsschrittweiten, Evolutionsgeschwindigkeit etc.

## Varianzanpassung des Zufallsvektors (Mutationsschrittweite)

- geringe Varianz  $\Rightarrow$  kleine Änderungen der Chromosomen  $\Rightarrow$  lokale Suche (Ausbeutung)
- hohe Varianz  $\Rightarrow$  große Änderungen der Chromosomen  $\Rightarrow$  globale Suche (Durchforstung)

Weitere Möglichkeiten, Parameter anzupassen:

- Wahl der Zahl der zu ändernden Gene (Vektorelemente)
- Wahl der Zahl  $\lambda$  der zu erzeugenden Nachkommen

# Globale Varianzadaptation

## Chromosomenunabhängige Varianz

- Wähle  $\sigma$ , sodass mittlere Konvergenzrate möglichst hoch  
Ansatz von [Rechenberg, 1973]: bestimme optimales  $\sigma$  für
  - $f_1(x_1, \dots, x_n) = a + bx_1$  und
  - $f_2(x_1, \dots, x_n) = \sum_{i=1}^n x_i^2$ ,indem Wahrscheinlichkeiten für erfolgreiche (d.h. verbessernde) Mutation bestimmt werden
- Ergebnisse bei dieser empirischen Untersuchung
  - für  $f_1$ :  $p_1 \approx 0.184$  und
  - für  $f_2$ :  $p_2 \approx 0.270$

### $\frac{1}{5}$ -Erfolgsregel

- heuristisch abgeleitet
- mit +-Strategie ist Mutationsschrittweite richtig, wenn etwa  $\frac{1}{5}$  der Nachkommen besser sind als die Eltern

# Globale Varianz Anpassung

Anpassung der Varianz  $\sigma^2$  auf Grundlage der  $\frac{1}{5}$ -Erfolgsregel:

- sind mehr als  $\frac{1}{5}$  der Nachkommen besser als die Eltern, vergrößere Varianz/Standardabweichung:

$$\sigma' = \sigma \cdot \alpha, \quad \alpha > 1$$

- sind weniger als  $\frac{1}{5}$  der Nachkommen besser als Eltern, verkleinere Varianz/Standardabweichung:

$$\sigma' = \sigma / \alpha$$

- bei größeren Populationen ist  $\frac{1}{5}$ -Erfolgsregel z.T. zu optimistisch
- analog zum **simulierten Ausglühen**: definiere Funktion, mit der Varianz proportional zu  $t$  sinkt

# Globale Varianz Anpassung

---

## Algorithm 1 ADAPTIVE-ANPASSUNG

---

**Input:** Standardabweichung  $\sigma$ , Erfolgsrate  $p_s$ , Schwellwert  $\theta = \frac{1}{5}$ , Modifikationsfaktor  $\alpha > 1$

**Output:** angepasste Standardabweichung  $\sigma$

```
1: if  $p_s > \theta$  {  
2:   return  $\alpha \cdot \sigma$   
3: }  
4: if  $p_s < \theta$  {  
5:   return  $\sigma/\alpha$   
6: }  
7: return  $\sigma$ 
```

---

---

## Algorithm 2 ES-Adaptiv

---

**Input:** Zielfunktion  $F$ , Populationsgröße  $\mu$ , Anzahl der Kinder  $\lambda$ , Modifikationshäufigkeit  $k$

```
1:  $t \leftarrow 0$ 
2:  $\sigma \leftarrow$  Wert für Anfangsschrittweite
3:  $s \leftarrow 0$ 
4:  $P(t) \leftarrow$  erzeuge Population mit  $\mu$  Individuen
5: bewerte  $P(t)$  durch  $F$ 
6: while Terminierungsbedingung nicht erfüllt {
7:    $P' \leftarrow \emptyset$  /* für Plus-Selektion  $P' \leftarrow P(t)$  */
8:   for  $i = 1, \dots, \lambda$  {
9:      $A \leftarrow$  selektiere Elternteil uniform zufällig aus  $P(t)$ 
10:     $B \leftarrow$  GAUSS-MUTATION( $A$ ) mit  $\sigma$ 
11:    bewerte  $B$  durch  $F$ 
12:    if  $B.F > A.F$  {
13:       $s \leftarrow s + 1$ 
14:    }
15:     $P' \leftarrow P' \cup \{B\}$ 
16:  }
17:   $t \leftarrow t + 1$ 
18:   $P(t) \leftarrow$  Selektion aus  $P'$  durch BESTEN-SELEKTION
19:  if  $\text{mod}(t, k) = 0$  {
20:     $\sigma \leftarrow$  ADAPTIVE-ANPASSUNG( $\sigma, \frac{s}{k \cdot \lambda}$ )
21:     $s \leftarrow 0$ 
22:  }
23: }
24: return bestes Individuum aus  $P(t)$ 
```

---

# Lokale Varianz Anpassung

## Chromosomenspezifische Varianz

**Varianz/Standardabweichung** wird in Chromosomen als Zusatzinformation aufgenommen:

- eine Varianz für alle Vektorelemente oder
- eine individuelle Varianz für jedes Vektorelement (doppelte Vektorlänge)

**Beachte:** zusätzlichen Vektorelemente für Varianz(en) haben *keinen direkten Einfluss* auf Fitness eines Chromosoms

**Erwartung:** Chromosomen mit „schlechten“ Varianzen, d.h.

- zu klein: Chromosomen entwickeln sich nicht schnell genug weiter oder
- zu groß: Chromosomen entfernen sich zu weit von ihren Eltern, erzeugen vergleichsweise mehr „schlechte“ Nachkommen  
ihre Gene (und damit auch ihre Varianzen) sterben leichter aus

# Lokale Varianz Anpassung

elementspezifische Mutationsschrittweiten  
(Standardabweichungen) werden nach folgendem Schema mutiert:

$$\sigma'_i = \sigma_i \cdot \exp(r_1 \cdot N(0, 1) + r_2 \cdot N_i(0, 1)).$$

$N(0, 1)$ : 1x je Chromosom zu bestimmende normalv. Zufallszahl

$N_i(0, 1)$ : für jedes Gen zu bestimmende normalverteilte Zufallszahl  
empfohlene Werte für  $r_1, r_2$  [Bäck and Schwefel, 1993]

$$r_1 = \frac{1}{\sqrt{2n}}, \quad r_2 = \frac{1}{\sqrt{2\sqrt{n}}},$$

wobei  $n$  Anzahl der Vektorelemente ist, oder [Nissen, 1997]

$$r_1 = 0.1, \quad r_2 = 0.2$$

oft: untere Schranke für Mutationsschrittweiten

---

## Algorithm 3 ES-Selbstadaptiv

---

**Input:** Zielfunktion  $F$ , Populationsgröße  $\mu$ , Anzahl der Kinder  $\lambda$

```
1:  $t \leftarrow 0$ 
2:  $P(t) \leftarrow$  erzeuge Population mit  $\mu$  Individuen
3: bewerte  $P(t)$  durch  $F$ 
4: while Terminierungsbedingung nicht erfüllt {
5:    $P' \leftarrow \emptyset$                                      /* für Plus-Selektion  $P' \leftarrow P(t)$  */
6:   for  $i = 1, \dots, \lambda$  {
7:      $A \leftarrow$  selektiere Elternteil uniform zufällig aus  $P(t)$ 
8:      $B \leftarrow$  SELBSTADAPTIVE-GAUSS-MUTATION( $A$ )
9:      $P' \leftarrow P' \cup \{B\}$ 
10:  }
11:  bewerte  $P'$  durch  $F$ 
12:   $t \leftarrow t + 1$ 
13:   $P(t) \leftarrow$  Selektion aus  $P'$  durch BESTEN-SELEKTION
14: }
15: return bestes Individuum aus  $P(t)$ 
```

---

## Erweiterungen der lokalen Varianz Anpassung

- Standardform der lokalen Varianz Anpassung: Varianzen der verschiedenen Vektorelemente sind unabhängig voneinander (formal: Kovarianzmatrix ist Diagonalmatrix)
- für Variationen eines Chromosoms bevorzugt in bestimmten Richtungen: mittels Einzelvarianzen nur genau dann, wenn achsenparallele Richtungen

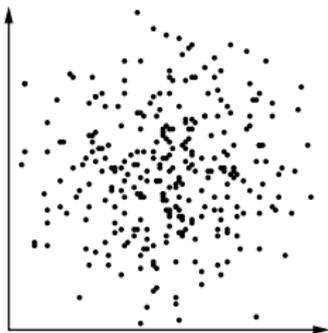
**Beispiel:** erzeuge Variationen von Chromosomen mit zwei Genen bevorzugt in Richtung der Hauptdiagonale, d.h. in Richtung (1, 1) mit Einzelvarianzen nicht beschreibbar

**Lösung:** benutze Kovarianzmatrix mit hoher Kovarianz, z.B.

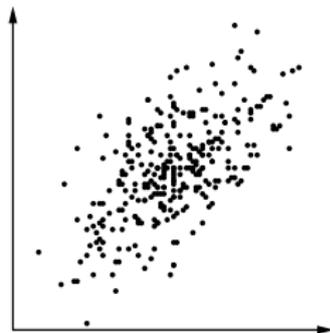
$$\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$

# Kovarianz und Korrelation

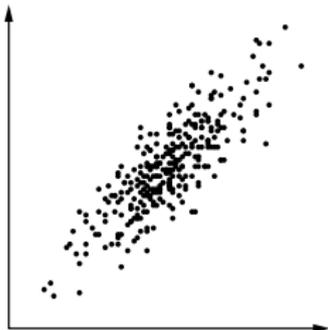
keine  
Korrelation



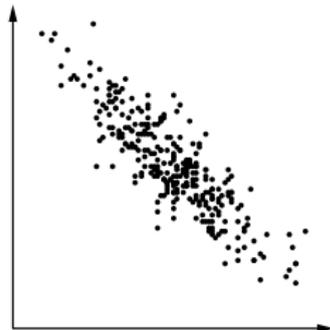
schwache  
positive  
Korrelation



starke  
positive  
Korrelation



starke  
negative  
Korrelation



# Cholesky-Zerlegung

**S** symmetrisch positiv definite Matrix (d.h. Kovarianzmatrix)

- symmetrisch:  $\forall 1 \leq i, j \leq m : s_{ij} = s_{ji}$
- positiv definit: für alle  $m$ -dim. Vektoren  $\mathbf{v} \neq \mathbf{0}$  gilt  $\mathbf{v}^T \mathbf{S} \mathbf{v} > 0$

Cholesky-Zerlegung: Berechnung der „Quadratwurzel“ von **S**

Berechne linke/untere Dreiecksmatrix **L**, sodass  $\mathbf{L} \mathbf{L}^T = \mathbf{S}$   
( $\mathbf{L}^T$  ist Transponierte der Matrix **L**)

$$l_{ii} = \left( s_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)^{\frac{1}{2}}$$

$$l_{ji} = \frac{1}{l_{ii}} \left( s_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right), \quad j = i + 1, i + 2, \dots, m$$

# Cholesky-Zerlegung

## Spezialfall: Zwei Dimensionen

Kovarianzmatrix

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$$

Cholesky-Zerlegung

$$\mathbf{L} = \begin{pmatrix} \sigma_x & 0 \\ \frac{\sigma_{xy}}{\sigma_x} & \frac{1}{\sigma_x} \sqrt{\sigma_x^2 \sigma_y^2 - \sigma_{xy}^2} \end{pmatrix}$$

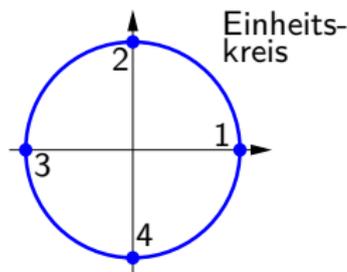
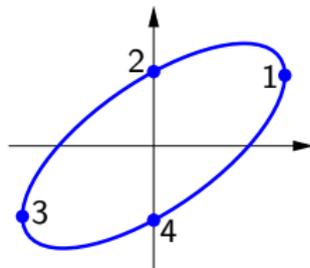


Abbildung mit  $\mathbf{L}$



# Eigenwertzerlegung

liefert ebenfalls **Analogon der Standardabweichung**

rechenaufwendiger als Cholesky-Zerlegung

**S** symmetrisch positiv definite Matrix (d.h. Kovarianzmatrix)

$$\mathbf{S} = \mathbf{R} \operatorname{diag}(\lambda_1, \dots, \lambda_m) \mathbf{R}^{-1},$$

wobei  $\lambda_j$ ,  $j = 1, \dots, m$ , Eigenwerte von **S** und Spalten von **R** (normierten) Eigenvektoren von **S** sind

- Eigenwerte  $\lambda_j$ ,  $j = 1, \dots, m$ , von **S** sind alle positiv und Eigenvektoren von **S** sind orthonormal ( $\Rightarrow \mathbf{R}^{-1} = \mathbf{R}^T$ )

**S** = **TT**<sup>T</sup> mit

$$\mathbf{T} = \mathbf{R} \operatorname{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_m})$$

# Eigenwertzerlegung

## Spezialfall: Zwei Dimensionen

Kovarianzmatrix

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$$

Eigenwertzerlegung (sei  $s = \sin \phi$ ,  $c = \cos \phi$ ,  $\phi = \frac{1}{2} \arctan \frac{\sigma_{xy}}{\sigma_y^2 - \sigma_x^2}$ )

$$\mathbf{T} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}, \quad \begin{aligned} \sigma_1 &= \sqrt{c^2 \sigma_x^2 + s^2 \sigma_y^2 - 2sc \sigma_{xy}}, \\ \sigma_2 &= \sqrt{s^2 \sigma_x^2 + c^2 \sigma_y^2 + 2sc \sigma_{xy}}. \end{aligned}$$

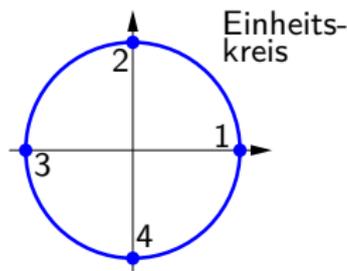
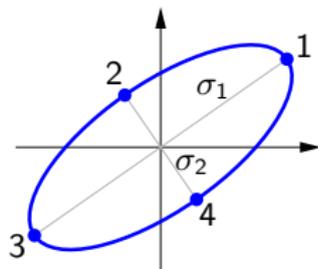


Abbildung mit  $\mathbf{T}$





# Varianz Anpassung

Kovarianzmatrix  $\Sigma$  ist i.A. chromosomenspezifisch  
(Chromosom hat dann  $n + \frac{n(n+1)}{2}$  Gene)

Mutation der Kovarianzen auf Rotationswinkeln und nicht direkt auf Einträgen der Matrix:

$$\varphi'_{ik} = \varphi_{ik} + r \cdot N(0, 1)$$

mit  $r \approx 0.0873$  ( $\approx 5^\circ$ )

$N(0, 1)$  in jedem Schritt neue normalverteilte Zufallszahl

**Nachteile** der korrelierten Mutation:

- deutlich mehr Parameter müssen angepasst werden
  - Varianzen und Rotationswinkel haben keinen direkten Einfluss auf Fitnessfunktion; ihre Anpassung geschieht eher „beiläufig“
- ⇒ fraglich, ob Anpassung der Winkel der Veränderung der eigentlich zu optimierenden Parameter schnell genug folgen kann

# Crossover/Rekombination

**zufällige Auswahl von Komponenten** der Eltern:

$$\begin{array}{l} (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{n-1}, \mathbf{x}_n) \\ (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_{n-1}, \mathbf{y}_n) \end{array} \Rightarrow (\mathbf{x}_1, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{x}_{n-1}, \mathbf{y}_n)$$

entspricht dem **uniformen Crossover**

(im Prinzip auch 1-, 2- oder  $n$ -Punkt-Crossover möglich)

**Mittelung** (blending, intermediäre Rekombination):

$$\begin{array}{l} (x_1, \dots, x_n) \\ (y_1, \dots, y_n) \end{array} \Rightarrow \frac{1}{2}(x_1 + y_1, \dots, x_n + y_n)$$

**Achtung:** bei Mittelung besteht Gefahr des **Jenkins Nightmare**

- völliges Verschwinden jeglicher Verschiedenheit in Population
- durch Mittelung begünstigt, da Gene mittlerem Wert zustreben

# Plus- versus Komma-Strategien

**Vorteil** der +-Strategie:

- wegen des strengen Eliteprinzips: nur Verbesserungen

**Nachteile:**

- Gefahr des Hängenbleibens in lokalen Minima
- für  $(\mu + \lambda)$ -Strategie mit  $\frac{\mu}{\lambda} \geq$  „beste W'keit für erfolgreiche Mutation“ ( $\approx \frac{1}{5}$ ) haben Chromosomen Selektionsvorteil, die ihre Varianz  $\sigma^2$  möglichst klein halten, da nicht genügend große Mutationen durchgeführt werden, um „echte“ Verbesserung zu erreichen („Beinahe-Stagnation“)  
übliche Wahl des Verhältnisses von  $\mu$  zu  $\lambda$  etwa 1:7

Wenn über mehrere Generationen keine Verbesserungen eintritt hilft es vorübergehend auf ,-Strategie umzuschalten um lokale Minima zu überwinden (erhöht Diversität in Population wieder)

# Übersicht

## 1. Evolutionsstrategien

## 2. Evolutionäre Algorithmen zur Verhaltenssimulation

Das Gefangenendilemma

Genetischer Algorithmus

Erweiterungen

# Evolutionäre Algorithmen zur Verhaltenssimulation

bisher: Verwendung von EAs um (numerische oder diskrete) Optimierungsprobleme zu lösen

jetzt: Verwendung von EAs um Verhalten zu simulieren (Populationsdynamik) und Verhaltensstrategien zu finden

## Grundlage: Spieltheorie

- dient der Analyse sozialer und wirtschaftlicher Situationen
- Modellierung von Handlungen als Spielzüge in festgelegtem Rahmen
- wichtigste theoretische Grundlage der Wirtschaftswissenschaften

## Allgemeiner Ansatz:

- kodiere Verhaltensstrategie eines Akteurs in Chromosom
- lasse Akteure miteinander interagieren und bewerte ihren Erfolg
- Akteure vermehren sich oder sterben aus, je nach erzieltm Erfolg

# Das Gefangenendilemma

bekanntestes Problem der Spieltheorie ist das **Gefangenendilemma** (engl.: prisoner's dilemma)

- 2 Personen haben Banküberfall begangen und werden verhaftet
- Beweise reichen nicht aus, um sie in Indizienprozess wegen Banküberfall zu verurteilen
- Beweise reichen jedoch aus, um sie wegen eines geringfügigeren Deliktes (z.B. unerlaubter Waffenbesitz) zu verurteilen (Strafmaß: 1 Jahr Gefängnis)

Angebot des Staatsanwaltes: Kronzeugenregelung

- gesteht einer der beiden, wird er Kronzeuge und nicht verurteilt
- der andere wird mit voller Härte bestraft (10 Jahre Gefängnis)
- Problem: gestehen beide, gilt Kronzeugenregelung nicht da sie beide geständig sind, erhalten sie mildernde Umstände (Strafe: je 5 Jahre Gefängnis)

# Das Gefangenendilemma

Analyse des Gefangenendilemmas durch **Auszahlungsmatrix**:

		B	
		schweigt	gesteht
A	schweigt	-1      -1	-10      0
	gesteht	0      -10	-5      -5

Kooperation (beide schweigen) ist insgesamt am günstigsten

**Aber:** doppeltes Geständnis ist **Nash-Gleichgewicht**:

keine der beiden Seiten kann ihre Auszahlung erhöhen, wenn nur sie ihre Aktion ändert (jede Auszahlungsmatrix hat mind. ein Nash-Gleichgewicht [Nash, 1950])

# Allg. Auszahlungsmatrix des Gefangenendilemmas

	B		
A \	cooperate	defect	
cooperate	R	S	T
defect	T	P	P

**R:** Reward for mutual cooperation    **P:** Punishment for mutual defection  
**T:** Temptation to defect                    **S:** Sucker's payoff

- genaue Werte für **R**, **P**, **T** und **S** sind nicht wichtig
  - es muss aber gelten  $T > R > P > S$  und  $2R > T + S$
2. Bedingung nicht erfüllt  $\Rightarrow$  wechselweises Ausbeuten besser

# Das Gefangenendilemma

Viele Alltagssituation: beschreibbar mit Gefangenendilemma

**aber:** obwohl doppelter Defekt = Nash-Gleichgewicht, auch anderes (kooperatives) Verhalten

Fragestellung (nach [Axelrod, 1980]):

**Unter welchen Bedingungen entsteht Kooperation in einer Welt von Egoisten ohne zentrale Autorität?**

Antwort von [Hobbes, 1651] (Leviathan):

- **Gar nicht!** Ehe staatliche Ordnung existierte, wurde Naturzustand dominiert von egoistischen Individuen, die so rücksichtslos gegeneinander wetteiferten, dass das Leben „solitary, poor, nasty, brutish, and short“ war.
- **aber:** Auf internationaler Ebene gibt es *de facto* keine zentrale Autorität, aber dennoch (wirtschaftliche und politische) Kooperation von Staaten.

# Das Gefangenendilemma

**Ansatz** von [Axelrod, 1980]: **iteriertes Gefangenendilemma**.  
(Gefangenendilemma von 2 Spielern mehrfach hintereinander gespielt, wobei sie vergangenen Züge des jeweils anderen Spielers kennen)

**Idee** dieses Ansatzes:

- wird Gefangenendilemma nur *einmal* gespielt, ist es am günstigsten, Nash-Gleichgewicht zu wählen
- wird es *mehrfach* gespielt, kann ein Spieler auf unkooperatives Verhalten des anderen reagieren  
(Möglichkeit der *Vergeltung* für erlittene Nachteile)

Fragestellungen:

1. **Entsteht im iterierten Gefangenendilemma Kooperation?**
2. **Was ist die beste Strategie im iterierten Gefangenendilemma?**

# Das Gefangenendilemma

[Axelrod, 1980] legte folgende **Auszahlungsmatrix** fest:

		B	
		cooperate	defect
A	cooperate	3, 3	0, 5
	defect	5, 0	1, 1

(kleinsten ganzen Zahlen  $\geq 0$ , die Bedingungen erfüllen)

- Wissenschaftler verschiedener Disziplinen (Psychologie, Sozial- und Politikwissenschaften, Wirtschaftswissenschaften, Mathematik) wurden eingeladen, Programme zu schreiben, die das iterierte Gefangenendilemma spielen
- Programm kann sich eigene und gegnerische Züge merken

# Turniere

Zur Beantwortung beider Fragen führte Axelrod 2 Turniere durch:

## 1. Turnier:

- 14 Programme plus ein Zufallsspieler (Fortran)
- Rundenturnier mit 200 Spielen je Paarung
- Sieger: A. Rapoport mit Tit-for-Tat (Wie du mir, so ich dir)

Programme und Ergebnisse des ersten Turniers wurden veröffentlicht

zu einem zweiten Turnier eingeladen

Idee: Ergebnisanalyse für ggf. bessere Programme

## 2. Turnier:

- 62 Programme plus ein Zufallsspieler (Fortran und Basic)
- Rundenturnier mit 200 Spielen je Paarung
- Sieger: A. Rapoport mit Tit-for-Tat (Wie du mir, so ich dir)

# Tit-for-Tat

Spielstrategie von **Tit-for-Tat** ist *sehr* einfach:

- kooperiere im ersten Spiel (spiele C)
- mache in allen folgenden Spielen den Zug des Gegners aus dem direkt vorangehenden Spiel

**beachte:** reines Tit-for-Tat ist nicht unbedingt beste Strategie, wenn gegen *einzelne* andere Strategien gespielt wird

- nur wenn es in Population Individuen gibt, mit denen Tit-for-Tat kooperieren kann, schneidet es insgesamt sehr gut ab
- **Problem** von Tit-for-Tat: Es ist **anfällig für Fehler** — spielt Tit-for-Tat gegen Tit-for-Tat und spielt einer der beiden Spieler „aus Versehen“ Defekt, so kommt es zu wechselseitigen Vergeltungsschlägen

Eine wichtige Alternative ist **Tit-for-Two-Tat**:

schlage erst nach zweimaligem Defekt des Gegners zurück

# Genetischer Algorithmus

**Kodierung der Spielstrategien:** [Axelrod, 1987]

betrachte alle Spielverläufe der Länge 3 ( $2^6 = 64$  Möglichkeiten)  
speichere für jeden Spielverlauf den im nächsten Spiel  
auszuführenden Zug (C – cooperate, D – defect, in 1 Bit):

		1. Spiel	2. Spiel	3. Spiel
1. Bit:	Antwort auf	(C,C),	(C,C),	(C,C):
2. Bit:	Antwort auf	(C,C),	(C,C),	(C,D):
3. Bit:	Antwort auf	(C,C),	(C,C),	(D,C):
⋮		⋮		⋮
64. Bit:	Antwort auf	(D,D),	(D,D),	(D,D):

C
D
C
⋮
D

(1. und 2. Element jedes Paares: eigener bzw. gegnerischer Zug)

Zusätzlich: 6 Bit zur Kodierung des Spielverlaufs vorm 1. Zug  
jedes Chromosom hat 70 binäre Gene (jeweils C oder D)

# Genetischer Algorithmus: Ablauf

initialisiere Anfangspopulation mit zufälligen Bitfolgen (70 Bit)

aus aktueller Population: wähle zufällig Paare von Individuen

sie spielen 200-mal Gefangendilemma gegeneinander

für die ersten 3 Spiele: nutze (ein Teil des) im Chromosom

abgespeicherten Anfangsspielverlaufs, um Zug zu bestimmen

(fehlende/zu kurze Historie wird ersetzt/aufgefüllt)

jedes Individuum spielt gegen gleiche Anzahl von Gegnern

(aus Rechenzeitgründen – 1987! – kein volles Rundenturnier)

Auswahl von Individuen für nächste Generation:

überdurchschnittliches Ergebnis ( $x \geq \mu + \sigma$ ): 2 Kinder

durchschnittliches Ergebnis ( $\mu - \sigma < x < \mu + \sigma$ ): 1 Kinder

unterdurchschnittliches Ergebnis ( $\mu - \sigma \geq x$ ): 0 Kinder

genetische Operatoren: Binärmutation, 1-Punkt-Crossover

## Genetischer Algorithmus: Ergebnis

sich ergebende Strategien sind **Tit-for-Tat** sehr ähnlich  
[Axelrod, 1987] identifizierte folgende allgemeine Muster:

**Don't rock the boat:** kooperiere nach drei Kooperationen  
(C,C), (C,C), (C,C)  $\rightarrow$  C

**Be provokable:** spiele Defekt nach plötzlichem Defekt des  
Gegners

(C,C), (C,C), (C,D)  $\rightarrow$  D

**Accept an apology:** kooperiere nach wechselseitiger Ausbeutung  
(C,C), (C,D), (D,C)  $\rightarrow$  C

**Forget:** (sei nicht nachtragend:) kooperiere nachdem Kooperation  
nach Ausbeutung wiederhergestellt (auch ohne Vergeltung)

(C,C), (C,D), (C,C)  $\rightarrow$  C

**Accept a rut:** (rut *fig.* ausgefahrenes Gleis, alter Trott)  
spiele Defekt nach dreimaligem Defekt des Gegners

(D,D), (D,D), (D,D)  $\rightarrow$  D

## Das Gefangenendilemma: Erweiterungen

- in Praxis: Auswirkungen von Handlungen nicht immer perfekt beobachtbar
- nicht genauer Zug des Gegners, sondern nur W'keiten sind bekannt
- oft  $\geq 2$  Akteure beteiligt: Mehr-Personen-Gefangenendilemma
- Berücksichtigung längerer Spielverläufe (mehr als drei Spiele)
- Hinzunahme einer Zufallskomponente für die Wahl des Spielzugs: W'keiten für Wahl von C und D statt eines festen Zuges
- Beschreibung der Spielstrategie durch Moore-Automaten oder allgemeine Programme, die dann in GA verändert werden

# Literatur zur Lehrveranstaltung I

-  Axelrod, R. (1980).  
More effective choice in the prisoner's dilemma.  
*Journal of Conflict Resolution*, 24:379–403.
-  Axelrod, R. (1987).  
The evolution of strategies in the iterated prisoner's dilemma.  
In Davis, L., editor, *Genetic Algorithms and Simulated Annealing*,  
pages 32–41. Morgan Kaufman, Los Altos, CA, USA.
-  Bäck, T. and Schwefel, H. (1993).  
An overview of evolutionary algorithms for parameter  
optimization.  
*Evolutionary Computation*, 1(1):1–23.
-  Hobbes, T. (1651).  
*Leviathan*.

## Literatur zur Lehrveranstaltung II

-  Nash, J. F. (1950).  
*Non-cooperative games.*  
PhD thesis, Princeton University.
  
-  Nissen, V. (1997).  
*Einführung in evolutionäre Algorithmen: Optimierung nach dem Vorbild der Evolution.*  
Vieweg, Braunschweig/Wiesbaden, Germany.
  
-  Rechenberg, I. (1973).  
*Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.*  
Fromman-Holzboog Verlag, Stuttgart, Germany.