

# Evolutionäre Algorithmen

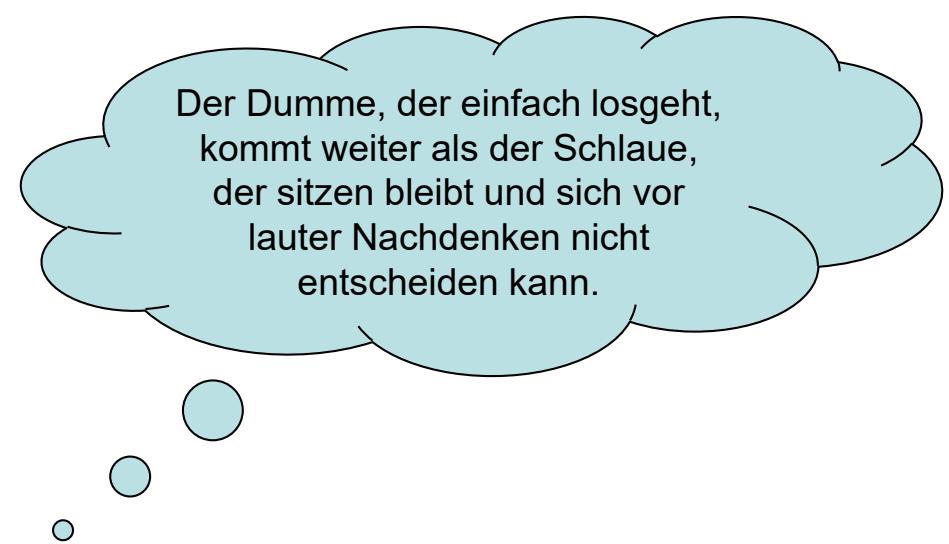
## Im Modul Advanced Algorithms

Prof. Dr. Katja Wengler

# Vorstellungsrunde

# Einleitung

## Evolutionäre Algorithmen



## Beispielproblem: Fahrplanauskunft

Anfrage: Nächste Verbindung von Karlsruhe Hbf bis Stuttgart

Problemfragen:

- Was genau möchte der Kunde?
  - Nächste Verbindung?
  - Kürzeste Verbindung?
  - Billigste Verbindung?
  - Wenig Umsteigen?

Lösung: Suchen nach dem (optimalen) Weg zum Ziel

---

## Suchen als Methodik der Informatik

### Suchen als Wiederfinden

- Datenbank
- Suchmaschine
- Knowledge Management
- Textsammlung

Vorgehen:

- Suche nach ähnlichen Begriffen
- Suchen nach Inhalten
- Erinnern als Wiederfinden oder als Rekonstruktion

Hilfsmittel: Suchbäume, Hashtabellen, ...

### Suchen als Problemlösen

- Existiert eine Lösung?
- Finde eine Lösung
- Was ist die beste Lösung? → Optimierung

Vorgehen:

- Systematisches Probieren
- z.B. Suchen in Graphen
  - Wege, Rundreise, Zyklen, spannende Bäume, ...

Systematisches Probieren

---

## Suchen als Problemlösen → Metaheuristik

- Rechentechnik zum näherungsweisen Lösen von numerischen und kombinatorischen Optimierungsproblemen in mehreren Schritten
- Vorgehen:
  - Abstrakte Folge von Operationen auf bestimmte Objekte
  - Schrittweise Verbesserung der Lösungskandidaten
  - **Geführte Zufallssuche** im Raum der Lösungskandidaten
  - Ende, wenn Abbruchkriterien erreicht (*anytime algorithm*)
- Anwendung:
  - Lassen sich auf fast beliebige Probleme anwenden
  - Probleme, die keinen effizienten Lösungsalgorithmus haben
- Schwierigkeit:
  - geeignete Abbildung des Problems auf abstrakte Strukturen und Operationen

## Lernziele der Vorlesung

Ziel: Problemlösung!

Sie lernen:

- Modellierung von komplexen Problemen
  - Naturinspirierte Algorithmen und Meta-Heuristiken
  - Evolutionäre Algorithmen und deren Varianten
  - Auswahl des geeigneten Algorithmus zur Problemlösung
  - Evaluierung der Algorithmen
  - Anwendung auf unterschiedliche Problemkategorien
-

## Inhalt der Vorlesung

- Einleitung und Überblick
- Grundkonzepte der natürlichen Evolution
- Optimierungsprobleme im Allgemeinen
- Standardprinzipien und Visualisierung
- Varianten von Evolutionären Algorithmen
- Übungen

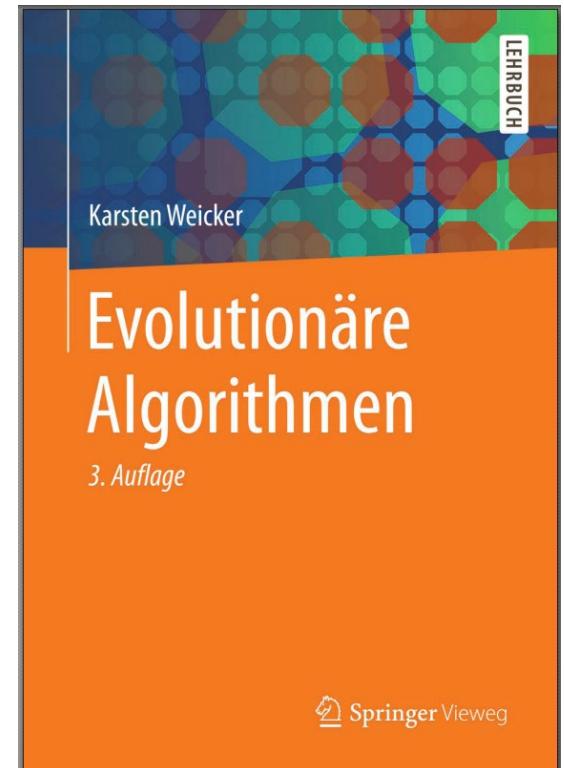


## Ablauf der Vorlesung

- Zeitlicher Rahmen: 16 Vorlesungsstunden
- Prüfung: kombinierte Prüfungsleistung
  - **Gruppenarbeit: 50%**
    - Einarbeiten in eine Software bzw. Lesen einer Fachpublikation, aneignen der Expertise bis zur nächsten Vorlesung und Begleitung der Lösungsfindung der Kommilitonen am letzten Vorlesungstag
  - **Mündliche Prüfung: 50%**
    - 15min zu Inhalten der Vorlesung
- Wechsel zwischen theoretischem Input und praktischer Anwendung

## Literatur

- Darwin, C. (1859). *On The Origin of Species by Means of natural Selection or the Preservation of favoured Races in the Struggle for Life*. John Murray.
- Weicker, K. (2015). *Evolutionäre Algorithmen*. Teubner Verlag.
- Pohlheim, H. (2000). *Evolutionäre Algorithmen – Verfahren, Operatoren und Hinweise für die Praxis*. Springer



## Weitere Informationsquellen

- Zeitschriften:
  - Evolutionary Computation, MIT Press
  - IEEE Transactions on Evolutionary Computation, IEEE Press
  - Journal of Heuristics, Springer
- Konferenzen:
  - Genetic and Evolutionary Computation Conference (GECCO)
  - Congress on Evolutionary Computation (CEC)
  - Parallel Problem Solving from Nature (PPSN)
  - Evo Star
  - Evolutionary Multi-criterion Optimization(EMO)
- Weitere Metaheuristiken:
  - Ant Colony Optimization: <http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>
  - Particle Swarm Optimization: <http://www.particleswarm.info/>
  - Traveling Salesman Problem: <http://www.tsp.gatech.edu/index.html>

# Überblick

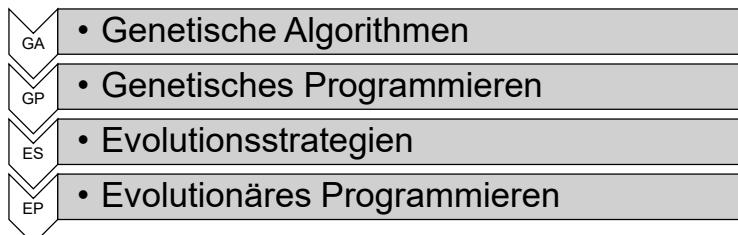
Evolutionäre Algorithmen

---

# Evolutionäre Algorithmen      vs.      Maschinelles Lernen

- Stochastische heuristische Optimierungs- und Suchverfahren, angelehnt an die biologische Evolution
- Nur geringes Problemwissen erforderlich
- Lösung vielfältiger Probleme für die keine spezifischen Algorithmen existieren

- Einsatzgebiete:
  - Konstruktion komplexer Bauteile, Erstellung von Fahr- / Stunden- oder Raumplänen
- Methoden:



- Künstliche Erzeugung von Wissen
- Computer lernt anhand von:

## **Beispieldaten + Beispielergebnisse → Erfahrung**

- Erkennt Gesetzmäßigkeiten der Beispieldaten
  - Beispiele werden nicht auswendig gelernt
  - Unbekannte Daten gleicher Art können beurteilt werden
- Einsatzgebiete:
    - Automatisch Diagnoseverfahren, Erkennung von Sprache, Schrift, Gesichtern, Klassifizierung
  - Methoden:
    - Support Vector Machine (SVM), Hidden Markov Models (HMM), Principal Component Analysis (PCA), ...

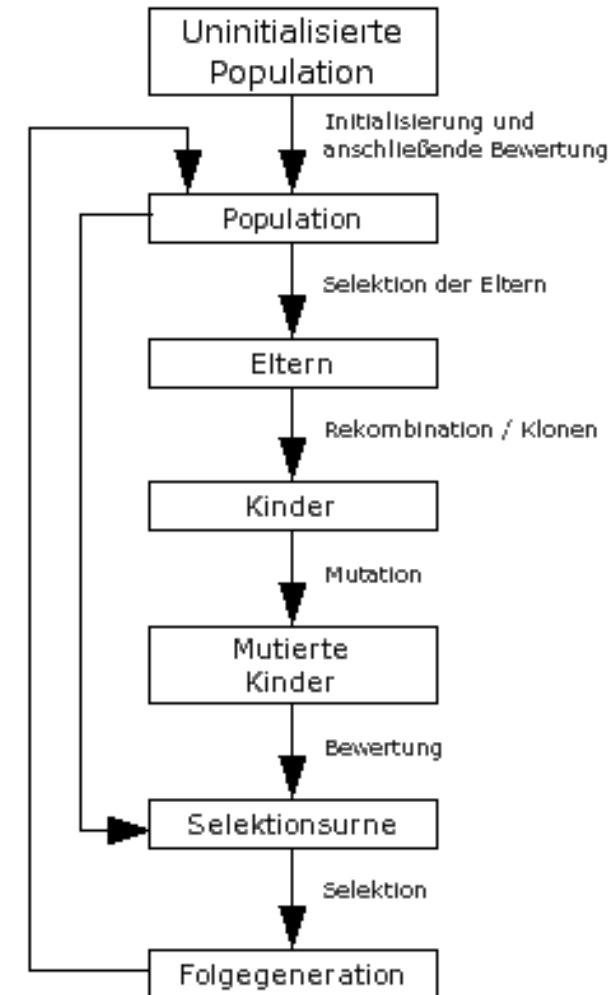
## Evolutionäre Algorithmen

- Stochastische Suchverfahren (heuristisch)
- Seit den 1950er Jahren dient die natürliche Evolution als Vorbild für die Lösung von Optimierungsproblemen
- Abstraktion der Mechanismen der natürlichen Evolution
  - Population – Sammlung von Lösungskandidaten = Individuen
  - Simulierte Evolution – Wechselspiel zwischen Modifikation und Auswahl besserer Individuen unter Verwendung von:
    - Mutation
    - Rekombination
    - Selektion

## Evolutionäre Algorithmen - Grundprinzip

Nachahmung der Evolution

- Durch Veränderung des Erbgutes werden hochkomplexe Lebensformen an (Veränderungen der) Umwelt angepasst

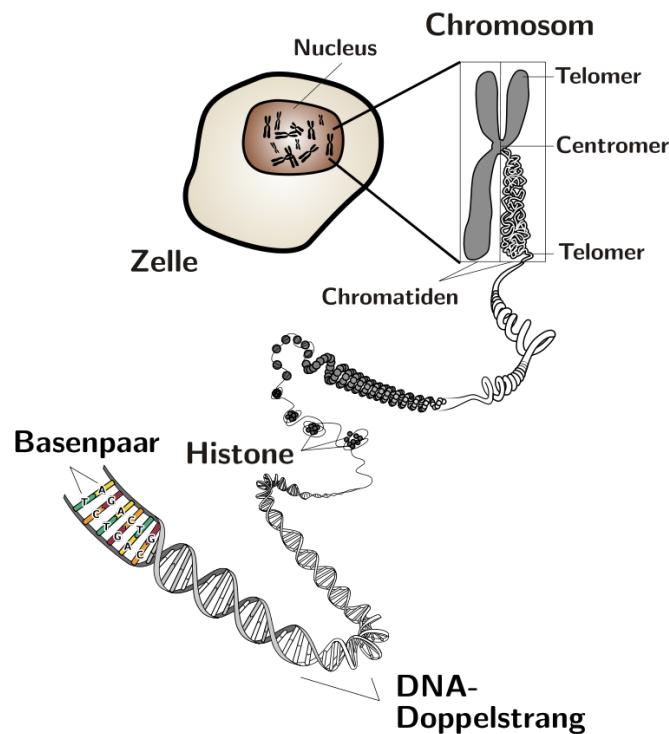


# Grundkonzepte der natürlichen Evolution

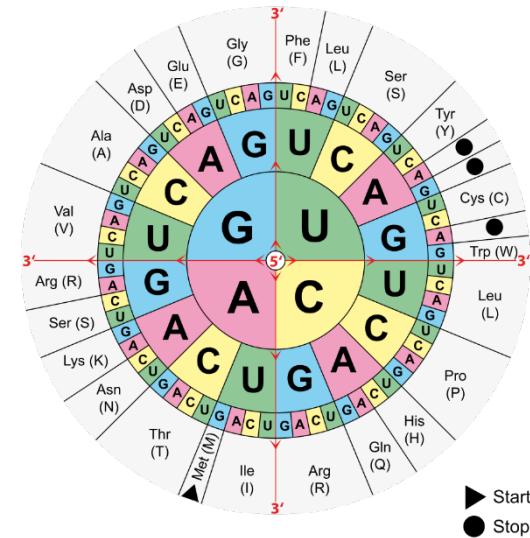
Evolutionäre Algorithmen

---

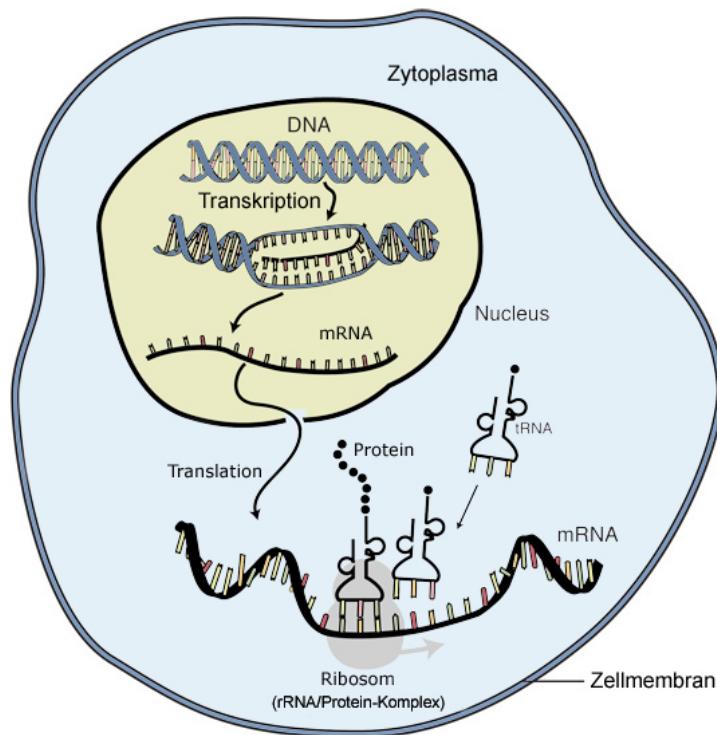
# Grundprinzipien



- Jede Zelle enthält alle Chromosomen
- Die DNA besteht aus 4 Nukleotiden
- Ein DNA Abschnitt, der als Bauplan für ein Protein genutzt wird, ist ein Gen

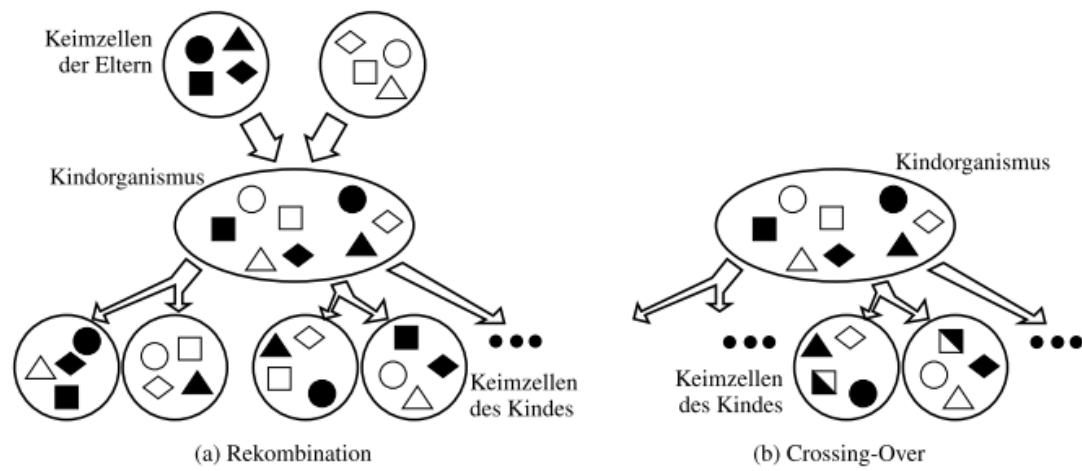


## Grundprinzipien



- Zur Synthese von Proteinen wird das Gen kopiert → mRNA (3 Ribonukleotide = eine der 20 Aminosäuren)
- Proteine in der jeweiligen Zelle bestimmen Struktur und Verhalten dieser Zelle
- Vervielfältigung der RNA-Moleküle arbeitet nicht fehlerfrei (Mutationsrate)

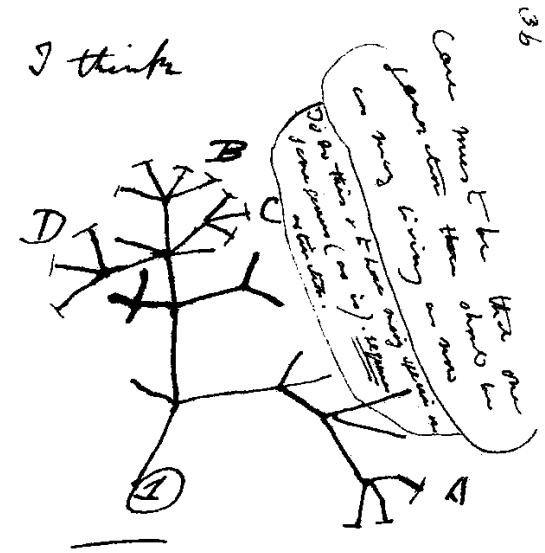
## Grundprinzipien



- **Nachwuchs:** Erbgut zweier Organismen wird vermischt → Verdopplung der Chromosomen und **Rekombination**
- **Crossing-Over:** Vermischung von Chromosomen an bestimmten Bruchstellen

## Biologische Evolution

- Idee von Charles Darwin entwickelt („Über die Entstehung der Arten“, 1859): **Aus zufälliger Variation hervorgehende vorteilhafte Eigenschaften werden durch natürliche Auslese bevorzugt.**
- „**Der Stärkere überlebt**“
- Prinzip der Selektion zeitgleich von Alfred Russel Wallace entwickelt



There between A + B. varies  
less & selection. C + B. The  
finer gradation, B & D  
rather greater distinction.  
Then genera would be  
formed. - binary division

## Biologische Evolution - Prinzipien

### Diversität

- Lebewesen sind verschieden (Vielfalt)

### Variation

- Durch Mutation und Rekombination
- Ökologische Nischen

### Anpassung

- Natürliche Auslese
- Evolution: Nicht nur Organismen, sondern auch Mechanismen, wie z.B. Sterberaten, Mutationsschritte, etc.

### Zufälligkeit

- Keine Ausrichtung auf bestimmte Merkmale
- **Nicht vorhersagbar**
- Irreversibel

### Gradualismus

- Veränderungen in kleinen Stufen
- Wachsende Komplexität

### Opportunismus

- Evolutive Prozesse arbeiten nur mit dem, was vorhanden ist, nicht mit dem, was es einmal gab oder geben könnte

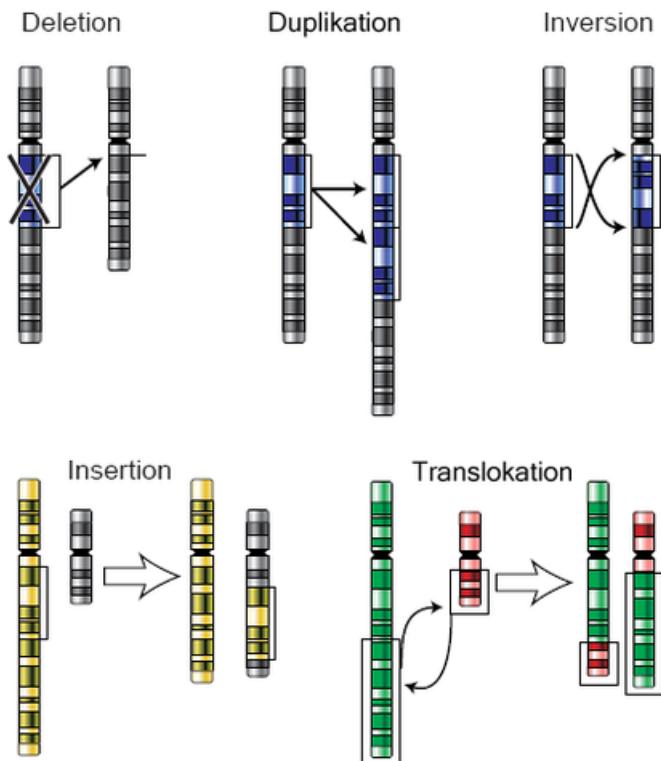
## Evolutionsfaktoren

- Genotyp (Genom) + Phänotyp = **Individuum**
- Ein Gen kann verschiedene Werte annehmen = **Allel**
- Gesamtheit aller Allele = **Genpool**
- **Evolutionsfaktoren** verändern die Häufigkeit von Allelen = die Gensequenz
  - **Mutationen**
  - **Selektion**
  - **Gendrift**
  - **Genfluss**

## Evolutionsfaktor - Mutation

### Zufällige Veränderung

- **Chromosomenmutation:** brechen und werden falsch zusammengesetzt bzw. einzelne Stücke gehen verloren
- **Genommutation:** Veränderung der Gesamtzahl de Chromosomen
- **Genmutation (Punktmutation):** Austausch / Verlust / Einschub einzelner Nukleotide in einem Gen



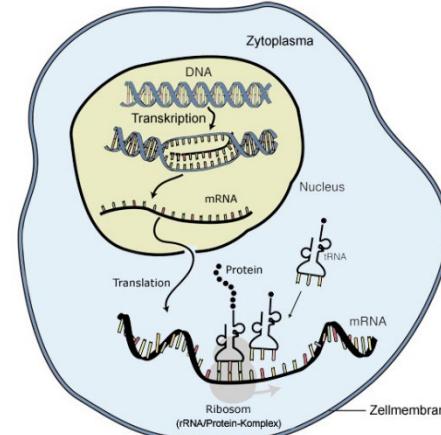
## Evolutionsfaktor - Mutation

**Stille Mutation:** Der Basenaustausch hat keine Folgen, da die gleiche Aminosäure codiert wird.

**Sinnmutation:** Es wird eine andere Aminosäure codiert. Der Austausch bleibt aber ohne Folgen, weil die neue Aminosäure die Proteinstruktur ohne Auswirkungen beeinflusst.

**Fehlsinnmutation:** Durch die neue Aminosäure wird der Proteinaufbau so beeinflusst, dass es zu Fehlfunktion kommt.

**Nichtsinnmutation:** Durch diese Mutation entsteht ein Stoppcodon. Es kommt zum Syntheseabbruch.



DNA	GGA CCT	GGT CCA	GCA CGT	GAA CTT	TGA ACT
-----	------------	------------	------------	------------	------------

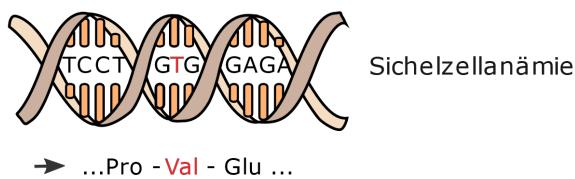
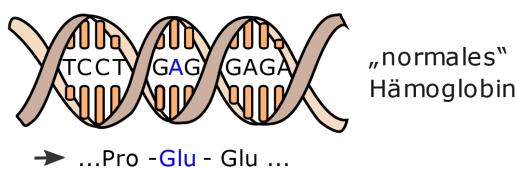
mRNA	GGA	GGU	GCA	GAA	UGA
------	-----	-----	-----	-----	-----

Aminosäure	Gly	Gly	Ala	Glu	Stop
------------	-----	-----	-----	-----	------

Auswirkung	normal	stille Mutation	Sinn-mutation	Fehlsinn-mutation	Nichtsinn-mutation
------------	--------	-----------------	---------------	-------------------	--------------------

# Evolutionsfaktor – Mutation → Auswirkungen

## 1. Punktmutation



## 2. Deletion

...ATCATC**CTT**TGGT... „normales“ CFTR  
 ...ATCATCTGGT... Mukoviszidose

## 3. Tripplett-Expansion

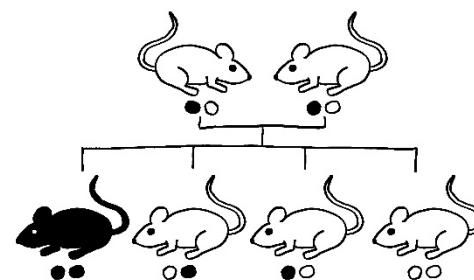
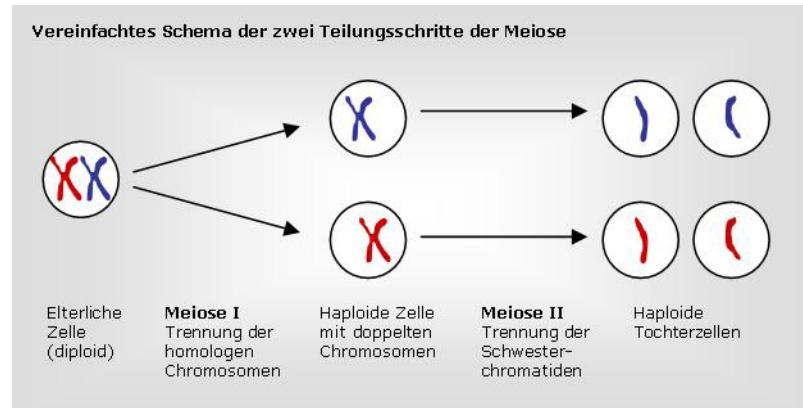
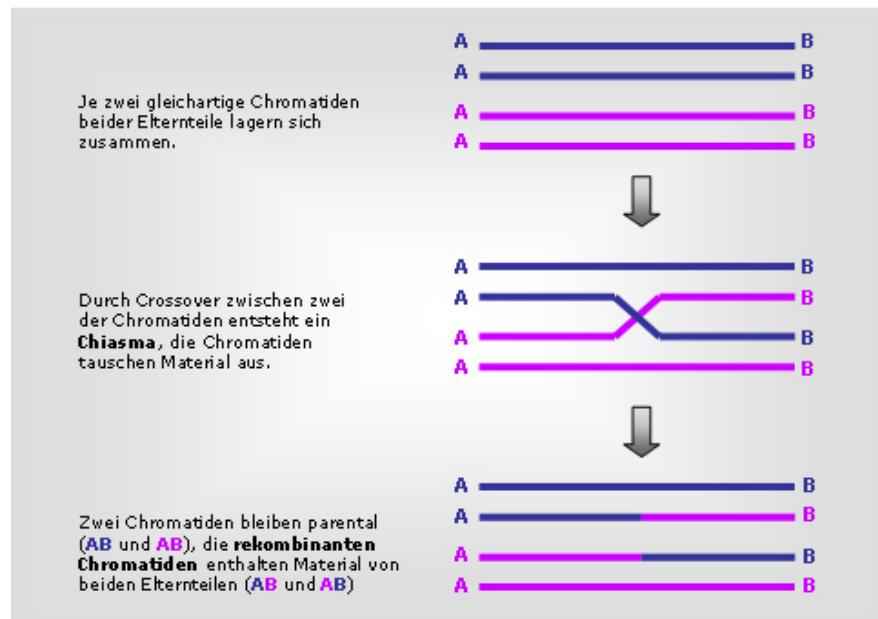
...TTC **CAG** ... CAG CAA... „normales“ Huntington  
 bis zu 34x  
 ...TTC **CAG...CAG...CAG** CAA... Chorea Huntington  
 35x und mehr

## 4. copy number variation



© wissenschaftsschau.de

## (Evolutionsfaktor) - Rekombination



## Evolutionsfaktor - Selektion

- Veränderung der Allelen-Häufigkeit durch unterschiedlich viele Nachkommen der einzelnen Allele:
  - Überlebenschancen → Umweltselektion,
  - Fähigkeit, einen Geschlechtspartner zu finden → sexuelle Selektion,
  - Fruchtbarkeit bzw. Fortpflanzungsraten oder
  - Länge der Generationsdauer (z.B. in Form des Zeitpunkts der Geschlechtsreife)
- Durch Selektionswert bzw. Fitnesswert gemessen:

$$\text{Fitness}(G) = \frac{\#\text{Nachkommen von } G}{\#\text{Nachkommen von } G'}$$



## Weitere Evolutionsfaktoren

### Genfluss

- Genhäufigkeiten in der Population werden direkt durch Zu- oder Abwanderung von Individuen einer anderen Population derselben Art verändert

### Gendrift

- insbesondere bei kleinen Populationen
- Aufgrund von Zufallseffekten sterben Allele einzelner Gene aus.
- somit deutliche Reduktion der Vielfalt in einer Population

## Übung

Beschreiben Sie einen passenden Algorithmus, um Schweine mit besonders großen Ohren zu züchten.

Zur Erinnerung - Vorgehen:

- Abstrakte Folge von Operationen auf bestimmte Objekte
  - Schrittweise Verbesserung der Lösungskandidaten
  - **Geführte Zufallssuche** im Raum der Lösungskandidaten
  - Ende, wenn Abbruchkriterien erreicht (*anytime algorithm*)
-

# Optimierungsprobleme

Evolutionäre Algorithmen

---

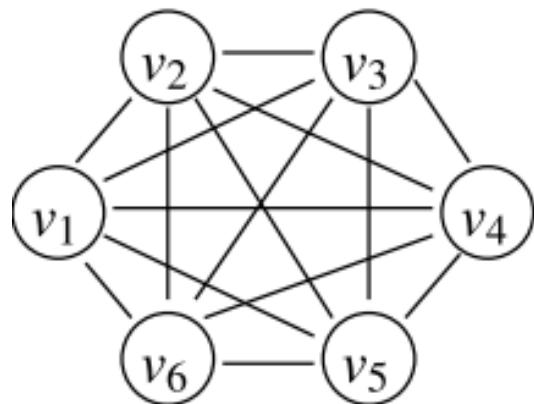
## Das *Travelling Salesman Problem*

**Route zwischen n Städten, die jede Stadt genau einmal besucht**

- NP-vollständig
- unwahrscheinlich effiziente Algorithmen zur Lösung dieses Problems zu finden
- Der „primitive“ Algorithmus, der einfach alle Möglichkeiten durchprobiert hat eine Komplexität von  $O(n!)$
- häufig „völlig ausreichend“ eine „gute“ Lösung zu finden



## Beispiel – Handlungsreisendenproblem



Kante $e$	$\gamma(e)$	Kante $e$	$\gamma(e)$	Kante $e$	$\gamma(e)$
$(v_1, v_2)$	5	$(v_2, v_3)$	10	$(v_3, v_5)$	17
$(v_1, v_3)$	8	$(v_2, v_4)$	4	$(v_3, v_6)$	8
$(v_1, v_4)$	11	$(v_2, v_5)$	9	$(v_4, v_5)$	6
$(v_1, v_5)$	3	$(v_2, v_6)$	12	$(v_4, v_6)$	5
$(v_1, v_6)$	7	$(v_3, v_4)$	6	$(v_5, v_6)$	11

Graph  $G = (V, E, \gamma)$

Knotenmenge  $V = \{v_1, \dots, v_n\}$

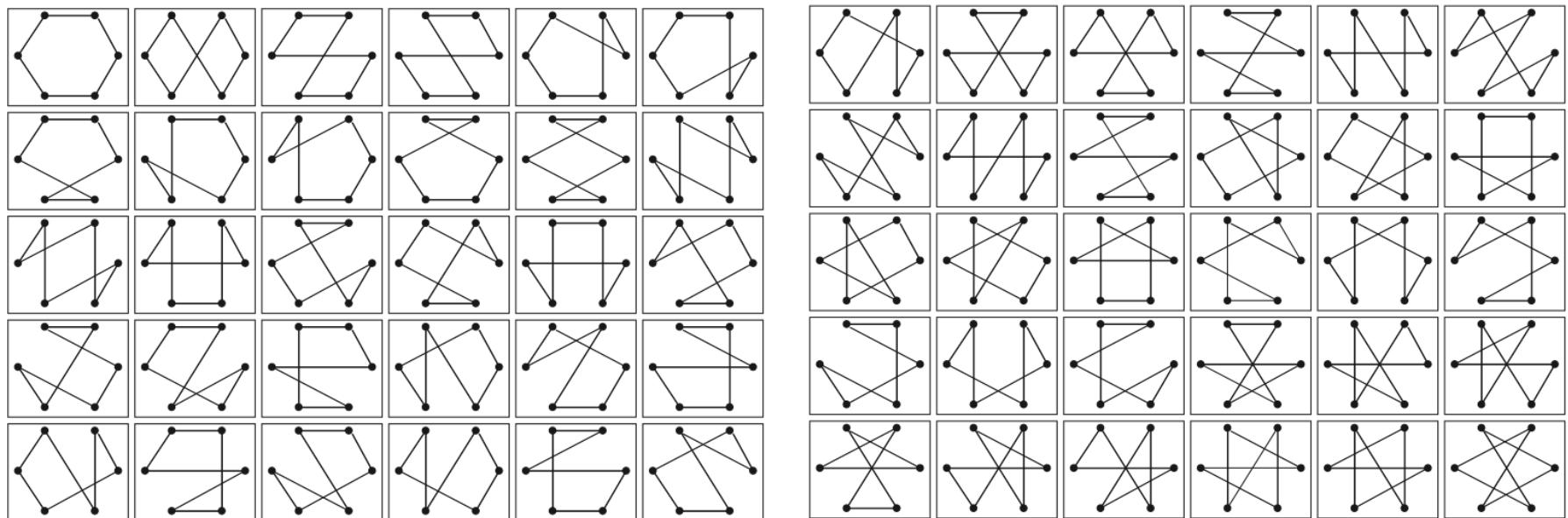
Kantenmenge  $E \subseteq V \times V$

Jede Straße hat Fahrtzeit:  $\gamma: E \rightarrow \mathbb{R}$

Problem: Tupel mit unterschiedlichen Besuchsreihenfolgen → Fahrtzeit minimieren

## Beispiel – Handlungsreisendenproblem

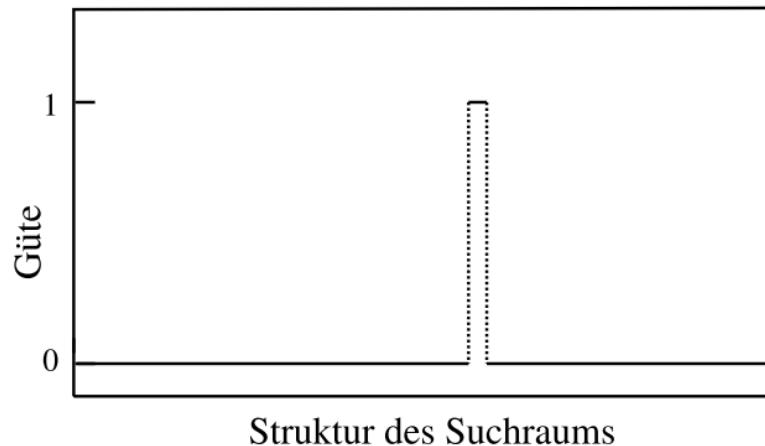
### Problemraum / Suchraum



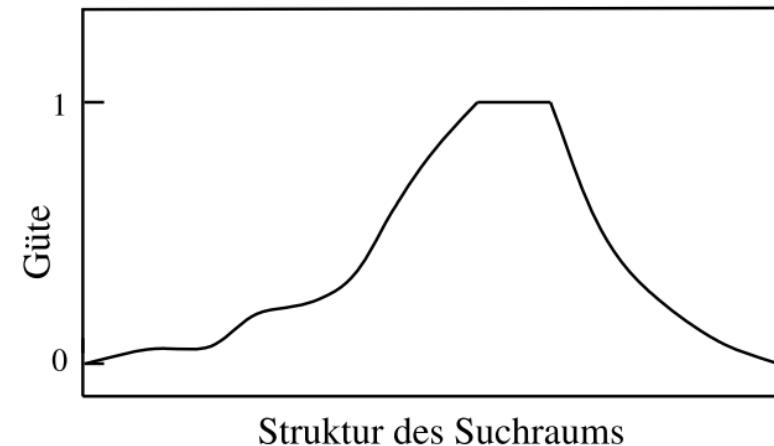
## Lösungsansätze für Optimierungsproblem

Brainstorming: Welche Ansätze fallen Ihnen ein?

## Gute Bewertungsfunktionen



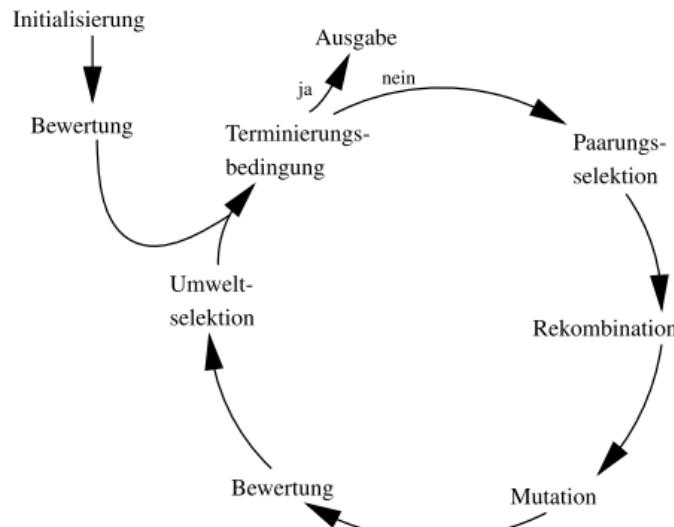
(a) Nadel im Heuhaufen



(b) leitende Bewertungsfunktion

# Optimierung mit Hilfe der Evolution

*Individuen* = Lösungskandidaten:  $A, B, \dots$



*Population*:  $P = \langle A^{(i)} \rangle_{1 \leq i \leq s}$

Population nach einer Iteration = neue Generation

*Terminierung*:

- Schwellwert für erwünschte Güte
- Anzahl der Generationen ohne Verbesserung und / oder
- maximale Anzahl an Iterationen

## Verwendung der biochemischen Begriffe

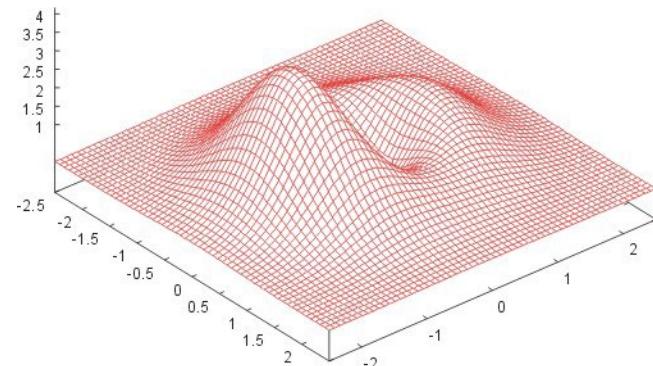
Begriff	Evolutionäre Algorithmen
Population	Sammlung mehrerer Individuen, meist allerdings mit fester Größe
Individuum	Lösungskandidat für ein Optimierungsproblem
Genotyp	steht analog für die gesamte im Individuum gespeicherte und durch die Evolution manipulierbare Information
Phänotyp	steht für das Individuum aus der Sicht des Optimierungsproblems
Mutation	steht für eine kleine Änderung am Genotyp
Rekombination	ist eine Operation, die zwei oder mehr Individuen miteinander kombiniert
Fitness	spezielle Berechnungsvorschrift als Hilfsmittel bei der Selektion = Güte eines Individuum
genetischer Code	meist direkte Abbildung (Dekodierungsfunktion) vom Genotyp auf den Phänotyp

## Übung

Finden Sie für die gegebene Funktion  
das Maximum.

$$f(x) = (x^2 + 2.5y^2 - y) * e^{1-(x^2+y^2)}$$

Partnerarbeit: Überlegen Sie sich zu zweit eine Mutation und eine Rekombination für diesen Optimierungsproblem.

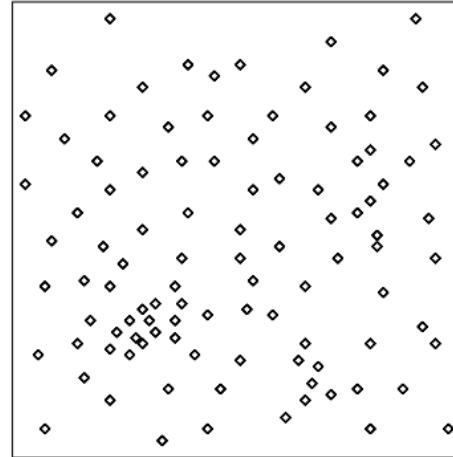


## Übung 2

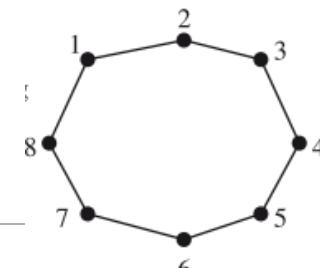
Handlungsreisendenproblem für 101 Städte

Ziel: schnell und mit ausreichender Qualität lösen

Vollständige Suche:  
 $4,6631 \cdot 10^{157}$  Rundreisen



Gruppenarbeit: Überlegen Sie sich eine Mutationen, eine Rekombination und eine Selektion für dieses Problem am Bsp. von 8 Städten. Präsentieren Sie Ihre Lösung.



Ausgangsindividuum

# Was unterscheidet EAs von traditionellen Optimierungsverfahren?

## Wann sollten EAs angewendet werden?

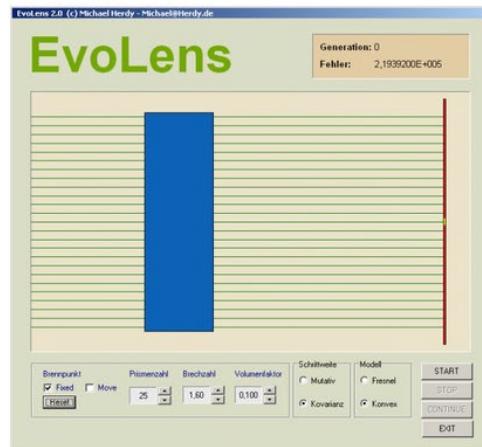
- Auf (fast) alle Probleme anwendbar
- Bei stark nichtlinearen oder diskontinuierlichen Zielfunktionen
- Bei Problemen mit unterschiedlichen Daten (binär, ganzzahlig, reell)
- ABER keine Erfolgs- oder Laufzeitgarantie

## Wann sollten EAs NICHT angewendet werden?

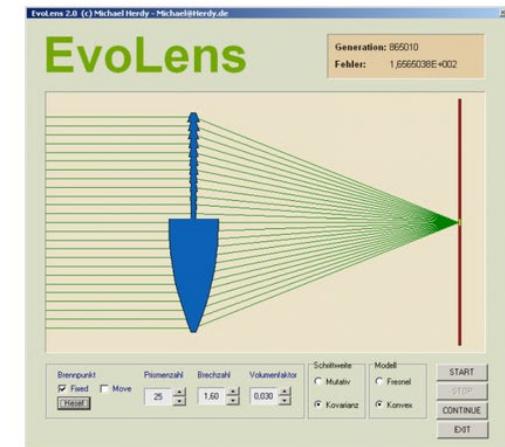
- KEIN Ersatz für klassische Optimierungsverfahren (nicht anwenden, wenn spezielles Lösungsverfahren vorhanden) oder
  - Zielfunktion sehr aufwendig zu berechnen ist
-

# Übung

- Welche Anwendungsgebiete fallen Ihnen ein?
- Nutzen Sie Brainstorming in einer Gruppe von bis zu 4 Studierenden.



Startkonfiguration für die evolutionäre Optimierung einer Linse ist eine Glasscheibe, auf deren blauer Kante wir hier schauen. Von links fällt ein Lichtbündel auf den Glaskörper, das auf der roten Leinwand fokussiert sein soll.



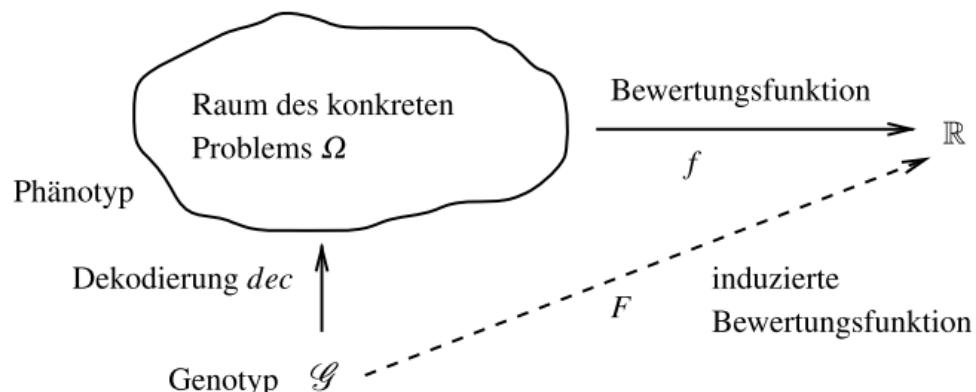
Linsenoptimierung mit EvoLens. Ergebnis der beiden Optimierungsläufe übereinander dargestellt. Die Fresnel-Linse im oberen Bildteil, die mit einem Modell mit mehr Freiheitsgraden optimiert wurde, besitzt nur ca. 20 % des Gewichts der Konvexlinse im unteren Bildteil.

# Formale Beschreibung

Evolutionäre Algorithmen

---

## Darstellung des Suchraums

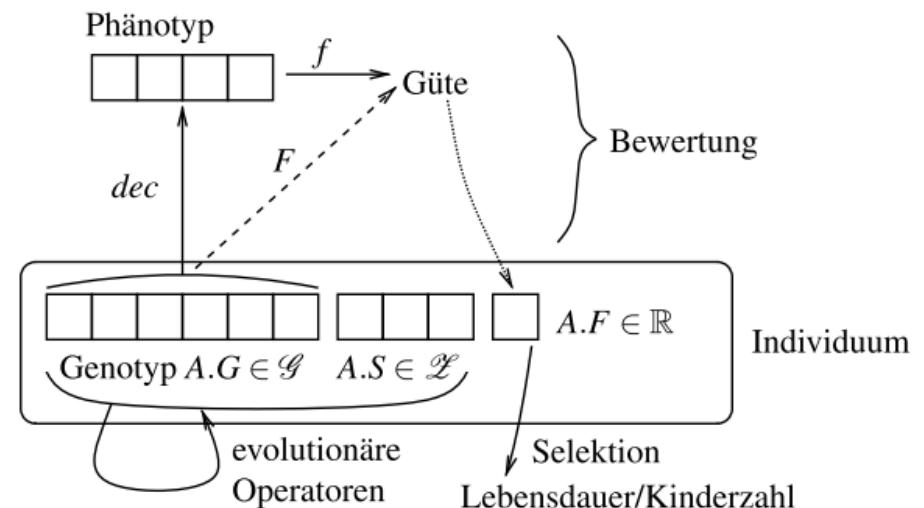


- Trennung **Phänotyp**  $\Omega$  (= natürliche Struktur des Suchraums) und **Genotyp**  $\mathcal{G}$  (Darstellung eines Lösungskandidaten in einem Individuum)
- Bewertungsfunktion ist auf Phänotyp definiert
- Dekodierung  $dec$ :  $\mathcal{G} \rightarrow \Omega$  (Abbildung vom Genotyp auf den Phänotyp) - optional

## Definition Individuum

Ein Individuum A ist ein Tupel  
 $(A.G, A.S, A.F)$  bestehend aus:

- **Genotyp:**  $A.G \in \mathcal{G}$  entspricht der genotypische Information
- Eventuell **Zusatzinformationen:**  $A.S \in \mathcal{Z}$  (= Raum aller möglichen Zusatzinformationen)
- **Güte:**  $A.F \in \mathbb{R}$



# Prinzipien

Evolutionäre Algorithmen

---

## Binäres Hillclimbing

- Binäre Zeichenketten

- $(b_1, \dots, b_l) \mapsto \sum_{1 \leq i \leq l} g(b_i, \hat{b}_i)$  mit  $g(b_i, \hat{b}_i) = \begin{cases} 1, & \text{falls } b_i = \hat{b}_i \\ 0, & \text{sonst} \end{cases}$
  - Zielfunktion wird maximiert

- Ein-Bit-Binäre Mutation

---

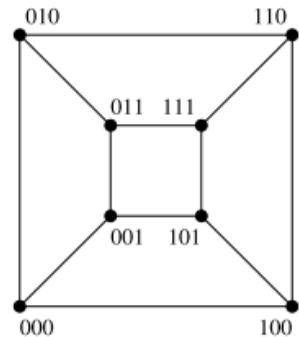
```
EIN-BIT-BINÄRE-MUTATION( Individuum A mit  $A.G \in \mathbb{B}^\ell$  )
1    $B \leftarrow A$ 
2    $i \leftarrow$  wähle zufällig gemäß  $U(\{1, \dots, \ell\})$ 
3    $B_i \leftarrow 1 - A_i$ 
4   return B
```

---

- **Population:** ein Individuum
- **Nächste Generation:**
  - ein neues Individuum durch Mutation erzeugt
  - Selektion: das bessere von beiden, beide gleich → Kind ersetze Elternindividuum

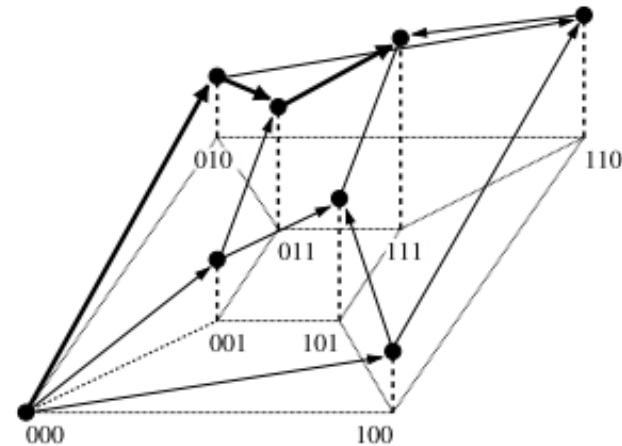
## Hillclimbing am Beispiel

Nachbarschaftsgraph für Mutation:



1 Kante = 1 Mutation (genau 1 Bit verändert)

Gütelandschaft auf  
Nachbarschaftsgraph



Selektion durch Pfeile dargestellt

## Hillclimbing - Einflüsse

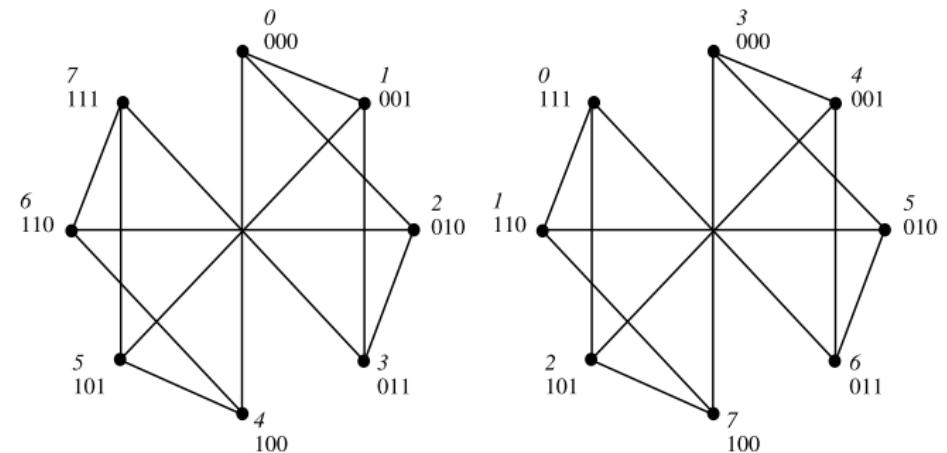
- Lokale Optima unüberwindbar, globales nicht mehr zu finden
- Plateaus können Optimierung behindern, da Richtungsinformationen fehlen, um Mutationen auf bessere Lösungskandidaten zu lenken → **random walk**

## Hillclimbing – Einflüsse – Random Walk

- Beispiel: geg: Zahlen von 0-7 = 3 Bits, ges: max

$$f_1(x) = \begin{cases} x + 3 & \text{falls } x < 5 \\ 7 - x & \text{sonst} \end{cases}$$

Bitmuster	000	001	010	011	100	101	110	111
dekodiert	0	1	2	3	4	5	6	7
in Funktion $f_1$	3	4	5	6	7	2	1	0



## Hillclimbing - Einflüsse

- Lokale Optima unüberwindbar, globales nicht mehr zu finden
- Plateaus können Optimierung behindern, da Richtungsinformationen fehlen, um Mutationen auf bessere Lösungskandidaten zu lenken → **random walk**
- **Hamming-Abstand:** Anzahl Änderungen, um eine Binärzahl in eine andere umzuwandeln, z.B.  $011 \rightarrow 100$  = Hamming-Abstand 3
  - Wenn alle Bits invertiert werden müssen = **Hamming-Klippe**
  - Hamming-Klippen können Suchraum zerklüften und Optimierung erschweren
  - *Gray-Kodierung* erschwert Bildung von Hamming-Klippen (bildet benachbarte Werte so ab, dass Sie Hamming Abstand 1 haben)

## Hillclimbing – Einflüsse - Gray-Kodierung

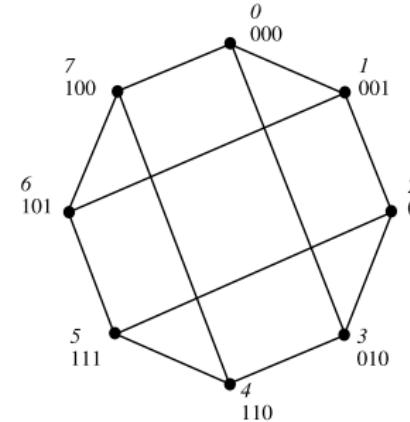
- Beispiel: geg: Zahlen von 0-7 = 3 Bits, ges: max

$$f_1(x) = \begin{cases} x+3 & \text{falls } x < 5 \\ 7-x & \text{sonst} \end{cases}$$

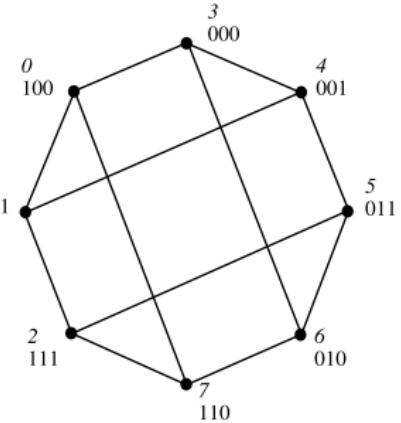
---

Gray-kod. Bitmuster	000	001	011	010	110	111	101	100
stdbin. Bitmuster	000	001	010	011	100	101	110	111
dekodiert	0	1	2	3	4	5	6	7
in Funktion $f_1$	3	4	5	6	7	2	1	0

---



(a) Gray-Kodierung


 (b) eingesetzt in  $f_1$

## Hillclimbing - Zusammenfassung

- **Lokale Optima** unüberwindbar, globales nicht mehr zu finden
- Plateaus können Optimierung behindern, da Richtungsinformationen fehlen, um Mutationen auf bessere Lösungskandidaten zu lenken → **random walk**
- **Hamming-Abstand:** Anzahl Änderungen, um eine Binärzahl in eine andere umzuwandeln, z.B. 011 → 100 = Hamming-Abstand 3
  - Wenn alle Bits invertiert werden müssen = **Hamming-Klippe**
  - Hamming-Klippen können Suchraum zerklüften und Optimierung erschweren
  - *Gray-Kodierung* erschwert Bildung von Hamming-Klippen (bildet benachbarte Werte im Phänotyp mit Hamming Abstand 1 ab)

## Rolle der Mutation

Zwei Funktionen:

- **Feinabstimmung** und  
stichprobenartiges **Erforschen** des  
Suchraums

---

**BINÄRE-MUTATION( Individuum  $A$  mit  $A.G \in \mathbb{B}^\ell$  )**

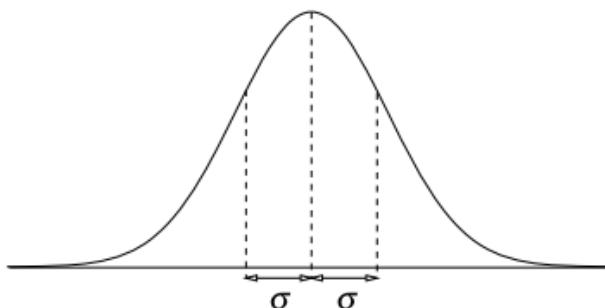
- 1  $B \leftarrow A$
  - 2 **for each**  $i \in \{1, \dots, \ell\}$
  - 3 **do**  $\lceil u \leftarrow$  wähle zufällig gemäß  $U([0, 1])$
  - 4     **if**  $u \leq p_m$  (**Mutationswahrscheinlichkeit**)
  - 5          $\lceil B.G_i \leftarrow 1 - A.G_i$
  - 6 **return**  $B$
- 

---

**GAUSS-MUTATION( Individuum  $A$  mit  $A.G \in \mathbb{R}^\ell$  )**

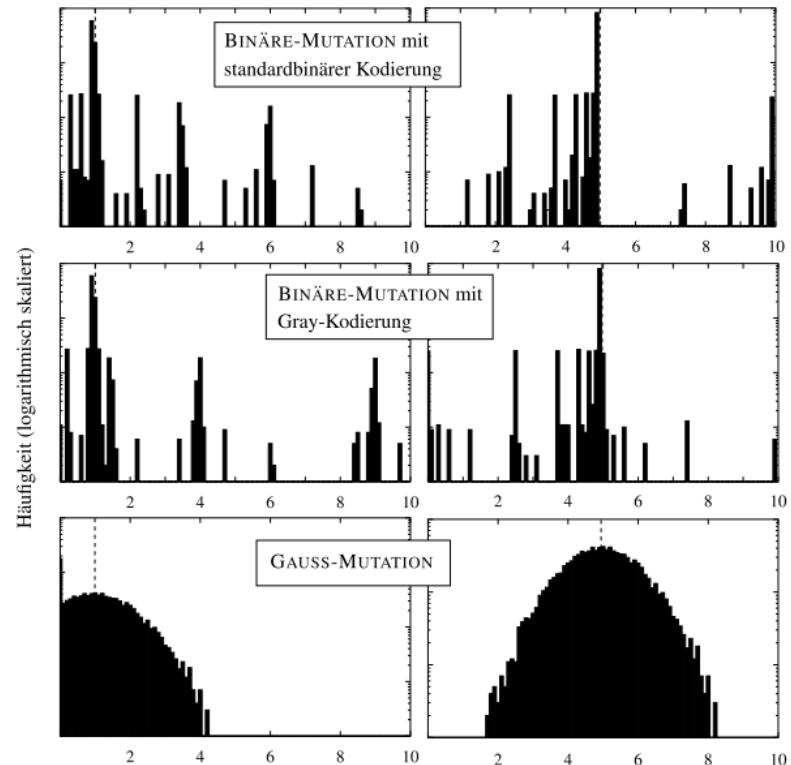
- 1 **for each**  $i \in \{1, \dots, \ell\}$
- 2 **do**  $\lceil u_i \leftarrow$  wähle zufällig gemäß  $\mathcal{N}(0, \sigma)$  (**Standardabweichung**)
- 3      $B_i \leftarrow A_i + u_i$
- 4      $B_i \leftarrow \max\{B_i, ug_i\}$  (**untere Wertebereichsgrenze**)
- 5      $\lfloor B_i \leftarrow \min\{B_i, og_i\}$  (**obere Wertebereichsgrenze**)
- 6 **return**  $B$

---



## Rolle der Mutation

- Vergleich hinsichtlich der Häufigkeit einzelner Mutationen
  - Gestrichelte Linie
    - Links: Elter = 1,0
    - Rechts: Elter = 4,99 (nah an Hamming-Klippe)
  - Jeweils 10.000 Mutationen pro Operator



## Rolle der Mutation – Anpassung der Mutationsstärke

### Anpassung der Schrittweite:

- Optimale Mutationsweite -> Maximaler Fortschritt
- Anpassung der Mutationsstärke ein Optimierungsproblem

### Algorithmus:

- a) bestimmen wie viele Nachkommen besser als die Eltern sind
- b) vergrößern / verkleinern der Schrittweite je nach Ergebnis

### 1/5 - Erfolgsregel

- basiert auf der Analyse der *Fortschrittsgeschwindigkeit* für das Korridormodell und Kugelmodell
- maximaler Fortschritt, wenn 1/5 der Nachkommen besser sind

### Mutationsoperator:

$$x_i(t+1) = x_i(t) + N(0, \sigma)$$

Schrittweite

Nachkomme      Eltern      zufällige Mutation

## Populationskonzept

Mehrere Lösungskandidaten einbeziehen  
 → Diversität fördern

---

**POPULATIONSBASIERTES-BINÄRES-HILLCLIMBING( Zielfunktion  $F$  )**

- 1  $t \leftarrow 0$
- 2  $P(t) \leftarrow$  erzeuge Population mit  $\mu$  (Populationsgröße) Individuen
- 3 bewerte  $P(t)$  durch  $F$
- 4 **while** Terminierungsbedingung nicht erfüllt
- 5 **do**  $\lceil P' \leftarrow P(t)$
- 6     **for each**  $i \in \{1, \dots, \mu\}$
- 7         **do**  $\lceil B \leftarrow$  EIN-BIT-BINÄRE-MUTATION( $A^{(i)}$ ) wobei  $P(t) = \langle A^{(k)} \rangle_{1 \leq k \leq \mu}$
- 8             bewerte  $B$  durch  $F$
- 9              $\lceil P' \leftarrow P' \circ \langle B \rangle$
- 10          $t \leftarrow t + 1$
- 11          $\lceil P(t) \leftarrow$  Selektion aus  $P'$  mittels **BESTEN-SELEKTION**
- 12     **return** bestes Individuum aus  $P(t)$

---



---

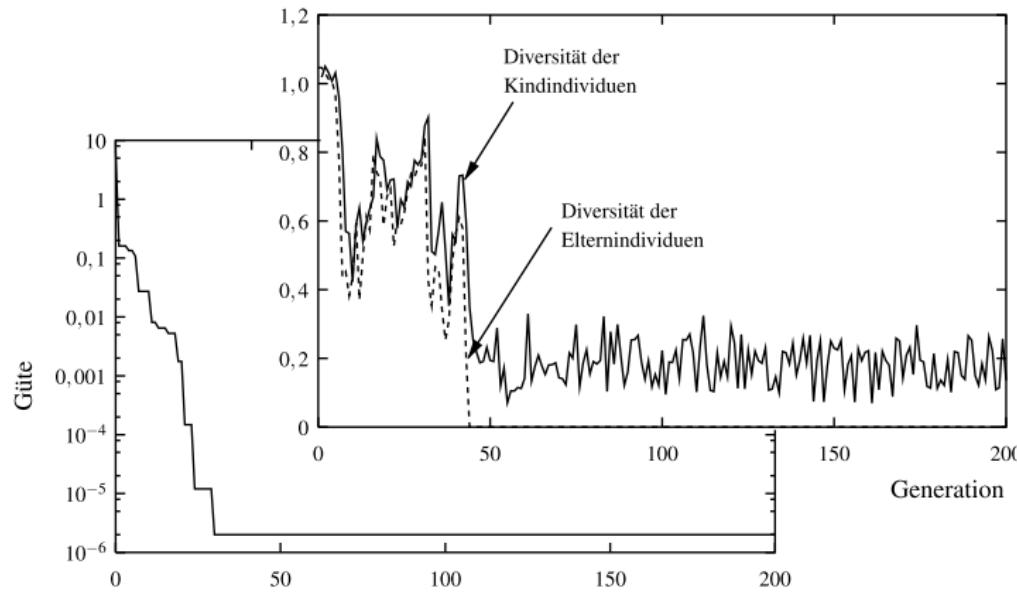
**BESTEN-SELEKTION( Gütwerte  $\langle A.F^{(i)} \rangle_{i=1, \dots, r}$  )**

- 1  $I \leftarrow \langle \rangle$
- 2 **for**  $j \leftarrow 1, \dots, s$  ( $\text{Anzahl der zu wählenden Individuen}$ )
- 3 **do**  $\lceil index_j \leftarrow$  derjenige Index aus  $\{1, \dots, r\} \setminus I$  mit dem besten Gütwert
- 4      $\lceil I \leftarrow I \circ \langle index_j \rangle$
- 5 **return**  $I$

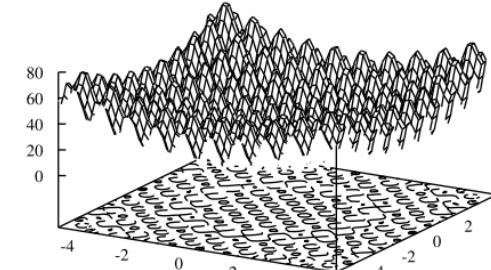
---

## Populationen - Umweltselektion

- Beste Lösungskandidaten aus Eltern- und Kindindividuen wählen
  - Vorteil: Schnelle Verbesserung in der Optimierung
  - Nachteil: vorzeitige Konvergenz in lokalen Optima



Rastrigin-Funktion  
mit  $n=2$ ,  $5,12 \leq X_1, X_2 \leq 5,12$



$$f(X) = 10 \cdot n + \sum_{i=1}^n (X_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot X_i))$$

## Populationen - Umweltselektion

- Beste Lösungskandidaten aus Eltern- und Kindindividuen wählen
  - *Vorteil:* Schnelle Verbesserung in der Optimierung
  - *Nachteil:* vorzeitige Konvergenz in lokalen Optima
- Regeln: deterministisch / probabilistisch, duplikatfrei
  - Alle Eltern sollen eine Chance haben, ausgewählt zu werden
    - Entweder jedes Individuum wird Elter für genau  $m > 0$  Kinder oder
    - Jedes Individuum wird mit einer Wahrscheinlichkeit als Elter ausgewählt
  - Möglichst große Vielfalt erhalten
    - Entweder nur die besten Individuen auswählen oder
    - Zufällige Auswahl, wobei bessere eine höhere Wahrscheinlichkeit haben und jedes nur einmal ausgewählt werden darf

## Populationen - Umweltselektion

- Deterministische Varianten
  - **Plus-Selektion ( $\mu + \lambda$ ):** überlappend, beste Individuen aus  $\mu$  Eltern und  $\lambda$  Kindern
  - **Komma-Selektion ( $\mu, \lambda$ ):** nicht überlappend, nur Kindindividuen
  - **Besten-Selektion (Elitist):** die besten Nachkommen werden behalten

*Selektionsdruck:  $s = \mu / \lambda$  bzw.  $\mu / (\mu + \lambda)$*

**$s=1$**  keine Selektion = zufällige Suche  
(kein Fortschritt)

**$s \rightarrow 0$**  hoher Fortschritt, geringer Einfluss  
des Zufalls (geringe  
Wahrscheinlichkeit, lokale Optima zu  
überwinden)

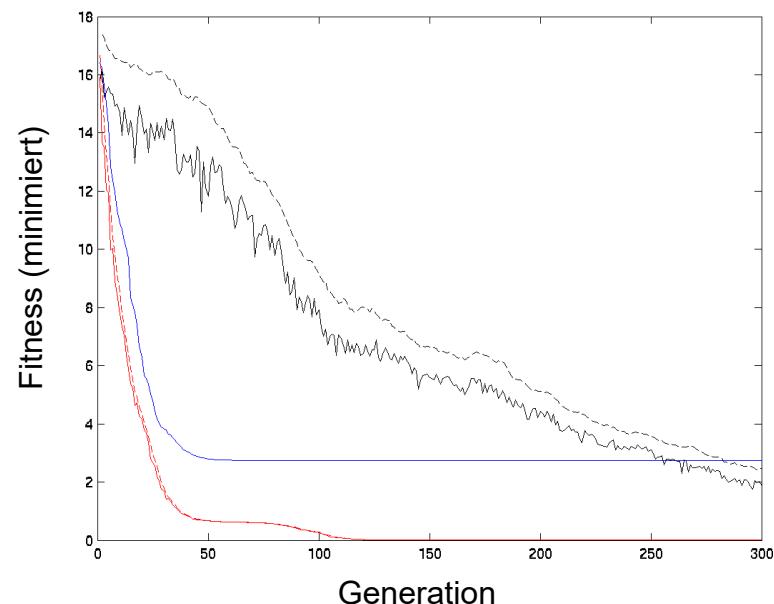
## Populationen – Selektionsdruck

### Hoher Selektionsdruck (Exploitation)

- + schnelle Qualitätssteigerung
- Mangelnde Diversität → schwierig lokale Minima zu verlassen

### Niedriger Selektionsdruck (Exploration)

- + hohe Diversität, globale Optimierung
- Zerfließen der Information, Zufallssuche



Evolutionsstrategie  
mit mutativer Selbstdadaptation

- 30 dimensionale Ackley Funktion

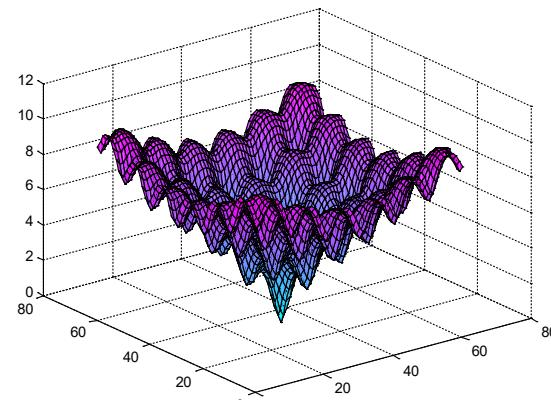
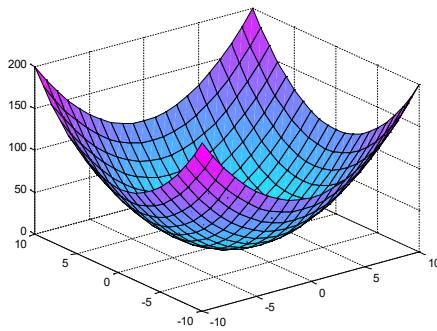
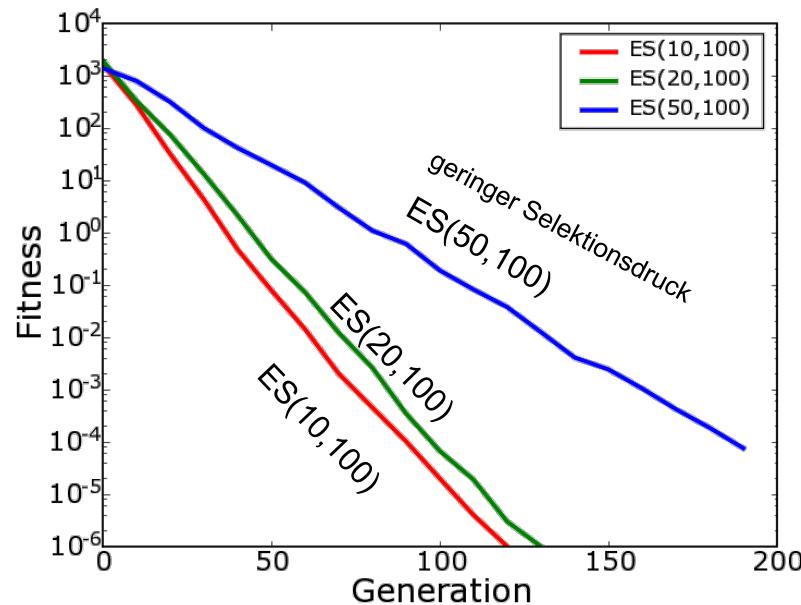
$(\mu, \lambda) = (75, 100)$ ; Selektionsdruck  $\sim 1.3$

$(\mu, \lambda) = (15, 100)$ ; Selektionsdruck  $\sim 6.7$

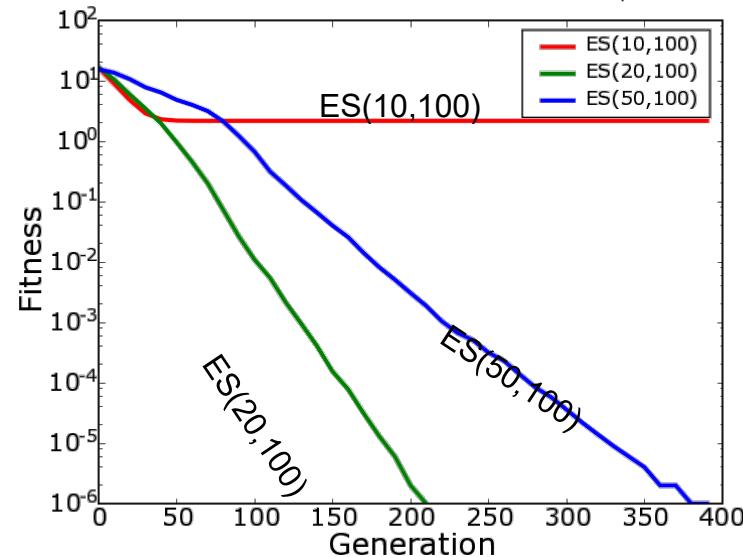
$(\mu, \lambda) = (1, 100)$ ; Selektionsdruck  $\sim 100$

— bestes Individuum  
- - - Populationsmittel

## unimodale Funktion



## multimodale Funktion (40 dimensional)



## Populationen - Umweltselektion

- Stochastische Varianten
  - **q-stufige Turnier-Selektion:**  
direkte Duelle mit q gleichverteilt zufällig gezogenen Individuen  
(Beispiel mit q=3, fünf Individuen auswählen)
  - **Rouletterad Selektion:**  
Gewichtung durch Fitnesswerte
  - **Ranking:** Rangfolge gemäß Fitnesswerte

Individuum	Turniere gegen Gegner			Siege	Wahl
$A^{(1)}.F = 3,1$	3	8✓	5	1	✓
$A^{(2)}.F = 1,0$	1	2	9	0	
$A^{(3)}.F = 4,5$	10✓	4✓	7✓	3	✓
$A^{(4)}.F = 2,4$	6✓	9✓	10	2	✓
$A^{(5)}.F = 3,6$	1✓	8✓	7✓	3	✓
$A^{(6)}.F = 2,1$	3	6	4	0	
$A^{(7)}.F = 2,7$	2✓	5	8✓	2	✓
$A^{(8)}.F = 1,8$	3	9	1	0	
$A^{(9)}.F = 2,2$	6✓	7	4	1	
$A^{(10)}.F = 3,5$	2✓	10	5	1	

# Populationen – Beispielalgorithmen zur Selektion

## Fitnessproportionale Selektion

---

**FITNESSPROPORTIONALE-SELEKTION(** Gütwerte  $\langle A^{(i)}.F \rangle_{1 \leq i \leq r}$  **)**

- 1  $Summe_0 \leftarrow 0$
- 2 **for**  $i \leftarrow 1, \dots, r$
- 3   **do**  $\lceil Fitness \leftarrow$  berechne Fitnesswert aus  $A^{(i)}.F$
- 4      $\lrcorner Summe_i \leftarrow Summe_{i-1} + Fitness$
- 5     $I \leftarrow \langle \rangle$
- 6   **for**  $i \leftarrow 1, \dots, s$  ( $\langle$ Anzahl der zu wählenden Individuen $\rangle$ )
- 7     **do**  $\lceil j \leftarrow 1$
- 8        $u \leftarrow$  wähle Zufallszahl gemäß  $U([0, Summe_r])$
- 9       **while**  $Summe_j < u$
- 10       **do**  $\lceil j \leftarrow j + 1$
- 11        $\lrcorner I \leftarrow I \circ \langle j \rangle$
- 12   **return**  $I$

---

## Stochastisch universelles Sampling

---

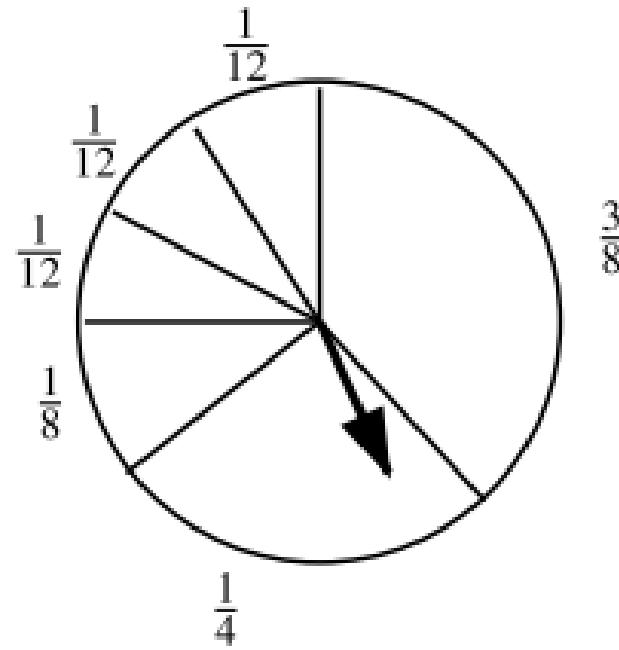
**STOCHASTISCHES-UNIVERSELLES-SAMPLING(** Gütwerte  $\langle A^{(i)}.F \rangle_{1 \leq i \leq r}$  **)**

- 1  $Summe_0 \leftarrow 0$
- 2 **for**  $i \leftarrow 1, \dots, r$
- 3   **do**  $\lceil Fitness \leftarrow$  berechne Fitnesswert aus  $A^{(i)}.F$
- 4      $\lrcorner Summe_i \leftarrow Summe_{i-1} + Fitness$
- 5    $u \leftarrow$  wähle Zufallszahl gemäß  $U([0, \frac{Summe_r}{s}])$
- 6    $j \leftarrow 1$
- 7    $I \leftarrow \langle \rangle$
- 8   **for**  $i \leftarrow 1, \dots, s$
- 9     **do**  $\lceil$  **while**  $Summe_j < u$
- 10       **do**  $\lceil j \leftarrow j + 1$
- 11        $u \leftarrow u + \frac{Summe_r}{s}$
- 12        $\lrcorner I \leftarrow I \circ \langle j \rangle$
- 13   **return**  $I$

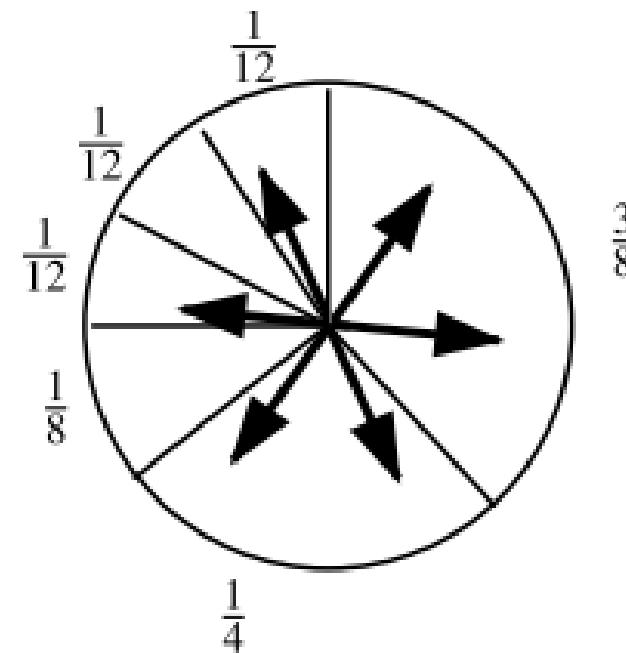
---

## Populationen – Beispielalgorithmen zur Selektion

### Fitnessproportionale Selektion



### Stochastisch universelles Sampling



## Populationsgröße

Mit folgenden Faktoren abstimmen:

- Schwierigkeiten und Charakter des Optimierungsproblems
- Involvierter Selektionsoperator
- Erforschung und Feinabstimmung durch Mutation und Rekombination

**KEINE** Formel vorhanden → **ausprobieren**

## Rekombination - Arten

- **Kombinierend:**
  - Details unterschiedlicher Individuen neu zusammensetzen
  - Idee: vorteilhafte Bestandteile der Eltern zusammenführen
  - Nachteil: hängt stark von Diversität der Population ab (nix neues)

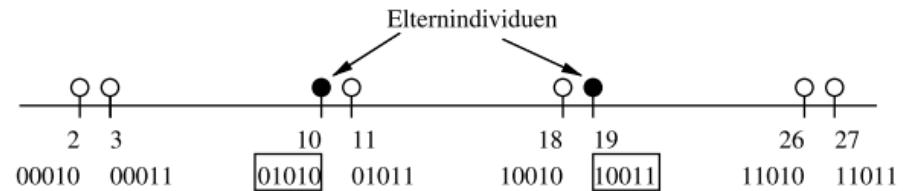
## Rekombination - kombinierend

---

**UNIFORMER-CROSSOVER( Individuum  $A$ , Individuum  $B$  )**

- 1 **for each**  $i \in \{1, \dots, \ell\}$
- 2 **do**  $\lceil b \leftarrow$  wähle zufällig gemäß  $U(\mathbb{B})$
- 3     **if**  $b$
- 4          $\lceil C.G_i \leftarrow A.G_i$
- 5          $\lceil D.G_i \leftarrow B.G_i$
- 6     **else**  $\lceil C.G_i \leftarrow B.G_i$
- 7          $\lceil D.G_i \leftarrow A.G_i$
- 8 **return**  $C, D$

---



## Rekombination - Arten

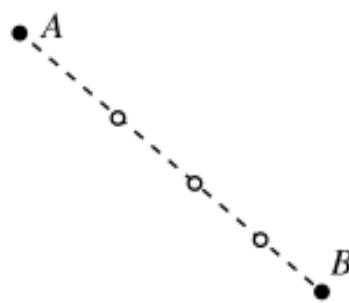
- Kombinierend:
  - Details unterschiedlicher Individuen neu zusammensetzen
  - Idee: vorteilhafte Bestandteile der Eltern zusammenführen
  - Nachteil: hängt stark von Diversität der Population ab (nix neues)
- Interpolierend
  - Vermischen Eigenschaften der Eltern zu Kindern mit neuen Eigenschaften
  - Idee: Konzentration auf gemeinsamen Nenner (Stabilität)
  - Nachteil: eventuell rasche Konvergenz

## Rekombination – interpolierend

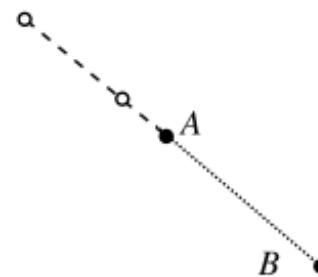
---

ARITHMETISCHER-CROSSOVER( Individuen  $A, B$  mit  $A.G, B.G \in \mathbb{R}^\ell$  )

- 1  $u \leftarrow$  wähle zufällig aus  $U([0, 1])$
  - 2 **for each**  $i \in \{1, \dots, \ell\}$
  - 3 **do**  $\lceil C.G_i \leftarrow u \cdot A.G_i + (1 - u) \cdot B.G_i$
  - 4 **return**  $C$
- 



interpolierend



Extrapolierend, wenn A.F > B.F

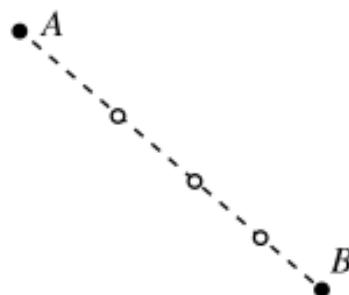
## Rekombination – interpolierend vs. extrapolierend

---

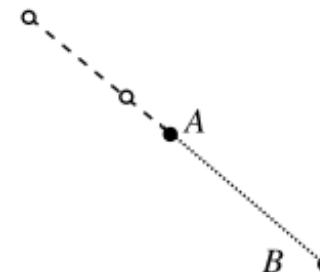
**ARITHMETISCHER-CROSSOVER( Individuen  $A, B$  mit  $A.G, B.G \in \mathbb{R}^\ell$  )**

- 1  $u \leftarrow$  wähle zufällig aus  $U([0, 1])$
  - 2 **for each**  $i \in \{1, \dots, \ell\}$
  - 3 **do**  $\lceil C.G_i \leftarrow u \cdot A.G_i + (1 - u) \cdot B.G_i$
  - 4 **return**  $C$
- 

Extrapolierend:  $u \geq 1$



interpolierend



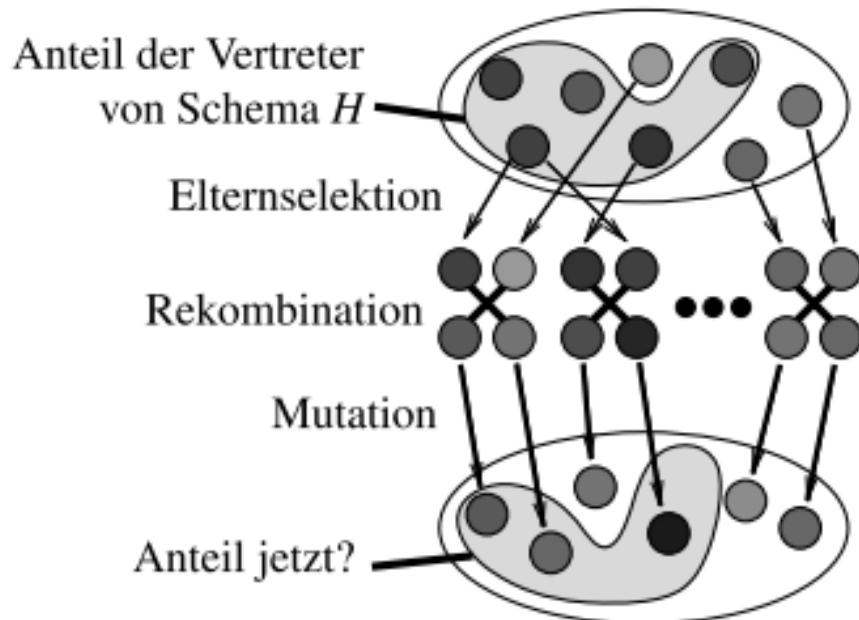
Extrapolierend, wenn  $A.F > B.F$

## Rekombination - Arten

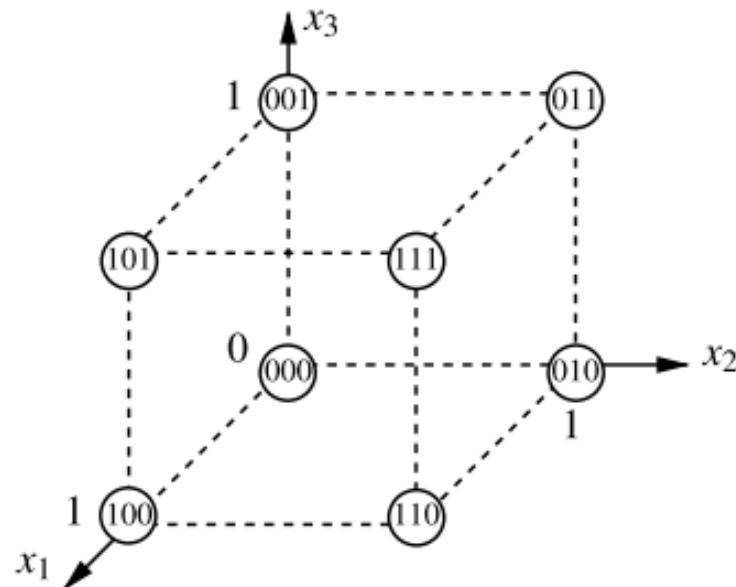
- Kombinierend:
  - Details unterschiedlicher Individuen neu zusammensetzen
  - Idee: vorteilhafte Bestandteile der Eltern zusammenführen
  - Nachteil: hängt stark von Diversität der Population ab (nix neues)
- Interpolierend
  - Vermischen Eigenschaften der Eltern zu Kindern mit neuen Eigenschaften
  - Idee: Konzentration auf gemeinsamen Nenner (Stabilität)
  - Nachteil: eventuell rasche Konvergenz
- Extrapolierend
  - Informationen von Geno- und Phänotyp, Kinder i.d.R. mit neuen Eigenschaften
  - Idee: Prognose, wo Güterverbesserung zu erwarten ist
  - Nachteil: abhängig von Gütewerten, Individuen Rücken an Rand des Suchbereichs

## Schematheorem

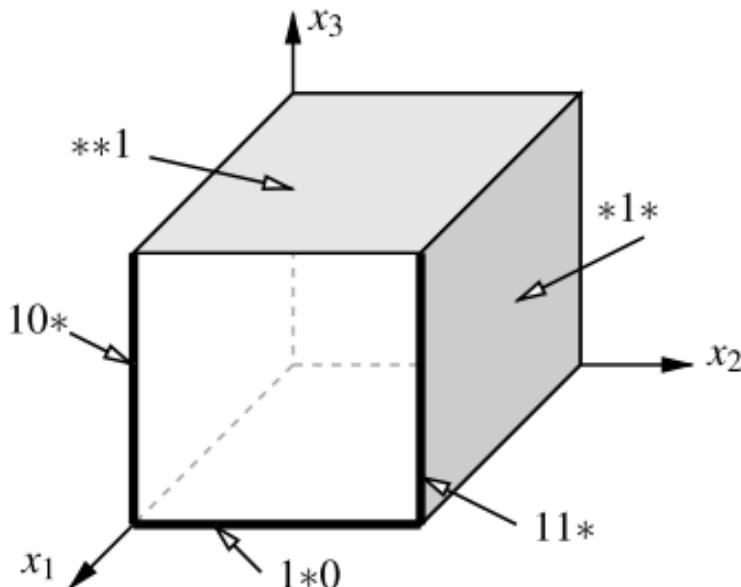
- Wie verändert die einmalige Anwendung von Elternselektion, Rekombination und Mutation den Anteil der Individuen eines Schematas in der Population?



## Schemata am Beispiel der 3 Bit Kodierung

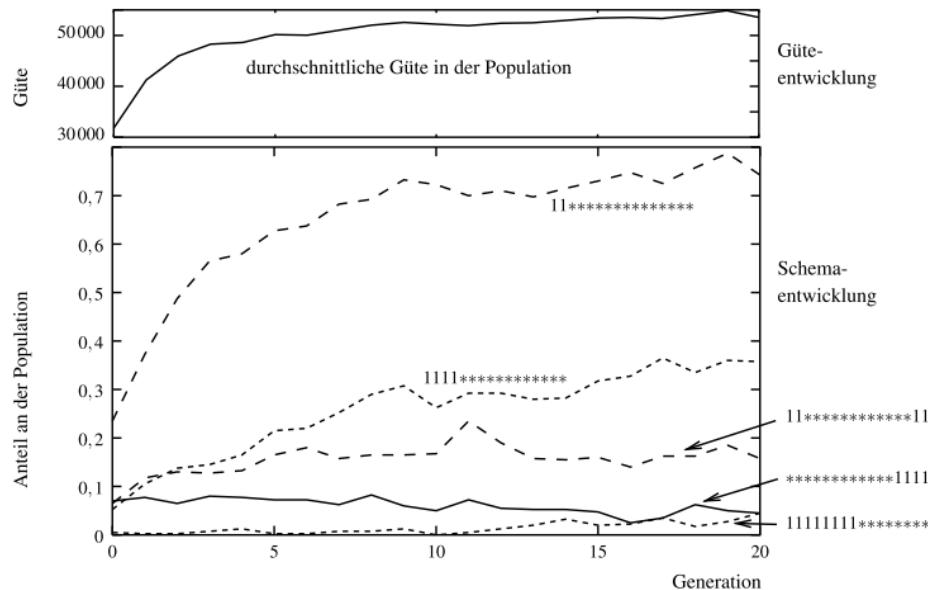


(a) Suchraum als Würfel



(b) Schemata als Hyperebenen

## Schemata-Entwicklung am Bsp. 16 Bits



Geg: 16 Bit Zahlen

Rekombinationswahrscheinlichkeit: 1.0

Mutationsrate: 1 / 16

Ges: Zahl maximieren → 111...111

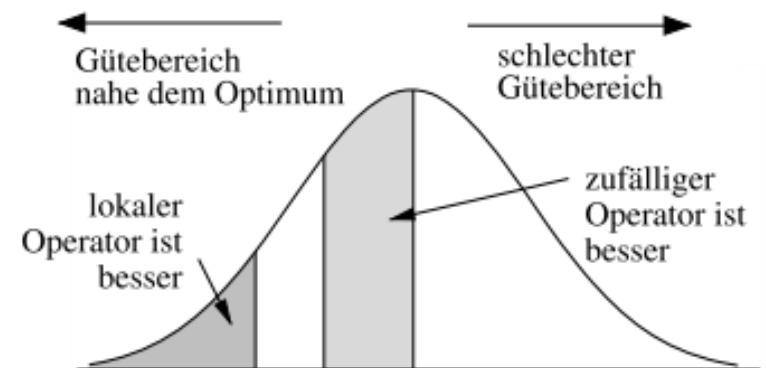
Mit dem Güte wert 65 536

Populationsgröße: 400

## Selbstanpassende Algorithmen

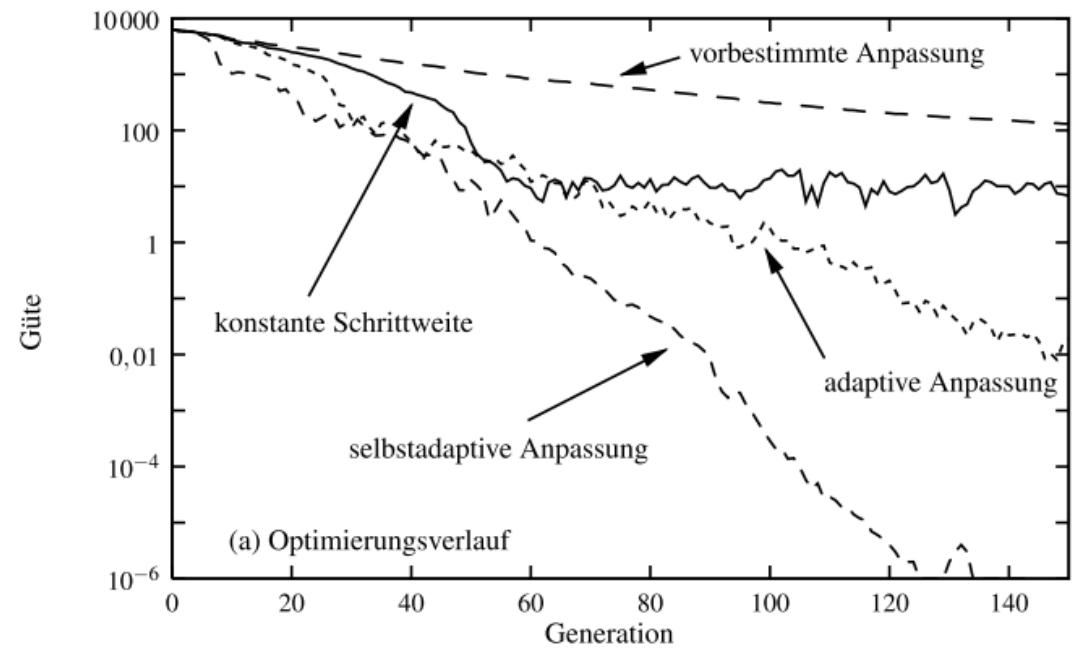
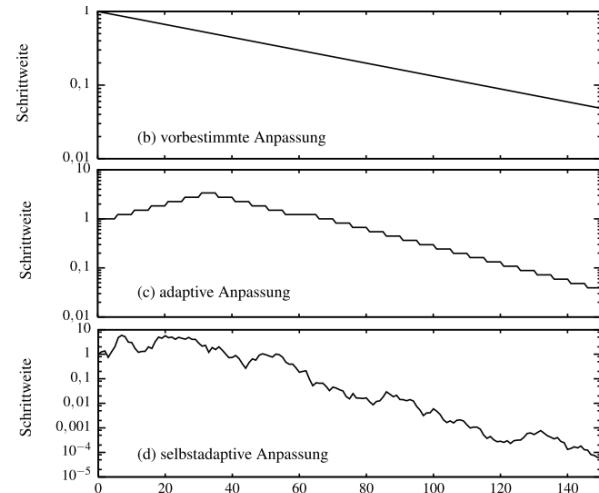
Hypothesen:

- Qualität der Mutationsoperatoren kann nicht unabhängig von aktuellem Güteniveau beurteilt werden
- Ein Operator ist niemals über den gesamten Verlauf einer Optimierung optimal

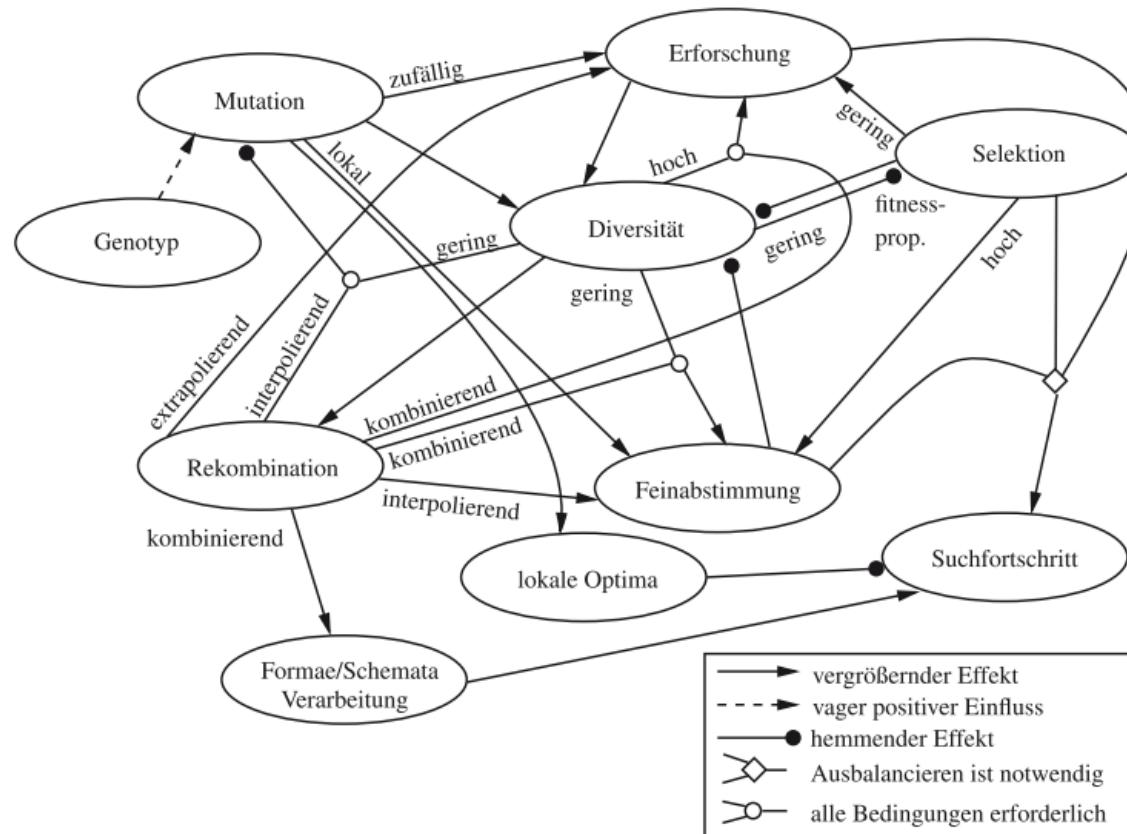


## Selbstanpassende Algorithmen - Strategien

- Vordefinierte Anpassung nach jeder Generation
- Adaptive Anpassung
- Selbstadaptive Anpassung



## Zusammenfassung – Faktoren beeinflussen sich gegenseitig



KEINE Formel vorhanden → ausprobieren

## Zusatz: eigener Zufallsgenerator – Box-Muller-Transformation

---

STANDARDNORMALVERTEILTE-ZUFALLSZAHL( Zustand  $u$ , Tabelle  $(t_i)_{1 \leq i \leq 32}$ , Index  $y$  )

- 1 **repeat**  $x_1, u, (t_i)_{1 \leq i \leq 32}, y \leftarrow -1 + 2 \cdot \text{UNIFORME-ZUFALLSZAHL}(u, (t_i)_{1 \leq i \leq 32}, y)$
- 2                    $x_2, u, (t_i)_{1 \leq i \leq 32}, y \leftarrow -1 + 2 \cdot \text{UNIFORME-ZUFALLSZAHL}(u, (t_i)_{1 \leq i \leq 32}, y)$
- 3                    $\sqcup rad \leftarrow x_1^2 + x_2^2$
- 4 **until**  $rad < 1,0$  und  $rad \neq 0$
- 5  $rad \leftarrow \sqrt{-\frac{2 \cdot \log rad}{rad}}$
- 6  $v \leftarrow x_1 \cdot rad$
- 7  $v' \leftarrow x_2 \cdot rad$
- 8 **return** Zufallszahlen  $v$  und  $v'$ , neuer Zustand  $u$ , Tabelle  $(t_i)_{1 \leq i \leq 32}$ , Index  $y$

---

## Übung 1 – Genotyp und Mutation

- Ein Produktionsplan soll für eine Fließbandproduktion optimiert werden. Es gibt  $n$  Aufträge, die alle  $m$  Stationen am Fließband in derselben Reihenfolge  $s_1, \dots, s_m$  durchlaufen. An jeder Station wird immer nur ein Auftrag gleichzeitig bearbeitet, Aufträge können sich nicht überholen. Der Auftrag  $a \in \{a_1, \dots, a_n\}$  benötigt an der Station  $s \in s_1, \dots, s_m$  genau  $t_{a,s} \in \mathbb{R}$  ( $t_{a,s} > 0$ ) Zeit.
- Gesucht wird ein Produktionsplan, der für jeden Auftrag die Startzeiten an den  $m$  Stationen und die Aufträge in der kürzesten Zeit bearbeitet.
- Der Genotyp soll die Reihenfolge der Aufträge festlegen. Wie kann daraus der Produktionsplan berechnet werden? Wie sieht ein möglicher Mutationsoperator aus?

## Übung 2 - Selektion

- Entwerfen Sie einen Selektionsoperator, der ähnlich zur proportionalen Selektion jedes Individuum mit einer bestimmten Wahrscheinlichkeit auswählt. Dabei sollen jedoch sowohl gute Individuen bevorzugt werden als auch die Diversität erhalten bleiben bzw. sogar vergrößert werden (indem der gesamte Gütebereich bis zum Ende der Optimierung repräsentiert wird).

# Visualisierung

Evolutionäre Algorithmen

---

## Welche Daten lassen sich darstellen?

- Einzelne Individuen (individuelle Daten und Eigenschaften)
- Eine / aktuelle Generation (lokale Daten und Eigenschaften)
- Alle Generationen eines Laufs (globale Daten und Eigenschaften)
- Mehrere Läufe (Vergleich globaler Daten und Eigenschaften)

## Welche Daten lassen sich darstellen?

### in einem bestimmten Zeitintervall

- Einzelne Individuen (individuelle Daten und Eigenschaften)
- Eine / aktuelle Generation (lokale Daten und Eigenschaften)
- Alle Generationen eines Laufs (globale Daten und Eigenschaften)
- Mehrere Läufe (Vergleich globaler Daten und Eigenschaften)

### Nach Anzahl

- Einzelne Individuen
- Alle Individuen einer (Unter-) Population
- Individuen mehrerer (Unter-) Populationen

### Nach Zweck

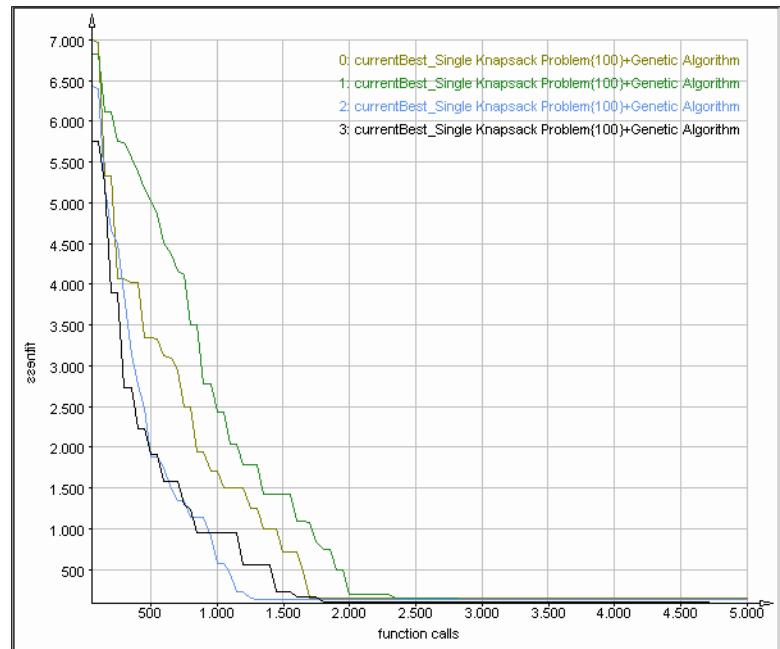
- Verlauf eines EA oder
- Einblick in verschiedene Eigenschaften der Zielfunktion

## Welche Arten von Daten lassen sich darstellen?

Direkte Werte der Individuen bzw. Populationen	Abgeleitet Werte aus den Individuen bzw. Populationen
Variablen der Individuen	Veränderung der Zielfunktionswerte (Ableitung, Differenz)
Zielfunktionswerte (Fitness) der Individuen	Verteilung oder Standardabweichung der Zielfunktionswerte
Anzahl der Individuen der Population	Abstände der Individuen voneinander (Distanz)
Anzahl der Generationen	Verteilung der Abstände / Distanzen zwischen den Individuen
Parameter des EA	
Erfolgreicher Abschluss eines Laufs	

## Konvergenzdiagramm

- Stellt Fitness des besten Individuums der Population über vergangene Generationen dar
- Zeigt wie schnell bessere Lösungen gefunden werden und wie groß der Fortschritt zwischen den Generationen ist
  - Kann mit Mittelwert und Standardabweichung aller Individuen einer Population ergänzt werden



# Übung

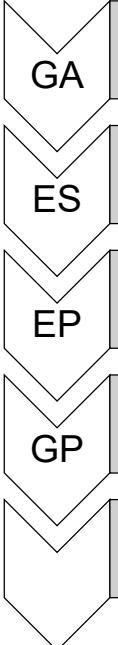
- Laden Sie sich das EvA2 base package binary (jar) herunter:
  - [http://www.ra.cs.uni-tuebingen.de/software/EvA2/down\\_direct.htm](http://www.ra.cs.uni-tuebingen.de/software/EvA2/down_direct.htm)
- Starten Sie EvA2.jar
- Laden Sie unter Optimization Parameters → Problem → Type F1-Probelm (hyper –parbola) und wählen Sie ES und setzen Sie EvaluationTerminator auf 10.000. Spielen Sie mit den Parameter herum, z.B.  $\mu$  (*Anzahl Eltern*) und  $\lambda$  (*Anzahl Kinder*) oder aktivieren Sie die Checkbox plusStrategy.
- Löschen Sie das Plot-Fenster und laden Sie das B1-benchmark Problem und lassen Sie es mit einem GA und einer ES laufen. Lassen Sie beide mehrfach laufen, welcher erzielt bessere Ergebnisse.
- Laden Sie nun **PSymbolicRegression** und spielen Sie mit verschiedenen Parametern herum und schauen sich die Auswirkungen an. Führen Sie die gleichen Parametereinstellungen mehrfach hintereinander aus. Was beobachten Sie?

# Standardalgorithmen

Evolutionäre Algorithmen

---

## Standardalgorithmen

- 
- Genetische Algorithmen
  - Evolutionsstrategien
  - Evolutionäres Programmieren
  - Genetisches Programmieren
  - Einfache lokale Suche

## Gruppen-Experten-Rallye – Phase 1

Bilden Sie vier Expertengruppen und suchen Sie eine Software ODER einen Fachartikel, der mit Hilfe eines EA ein Optimierungsproblem löst.

- a) Arbeiten Sie sich in die Software ein. Bereiten Sie für die nächste Vorlesung ein hands-on der Software und Übungen zum Testen der Software vor. Stellen Sie den Studierenden vor der nächsten Vorlesung eine Installationsanleitung zur Verfügung.
- b) Lesen Sie den gewählten Fachartikel. Bereiten Sie für die nächste Vorlesung eine Präsentation vor, um das Optimierungsproblem vorzustellen. Nennen Sie die Art des Evolutionären Algorithmus, aber keine Details der Umsetzung. Erstellen Sie eine zweite Präsentation zur Erklärung der Lösung am End der Gruppenarbeit.
  - Um welches Optimierungsproblem handelt es sich?
  - Mit welcher Art von evolutionären Algorithmus wurde es gelöst? Erklären Sie diesen?
  - Welche Mutationen und Rekombinationen wurden verwendet?
  - Wie gut sind die Ergebnisse? Welche Erkenntnisse wurden aus diesen gezogen?

Mögliche Ideen:

<https://openai.com/blog/evolution-strategies/>

<https://dl.acm.org/doi/pdf/10.1145/3449639.3459307>

<https://dev.heuristiclab.com/trac.fcgi/wiki>

<https://meiyi1986.github.io/publication/al-2019-survey/al-2019-survey.pdf>

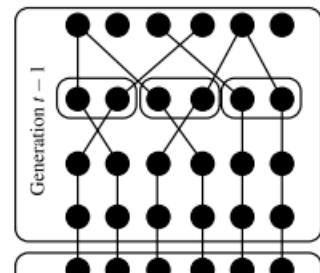
<https://link.springer.com/article/10.1007/s10710-021-09413-9>

---

## Genetische Algorithmen – John Holland (1970er Jahre)

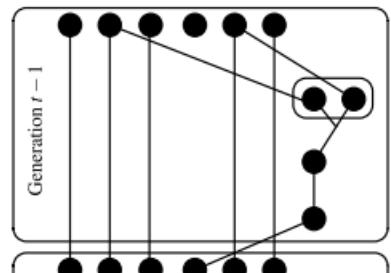
- Klassisch: Codierung durch binäre Zeichenketten
- Probabilistische Elternselektion
  - Standard GA: stochastisch universelles Sampling
  - Steady-state GA: q-fasche Turnier-Selektion
- Primärer Suchoperator: Rekombination (~70% neue Individuen)
  - Ein-Punkt-Crossover
  - Uniformer-Crossover
  - K-Punkt Crossover
- Binäre Mutation

Parameter	Wertebereich
Populationsgröße:	30–100
Rekombinationswahrscheinlichkeit:	0,6–0,9
Mutationsrate:	$0,001–0,01, 1/\ell$



(a) Genetischer Algorithmus

probabilistische  
Elternselektion  
Rekombination  
Mutation  
Identität als  
Umweltselektion



(b) steady-state GA

# Genetische Algorithmen – Steady-state GA

---

Algorithmus 4.1 (Steady state genetischer Algorithmus)

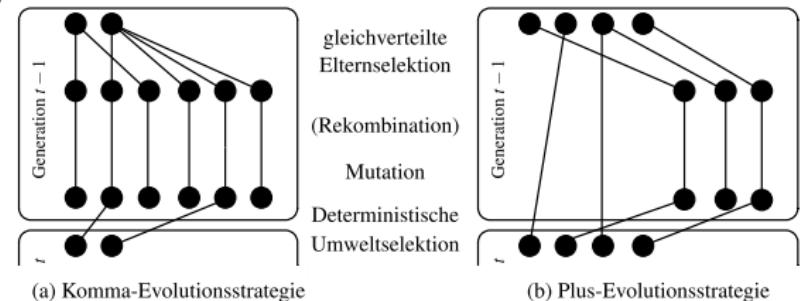
---

STEADY-STATE-GA( Zielfunktion  $F$  )

- 1  $t \leftarrow 0$
  - 2  $P(t) \leftarrow$  erzeuge Population mit  $\mu$  (Populationsgröße) Individuen
  - 3 bewerte  $P(t)$  durch  $F$
  - 4 **while** Terminierungsbedingung nicht erfüllt
  - 5 **do**  $\lceil A, B \rceil \leftarrow$  Selektion aus  $P(t)$  mittels FITNESSPROPORTIONALE-SELEKTION
  - 6      $u \leftarrow$  wähle Zufallszahl gemäß  $U([0, 1])$
  - 7     **if**  $u \leq p_x$  (Rekombinationswahrscheinlichkeit)
  - 8          $\lceil C \leftarrow$  EIN-PUNKT-CROSSOVER( $A, B$ )
  - 9     **else**  $\lceil C \leftarrow B$
  - 10     $D \leftarrow$  BINÄRE-MUTATION( $C$ )
  - 11    bewerte  $D$  durch  $F$
  - 12     $P' \leftarrow$  entferne das schlechteste Individuum aus  $P(t)$
  - 13     $t \leftarrow t + 1$
  - 14     $\lceil P(t) \leftarrow P' \circ \langle D \rangle$
  - 15 **return** bestes Individuum aus  $P(t)$
-

## Evolutionsstrategien – Rechenberg (1973), Schwefel (1981)

- Klassisch: reellwertiger Genotyp
- Elternselektion: ohne Selektionsdruck (gleichverteilt zufällig gewählt)
- Umweltselektion: Besten-Selektion (Komma- oder Plus-Selektion)
- Selektionsdruck über Populationsgrößen der Eltern und Kinder geregelt
- Primärer Suchoperator: Mutation
  - Gauss-Mutation
- Rekombination (teilweise nicht verwendet)



## Evolutionsstrategien - Beispiel

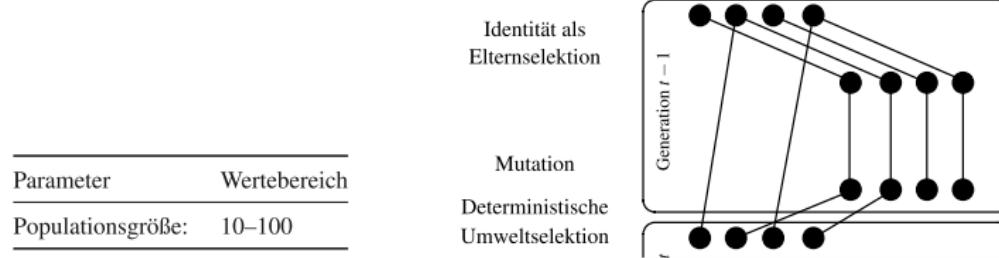
---

```
ES-ADAPTIV( Zielfunktion F )
1    $t \leftarrow 0$ 
2    $\sigma \leftarrow$  Wert für Anfangsschrittweite
3    $s \leftarrow 0$ 
4    $P(t) \leftarrow$  erzeuge Population mit  $\mu$  (Populationsgröße) Individuen
5   bewerte  $P(t)$  durch  $F$ 
6   while Terminierungsbedingung nicht erfüllt
7   do  $P' \leftarrow \langle \rangle$ 
8     for each  $i \in \{1, \dots, \lambda\}$  (Anzahl der Kinder)
9       do  $A \leftarrow$  selektiere Elter uniform zufällig aus  $P(t)$ 
10       $C \leftarrow$  GAUSS-MUTATION( $A$ ) mit  $\sigma$ 
11      bewerte  $C$  durch  $F$ 
12      if  $C.F \succ A.F$ 
13        then  $\lceil s \leftarrow s + 1$ 
14         $P' \leftarrow P' \circ \langle C \rangle$ 
15       $t \leftarrow t + 1$ 
16       $P(t) \leftarrow$  Selektion aus  $P'$  mittels BESTEN-SELEKTION
17      if  $t \bmod k$  (Modifikationshäufigkeit) = 0
18        then  $\lceil \sigma \leftarrow$  ADAPTIVE-ANPASSUNG( $\sigma, \frac{s}{k \cdot \lambda}$ )
19         $\lceil \lceil s \leftarrow 0$ 
20  return bestes Individuum aus  $P(t)$ 
```

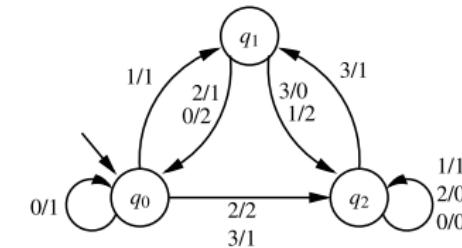
---

## Evolutionäres Programmieren

- Fokus auf Verhalten (Genetik uninteressant) – phänotypisch beobachtbare Ähnlichkeit zum Elternteil
- Keine Rekombination (nur Mutation + Reproduktion)
- Zwei Ansätze:
  - 1960er Jahre – **endliche Automaten**



Automat mit drei Zuständen

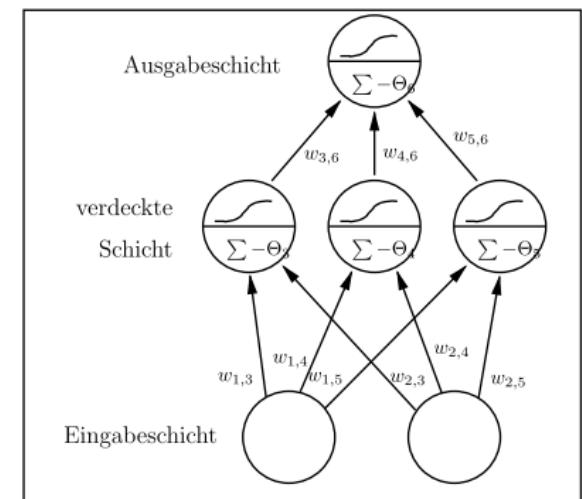


Zeitreihe	1	0	2	3	1	2
Zustand	$q_0$	$q_1$	$q_0$	$q_2$	$q_1$	$q_2$
Vorhersage	1	2	2	1	2	2

## Evolutionäres Programmieren

- Fokus auf Verhalten (Genetik uninteressant) – phänotypisch beobachtbare Ähnlichkeit zum Elternteil
- Keine Rekombination (nur Mutation + Reproduktion)
- Zwei Ansätze:
  - 1960er Jahre – endliche Automaten
  - 1980er Jahre – **neuronale Netze**

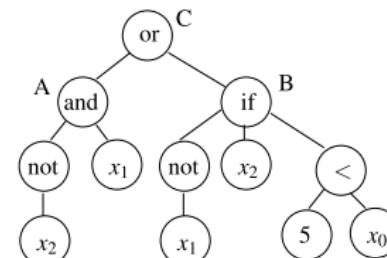
Parameter	Wertebereich
Populationsgröße:	20–200, selten bis 500
Anpassungsstärke $\alpha$ :	0,1–0,4, $1/\ell^2$
minimale Standardabweichung $\varepsilon$ :	$10^{-5}$ – $10^{-3}$
Turniergröße:	5–10



## Genetisches Programmieren - deGaris (1990), Fujiko & Dickinson (1987), ...

- Klassisch: Syntaxbäume (Programmcode)
- Primärer Suchoperator: Rekombination (z.B. Baum-Tausch)
- Hintergrundoperator: Mutation (z.B. Einfügen einzelner Operatoren)

Parameter	Wertebereich
Populationsgröße:	200–5 000
Rekombination/Mutation/Klonen:	80/10/10

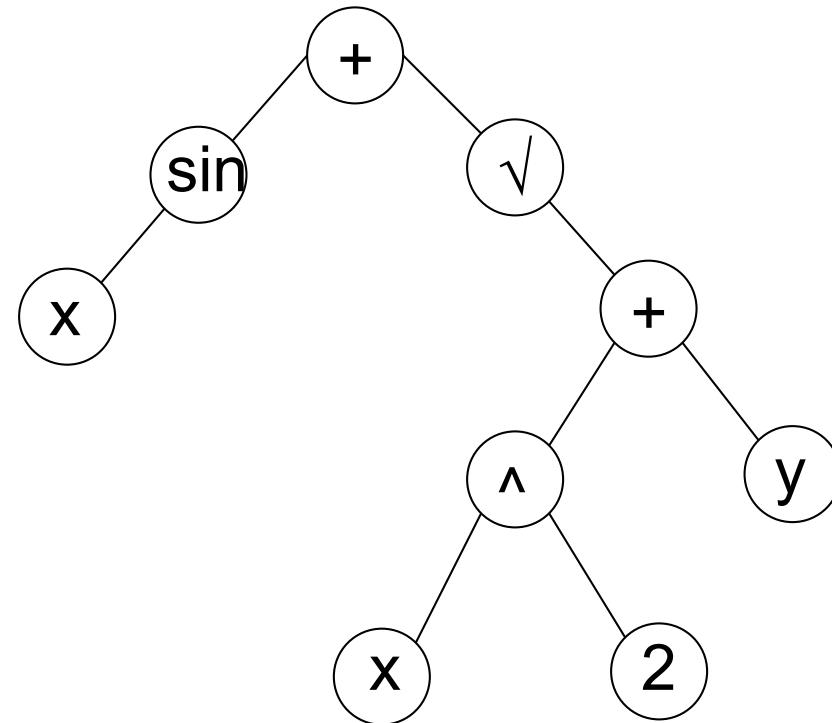


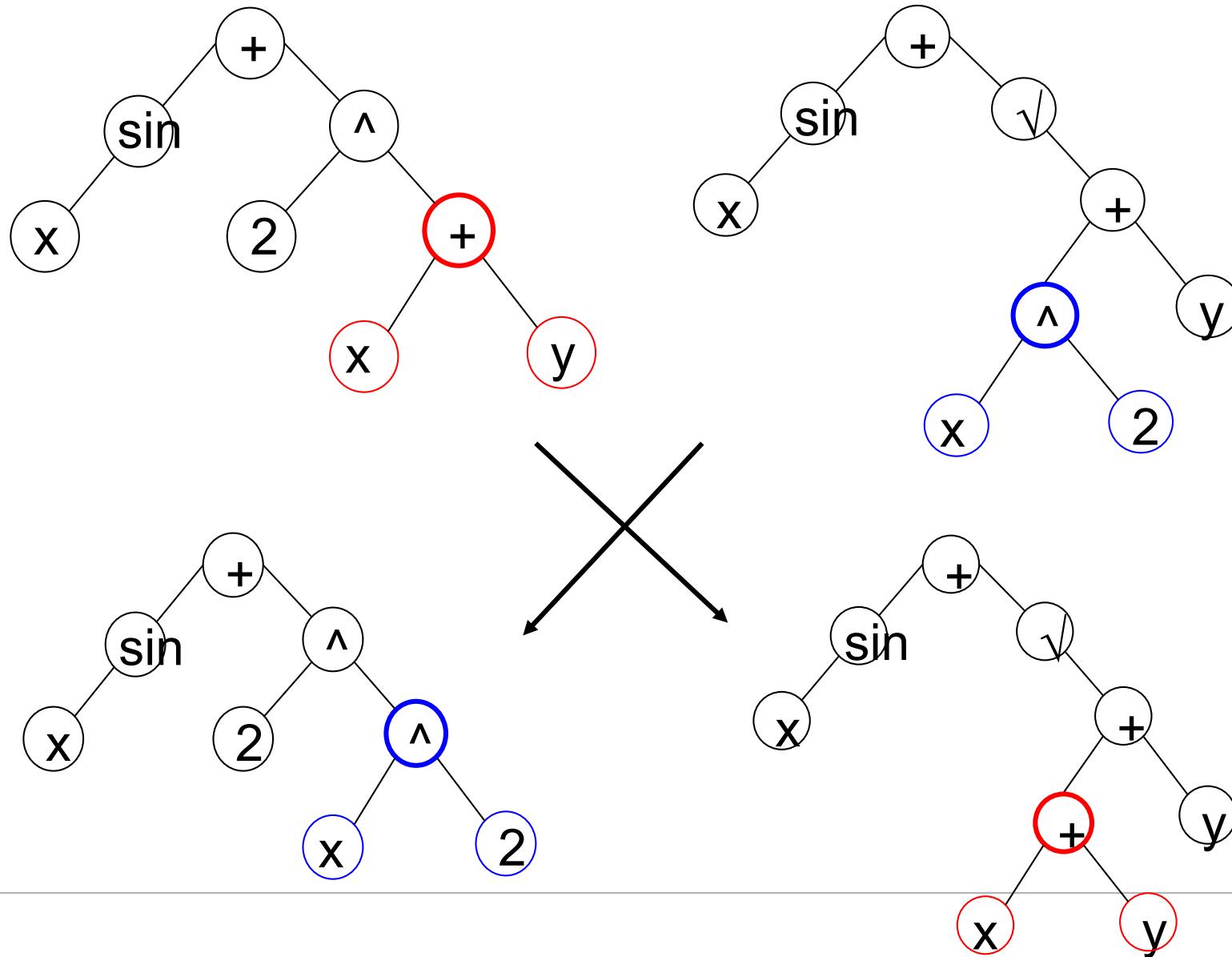
$5 < x_0$	$x_1$	$x_2$	A	B	C
false	false	false	false	false	false
false	false	true	false	true	true
false	true	false	true	false	true
false	true	true	false	false	false
true	false	false	false	false	false
true	false	true	false	true	true
true	true	false	true	true	true
true	true	true	false	true	true

(b) Wertetabelle

## Genetische Programmierung – Beispiel

$$\sin(x) + \sqrt{x^2 + y}$$





## Genetisches Programmieren - Beispiel

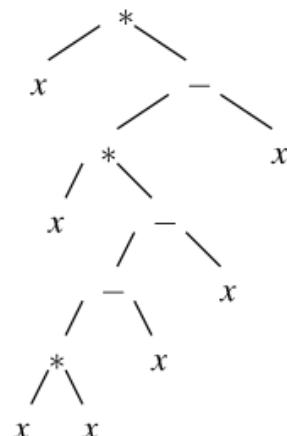
### Symbolic-Regression-Problem

- Rekonstruktion einer Funktion anhand von vorgegebenen Stützstellen
- Fitness: min Summe des quadratischen Fehlers

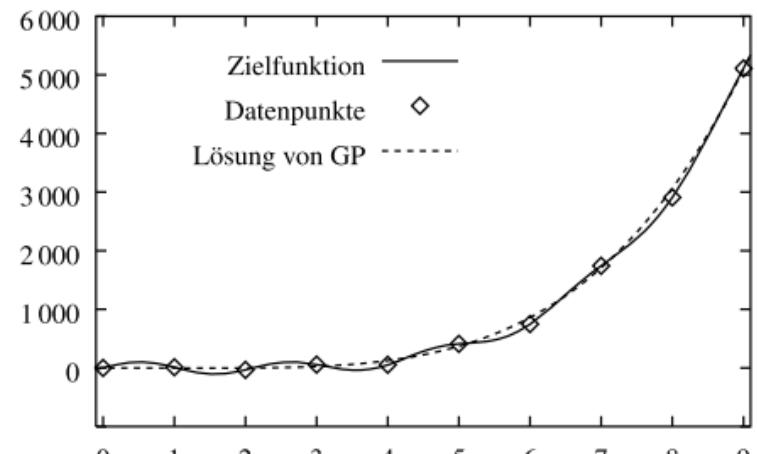
Zielfunktion:

$$x^4 - 2x^3 - x^2 - x + 100 \sin(3x)$$

an den Stützstellen  $\{0, 1, \dots, 9\}$



(a) bestes Individuum



(b) Vergleich mit der gesuchten Funktion

## Einfache lokale Suche

- Sonderfall: EA mit nur einem Individuum
- Selektion: ist Kind besser als Elter?
- Nur Mutation

---

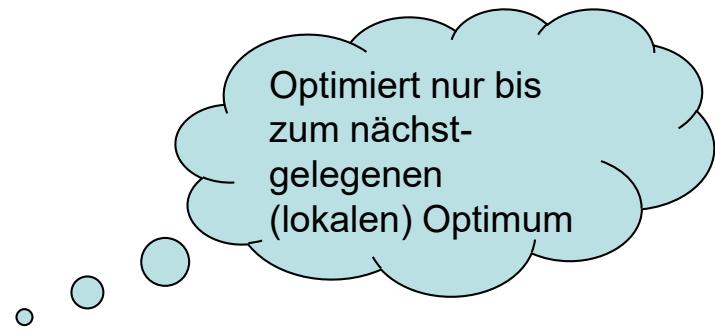
**LOKALE-SUCHE( Zielfunktion  $F$  )**

```

1    $t \leftarrow 0$ 
2    $A(t) \leftarrow$  erzeuge Lösungskandidat
3   bewerte  $A(t)$  durch  $F$ 
4   while Terminierungsbedingung nicht erfüllt
5   do  $\lceil B \leftarrow$  variiere  $A(t)$ 
6       bewerte  $B$  durch  $F$ 
7        $t \leftarrow t + 1$ 
8       if Akz( $A(t - 1).F, B.F, t$ ) (Akzeptanzbedingung)
9           then  $\lceil A(t) \leftarrow B$ 
10      else  $\lceil A(t) \leftarrow A(t - 1)$ 
11   return  $A(t)$ 

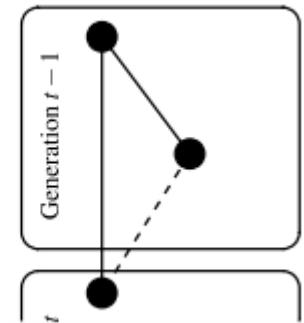
```

---



Identität als  
Elternselektion  
und Mutation

Akzeptanz-  
kriterium



## Einfache lokale Suche – beispielhafte Problemlösungen

### Ausgang: Hill-Climbing

---

```
AKZEPTANZ-HC( Elterngüte A.F, Kindgüte B.F, Generation t )
1  return B.F  $\succ$  A.F
```

---

### Threshold Accepting

---

```
AKZEPTANZ-TA( Elterngüte A.F, Kindgüte B.F, Generation t )
1  if B.F  $\succ$  A.F oder  $d_{euk}(A.F, B.F) \leq Temp_t$ 
2  then  $\sqcap$  return wahr
3  else  $\sqcap$  return falsch
```

---

### Simulated Annealing

---

```
AKZEPTANZ-SA( Elterngüte A.F, Kindgüte B.F, Generation t )
1  if B.F  $\succ$  A.F
2  then  $\sqcap$  return wahr
3  else  $\lceil u \leftarrow$  wähle zufällig aus  $U([0, 1])$ 
4    if  $u \leq \exp\left(-\frac{d_{euk}(A.F, B.F)}{Temp_{t-1}}\right)$ 
5      then  $\sqcap$  return wahr
6     $\sqcup$  else  $\sqcap$  return falsch
```

---

## Vergleich einzelner Ansätze

### ES

- Schnelle lokale Optimierung
- Erreicht schnell homogene Population
- Starke Mutation
- Starke Selektion

### GA

- Durchforstung des Suchraums
- Vermeidet genau diesen Effekt
- Vermeiden von Mutation
- Schwache Selektion

# Vorgehen bei der Entwicklung

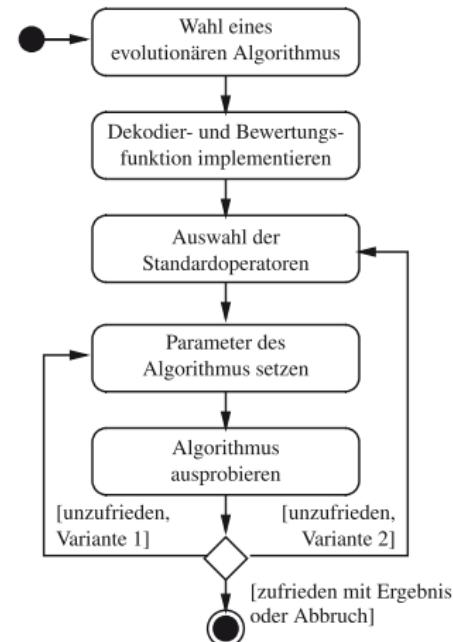
Evolutionäre Algorithmen

---

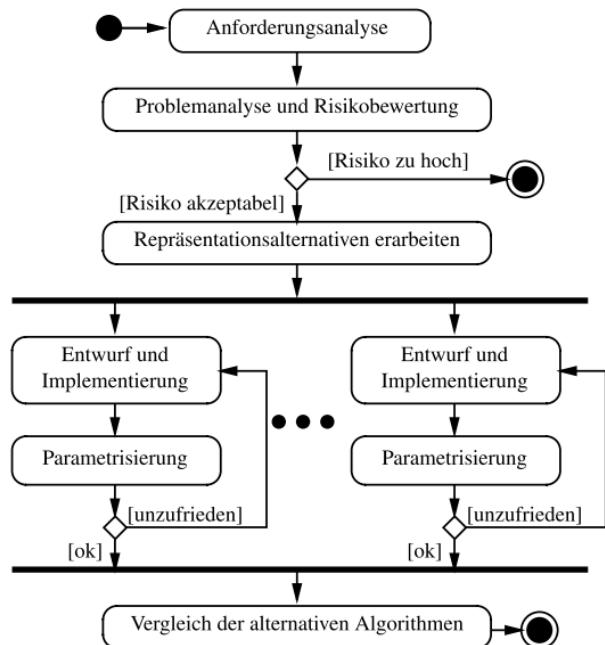
## Allgemeine Vorgehensweise

Wiederverwendung prüfen?

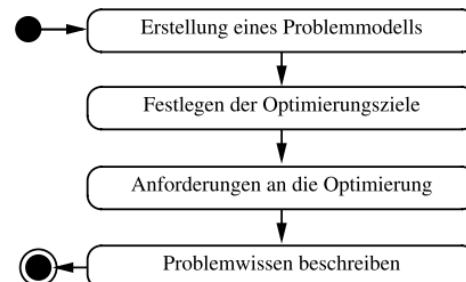
- Standardalgorithmus einer Bibliothek anwenden oder
- Entwurf aus der Literatur einsetzen



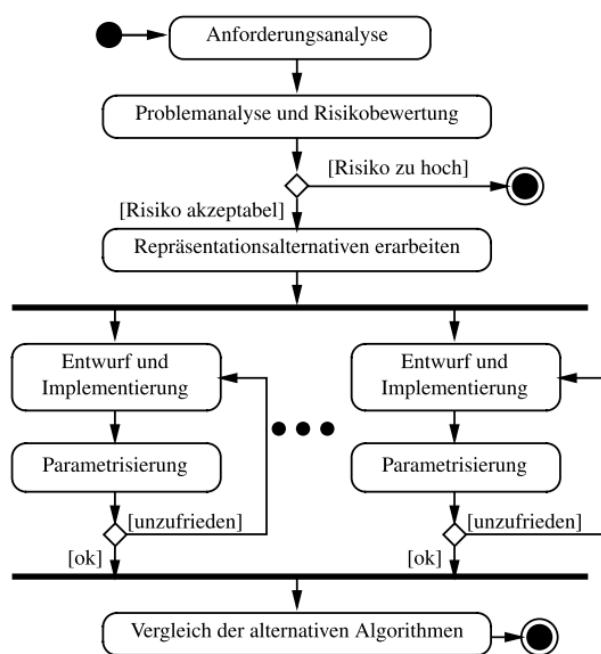
## Analysebasierter Ansatz



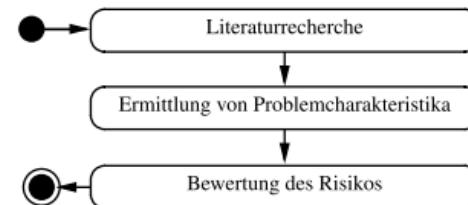
### Anforderungsanalyse:



## Analysebasierter Ansatz

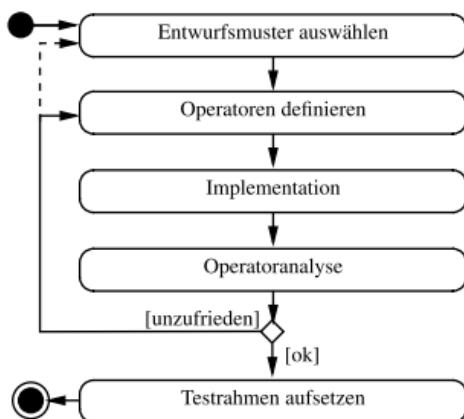


### Problemanalyse:



## Analysebasierter Ansatz

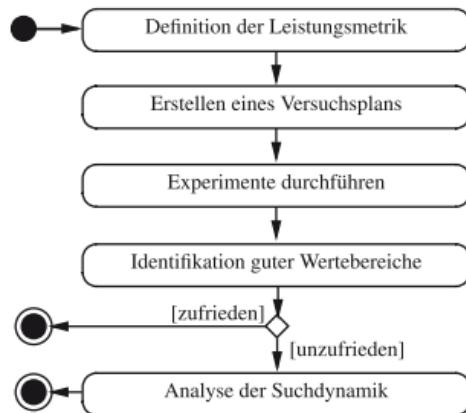
Entwurf und Implementierung:



Aspekt	Beschreibung
Name des Muster Kontext	eindeutiger Bezeichner Problemstellungen, die eine Anwendbarkeit des Musters nahe legen
Mutationsrolle	Vorgaben zur Wahl oder zum Entwurf des Mutationsoperators (oder der Mutationsoperatoren)
Rekombinationsrolle	Vorgaben zur Wahl oder zum Entwurf des Rekombinationsoperators (oder der Rekombinationsoperatoren)
Selektionsrolle	Vorgaben zur Wahl oder zum Entwurf des Selektionsoperators (oder der Selektionsoperatoren)
Erfolgsfaktoren	Konkrete Eigenschaften des Optimierungsproblems oder sonstiger Anforderungen, die nach bisherigen Erfahrungen das Entwurfsmuster positiv beeinflussen
Metriken zum Test	konkrete Angaben zu Messungen, die dieses Muster aufweisen sollte, wenn die Operatoranalyse durchgeführt wird

## Analysebasierter Ansatz

### Parametrisierung:



Leistungsmetrik	Beschreibung
beste Güte mittlere beste Güte	Der beste Gütwert in allen betrachteten Generationen. In jeder Generation wird der beste Gütwert bestimmt, aus denen dann der Mittelwert berechnet wird. Dies betont die Konvergenzgeschwindigkeit, falls evtl. nur wenig Zeit für eine Optimierung zur Verfügung steht.
durchschnittliche Güte	Der Durchschnitt über die Güte aller evaluierten Individuen wird berechnet. Zusätzlich zur Konvergenzgeschwindigkeit wird noch die Anzahl der evaluierten Individuen berücksichtigt.
Erfolgswahrscheinlichkeit	Es wird in mehreren Experimenten gemessen, wieviel Prozent davon einen vorgegebenen Gütwert erreichen konnten.
benötigte Bewertungen	Es wird gemessen, wieviele Individuen bewertet werden mussten, bis ein vorgegebener Gütwert erreicht wurde.

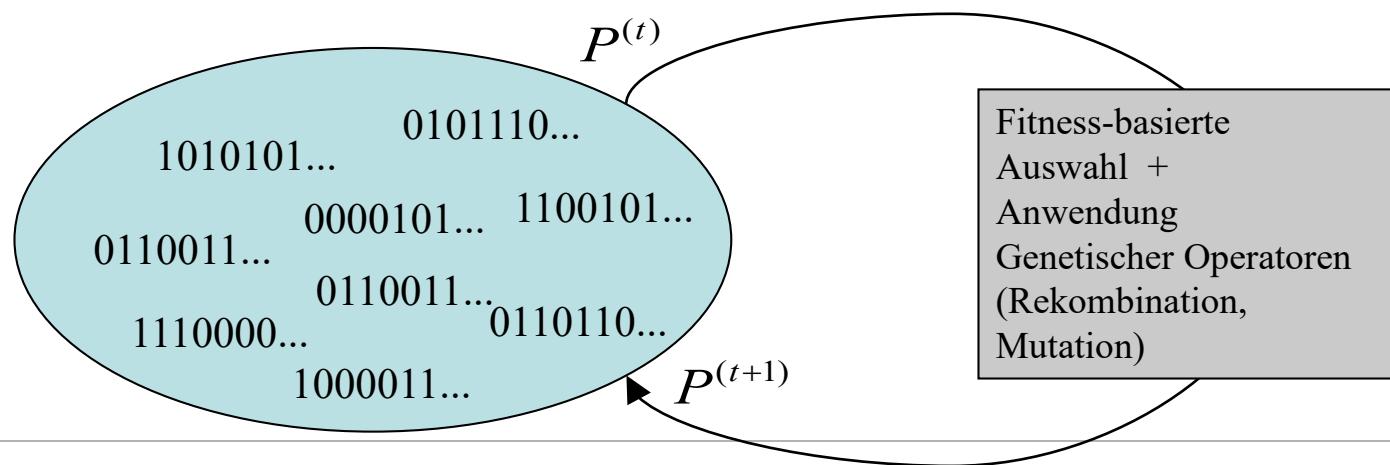
# Wiederholung

Evolutionäre Algorithmen

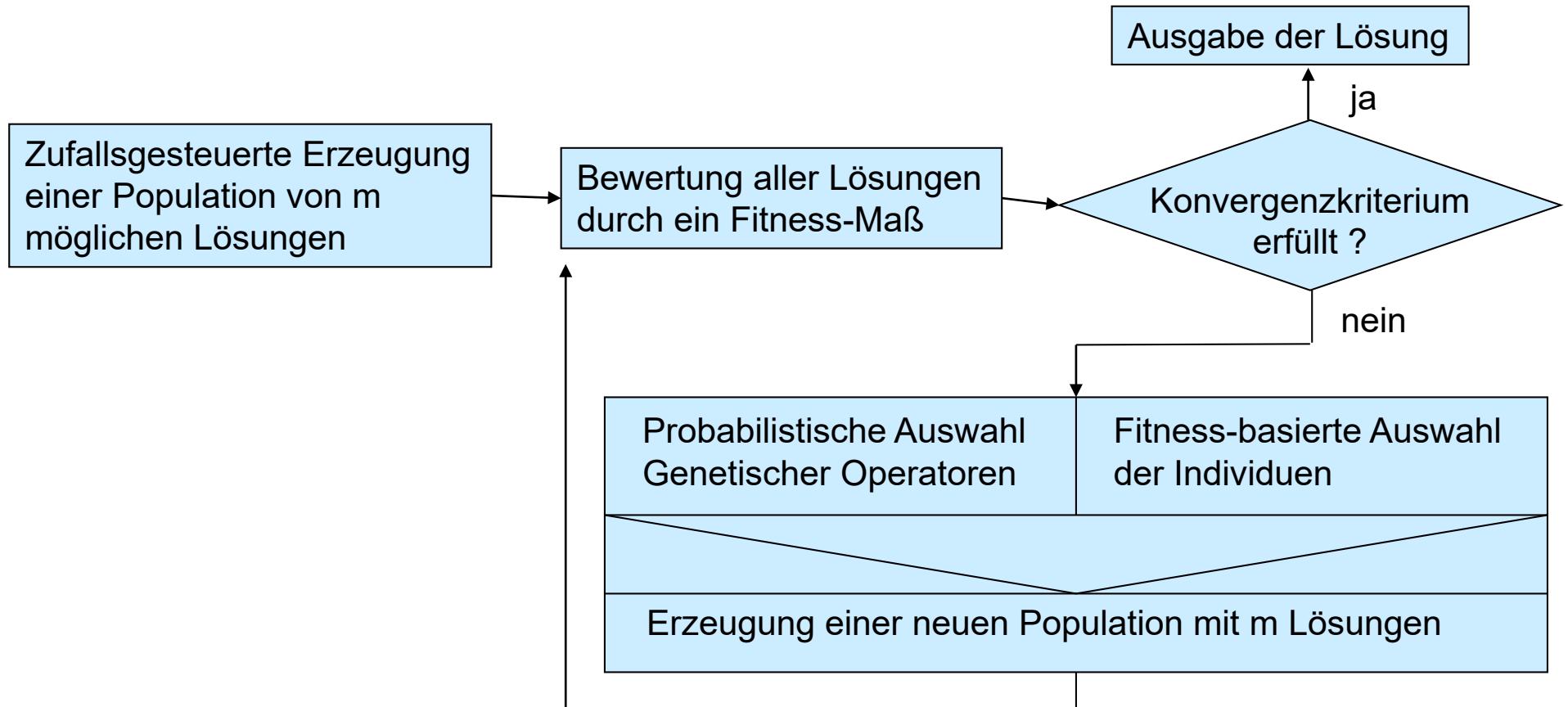
---

## Zusammenfassung EA

- Stochastische Suchverfahren (heuristisch)
- Basierend auf Rekombination, Mutation und Selektion
- Einfaches Grundprinzip, viele Varianten
- Hauptvorteil gegenüber anderen Verfahren: Parallelisierbarkeit
- Zahlreiche Anwendungen in den Ingenieur- und Naturwissenschaften



## Zusammenfassung - Vorgehensweise



## Zusammenfassung - No Free Lunch

- Ohne Berücksichtigung problemspezifischer Randbedingungen kommt ein evolutionärer Algorithmus nicht schneller zu einer optimalen Lösung als eine zufallsbasierte Suche.
  - im Mittel ist die Performance über alle Probleme gleich
  - wenn bestimmte Probleme besser gelöst werden, werden andere dafür schlechter gelöst
- Das gilt für alle Lösungsverfahren für kombinatorische Optimierungsprobleme
- Es gibt keinen heiligen Gral !

## Zusammenfassung – Weitere Effekte

**Lamarck'sche Hypothese:** Individuen können die während ihres Lebens erlernten Fähigkeiten vererben.

**Baldwin-Effekt:** Lernfähige (und damit besonders anpassungsfähige) Individuen haben eine höhere Reproduktionsrate

“Intelligente” Initialisierung der Population kann

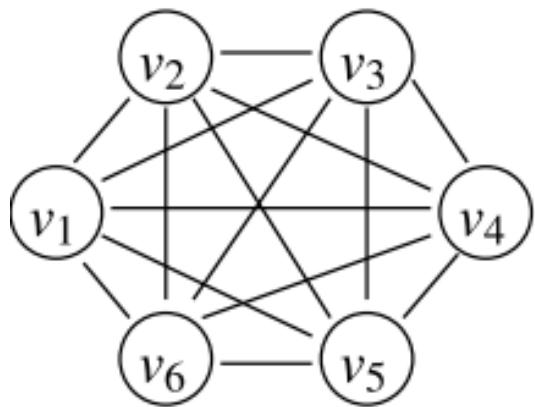
- ungünstigen Bias vermeiden und dementsprechend
- vorzeitige Konvergenz verhindern und
- Konvergenzverhalten insgesamt verbessern

---

Beschränkte Partnerwahl – restricted mating

# Prüfungsvorbereitung

## Übung 1 – Handlungsreisendenproblem



Kante $e$	$\gamma(e)$	Kante $e$	$\gamma(e)$	Kante $e$	$\gamma(e)$
$(v_1, v_2)$	5	$(v_2, v_3)$	10	$(v_3, v_5)$	17
$(v_1, v_3)$	8	$(v_2, v_4)$	4	$(v_3, v_6)$	8
$(v_1, v_4)$	11	$(v_2, v_5)$	9	$(v_4, v_5)$	6
$(v_1, v_5)$	3	$(v_2, v_6)$	12	$(v_4, v_6)$	5
$(v_1, v_6)$	7	$(v_3, v_4)$	6	$(v_5, v_6)$	11

Geb: Elternpopulation mit den Individuen:

- (1, 4, 2, 5, 6, 3)
- (4, 5, 3, 2, 6, 1)

Berechnen Sie zwei Generationen in dem Sie ein Individuum durch Kantenrekombination und auf beiden Eltern und ein Individuum invertierende Mutation auf einem der beiden Eltern erzeugen. Selektieren Sie die zwei besten.

## Übung 2 - Implementierung

Implementieren Sie den beschriebenen Algorithmus EA-HANDLUNGSREISENDENPROBLEM und wenden Sie ihn auf ein Problem mit 100 zufällig im zweidimensionalen Raum verteilten Städten an. Die Kantengewichte sollen der euklidischen Distanz entsprechen:

$$\gamma((u, v)) = \sqrt{(x - x')^2 + (y - y')^2}$$

für  $u = (x, y)$  und  $v = (x', y')$ .

---

## Übung 3 - Eignung evolutionärer Algorithmen

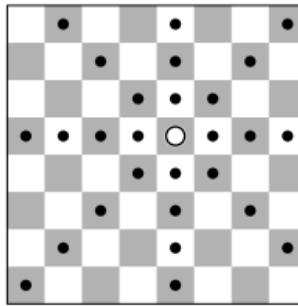
Entscheiden Sie für die folgenden Probleme, ob sich der Einsatz eines evolutionären Algorithmus lohnt:

- Planungsproblem: Zwei Produkte A und B können auf drei Maschinen gefertigt werden, wobei sie jeweils unterschiedliche Laufzeiten benötigen. Ferner ist die Laufzeit der Maschinen pro Tag beschränkt und jedes Produkt erzielt einen gegebenen Preis. Gesucht ist ein Verfahren, das bestimmt, wie viele Exemplare der Produkte auf den jeweiligen Maschinen zu produzieren sind, damit die Firma einen maximalen Gewinn erzielt.
- Hamiltonkreis: In einem beliebigen Graphen mit  $E \neq V \times V$  ist ein Weg gesucht, der jeden Knoten nur einmal besucht – im Gegensatz zum Handlungsreisendenproblem interessiert hier nur die reine Existenz eines Weges.
- In einem Graphen ist der zweitkürzeste Weg zwischen zwei gegebenen Knoten gesucht.

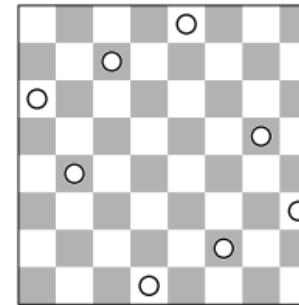
## Übung 4 – Entwicklung eines Algorithmus

Überlegen Sie, wie mehrere Populationen in einem evolutionären Algorithmus zusammenwirken können. Skizzieren Sie einen Algorithmus, der auch Genfluss als Evolutionsfaktor nutzt.

## Übung 5



Zugmöglichkeiten einer Dame



Lösung des 8-Damen-Problems

Entwickeln Sie einen EA für das 8-Damen-Problem:

- finden Sie eine geeignete Repräsentation
- bestimmen Sie passende genetische Operatoren
- definieren Sie eine geeignete Fitnessfunktion
- wählen Sie ein Selektionsverfahren und legen die Parameter fest
- zeichnen sie einen Generationsschritt (Initialisierung einer Population, Bewertung, Auswahl der Operatoren + Individuen, Aufstellung der neuen Population) auf

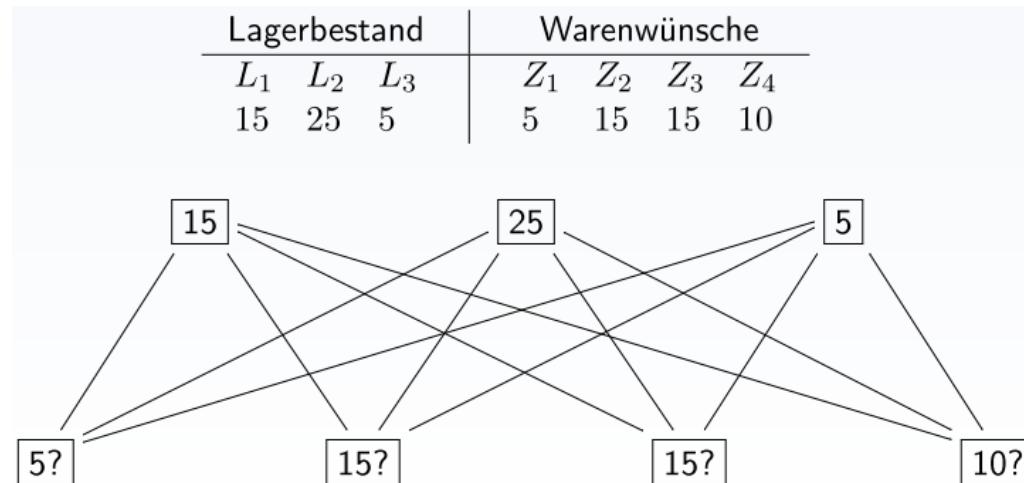
Diskutieren Sie, ob das Verfahren immer erfolgreich ist oder nicht? Warum ist dies der Fall?

---

## Übung 6 - Transportproblem (Michalewicz 1992)

Gegeben:

- 3 Warenlager: L1, L2, L3
- 4 Zielpunkten Z1, . . . , Z4.
- Kosten für den Transport von Waren von Li nach Zi pro Einheit.



## Übung 6 - Transportproblem (Michalewicz 1992)

Die Kosten pro Einheit sind:

	$Z_1$	$Z_2$	$Z_3$	$Z_4$
$L_1$	10	0	20	11
$L_2$	12	7	9	20
$L_3$	0	14	16	18

Eine optimale Lösung ist:

	5	15	15	10
15	0	5	0	10
25	0	10	15	0
5	5	0	0	0

Gesucht:  
minimale  
Transportkosten

Die Kosten sind:

$$5 * 0 + 10 * 11 + 10 * 7 + 15 * 9 + 5 * 0 = 315$$

---

## Übung 7 - Erklären Sie z.B. die folgenden Begriffe?

- Random walk
  - Hamming-Klippe
  - Mutation als Hintergrundoperator
  - Interpolierende Rekombination
  - Simulated Annealing
  - ...
- 
- Wie ist der Hamming-Abstand für 110 und 011?
  - Welche Arten von Mutation / Rekombination / Selektion kennen Sie?
  - Welche Effekte haben die Evolutionsfaktoren?

## Interessante Projekte

- [EA Library in Java - https://jenetics.io/](https://jenetics.io/)
  - <http://www.ra.cs.uni-tuebingen.de/software/EvA2/index.html>
  - <http://geneticprogramming.com/>
  - <https://sourceforge.net/projects/opendino/> (ES, Simplex, Neural Network)
  - <https://sourceforge.net/projects/popt4jlib/> (Java)
  - <https://sourceforge.net/projects/ecfproject/> (C++)
  - <https://sites.google.com/site/gptips4matlab/> (Matlab)
  - <https://openai.com/blog/evolution-strategies/>
-