

UNIVERSITÄT

BONN

AIS

COMPUTATIONAL INTELLIGENCE

9. EVOLUTIONÄRE ALGORITHMEN CONT.

Prof. Dr. Sven Behnke

LETZTE VORLESUNG

■ Evolutionäre Algorithmen

- Optimierung einer Funktion
- Ableitungsfrei

■ EA- Grundelemente

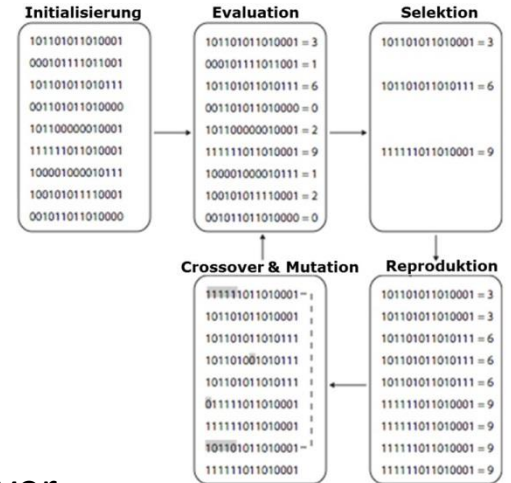
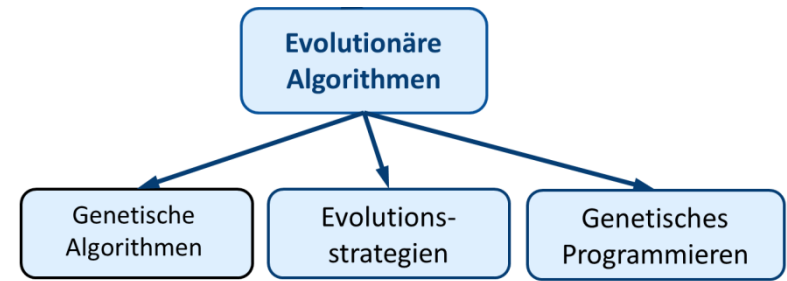
- Kodierungsvorschrift
- Initialisierungsmethode
- Fitnessfunktion
- Selektionsoperator
- Reproduktionsoperator
- Abbruchkriterium

■ Genetische Operatoren: Mutation,

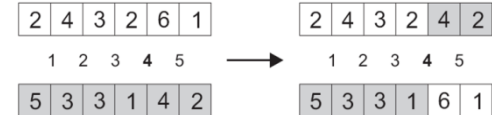


■ Fitnessfunktion:

- Genotyp => Phenotyp => Evaluation



Crossover



KODIERUNG

■ Kodierung der Lösungskandidaten

- Kodierung ist **problemspezifisch**
- Es gibt also kein Verfahren um automatisch gute Kodierungen zu finden
- Es gibt allerdings Prinzipien, die bei der Kodierung geachtet werden sollten

■ Eigenschaften einer guten Kodierung

- Ähnliche Lösungskandidaten sollten ähnliche Fitness haben (Prinzip der starken Kausalität)
- Ähnliche Genotypen sollten ähnliche Phänotypen repräsentieren
- Suchraum sollte möglichst unter den verwendeten genetischen Operatoren abgeschlossen sein

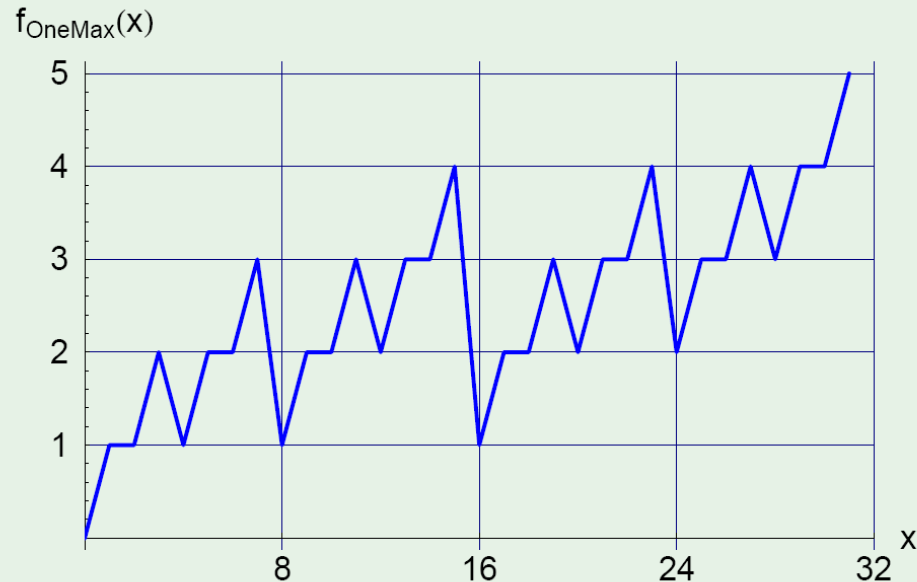
KODIERUNG KLASSISCHER GENETISCHER ALGORITHMEN

- **Binäre Vektoren** fester Länge L
- Repräsentation direkt geeignet für Probleme mit boolschem Definitionsbereich
- Unterschiedlichste Interpretationen möglich
 - L 1-stellige binäre Entscheidungsvariablen
 - M N -stellige binär kodierte Variablen mit $MN=L$
 - Eine L -stellige binär kodierte Variable
 - ...

MÖGLICHE INTERPRETATION: ANZAHL DER EINSEN

Beispiel (5-stellige 'OneMax'-Funktion)

$$f_{\text{OneMax}} : \{0,1\}^5 \rightarrow \mathbb{N} \quad \text{mit} \quad f_{\text{OneMax}}(\vec{x}) = \sum_{i=1}^5 x_i$$



BINÄRKODIERUNG REELLER ZAHLEN

- Gegeben: Intervall reeller Zahlen $[u, v]$ und Kodierungsgenauigkeit ϵ
- Gesucht: Kodierung einer reellen Zahl $x \in [u, v]$ als Binärzahl b mit Fehler kleiner ϵ
- Ansatz:
Teile das Intervall $[u, v]$ in Abschnitte mit einer Länge $\leq \epsilon$. Wähle dazu 2^k Abschnitte, mit Index $0, \dots, 2^k-1$, mit $k = \lceil \log_2 \frac{v-u}{\epsilon} \rceil$

- Kodierung:

$$b = \left\lfloor \frac{x - u}{v - u} (2^k - 1) \right\rfloor$$

- Dekodierung:

$$x = u + b \frac{v - u}{2^k - 1}$$

BINÄRKODIERUNG REELLER ZAHLEN

Beispiel

- Gegeben sei das Intervall $[-1, 4]$,
die Kodierungsgenauigkeit $\epsilon = 10^{-6}$ und
die zu kodierende Zahl $x = 3.141593_{10}$
- Es werden

$$k = \left\lceil \log_2 \frac{4 - (-1)}{10^{-6}} \right\rceil = \lceil \log_2(5 \cdot 10^6) \rceil = 23$$

Stellen für die Binärkodierung benötigt

- Die Binärdarstellung ist somit

$$\begin{aligned} b &= \left\lfloor \frac{3.141593 - (-1)}{4 - (-1)} (2^{23} - 1) \right\rfloor = 6948439_{10} \\ &= 11010100000011001010111_2 \end{aligned}$$

STARKE KAUSALITÄT DER FITNESS

- Prinzip: Ähnliche Lösungskandidaten sollen ähnliche Fitness haben
- Motivation
 - Ähnliche Ursachen haben ähnliche Wirkungen
 - Grundlage für die Anwendbarkeit Evolutionärer Algorithmen
- Problem der Epistase
 - Zwischen den Genen eines Chromosoms können Wechselwirkungen bestehen
 - Ein Allel eines epistatischen Gens unterdrückt die Wirkung aller möglichen Ausprägungen eines oder mehrerer anderer Gene
 - => Fitnesswirkung eines Gens kann stark von den Ausprägungen anderer (epistatischer) Gene abhängen

EPISTASE IN DER BIOLOGIE

- Häufig die Ursache für die Abweichungen von den Mendelschen Gesetzen

Beispiel (Fellfarben bei Tieren)

- Ein Gen für die Fellfarbe: Schwarz (B) / braun (b), wobei schwarz gegenüber braun dominant ist
- Ein Gen für farbig (C) / nicht farbig (c)
- Das zweite Gen steht epistatisch über dem ersten
- Kreuzen zweier Tiere mit Genotyp BbCc ergibt
 - zu 9/16 schwarze Tiere (Bb oder BB, Cc oder CC)
 - zu 3/16 braune Tiere (bb und Cc oder CC)
 - zu 4/16 weiße Tiere (cc, unabhängig vom Farbtyp)

KODIERUNG BESTIMMT EPITASE

Beispiel (Epistase beim n -Damen-Problem)

- **Kodierung 1**

Brettbelegung wird durch eine *Permutation* der möglichen Spaltenpositionen zeilenweise dargestellt

Geringe Epistase: Austausch zweier Zeilen stellt eine lokale Änderung dar und ändert die Fitness im Allgemeinen etwa gleich stark

- **Kodierung 2**

Brettbelegung erfolgt zeilenweise durch *Auswahl von Elementen* aus der Liste *der nichtbelegten Spalten*

Hohe Epistase: Änderung eines Gens, speziell der im Chromosom vorn liegenden, kann die Brettbelegung (fast) vollständig ändern und entsprechend auch die Fitness. Änderungen sind insofern global.

Beispiel (2. Kodierung ▷ Wirkung einer Mutation)

- Vor der Mutation – Kollisionen: 4

Chromosom

| |
|---|
| 1 |
| 3 |
| 3 |
| 2 |
| 1 |

Liste zu belegender Spalten

1, 2, 3, 4, 5
2, 3, **4**, 5
2, 3, **5**
2, **3**
2

Brettbelegung

| |
|---|
| 1 |
| 4 |
| 5 |
| 3 |
| 2 |

- Nach der Mutation – Kollisionen: 8

Chromosom

| |
|----------|
| 5 |
| 3 |
| 3 |
| 2 |
| 1 |

Liste zu belegender Spalten

1, 2, 3, 4, **5**
1, 2, **3**, 4
1, 2, **4**
1, **2**
1

Brettbelegung

| |
|---|
| 5 |
| 3 |
| 4 |
| 2 |
| 1 |

EPISTASE: KONSEQUENZEN

■ Kodierung mit hoher Epistase

- Evolutionärer Algorithmus hat Schwierigkeiten, den Zusammenhang zwischen Genom und Fitness auszunutzen
- Genetische Operatoren führen zu fast zufälligen Fitnessänderungen

■ Kodierung mit sehr geringer Epistase

- Einfache, direkte Zusammenhänge zwischen Genom und Fitness
- Benötigt keinen Evolutionären Algorithmus, da schon einfache Verfahren wie Zufallsaufstieg zur Optimierung ausreichen

ÄHNLICHKEIT GENOTYP – PHÄNOTYP

- **Prinzip:** Ähnlichen Genotypen sollten ähnliche Phänotypen entsprechen
- Mutationen (Alleländerungen einzelner Gene) stellen kleine Änderungen des Genoms dar und führen somit zu ähnlichen Genotypen
- Bewertung der Fitness findet beim Phänotyp statt
- Entsprechen ähnlichen Genotypen nicht auch ähnliche Phänotypen, können naheliegende Verbesserungen des Phänotyps eventuell nicht erzeugt werden, da dafür große Änderungen des Genotyps erforderlich sein können
- Beispiel: Naive Kodierung als Binärzahl führt zu Hamming-Klippen

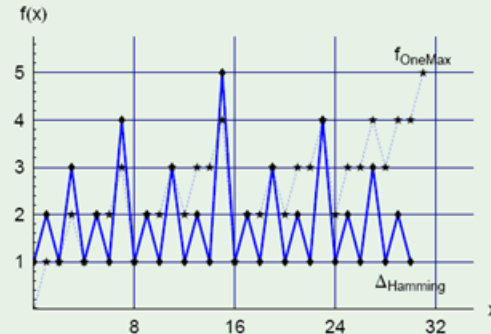
HAMMING-KLIPPEN

■ Problem:

- Benachbarte Zahlen können sich in ihrer Kodierung stark unterscheiden
- Anzahl verschiedener Bits wird als Hamming-Abstand bezeichnet
- Große Hamming-Abstände können durch genetische Operatoren nur schwer überwunden werden

Beispiel (Analytische Optimierung einer Zielfunktion)

- Stelle die Zahlen $0, \dots, 31$ durch 5-bit Binärzahlen dar
- Die Zahlen $15_{10} = 01111_2$ und $16_{10} = 10000_2$ haben den Hamming-Abstand 5, da jedes Bit verschieden ist



VERMEIDUNG VON HAMMING-KLIPPEN

■ Idee

- Nutze eine Kodierung, bei der sich je zwei benachbarte Zahlen nur um ein Bit unterscheiden, d.h. einen Gray-Kode

■ Gray-Kode

- Nicht eindeutig
- Häufigste Form

□ Kodierung: $g = b \oplus \left\lfloor \frac{b}{2} \right\rfloor$ \oplus Exklusiv-Oder-Operator

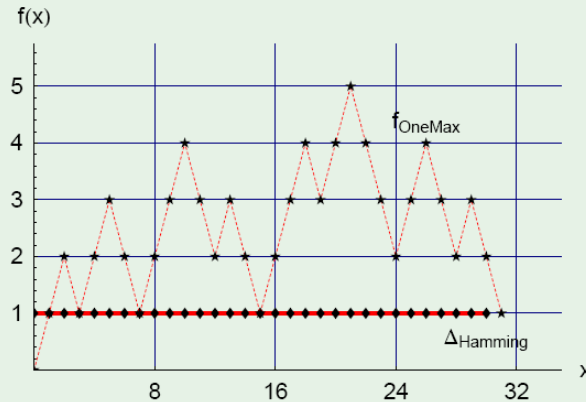
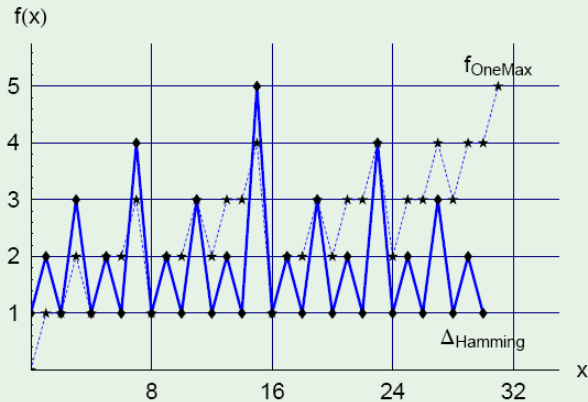
□ Dekodierung: $b = \bigoplus_{i=0}^{k-1} \left\lfloor \frac{g}{2^i} \right\rfloor$

| dezimal | binär | Gray |
|---------|-------|------|
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |
| 3 | 0011 | 0010 |
| 4 | 0100 | 0110 |
| 5 | 0101 | 0111 |
| 6 | 0110 | 0101 |
| 7 | 0111 | 0100 |
| 8 | 1000 | 1100 |
| 9 | 1001 | 1101 |
| 10 | 1010 | 1111 |
| 11 | 1011 | 1110 |
| 12 | 1100 | 1010 |
| 13 | 1101 | 1011 |
| 14 | 1110 | 1001 |
| 15 | 1111 | 1000 |

Beispiel (Berechnung eines Gray-Kodes)

- Berechne Gray-Code für die bereits bekannte Binärdarstellung b der Zahl $x = 3.141593_{10}$ (Intervall $[-1, 4]$, Kodierungsgenauigkeit $\epsilon = 10^{-6}$) mit $b = 11010100000011001010111_2$
- Gray-Code ist somit

$$\begin{aligned} g &= 11010100000011001010111_2 \\ \oplus \quad & 11010100000011001010111_2 \\ = \quad & 10111110000010101111100_2 \end{aligned}$$



ABGESCHLOSSENER SUCHRAUM

- **Prinzip:** Der Suchraum sollte möglichst unter den verwendeten genetischen Operatoren abgeschlossen sein.
- Was unter Verlassen des Suchraums zu verstehen ist, muss definiert werden
- Allgemein: Suchraum wird verlassen falls
 - Neues Chromosom kann nicht sinnvoll interpretiert (dekodiert) werden
 - Lösungskandidat verletzt Nebenbedingung(en)
 - Lösungskandidat kann nicht sinnvoll durch Fitnessfunktion bewertet werden
- Problem der Abhängigkeit von Kodierung und genetischen Operatoren
 - Genetische Operatoren sind kodierungsspezifisch
 - => Reperaturmechanismen für Chromosomen
 - => Strafterme, die Fitness von Individuen außerhalb des zulässigen Suchraums verringern

Beispiel (Verlassen des Suchraums beim n -Damen-Problem)

• Kodierung 1

- Kodiere die Spaltenpositionen der Damen für jede Zeile in das Chromosom (Länge n , Allele $0, \dots, n - 1$; wie im einführenden Beispiel)
- Mit den Operatoren *Ein-Punkt-Crossover* und *Standardmutation* entstehen stets wieder gültige Lösungskandidaten

→ Suchraum wird nicht verlassen

• Kodierung 2

- Kodiere die Feldpositionen der Damen in das Chromosom (Länge n , Allele $0, \dots, n^2 - 1$)
- Mit den Operatoren *Ein-Punkt-Crossover* und *Standardmutation* können ungültige Lösungskandidaten entstehen, die mehrere Damen auf ein Feld setzen

→ Suchraum wird verlassen

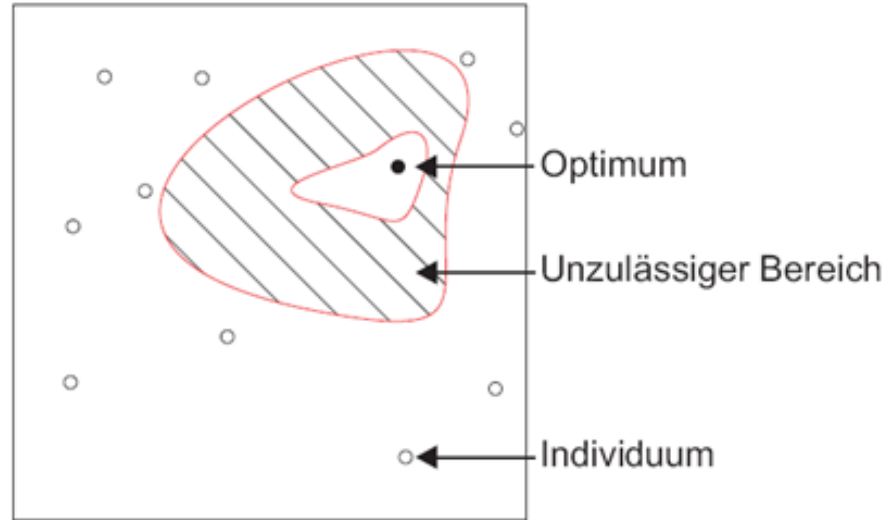
VERLASSEN DES SUCHRAUMS

Lösungsstrategien für die 2. Kodierung:

- Verwende andere Kodierung, die den Suchraum nicht verlässt, z.B. Kodierung 1
- Kodierungsspezifische genetische Operatoren
 - Mutation: Verhindere Erzeugung von Allelduplikaten
 - Crossover: Filtere die Feldnummer je Chromosom, die im jeweils anderen nicht vorkommt und wende auf das Resultat das Ein-Punkt-Crossover an
- Reperaturmechanismen
 - Ersetze mehrfach vorkommende Feldnummern derart, dass alle Feldnummern verschieden sind
- Strafterm
 - Verringere die Fitness pro Mehrfachbelegung, eventuell mit Gewichtungsfaktor

NICHT ZUSAMMENHÄNGENDE SUCHRÄUME

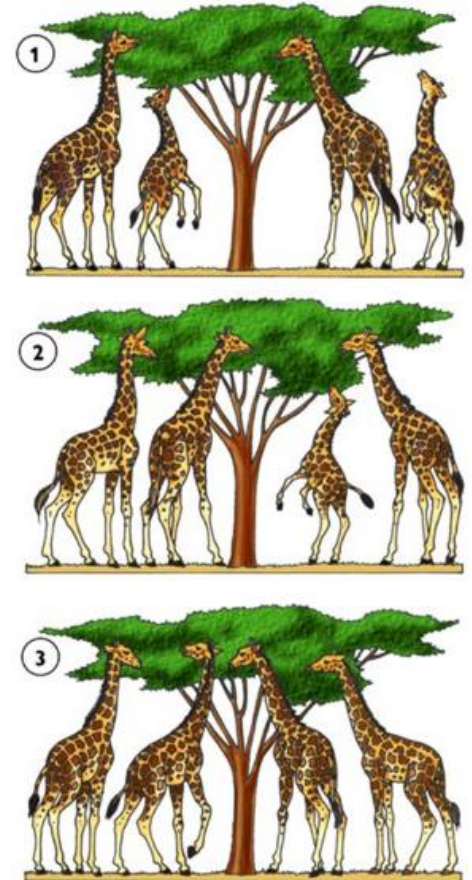
- Bei nicht zusammenhängenden Suchräumen können Reparaturmaßnahmen das Erreichen des Optimums verhindern, da der unzulässige Bereich nicht überwunden werden kann



- Hier können Strafterme sinnvoll sein, da sie unzulässige Lösungen nicht generell ausschließen

SELEKTION

- **Prinzip:** Bessere Individuen (entsprechend Fitness) sollen mit höherer Wahrscheinlichkeit Nachkommen haben (differentielle Reproduktion)
- Reduziert den Überschuss an Lösungskandidaten
- Verringert die Diversität (Varianz) innerhalb der Population
- Gibt der Optimierung eine Richtung
- Intensität des Konkurrenzkampfes wird als **Selektionsdruck** bezeichnet



SELEKTIONSDRUCK

- Wahl des Selektionsdrucks wird durch zwei gegensätzliche Ziele bestimmt:
- Erkundung des Suchraums (Exploration)
 - Der Suchraum soll möglichst weiträumig durchsucht werden, damit die Chancen steigen, das globale Optimum zu finden
 - D.h. die Individuen sollten möglichst breit über den Suchraum verteilt sein
=> geringer Selektionsdruck
- Ausbeutung guter Individuen (Exploitation)
 - Der Suchraum soll möglichst zielstrebig durchschritten werden, hin zu (eventuell lokalem) Optimum
 - D.h. die Individuen sollten möglichst gut dem Gradienten der Fitness folgen
=> hoher Selektionsdruck

BESTIMMUNG DES SELEKTIONSDRUCKS

- Grenzfälle des Selektionsdrucks sollten vermieden werden
- Zu niedriger Selektionsdruck
 - Gute Individuen vermehren sich kaum, schlechte bleiben erhalten
 - Evolutionärer Algorithmus degeneriert zur Zufallssuche
 - Keine oder sehr langsame Konvergenz
- Zu hoher Selektionsdruck
 - Variabilität der Population nimmt rasch ab
 - „Superindividuen“ dominieren
 - Suchraum wird nur eingeschränkt durchsucht
 - Vorzeitige Konvergenz zu einem lokalen Optimum

BESTIMMUNG DES SELEKTIONSDRUCKS

■ Erfolgversprechende Strategie: **Zeitabhängiger Selektionsdruck:**

- Am Anfang der Optimierung geringer Selektionsdruck
- Später höherer Selektionsdruck

=> Zunächst großräumige Erkundung des Suchraums, später gezielte Verbesserung in erfolgsversprechendsten Regionen

■ Selektionsdruck wird durch Skalierung der Fitnessfunktion oder durch Art/Parameter des Selektionsverfahrens gesteuert

■ Wichtige Selektionsverfahren

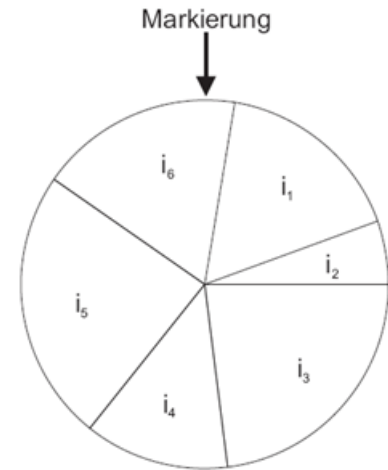
- Glückradauswahl
- Rangauswahl
- Turnierauswahl

■ Skalierungsmethoden

- Anpassung der Fitnessfunktion
- Linear-dynamische Skalierung
- σ -Skalierung

GLÜCKSRADAUSWAHL

- Bekanntestes Selektionsverfahren
- Sektorgrößen entsprechen relativen Fitnesswerten den Individuen
- Auswahl der Zwischenpopulation:
 - Drehe das Rad und wähle das Individuum, das dessen Sektor unter der Markierung liegt
 - Drehe für jedes benötigte Individuum (Populationsgröße) ein Mal
- **Fitnessproportionale Selektion**
- Fitness muss nichtnegativ sein (oder gemacht werden)



$$f_{\text{rel}}(i) = \frac{f_{\text{abs}}(i)}{\sum_{i' \in \text{pop}(t)} f_{\text{abs}}(i')}$$

DOMINANZPROBLEM

- Ein Individuum mit sehr hoher Fitness kann die Auswahl dominieren
- Dominanz wird in jeder Generation stärker
- Population besteht dann nur noch aus gleichen oder sehr ähnlichen Individuen (Crowding)
- Crowding
 - Vorzeitige Konvergenz gegen (lokale) Optima
 - Nachteil: Diversität der Population geht verloren
=> Keine Erkundung des Suchraums, sondern nur lokale Optimierung



VARIANZPROBLEM

- Problem: Auswahl der Individuen ist zwar fitnessproportional, aber dennoch zufällig
- Zahl der Nachkommen kann stark vom Erwartungswert abweichen
=> es ist nicht garantiert, dass das beste Individuum in die nächste Generation kommt
- Einfache Lösung:
 - Diskretisierung des Fitnesswertbereichs
 - Berechne Mittelwert $\mu_f(t)$ und Standardabweichung $\sigma_f(t)$ der Population
 - Deterministische Auswahl:
 - 0 Nachkommen falls $f(i) < \mu_f(t) - \sigma_f(t)$
 - 1 Nachkomme falls $\mu_f(t) - \sigma_f(t) \leq f(i) \leq \mu_f(t) + \sigma_f(t)$
 - 2 Nachkommen falls $f(i) > \mu_f(t) + \sigma_f(t)$

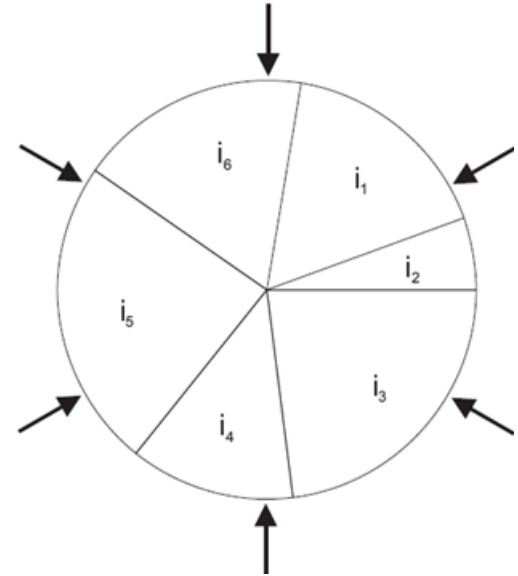
ERWARTUNGSWERTMODELL

- Erwartungswertmodell ist weitere Lösungsmöglichkeit für das Varianzproblem bei der Glücksradauswahl
- Erzeuge für jeden Lösungskandidaten $\lfloor f_{\text{rel}}(i) \cdot \text{popsize} \rfloor$ Individuen
- Fülle Zwischenpopulation durch Glücksradauswahl auf
- Variante: Modifizierte Glücksradauswahl
 - Pro erzeugtem Nachkommen wird die Fitness des Individuums um festen Betrag Δf verringert
 - Wird Fitness eines Individuums negativ, kann es keine weiteren Nachkommen erzeugen
 - Wahl von Δf so, dass bestes Individuum höchstens k Nachkommen erzeugen kann:
$$\Delta f = \frac{1}{k} \max\{f(i) \mid i \in \text{pop}(t)\}$$

STOCHASTIC UNIVERSAL SAMPLING

- Alternative zum Erwartungswertmodell
- Mehrere gleichverteilte Markierungen
- Glücksrad wird nur ein Mal gedreht
- Auswahl der markierten Individuen

=> überdurchschnittlich gute Individuen kommen sicher in die Zwischenpopulation



RANGAUSWAHL

■ Idee:

- Sortiere Individuen absteigend nach Fitness
- Rangliste definiert Wahrscheinlichkeitsverteilung:
Auswahlwahrscheinlichkeit sinkt mit steigendem Rang
- Glücksradauswahl mit dieser Verteilung

■ Vorteile:

- Dominanzproblem vermieden, da absolute Fitnesswerte nicht direkt in Auswahlwahrscheinlichkeit eingehen
- Durch Abfallgeschwindigkeit der Auswahlwahrscheinlichkeit in der Rangliste ist Selektionsdruck steuerbar

■ Nachteil:

- Zusätzlicher Aufwand für Sortierung

TURNIERAUSWAHL

■ Idee:

- Wähle n Individuen als Teilnehmer eines Turniers zufällig (z.B. uniform, mit zurücklegen) aus der Population aus
- Bestes Individuum des Turniers gewinnt und erzeugt Nachkommen
- Turniergröße n : $2 \leq n \leq \text{Populationsgröße}$

■ Vorteile:

- Dominanzproblem vermieden, da absolute Fitnesswerte nicht direkt in Auswahlwahrscheinlichkeit eingehen
- Selektionsdruck durch Wahl der Turniergröße steuerbar

* Extremfälle:

- $n=2$ => geringer Selektionsdruck
- $n = \text{Populationsgröße}$ => maximaler Selektionsdruck

ELITISMUS

■ Problem:

- Bei den meisten Selektionsverfahren (Ausnahme: Erwartungswertmodell und seine Varianten) gelangt das beste Individuum nicht sicher in die Zwischenpopulation
- Alle Individuen, auch das Beste, werden durch die genetischen Operatoren modifiziert
=> Fitness des besten Individuums kann sich verschlechtern, d.h. gute Lösungen können verloren gehen

■ Lösung: **Elitismus**

- Übernimm die Elite der Population, d.h. die besten n Individuen direkt in die nächste Generation
- Selektiere die verbleibenden Individuen mit beliebigem Verfahren

LOKALER ELITISMUS

■ Idee:

- Elitismus nur zwischen Eltern und ihren Nachkommen
- Mutation: Nur mutierter Nachkomme mit mindestens der Fitness seines Eltern wird in die nächste Generation übernommen
- Crossover: Nur die beiden besten der vier am Crossover beteiligten Individuen werden in die nächste Generation übernommen

■ Vorteil:

- Hohe Konvergenzgeschwindigkeit gegen (lokales) Optimum

■ Nachteil:

- Hohe Wahrscheinlichkeit der vorzeitigen Konvergenz gegen lokales Optimum, da keine (temporären) Verschlechterungen erlaubt werden

DETERMINISTISCHES CROWDING

■ Idee:

- Erzeuge pro Elternpaar ein Nachkommenpaar mittels Crossover und Mutation
- Bilde dann Paare aus je einem Elter und dem ihm ähnlichsten Nachkommen
- Das jeweils fittere Individuum eines Paares wird in die nächste Generation übernommen

■ Eigenschaften:

- Umfasst Selektion (innerhalb des Eltern-Nachkommenpaares), Crossover und Mutation
- Diversität nimmt nur langsam ab
=> Langsame Konvergenz zu einem Optimum

SKALIERUNGSMETHODEN — VON DER ZIEL- ZUR FITNESSFUNKTION

- Evolutionäre Algorithmen unterscheiden zwischen der aus den Zielkriterien zusammengesetzten Zielfunktion (auch Bewertungsfunktion) und der Fitnessfunktion
- Zielfunktion $f(x) : \mathcal{X} \rightarrow \mathbb{R}$ gibt an, wie gut ein Chromosom die definierten Ziele erfüllt
- Fitnessfunktion $\Phi(x) : \mathcal{X} \rightarrow \mathbb{R}^+$ gibt Chance des Individuums an, sich in der nächsten Generation zu reproduzieren
- Beide Funktionen sind oft identisch, oder es besteht ein einfacher funktionaler Zusammenhang $\Phi(x) = g(f(x))$
- Bei Minimierung der Zielfunktion ist eine Transformation immer erforderlich,

z.B.:

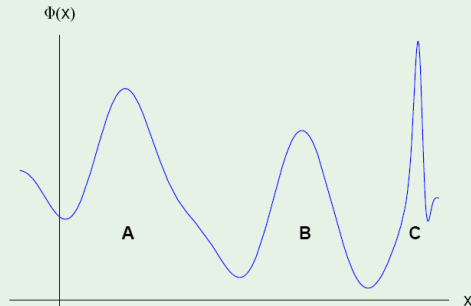
$$\Phi(x) = \begin{cases} c_{\max_{\text{est}}} - f(x) & \text{falls } c_{\max_{\text{est}}} - f(x) > 0 \\ 0 & \text{sonst} \end{cases}$$

EINFLUSS DER FITNESSFUNKTION

- Das Dominanzproblem resultiert aus den Eigenschaften der Fitnessfunktion
- Fitnessproportionale Selektion nutzt relative Fitness als Reproduktionswahrscheinlichkeit
- **Problem:** Variiert die Fitnessfunktion sehr stark, kann es zu vorzeitiger Konvergenz (Steckenbleiben in lokalem Optimum) kommen

Beispiel (Vorzeitige Konvergenz)

- Deckt zu Beginn kein Chromosom den Bereich **C** ab, wird die Population aufgrund der Selektion in den Bereichen **A** und **B** verbleiben
- Individuen, die sich dem Übergangsbereich von **B** zu **C** nähern haben nur mit geringer Wahrscheinlichkeit Nachkommen



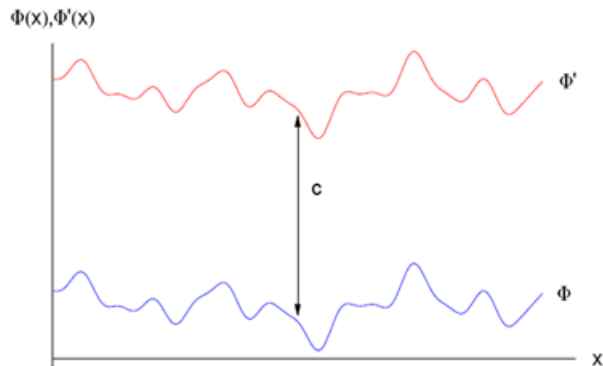
VERSCHWINDENDER SELEKTIONSDRUCK

- Verhältnis von absoluter Höhe der Fitnesswerte und ihrer Varianz entscheidend für Selektionsdruck
- Verschwindender Selektionsdruck durch additive Konstante

Maximierung der Funktion $\Phi' : \mathcal{X} \rightarrow \mathbb{R}$ ist äquivalent zur Maximierung der Funktion $\Phi : \mathcal{X} \rightarrow \mathbb{R}$, da gilt $\Phi'(x) = \Phi(x) + c, c \in \mathbb{R}$

Im Falle $c \gg \sup_{x \in \mathcal{X}} \Phi(x)$ gilt für die relative Fitness

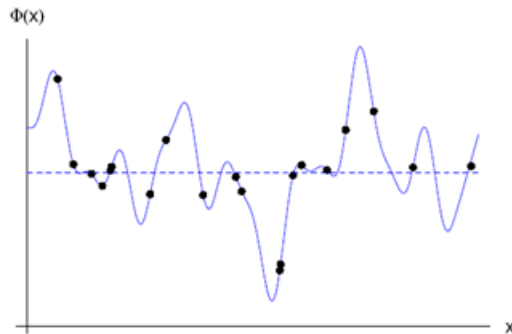
$\forall x \in \mathcal{X} : \Phi'_{\text{rel}}(x) \approx \frac{1}{\text{popsize}}$, d.h. der Selektionsdruck ist (zu) gering



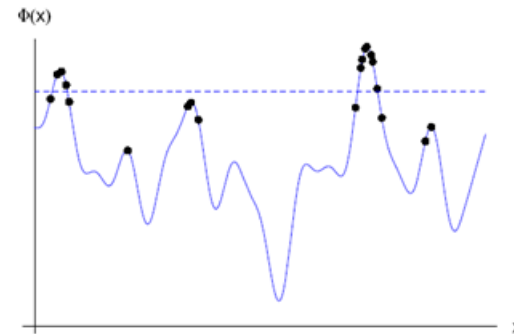
Obwohl die Maxima an gleichen Stellen liegen, sind sie für einen Genetischen Algorithmus dennoch unterschiedlich schwer zu finden, da die relativen Unterschiede der Fitness der Funktion Φ' (zu) gering sind.

VERSCHWINDENDER SELEKTIONSDRUCK

- Optimierung kann verschwindenden Selektionsdruck selbst erzeugen, indem durchschnittliche Fitness steigt
 - Beobachtung:
 - Zu Beginn Selektionsdruck, da Fitnesswerte zufällig verteilt
 - Später geringer Selektionsdruck, da alle Fitnesswerte hoch
- => Das Gegenteil wäre gut



Beginn der Optimierung



Spätes Stadium der Optimierung

SKALIERUNG DER FITNESSFUNKTION

- Die beschriebenen Probleme können durch eine Skalierung der Fitnessfunktion behoben werden
- Ansätze:
 - Linear-dynamische Skalierung
 - σ -Skalierung
- Linear-dynamische Skalierung
 - Benutze Minimum der letzten b Generationen

$$\Phi_{\text{lds}}(x) = \alpha f(x) - \min\{f(x') \mid x' \in \bigcup_{i=t-b}^t \text{pop}(i)\}, \quad \alpha > 0,$$

- Üblicherweise ist $\alpha > 1$

■ Idee:

- Ziehe unterdurchschnittliche Fitness ab:

$$\Phi_{\sigma}(x) = f(x) - (\mu_f(x) - \beta \cdot \sigma_f(x)), \quad \beta > 0$$

- Mittelwert der Fitness: $\mu_f(t) = \frac{1}{\text{popsize}} \sum_{x \in \text{pop}(t)} f(x)$

- Standardabweichung:

$$\sigma_f(t) = \sqrt{\frac{1}{\text{popsize} - 1} \sum_{x \in \text{pop}(t)} (f(x) - \mu_f(t))^2}$$

- Problem: Wie soll Parameter β gewählt werden?

Σ-SKALIERUNG

■ Ansatz:

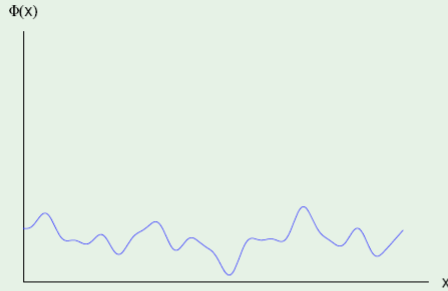
- Analysiere den Variationskoeffizienten der Fitness

$$v = \frac{\sigma_f}{\mu_f} = \frac{\sqrt{\frac{1}{|\mathcal{X}|-1} \sum_{x' \in \mathcal{X}} \left(f(x') - \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} f(x) \right)^2}}{\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} f(x)}$$

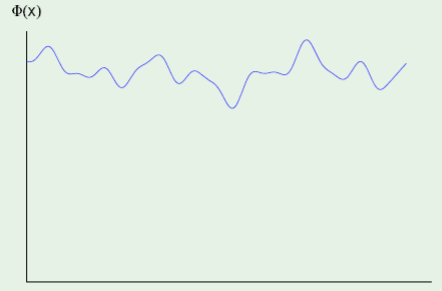
- Ein Variationskoeffizient von etwa 0,1 erzeugt eine gute Balance zwischen Erkundung des Suchraums und lokaler Optimierung
- Falls v von diesem Wert abweicht wird die Fitnessfunktion angepasst, um ihn wieder zu erreichen (Skalierung, Exponentiation)
- Variationskoeffizient wird für aktuelle Population berechnet
- σ-Skalierung der Fitness mit $\beta=1/v^*$, wobei $v^*=0,1$

Σ -SKALIERUNG

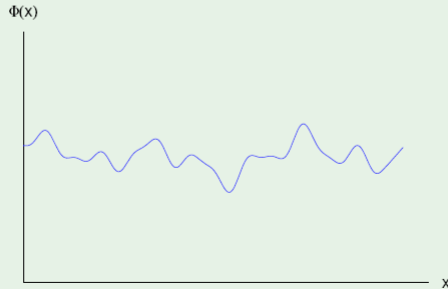
Beispiel



Variationskoeffizient $v \approx 0.2$



Variationskoeffizient $v \approx 0.05$



Variationskoeffizient $v \approx 0.1$

Ein zu hoher Variationskoeffizient führt zu vorzeitiger Konvergenz, ein zu niedriger zu verschwindendem Selektionsdruck. Ein guter Wert ist erfahrungsgemäß $v \approx 0.1$.

ZEITABHÄNGIGE FITNESSEXPOSITIONATION

■ Boltzmann-Selektion

- Exponentiation der Fitness: $g(x) = \exp\left(\frac{f(x)}{kT}\right)$
- Selektionsdruck wird durch Temperaturparameter T gesteuert
- Temperatur nimmt im Laufe der Optimierung ab, z.B. linear

■ Idee der Boltzmann-Selektion ähnelt Simulated Annealing

- Zu Beginn ist die Temperatur hoch, die relativen Unterschiede zwischen den exponierten Fitnesswerten sind also gering
- Im Verlauf der Optimierung sinkt Temperatur, d.h. die relativen Fitnessunterschiede werden immer wichtiger
- Im Extremfall ($T=0$) werden nur die besten Individuen ausgewählt

EVOLUTIONSSTRATEGIEN (ES)

■ Repräsentation

[Rechenberg, Schwefel]

- Vektor reeller Zahlen
- Jede Komponente wird als Koordinate eines n-dimensionalen Optimierungsproblems interpretiert
=> Beschränkung auf numerische Optimierung

■ Genetische Operatoren

- Mutation durch Addition normalverteilter Zufallszahlen
- Rekombinationen mehrere Vektoren (anstatt Crossover)

■ Selektion

- Elitistisch
- Deterministisch

ELEMENTARE EVOLUTIONSSTRATEGIE (1+1)-ES

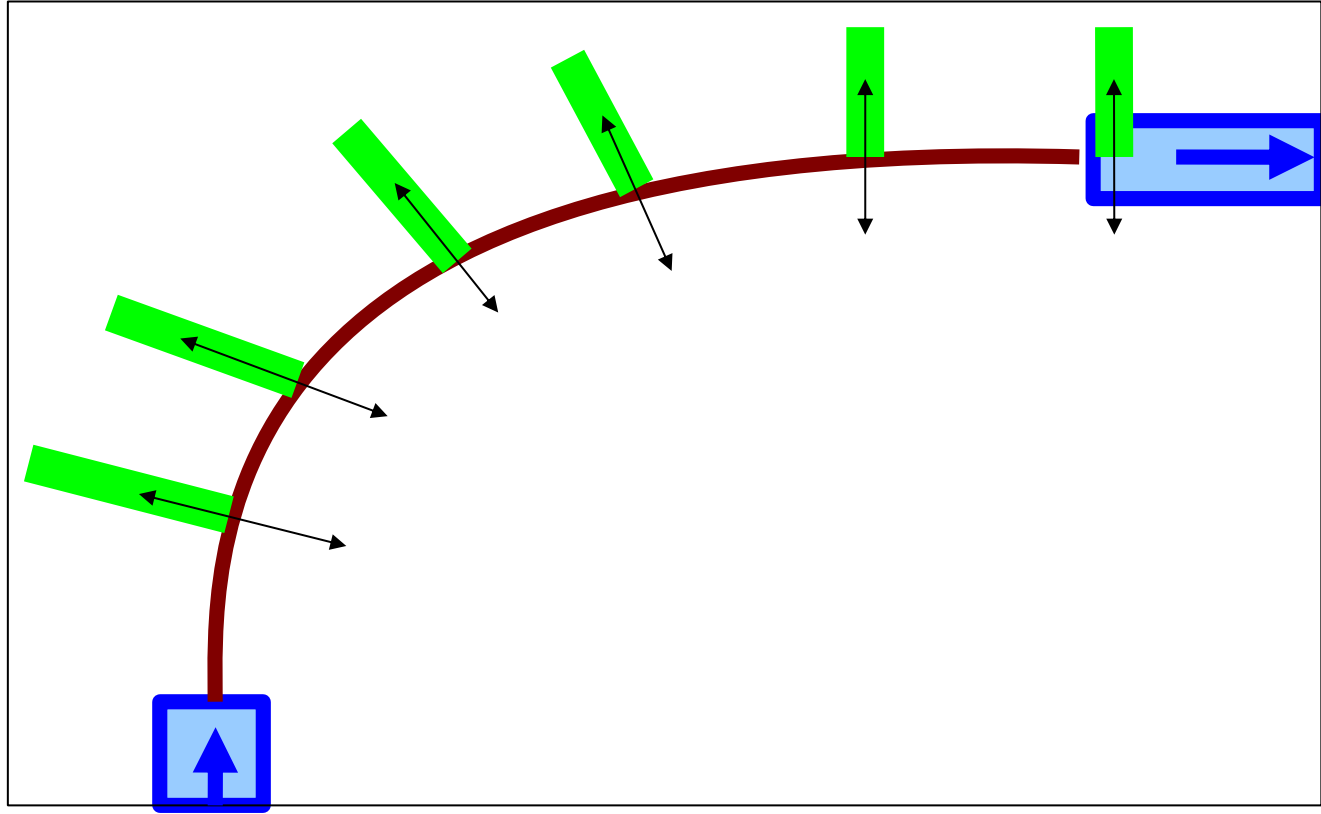
Beispiel ((1+1)-Evolutionstrategie)

- Ein Elter erzeugt einen Nachkommen, das bessere Individuum überlebt
- (1+1)-Evolutionstrategie entspricht dem Verfahren des *Zufallsabstieg*
- Erste Variante einer Evolutionstrategie (Rechenberg, 1964)
- Genutzt zur experimentellen Optimierung
 - Tragflächenprofil
 - Rohrkrümmer
- Frage: Wie stark soll das Individuum mutiert werden?

BEISPIEL: RECHENBERGS TUBING PROBLEM

Der Treibstoff soll von dem Rohr (links unten) über eine Leitung in das Rohr rechts oben geführt werden.

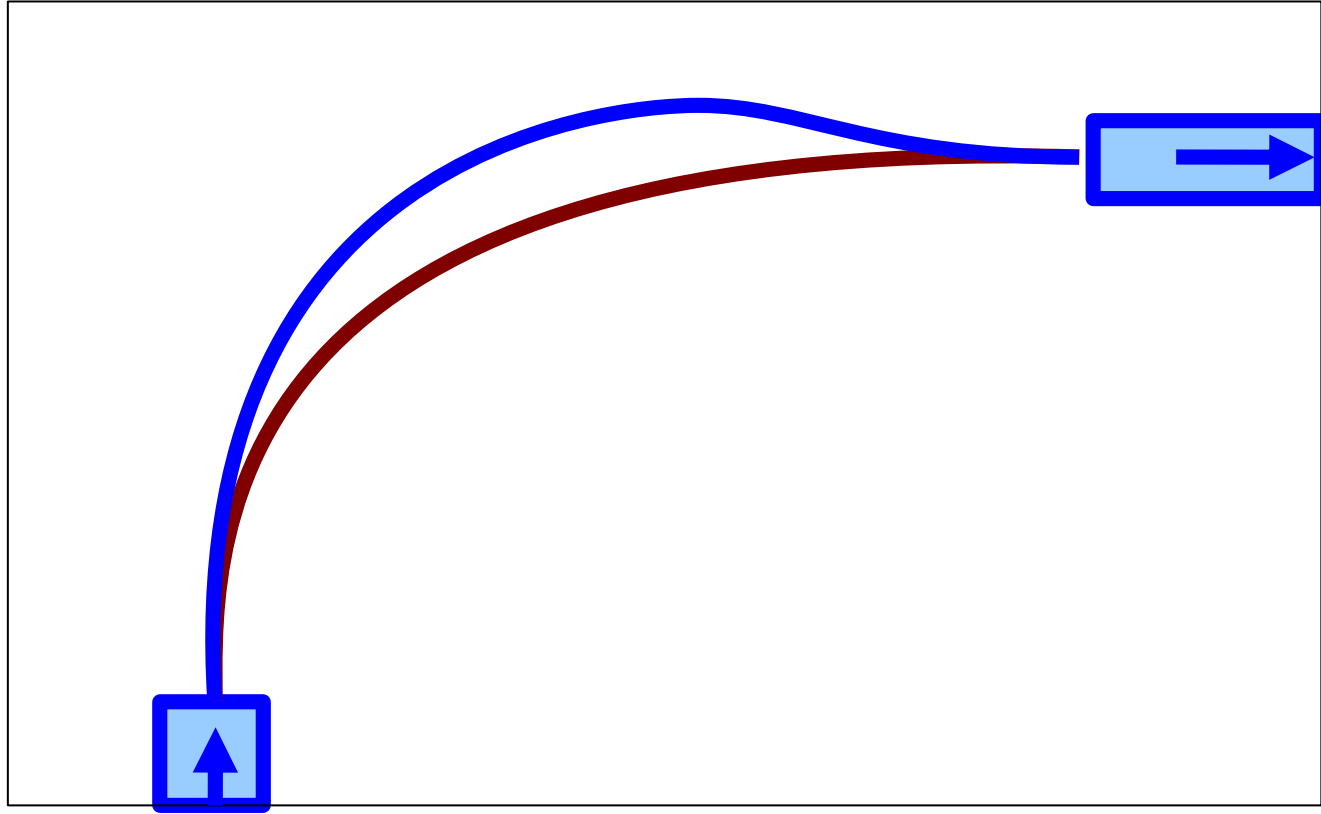
Wie muss das Verbindungsstück geformt sein, dass der Strömungswiderstand minimal wird?



BEISPIEL: RECHENBERGS TUBING PROBLEM

Der Treibstoff soll von dem Rohr (links unten) über eine Leitung in das Rohr rechts oben geführt werden.

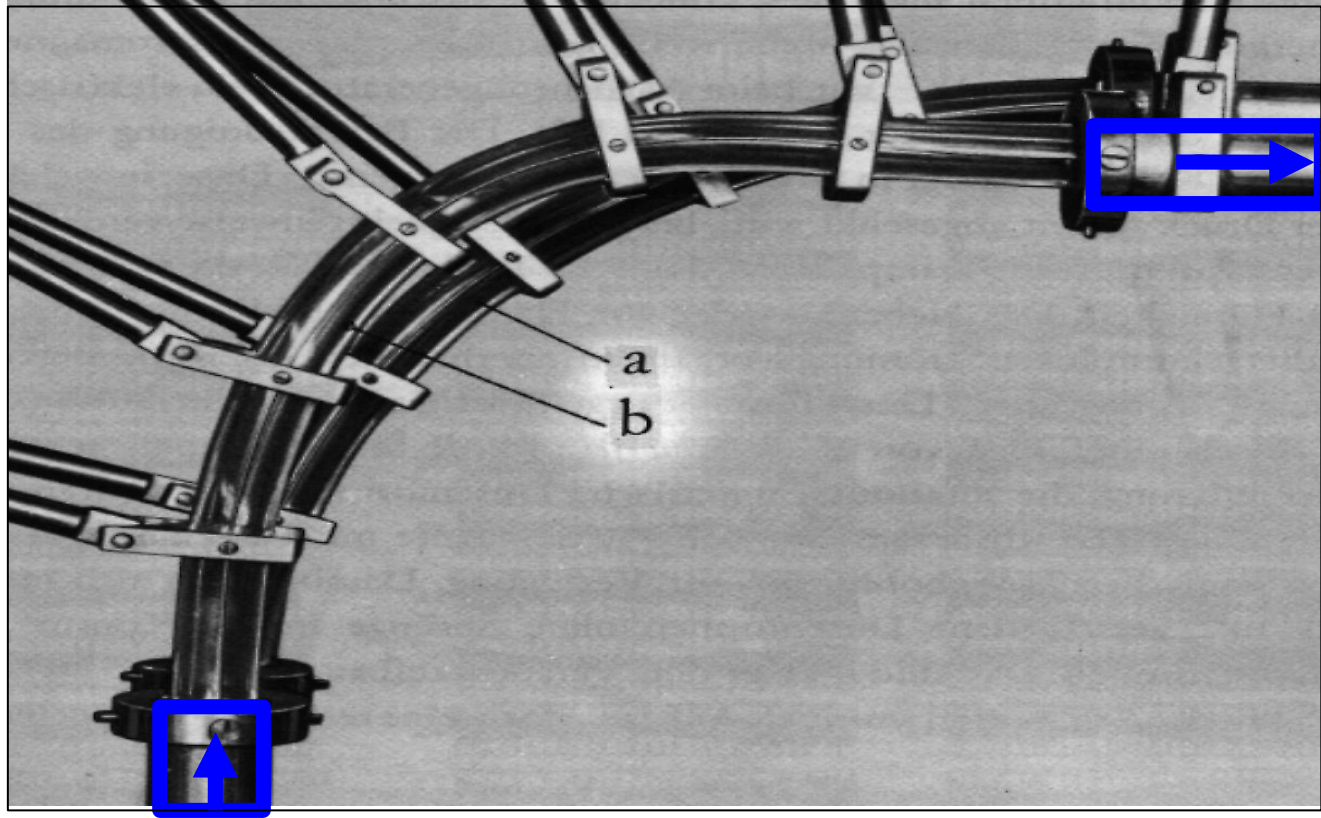
Wie muss das Verbindungsstück geformt sein, dass der Strömungswiderstand minimal wird?



BEISPIEL: RECHENBERGS TUBING PROBLEM

Der Treibstoff soll von dem Rohr (links unten) über eine Leitung in das Rohr rechts oben geführt werden.

Wie muss das Verbindungsstück geformt sein, dass der Strömungswiderstand minimal wird?



ABLAUF DER (1+1)-ES

Algorithmus

- 1 Erzeuge eine zufällige ‚Anfangspopulation‘, d.h. Startpunkt \vec{x}^0
- 2 Erzeuge einen normalverteilten Zufallsvektor \vec{z} und bestimme den Nachkommen als $\vec{x}' = \vec{x}^t + \vec{z}$
- 3 Berechne die Fitness der Individuen mit der Zielfunktion
- 4 Selektiere den Elter der nächsten Population

$$\vec{x}^{(t+1)} = \begin{cases} \vec{x}^{(t)} & \text{falls } f(\vec{x}^{(t)}) \leq f(\vec{x}') \\ \vec{x}' & \text{sonst} \end{cases}$$

- 5 Gehe zu Schritt 2 bis ein Abbruchkriterium erfüllt ist

ANMERKUNGEN ZUR (1+1)-ES

- Erlaubt keine Rekombination
- Erweiterung auf μ Individuen in der Population möglich
- Erweiterung auf λ Nachkommen in der Population möglich
- Dann:
 - Neben Mutation auch Rekombination möglich
 - Alternatives Selektionsverfahren möglich

ALLGEMEINER ABLAUF EINER ES

Algorithmus

- ① Erzeuge eine zufällige Anfangspopulation von μ Individuen
- ② Erzeuge λ Nachkommen. Dazu jeweils:
 - ① Wähle ρ Eltern zur Erzeugung eines Nachkommen aus
 - ② Rekombiniere diese Eltern zu einem neuen Individuum
 - ③ Mutiere das Individuum
- ③ Bewerte die Individuen gemäß ihrer Fitness
- ④ Selektiere die μ Eltern der nächsten Population
- ⑤ Gehe zu Schritt 2 bis ein Abbruchkriterium erfüllt ist

SELEKTION BEI ES

■ Prinzip

- Übernahme nur die besten Individuen in die nächste Generation (deterministische, streng elitäre Selektion)

■ Varianten

- „+“-Strategie (Plus-Strategie)
Wähle aus den μ Eltern und λ Nachkommen die μ besten Individuen aus;
Notation: $(\mu+\lambda)$
- „ , “-Strategie (Komma-Strategie)
Wähle nur aus den λ Nachkommen die μ besten Individuen aus; Notation: (μ,λ)

■ Vergleich

- Plus-Strategie garantiert, dass sich die Population nicht verschlechtert, erhöht aber auch die Gefahr der vorzeitigen Konvergenz gegen lokale Optima

REPRODUKTION BEI ES

■ Prinzip

- Wähle uniform $1 \leq \rho \leq \mu$ Eltern aus der Population, die ein neues Individuum durch Rekombination erzeugen

■ Anmerkungen

- Im Falle $\rho > 2$ liegt Multirekombination vor
- Bereits gewählte Eltern werden häufig nicht aus der Menge der potentiellen Eltern entfernt, können also mehrfach gewählt werden

REKOMBINATION

■ Prinzip

- Mische die genetischen Informationen der ρ gewählten Eltern

■ Intermediäre Rekombination

- Mittele über die Eltern, d.h. der Nachkomme liegt im Schwerpunkt der Eltern

$$\vec{r} := \frac{1}{\rho} \sum_{i=1}^{\rho} \vec{v}_i$$

■ Diskrete Rekombination

$$\vec{r} := \sum_{i=1}^{\dim \vec{x}} (\vec{e}_i^T \vec{v}_{m_i}) \vec{e}_i$$

- Wähle jede Komponente zufällig aus einem der Eltern

mit $m_i := \text{rand}\{1, \dots, \rho\}$ gleichverteilte Zufallszahlen,

\vec{e}_i Einheitsvektor in i-ter Dimension

MUTATION DURCH ADDITION EINES ZUFALLSVEKTORS

■ Prinzip

- Variiere genetische Information durch Addition normalverteilter Zufallszahlen

■ Anforderungen

- Kleine Änderungen sollen mit hoher Wahrscheinlichkeit auftreten
- Große Änderungen sollen nur selten auftreten
- Stärke der Mutation soll variabel sein
- Stationarität unter Abwesenheit von Selektion

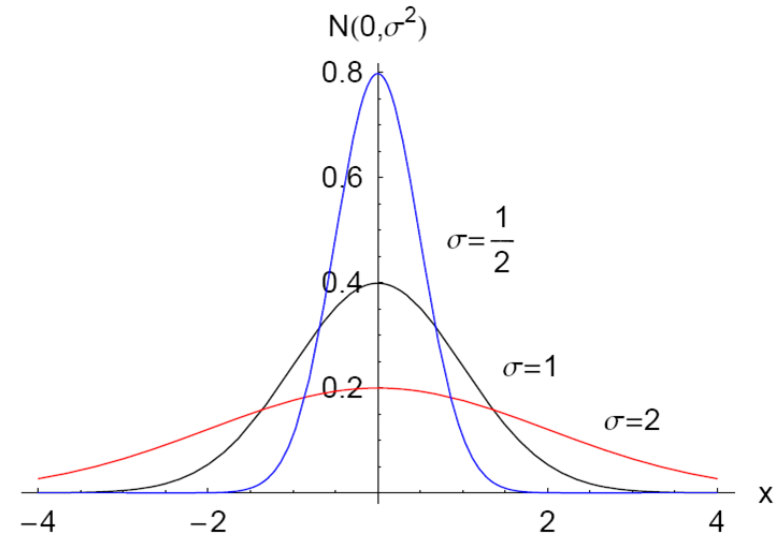
■ Anmerkung

- Mutationsrate in ES wesentlich höher als in Genetischen Algorithmen

NORMALVERTEILTE MUTATIONEN

■ Prinzip

- Kleine Varianz: Kleine Änderungen des Chromosoms
=> lokale Suche (Ausbeutung guter Individuen)
- Große Varianz: Große Änderungen des Chromosoms
=> globale Suche (Erkundung des Suchraums)



ANPASSEN DER MUTATIONSSTÄRKE

■ 1/5 Erfolgsregel

- Wähle globale Mutationsstärke (Standardabweichung σ) so, dass die Konvergenzgeschwindigkeit maximal ist
- Überprüfe dazu in regelmäßigen Abständen die Erfolgsrate p_s , d.h. das Verhältnis der erfolgreichen Nachkommen zur Gesamtzahl der erzeugten Nachkommen
- p_s misst die Wahrscheinlichkeit dass ein Nachkomme den Elter ersetzt
- Um gute lokale Konvergenzgeschwindigkeit zu erreichen, sollte die gemessene Erfolgsrate bei ungefähr 1/5 liegen
- Heuristisch aus Kugelmodell abgeleitet

■ Anmerkungen

- Wertet globale, über Generationen gesammelte Informationen aus
- Steuerung einer globalen Mutationsrate möglich

1/5 ERFOLGSREGEL

Definition

$$\sigma^{(g+1)} = \begin{cases} c \sigma^{(g)}, & \text{if } p_s < 1/5 \\ \sigma^{(g)}, & \text{if } p_s = 1/5 \\ \sigma^{(g)}/c, & \text{if } p_s > 1/5, \end{cases}$$

- Die Wahl von c hängt von der Zielfunktion, der Dimension und der Anzahl der Generationen ab
- Empfehlung von Rechenberg: $0.85 \leq c \leq 1$

■ Interpretation

- Ist die Erfolgsrate zu hoch, erhöhe die Mutationsstärke.
Annahme: Schnelleres Vorgehen in Richtung des Optimums ist möglich
- Ist die Erfolgsrate zu niedrig, senke die Mutationsstärke.
Annahme: Feinere Analyse des Suchraums nötig

SELBSTADAPTION

■ Prinzip

- Alle Individuen besitzen eigene Strategieparameter, wie beispielsweise die Mutationsstärke
- Die Strategieparameter werden nicht explizit eingestellt, sondern sind selbst Teil des evolutionären Geschehens
- Strategieparameter entwickeln sich mit den Individuen im Verlauf der Optimierung (Koevolution)

■ Realisierung

- Die Strategieparameter werden wie auch die Objektparameter im Chromosom kodiert
- Ein Individuum besteht somit aus dem Tupel (\vec{x}, \vec{s})
- Die Strategieparameter werden ebenso den genetischen Operatoren unterzogen wie die Objektparameter
- Zunächst werden Strategiekomponenten eines Nachkommen erzeugt (Rekombination und Mutation) und dann für die Erzeugung der Objektparameter verwendet

SELBSTADAPTION

■ Anmerkungen

- Strategieparameter haben keinen direkten Einfluss auf die Fitnessberechnung
- Gute Strategieparameter pflanzen sich indirekt durch die gute Fitness ihrer Individuen fort
- Strategieparameter parametrisieren die genetischen Operatoren, im Wesentlichen die Mutation

VARIANTEN VON MUTATIONSOPERATOREN

- Verschiedene Varianten von Mutationsoperatoren
 - **Isotropische** Mutation:
Eine einzige Schrittweite für alle Dimensionen des Optimierungsproblems
 - **Skalierte** Mutation:
Eine eigene Schrittweite für jede Dimension
 - **Korrelierte** Mutation:
Variiere neben den Mutationsschrittweiten auch die Korrelation der Zufallsvariablen
 - **Gerichtete** Mutation:
Bevorzugung von bestimmten Richtungen durch Einsatz schiefsymmetrischer Verteilungen
- Isotropische Mutation ist älteste Variante
- Eingesetzt wird häufig die skalierte Mutation
- Ziel sind möglichst gut an die Topologie des zu optimierenden Problems angepasste Operatoren

EVOLUTIONSSTRATEGIEN VS. GENETISCHE ALGORITHMEN

| | ES | GA |
|------------------|-------------------------|---------------------|
| Darstellung | reellwertig | binär |
| Selektion | deterministisch, elitär | stochastisch |
| Reproduktion | Rekombination, Mutation | Crossover, Mutation |
| Mutation | häufig | selten |
| Selbstadaptation | + | - |

■ Repräsentation

- Individuen repräsentieren Funktionen oder Programme
- Komplexe Chromosomen variable Länge

■ Genetische Operatoren

- Mutation durch Ersetzen von Teilbäumen
- Crossover durch Austausch von Teilbäumen verschiedener Individuen

■ Selektion

- Wie bei Genetischen Algorithmen

GP-REPRÄSENTATION

■ **Grundlage:** Formale Grammatiken

- \mathcal{T} : Menge der Terminalsymbole
- \mathcal{F} : Menge der Funktionssymbole
- Wahl von \mathcal{T} und \mathcal{F} ist problemspezifisch

■ **Terminalsymbole:** Benötigen keine Argumente

- Benutzereingaben
- Konstanten
 - Zufallszahlen: werden zu Beginn der GP mit zulässigen Werten initialisiert und behalten diese
 - Veränderliche Konstanten: können durch genetische Operatoren verändert werden

■ **Funktionssymbole:** Nichtterminale, benötigen Argumente

- Funktionen
- Operatoren (Wertzuweisungen, ...)
- Kontrollstrukturen (if-then-else, for, do-while, goto, ...)

BEISPIELE FÜR GENETISCHE PROGRAMMIERUNG

■ Symbolische Regression

- Bestimme zu gegebenen Daten eine Näherungsfunktion, welche die Summe der Fehlerquadrate minimiert
- Terminalsymbole $\mathcal{T} = \{x_1, \dots, x_m\} \cup \mathbb{R}$
- Funktionssymbole $\mathcal{F} = \{+, -, \cdot, /, \sqrt{}, \sin, \cos, \exp, \log, \dots\}$

■ Boolesche Funktionen

- Finde eine Funktion, die eine gegebene Wahrheitstabelle abbildet
- Terminalsymbole $\mathcal{T} = \{b_1, \dots, b_m\} \cup \{0, 1\}$
- Funktionssymbole $\mathcal{F} = \{\text{and, or, not, } \dots\}$

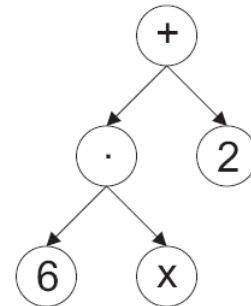
ABGESCHLOSSENHEIT DER FUNKTIONEN

- Die verwendeten Funktionen müssen für alle Belegungen ihrer Argumente auswertbar sein
- Bei einigen Funktionen muss der Definitionsbereich sinnvoll erweitert werden
- Beispiel: $x/0 \Rightarrow$ liefere maximalen Wert zurück
- Weitere problematische Funktionen:
 - Logarithmus
 - Wurzel
 - Tangens
 - ...

KODIERUNG DER INDIVIDUEN

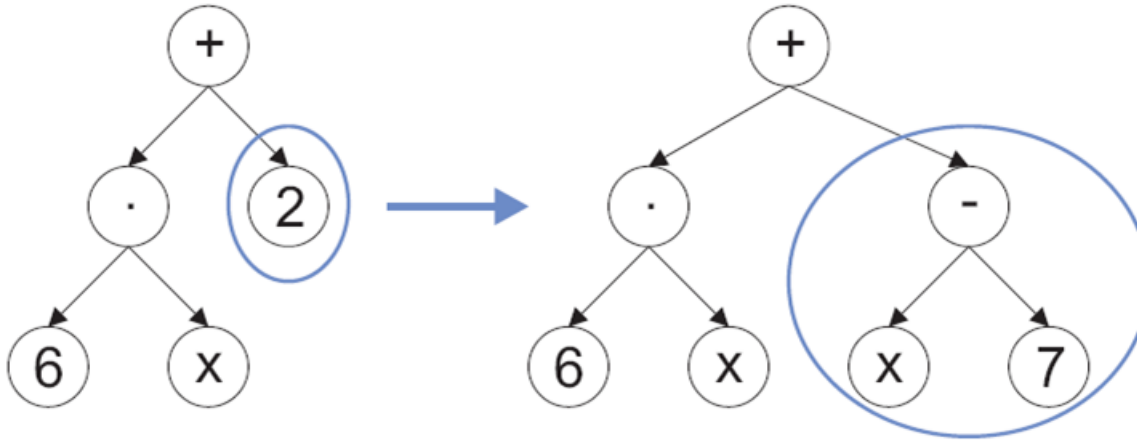
- Die Gene der Chromosomen eines Individuums enthalten Elemente der Menge
- Zugelassen werden nur Ausdrücke gemäß folgender rekursiver Definition (Prefixnotation):
 - Terminalsymbole sind symbolische Ausdrücke $\mathcal{C} = \mathcal{F} \cup \mathcal{T}$
 - Sind t_1, \dots, t_n symbolische Ausdrücke und ist $f \in \mathcal{F}$ ein n -stelliges Funktionssymbol, dann ist auch $(f t_1 \dots t_n)$ ein symbolischer Ausdruck
- Symbolische Ausdrücke in Prefixnotation können als Bäume dargestellt werden
- Beispiele:
 - $+ \cdot 6 \times 2$ ist ein symbolischer Ausdruck, der repräsentiert.
 - $4 \cdot 1 \times -$ ist kein symbolischer Ausdruck

$6 \cdot x + 2$
Als Baum:



MUTATION

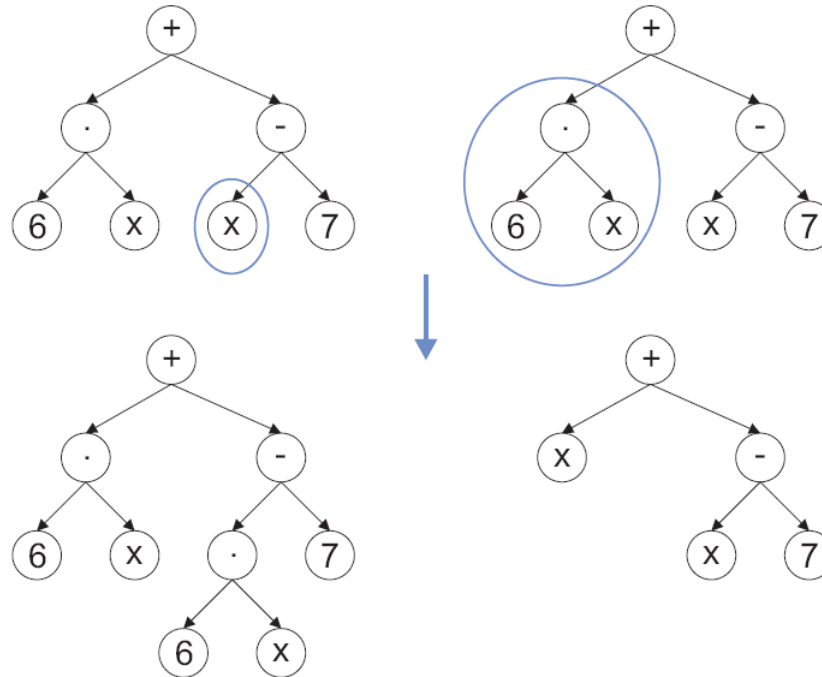
- Ersetzt Teilausdruck (Teilbaum) durch einen zufällig erzeugten neuen



- Mutation sollte selten vorkommen
- Zu ersetzenden Teilbäume sollten klein sein

CROSSOVER

- Tausche Teilausdrücke (Teilbaume) zwischen Individuen
- Operation ist variantenreicher als auf Vektoren, da selbst identische Individuen neuartige Nachkommen erzeugen können



ABLAUF DER GENETISCHEN PROGRAMMIERUNG

Algorithmus

- ① Erzeuge eine zufällige Anfangspopulation symbolischer Ausdrücke
Parameter:
 - Maximale Verschachtelungstiefe
 - Wahrscheinlichkeiten der Terminalsymbole
- ② Bewerte die Ausdrücke gemäß ihrer Fitness
 - Symbolische Regression: Summe der Fehlerquadrate zu allen Datenpunkten
 - Boolesche Funktionen: Anteil korrekter Ausgaben für die Menge der Eingaben
- ③ Selektiere die Individuen der Zwischenpopulation
- ④ Wende die genetischen Operatoren an
- ⑤ Gehe zu Schritt 2 bis ein Abbruchkriterium erfüllt ist

ZUSAMMENFASSUNG

■ Selektionsverfahren

- Monotone steigende Abbildung von Fitnessfunktion auf Auswahlwahrscheinlichkeit
- Selektionsdruck muss angepasst werden
- Rangordnungsverfahren, Skalierung der Fitnessfunktion

■ Evolutionsstrategien

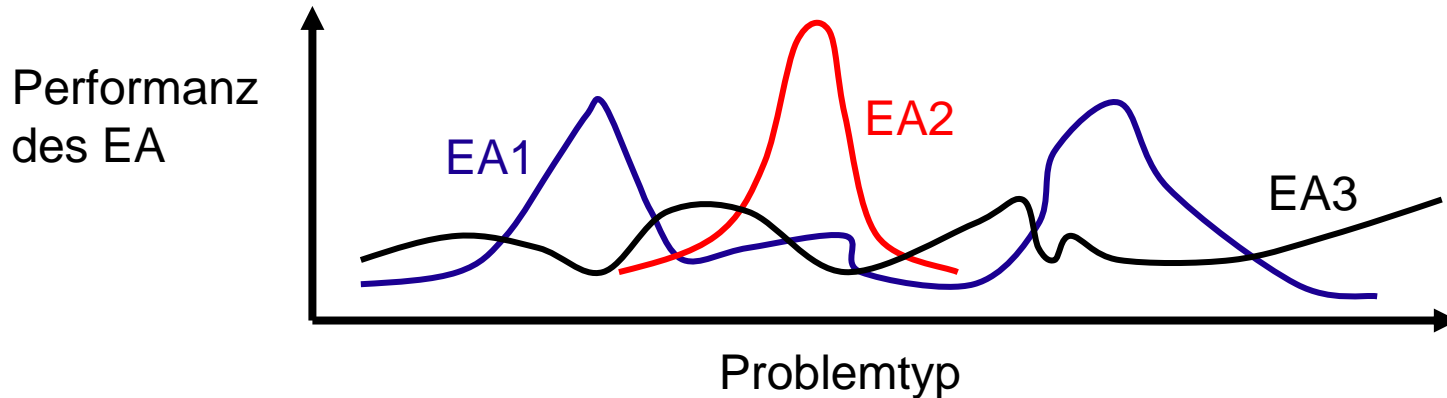
- Repräsentation durch Vektor reeller Zahlen
- Elitistische, deterministische Selektion
- Mutation durch Addition normalverteilter Zufallszahlen
- Mutation ist wichtigster Operator
- Selbstadaption zur Anpassung der Mutationsrate
- Rekombination mehrerer Vektoren

■ Genetische Programmierung

- Individuen repräsentieren Funktionen oder Programme
- Komplexe Chromosomen variabler Länge
- Mutation durch Ersetzen von Teilbäumen
- Crossover durch Tausch von Teilbäumen zwischen Individuen

ANWENDBARKEIT

- Evolutionäre Algorithmen können für vielfältige Probleme genutzt werden
- Biologische Inspiration häufig nicht mehr direkt sichtbar
- Verschiedene Probleme können unterschiedliche Repräsentationen/Operatoren erfordern



- => Wissen über Domäne kann für Auswahl oder Design des EA genutzt werden