# Evolutionäre Algorithmen

**Vorlesung 2**

**Optimierungsprobleme**
**Beispiel eines EA**
**Geschichte**

## Evolution und Optimierung

- **Evolution** produziert Lösungen

  Aufnahme von Energie – Produktion von genügend Nachkommen –

  Partnerfindung – Tarnung

- **Computer** sind universell

  Turing: Computer lösen jedes prinzipiell lösbare Problem

  Frage nach konkretem Algorithmus und Laufzeit

- **Evolutionäre Algorithmen** kombinieren beides



## Optimierungsprobleme

**Definition:** Optimierungsproblem

- Suchraum $\mathcal{S}$
- Bewertungsfunktion $f : \mathcal{S} \to \mathbb{R}$
- Vergleichsrelation „ $\succ$ “ $\in \{<, >\}$
- Menge der *globalen Optima*:

  $$\mathcal{X} = \{x \in \mathcal{S} \mid \forall_{x' \in \mathcal{S}} f(x) \succeq f(x')\}$$



## Beispiel: symmetrisches TSP

- $n$ Städten $V = \{v_1, \ldots, v_n\}$
- Straßen $E \subset V \times V$
- Fahrtzeit $\gamma : E \to \mathbb{R}$
- Gesucht: kostenminimale Rundreise, d.h. Permutation $(i_1, \ldots, i_n) \in \Pi_{1,n}$ mit minimalen Kosten $((i_1, \ldots, i_n)) =$

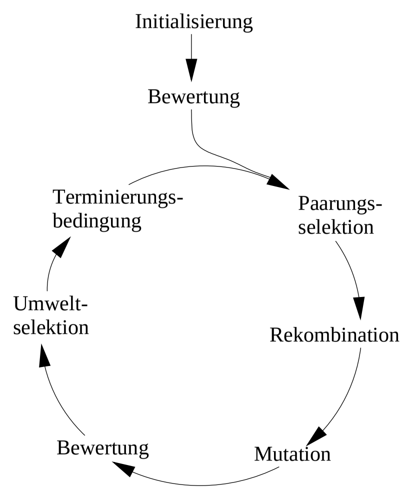
  $$\gamma((v_{i_n}, v_{i_1})) + \sum_{j=2}^{n} \gamma\left((v_{i_{j-1}}, v_{i_j})\right)$$

# Weitere Aspekte

- ▷ mehrere Zielgrößen
- ▷ Bewertungsfunktion nicht klar definiert
- ▷ keine vorgegebene Struktur für Lösungen
- ▷ verrauschte Zielgrößen
- ▷ zeitabhängige Bewertungsfunktion
- ▷ Güte nur im Vergleich meßbar
- ▷ Stabilität einer Lösung



# Evolutionärer Zyklus



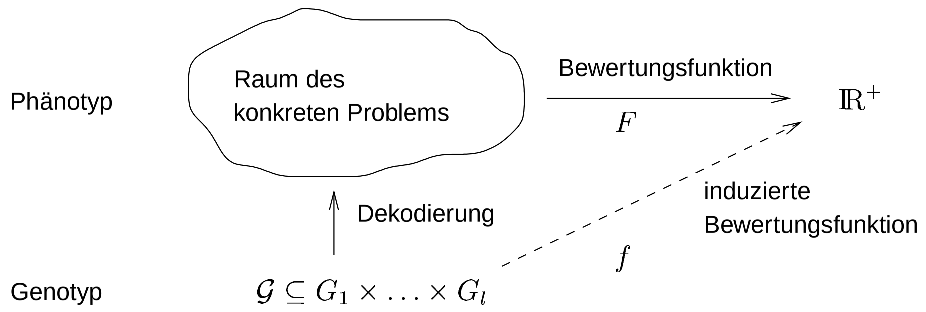



# EA in Pseudo-Notation

**EINGABEN:** Parameter $\mu$, $\lambda$, ..., Zielfunktion $F$
$t \leftarrow 0$
setze Parameter
erzeuge Population $P(t)$ der Größe $\mu$
bewerte Population $P(t)$ mit Zielfunktion $F$
**while** Terminierungsbedingung nicht erfüllt **do**
  selektiere Eltern $E$ für $\lambda$ Nachkommen aus $P(t)$
  $P' \leftarrow$ erzeuge Nachkommen durch Rekombination aus $E$
  $P'' \leftarrow$ mutiere die Individuen in $P'$
  bewerte Individuen in $P''$
  $t \leftarrow t + 1$
  $P(t) \leftarrow$ selektiere $\mu$ Individuen aus $P''$ (oder $P'' \circ P(t-1)$)
**end while**
**AUSGABE:** bestes Individuum aus $P(t)$



# Genotyp und Phänotyp



## Operatoren

- ▷ Mutationsoperator $M^{\xi} : \mathcal{G} \to \mathcal{G}$
- ▷ Rekombinationsoperator ($r \geq 2$ Eltern und $s > 1$ Kinder)
  $R^{\xi} : \mathcal{G}^r \to \mathcal{G}^s$
- ▷ Dekodierungsfunktion $dec : \mathcal{G} \to \mathcal{S}$
- ▷ Selektionsoperator $S^{\xi,dec,f} : \mathcal{G}^r \to \mathcal{G}^s$
  wobei für $S^{\xi,dec,f}(P) = \langle B_1, \ldots, B_s \rangle$ gilt: $B_i \in$ **set**$(P)$



## Beispiel: EA für TSP

- ▷ geeignete Repräsentation?
- ▷ wählen: $\mathcal{G} = \mathcal{S} = \Pi_{1,n}$
- ▷ geeignete Operatoren?



## Vertauschende Mutation

**EINGABEN:** Individuum $A = (A_1, \ldots, A_n)$
$B \leftarrow A$
$i_1 \leftarrow$ Zufallszahl aus $U(\{1, \ldots, n\})$
$i_2 \leftarrow$ Zufallszahl aus $U(\{1, \ldots, n\})$
$B_{i_1} \leftarrow A_{i_2}$
$B_{i_2} \leftarrow A_{i_1}$
**AUSGABE:** $B$

- ▷ Was verändert der Operator?



## Invertierende Mutation

**EINGABEN:** Individuum $A = (A_1, \ldots, A_n)$
$B \leftarrow A$
$i_1 \leftarrow$ Zufallszahl aus $U(\{1, \ldots, n\})$
$i_2 \leftarrow$ Zufallszahl aus $U(\{1, \ldots, n\})$
**if** $i_1 > i_2$ **then**
  vertausche $i_1$ und $i_2$
**end if**
**for** $j \in \{i_1, \ldots, i_2\}$ **do**
  $B_{i_2+1-j} \leftarrow A_j$
**end for**
**AUSGABE:** $B$

- ▷ Was verändert der Operator?

# Rekombination

▷ Welche Eigenschaften hat ein „guter“ Rekombinationsoperator?

▷ Wie können Eigenschaften von zwei Eltern in einem Nachkommen sinnvoll kombiniert werden?

▷ Können nur Kanten aus den Eltern genutzt werden?



# Rekombination

**EINGABEN:** Permutationen $A = (A_1, \ldots, A_n)$ und $B = (B_1, \ldots, B_n)$
**for** Knoten $k \in \{1, \ldots, n\}$ **do**
  $i \leftarrow$ Index aus $\{1, \ldots, n\}$ mit $k = A_i$
  $j \leftarrow$ Index aus $\{1, \ldots, n\}$ mit $k = B_j$
  $Adj(k) \leftarrow \{A_{((i+n-2)\mathbf{mod}n)+1}, A_{(i\mathbf{mod}n)+1}, B_{((j+n-2)\mathbf{mod}n)+1}, B_{(j\mathbf{mod}n)+1}\}$
**end for**
$C_1 \leftarrow$ wähle zufällig aus $U(\{i_1, j_1\})$
**for** $i = 1$ to $n-1$ **do**
  $K \leftarrow \{m \in Adj(C_i) \mid \|Adj(m) \setminus \{C_1, \ldots, C_i\}\|$ minimal $\}$
  **if** $K \neq \emptyset$ **then**
    $C_{i+1} \leftarrow$ wähle gleichverteilt zufällig aus $K$
  **else**
    $C_{i+1} \leftarrow$ wähle gleichverteilt zufällig aus $\{1, \ldots, n\} \setminus \{C_1, \ldots, C_i\}$
  **end if**
**end for**
**AUSGABEN:** $C$

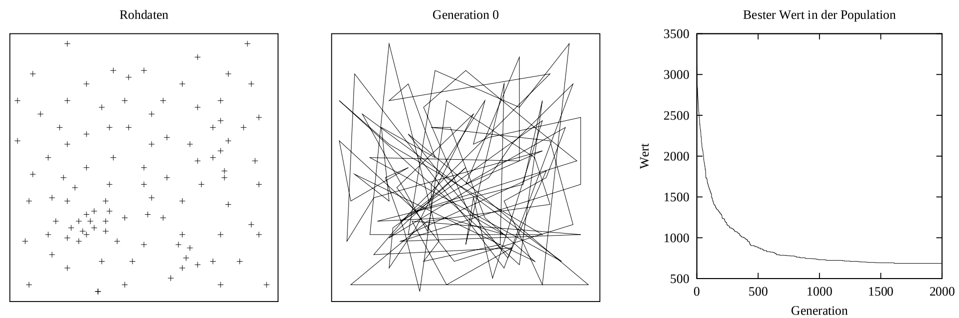


# Gesamtalgorithmus

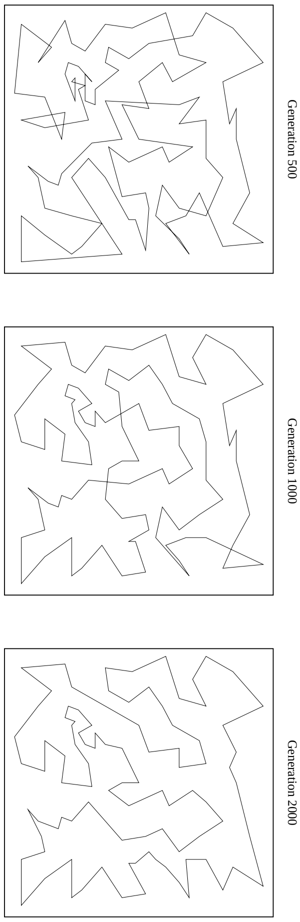
**EINGABEN:** Dimension $n$, Zielfunktion $F$
$t \leftarrow 0$
$P(t) \leftarrow$ Liste mit 10 gleichverteilt zufälligen Permutationen aus $U(\Pi_{1,n})$
bewerte alle $A \in P(t)$ mit Zielfunktion $F$
**while** $t \leq 2000$ **do**
  $P' \leftarrow \langle\rangle$
  **for** $i = 1$ to $40$ **do**
    wähle zufällig Eltern $A, B \in P(t)$
    **if** Zufallszahl $u$ ($\in U([0,1))) < 0.3$ **then**
      $A \leftarrow R_{kanten}(A, B)$
    **end if**
    $A \leftarrow M_{invers}(A)$
    $P' \leftarrow P' \circ \langle A \rangle$
  **end for**
  bewerte alle $A \in P(t)$ mit Zielfunktion $F$
  $t \leftarrow t + 1$
  $P(t) \leftarrow$ 10 beste Individuen aus $P' \circ P(t-1)$
**end while**
**Ausgabe:** bestes Individuum aus $P(t)$



# Ergebnisse

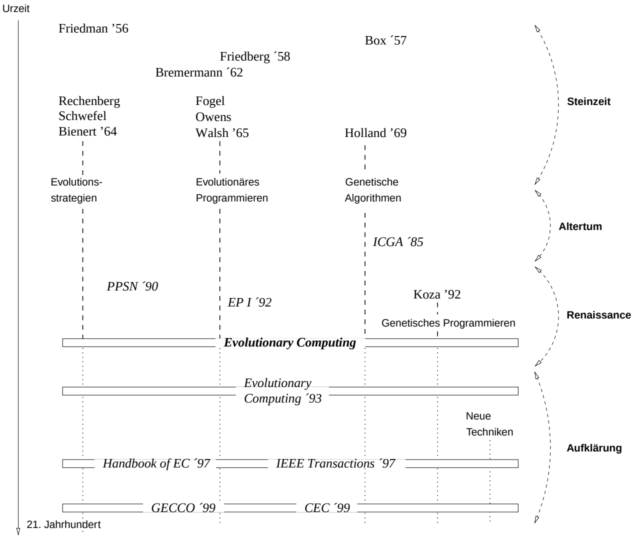


# Ergebnisse






# Zeittafel






# Grundsteine

- L. Fogel, 1965, Vorhersage von Zeitreihen
  ⇒ **evolutionäres Programmieren**
- Bienert, Rechenberg und Schwefel, 1964, experimentelle Optimierung eines Widerstandskörpers
  ⇒ **Evolutionsstrategien**
- Holland, 1969, Analyse adaptiver Systeme
  ⇒ **genetische Algorithmen**
- Koza, 1989, Erzeugung von Syntaxbäumen
  ⇒ **genetisches Programmieren**