

Evolutionäre Algorithmen: Erste Grundlagen

Karsten Weicker

HTWK Leipzig

5. Januar 2013

Überblick

- 1 Einführung
 - Probleme und Algorithmen
 - Ableitung aus der Biologie
- 2 Formale Beschreibung
- 3 Ein erster evolutionärer Algorithmus
 - Das Handlungsreisendenproblem
 - Operatoren
- 4 Kurzer geschichtlicher Abriss

Überblick

- 1 Einführung
 - Probleme und Algorithmen
 - Ableitung aus der Biologie
- 2 Formale Beschreibung
- 3 Ein erster evolutionärer Algorithmus
 - Das Handlungsreisendenproblem
 - Operatoren
- 4 Kurzer geschichtlicher Abriss

Was sind Optimierungsprobleme?

Beispiele

- kürzeste Route für einen Briefträger
- Auftragsreihenfolge für ein Fließband
- Platzierung von Antennen für Mobilfunk
- Anpassung einer Ampelsteuerung an das aktuelle Verkehrsaufkommen
- Stundenplanung
- Regression

Was sind Optimierungsprobleme?

Definition

Optimierungsproblem (Ω, f, \succ) ist gegeben durch

- Suchraum Ω
- Bewertungsfunktion $f : \Omega \rightarrow \mathbb{R}$
- Vergleichsrelation $\succ \in \{<, >\}$

Was sind Optimierungsprobleme?

Definition

Optimierungsproblem (Ω, f, \succ) ist gegeben durch

- Suchraum Ω
- Bewertungsfunktion $f : \Omega \rightarrow \mathbb{R}$
- Vergleichsrelation $\succ \in \{<, >\}$

Menge der globalen Optima:

$$\mathcal{X} = \{x \in \Omega \mid \forall x' \in \Omega : f(x) \succeq f(x')\}.$$

Was sind Optimierungsalgorithmen?

Was sind Optimierungsalgorithmen?

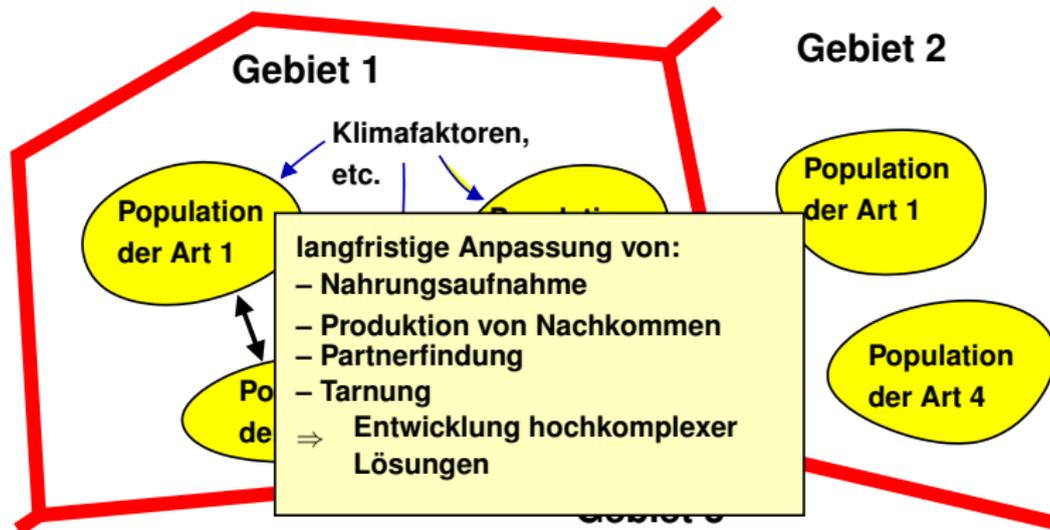
Beispiele

- Gradientenabstieg (z.B. neuronale Netze)
- Simplex/lineares Programmieren (OR)
- spezielle Algorithmen (Dijkstra, Prim, A*-Suche etc.)
- Backtracking/Branch&Bound

Evolution in der Biologie

- Theorie, dass sich Lebewesen allmählich aus einfachen zu immer komplizierteren Formen entwickelt haben
- reine Beobachtung und Hypothesenbildung: Lamarck (1809), Darwin (1859)
- wissenschaftliche Erklärungen auf der Ebene der Populationen: Populationsgenetik (1908)
- Erklärungen aus der Molekulargenetik: Watson und Crick (1953)

Modell in der Biologie



Ideen aus der Biologie

Konzepte und Begriffe

- Population und Individuum
- Genotyp
- Fitness

Evolutionsfaktoren (nach Hardy-Weinberger)

- Mutation
- Genfluss
- Selektion
- Gendrift

Moderne Biologie: genregulierende Netzwerke

- Rekombination

Fazit

Zunächst unberücksichtigt bleiben

- dynamische Populationsgrößen
- selbstorganisierte Interpretation der DNA
- Sexualität
- Nischenbildung
- Koevolution
- Genfluss
- dynamische Umgebungen

Fazit

Umgesetzte Konzepte

- individuelle Lösungsversuche werden betrachtet
- Konzept der Population
- randomisierte Operationen
 - Mutation
 - Selektion
- Lösungsversuche unterliegen einer Selektion
- Güte/Fitness steht in einem Zusammenhang zur Weitervererbung

Fazit

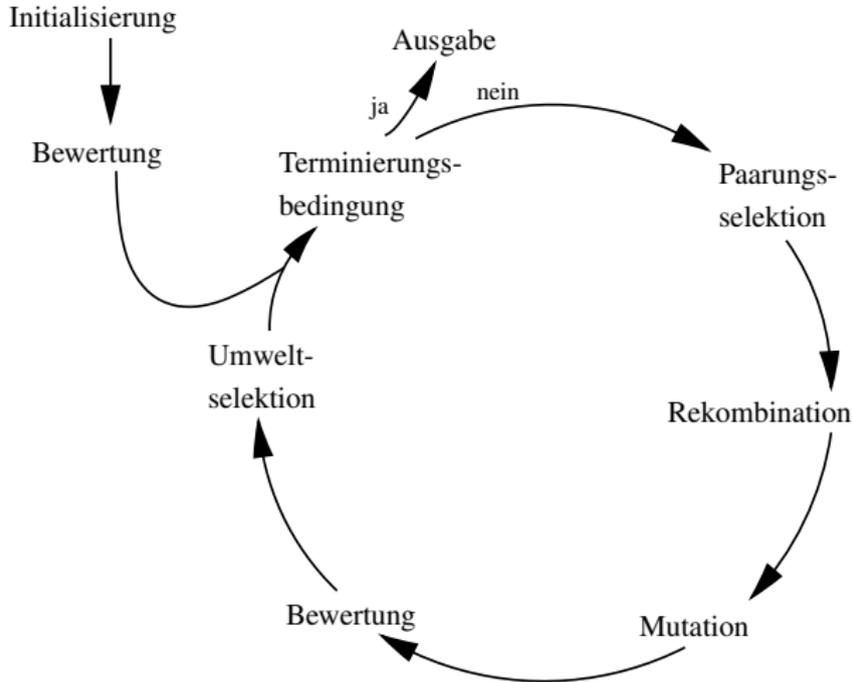
Umgesetzte Konzepte

- individuelle Lösungsversuche werden betrachtet
- Konzept der Population
- randomisierte Operationen
 - Mutation
 - Selektion
- Lösungsversuche unterliegen einer Selektion
- Güte/Fitness steht in einem Zusammenhang zur Weitervererbung

Aber:

teilweise andere Verwendung von Begriffen!

Der EA-Zyklus



Überblick

- 1 Einführung
 - Probleme und Algorithmen
 - Ableitung aus der Biologie
- 2 **Formale Beschreibung**
- 3 Ein erster evolutionärer Algorithmus
 - Das Handlungsreisendenproblem
 - Operatoren
- 4 Kurzer geschichtlicher Abriss

Suchraum

auf Problemseite

- Raum aller Lösungskandidaten Ω
- Bewertungsfunktion $f : \Omega \rightarrow \mathbb{R}$
- man spricht auch vom *Phänotyp*

Suchraum

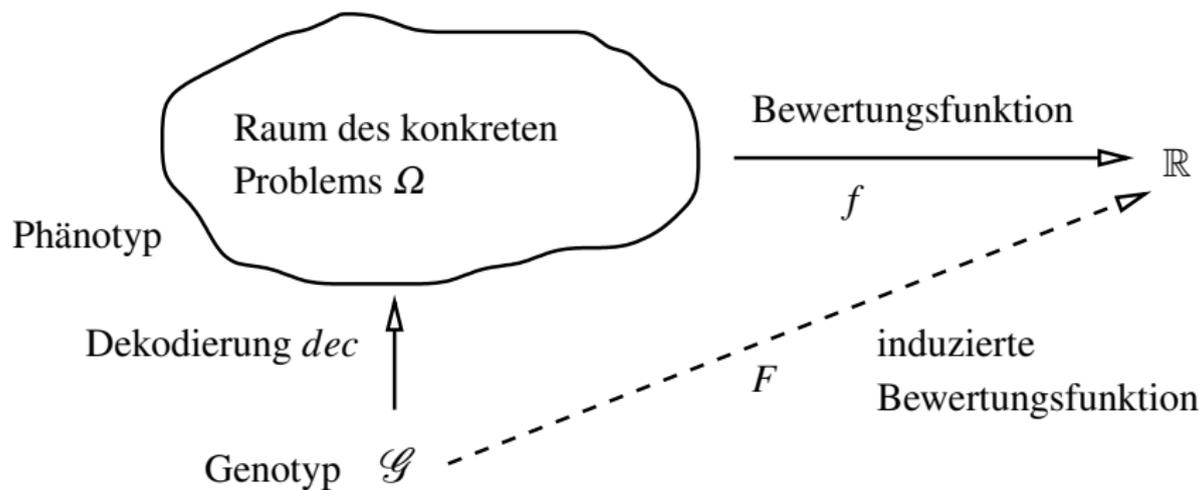
auf Problemseite

- Raum aller Lösungskandidaten Ω
- Bewertungsfunktion $f : \Omega \rightarrow \mathbb{R}$
- man spricht auch vom *Phänotyp*

auf Algorithmuseite

- Darstellung der Lösungskandidaten im Raum \mathcal{G}
- Dekodierungsfunktion $dec : \mathcal{G} \rightarrow \Omega$
- man spricht auch vom *Genotyp*
- induzierte Bewertungsfunktion $F : \mathcal{G} \rightarrow \mathbb{R}$

Suchraum

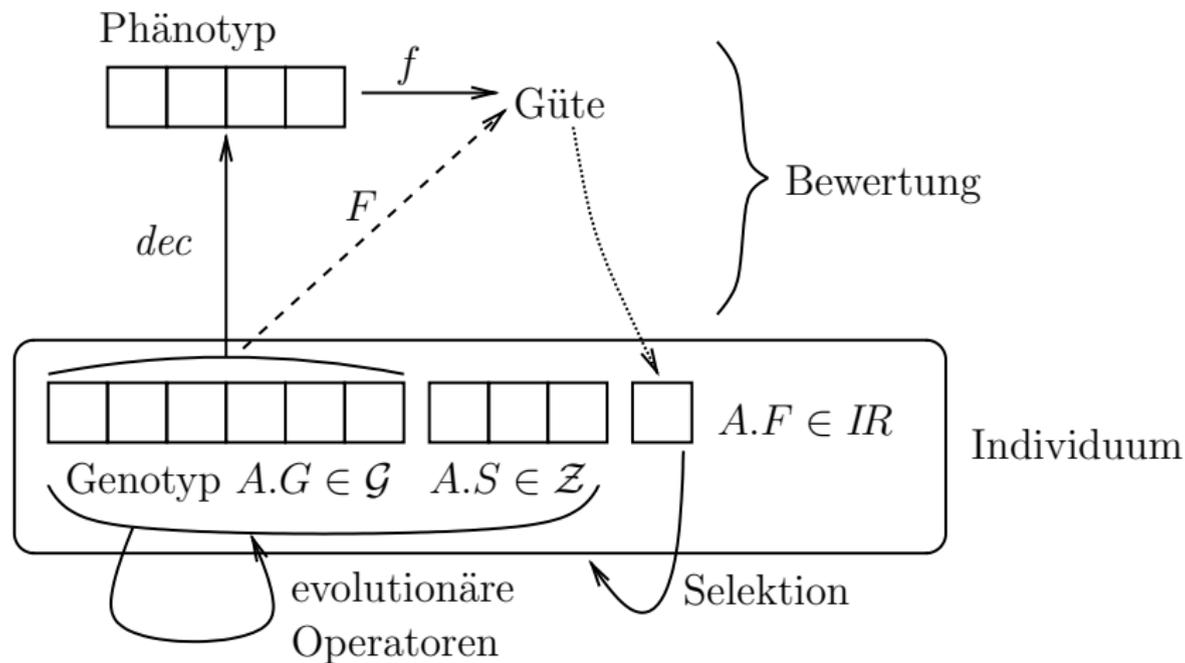


Teile eines Individuums

Individuum A enthält

- Genotyp $A.G \in \mathcal{G}$
- Güte $A.F \in \mathbb{R}$
wird gespeichert, sie berechnet ist (falls die Berechnung teuer ist)
- evtl. Zusatzinformationen $A.S \in \mathcal{Z}$,
die von Mutation und Rekombination genutzt werden können

Teile eines Individuums



Operatoren

Zufallszahlen

- Ξ (X_i) bezeichne die Menge aller Zustände des Zufallszahlengenerators
- die Veränderung des aktuellen Zustands $\xi \in \Xi$ wird in den Algorithmen nicht beschrieben

Mutation

$$Mut^\xi : \mathcal{G} \times \mathcal{Z} \rightarrow \mathcal{G} \times \mathcal{Z}$$

Rekombination

$$Rek^\xi : (\mathcal{G} \times \mathcal{Z})^r \rightarrow (\mathcal{G} \times \mathcal{Z})^s$$

mit $r \geq 2$ Eltern und $s \geq 1$ Kindern

Selektion

Grundidee

- Selektion wählt Individuen aus (unverändert!)
- Basis sind die Gütwerte

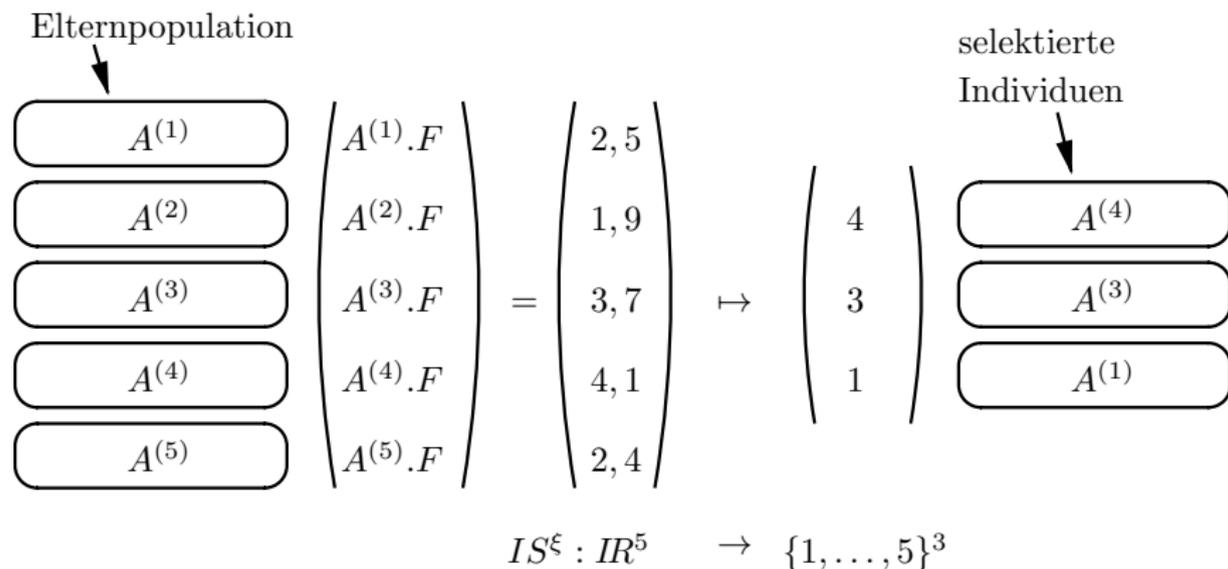
Formalisierung

$$\begin{aligned}
 \text{Sel}^\xi : (\mathcal{G} \times \mathcal{Z} \times \mathbb{R})^r &\rightarrow (\mathcal{G} \times \mathcal{Z} \times \mathbb{R})^s \\
 \langle A^{(i)} \rangle_{1 \leq i \leq r} &\mapsto \langle A^{(IS^\xi(c_1, \dots, c_r)_k)} \rangle_{1 \leq k \leq s}
 \end{aligned}$$

mit $A^{(i)} = (a_i, b_i, c_i)$ und Indexselektion

$$IS^\xi : \mathbb{R}^r \rightarrow \{1, \dots, r\}^s$$

Beispiel für die Selektion



Generischer Grundalgorithmus

Definition

Ein *einfacher evolutionärer Algorithmus* zu einem Optimierungsproblem (Ω, f, \succ) ist ein 8-Tupel $(\mathcal{G}, dec, Mut, Rek, IS_{Eltern}, IS_{Umwelt}, \mu, \lambda)$. Dabei bezeichnet μ die Anzahl der Individuen in der Elternpopulation und λ die Anzahl der erzeugten Kinder pro Generation. Ferner gilt

$$Rek : (\mathcal{G} \times \mathcal{Z})^k \rightarrow (\mathcal{G} \times \mathcal{Z})^{k'}$$

$$IS_{Eltern} : \mathbb{R}^\mu \rightarrow (1, \dots, \mu)^{\frac{k}{k'}\lambda} \quad \text{mit } \frac{k}{k'}\lambda \in \mathbb{N}$$

$$IS_{Umwelt} : \mathbb{R}^{\mu+\lambda} \rightarrow (1, \dots, \mu + \lambda)^\mu.$$

Generischer Grundalgorithmus

EA-SCHEMA(Optimierungsproblem (Ω, f, \succ))

- 1 $t \leftarrow 0$
- 2 $P(t) \leftarrow$ erzeuge Population der Größe μ
- 3 bewerte $P(t)$
- 4 **while** Terminierungsbedingung nicht erfüllt
- 5 **do** $\lceil P' \leftarrow$ selektiere Eltern für λ Nachkommen aus $P(t)$
- 6 $P'' \leftarrow$ erzeuge Nachkommen durch Rekombination aus P'
- 7 $P''' \leftarrow$ mutiere die Individuen in P''
- 8 bewerte P'''
- 9 $t \leftarrow t + 1$
- 10 $\lfloor P(t) \leftarrow$ selektiere μ Individuen aus $P''' \circ P(t - 1)$
- 11 **return** bestes Individuum aus $P(t)$

ZWEI MINUTEN

Welche Aussagen sind richtig?

- alle Individuen einer Population sind unterschiedlich
- die Selektion kann Individuen verdoppeln
- die Dekodierfunktion erlaubt die generelle Anwendbarkeit der Algorithmen
- durch die Selektion werden es immer weniger Individuen
- die Rekombination erzeugt immer ein Kindindividuum
- die Dekodierfunktion muss bijektiv sein

ZWEI MINUTEN

Welche Aussagen sind richtig?

- alle Individuen einer Population sind unterschiedlich
- die Selektion kann Individuen verdoppeln
- die Dekodierfunktion erlaubt die generelle Anwendbarkeit der Algorithmen
- durch die Selektion werden es immer weniger Individuen
- die Rekombination erzeugt immer ein Kindindividuum
- die Dekodierfunktion muss bijektiv sein

Überblick

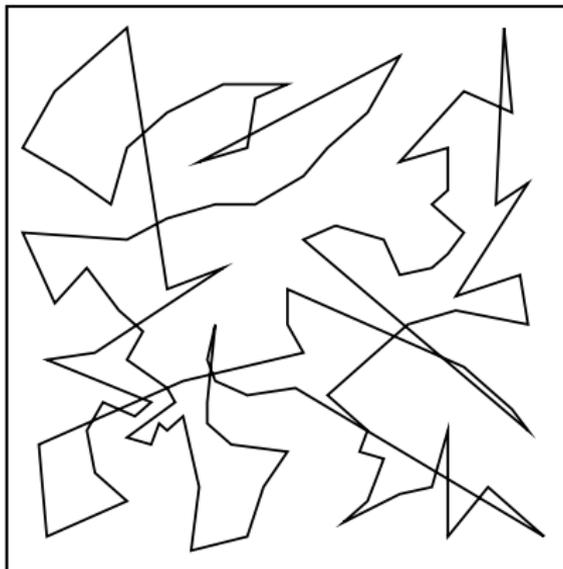
- 1 Einführung
 - Probleme und Algorithmen
 - Ableitung aus der Biologie
- 2 Formale Beschreibung
- 3 Ein erster evolutionärer Algorithmus
 - Das Handlungsreisendenproblem
 - Operatoren
- 4 Kurzer geschichtlicher Abriss

Handlungsreisendenproblem

Definition

- Geg.: Graph $G = (V, E, \gamma)$ mit „Straßen“
 $E \subseteq V \times V$ und „Fahrzeit“ $\gamma : E \rightarrow \mathbb{R}$
- *Handlungsreisendenproblem* ist $(\mathcal{S}_n, f_{\text{TSP}}, <)$ mit
Raum aller Permutationen \mathcal{S}_n
- Bewertungsfunktion
$$f_{\text{TSP}}((i_1, \dots, i_n)) = \gamma((v_{i_n}, v_{i_1})) + \sum_{j=2}^n \gamma((v_{i_{j-1}}, v_{i_j}))$$
- heißt *symmetrisch*, wenn für alle $(v_i, v_j) \in E$ sowohl
 $(v_j, v_i) \in E$ als auch $\gamma((v_i, v_j)) = \gamma((v_j, v_i))$ erfüllt
sind.

Ergebnis der MSB-Approximation



Mutation: Vertauschen

VERTAUSCHENDE-MUTATION(Permutation $A = (A_1, \dots, A_n)$)

1 $B \leftarrow A$

2 $u_1 \leftarrow$ wähle Zufallszahl gemäß $U(\{1, \dots, n\})$

3 $u_2 \leftarrow$ wähle Zufallszahl gemäß $U(\{1, \dots, n\})$

4 $B_{u_1} \leftarrow A_{u_2}$

5 $B_{u_2} \leftarrow A_{u_1}$

6 **return** B

Mutation: Invertieren eines Teilabschnitts

INVERTIERENDE-MUTATION(Permutation $A = (A_1, \dots, A_n)$)

```
1  $B \leftarrow A$ 
2  $u_1 \leftarrow$  wähle Zufallszahl gemäß  $U(\{1, \dots, n\})$ 
3  $u_2 \leftarrow$  wähle Zufallszahl gemäß  $U(\{1, \dots, n\})$ 
4 if  $u_1 > u_2$ 
5   then  $\square$  vertausche  $u_1$  und  $u_2$ 
6   for  $j \in \{u_1, \dots, u_2\}$ 
7     do  $\square B_{u_2+u_1-j} \leftarrow A_j$ 
8   return  $B$ 
```

Ordnungsrekombination

```
ORDNUNGSREKOMBINATION(Permutationen  $A, B$ )
1   $j \leftarrow$  wähle zufällig gemäß  $U(\{1, \dots, n - 1\})$ 
2  for  $i \in \{1, \dots, j\}$ 
3  do  $C_i \leftarrow A_i$ 
4  for  $i \in \{1, \dots, n\}$ 
5  do  $\lceil$  if  $B_i \notin \{C_1, \dots, C_j\}$ 
6      then  $\lceil j \leftarrow j + 1$ 
7           $\lfloor$   $C_j \leftarrow B_i$ 
8  return  $C$ 
```

Kantenrekombination

KANTENREKOMBINATION(Permutationen A, B)

```

1  for Knoten  $k \in \{1, \dots, n\}$ 
2  do  $\lceil Adj(k) \leftarrow \emptyset$ 
3  for  $i = 1, \dots, n$ 
4  do  $\lceil Adj(A_i) \leftarrow Adj(A_i) \cup \{A_{(i \bmod n)+1}\}$ 
5       $Adj(A_{(i \bmod n)+1}) \leftarrow Adj(A_{(i \bmod n)+1}) \cup \{A_i\}$ 
6       $Adj(B_i) \leftarrow Adj(B_i) \cup \{B_{(i \bmod n)+1}\}$ 
7       $\lceil Adj(B_{(i \bmod n)+1}) \leftarrow Adj(B_{(i \bmod n)+1}) \cup \{B_i\}$ 
8   $C_1 \leftarrow$  wähle zufällig gemäß  $U(\{A_1, B_1\})$ 
9  for  $i \in \{1, \dots, n-1\}$ 
10 do  $\lceil K \leftarrow \{m \in Adj(C_i) \mid \#(Adj(m) \setminus \{C_1, \dots, C_i\}) \text{ minimal}\}$ 
11     if  $K \neq \emptyset$ 
12     then  $\lceil C_{i+1} \leftarrow$  wähle gleichverteilt zufällig aus  $K$ 
13      $\lceil$  else  $\lceil C_{i+1} \leftarrow$  wähle gleichverteilt zufällig aus  $\{1, \dots, n\} \setminus \{i\}$ 
14 return  $C$ 

```

Gesamtalgorithmus

EA-HANDLUNGSREISENDENPROBLEM(Zielfunktion F , Anzahl der Städte n)

```

1   $t \leftarrow 0$ 
2   $P(t) \leftarrow$  Liste mit 10 gleichverteilt zufälligen Permutationen aus  $U(\mathcal{S}_n)$ 
3  bewerte alle  $A \in P(t)$  mit Zielfunktion  $F$ 
4  while  $t \leq 2000$ 
5  do  $\lceil P' \leftarrow \langle \rangle$ 
6      for  $i \in \{1, \dots, 40\}$ 
7      do  $\lceil A, B \leftarrow$  wähle gleichverteilt zufällig Eltern aus  $P(t)$ 
8          if  $u < 0.3$  für eine Zufallszahl  $u \in U([0, 1])$ 
9          then  $\lfloor A \leftarrow$  KANTENREKOMBINATION( $A, B$ )
10              $A \leftarrow$  INVERTIERENDE-MUTATION( $A$ )
11              $\lfloor P' \leftarrow P' \circ \langle A \rangle$ 
12             bewerte alle  $A \in P'$  mit Zielfunktion  $F$ 
13              $t \leftarrow t + 1$ 
14              $\lfloor P(t) \leftarrow$  10 beste Individuen aus  $P' \circ P(t - 1)$ 
15  return bestes Individuum aus  $P(t)$ 

```

Überblick

- 1 Einführung
 - Probleme und Algorithmen
 - Ableitung aus der Biologie
- 2 Formale Beschreibung
- 3 Ein erster evolutionärer Algorithmus
 - Das Handlungsreisendenproblem
 - Operatoren
- 4 Kurzer geschichtlicher Abriss

Kurzer geschichtlicher Abriss

Friedmann 1956

- Nachempfinden der natürlichen Selektion
- „Selective Feedback Computer“
- Schaltkreise zur Steuerung, die eine konstante Temperatur suchen

Box 1957

- „Evolutionary Operation“
- Produktivitätssteigerung in Fertigungsprozessen durch Parameteroptimierung

Kurzer geschichtlicher Abriss

Friedberg 1958

- „Learning Machine“
- tabellarische Programme, die Eingabe/Ausgabe-Daten zur Deckung bringen sollen

Bremermann 1962

- Minimierung einer einfachen mathematischen Funktion
- 1-Bit-Mutation

Kurzer geschichtlicher Abriss

Rechenberg/Schwefel 1964

- begründen *Evolutionstrategie*
- experimentelle Optimierung eines Widerstandskörpers

L. Fogel/Owens/Walsh 1965

- begründen *evolutionäres Programmieren*
- Zeitreihenprognose durch endliche Automaten
- Anfang 1990er: erneuert durch David B. Fogel

Kurzer geschichtlicher Abriss

Holland 1969

- begründet *genetische Algorithmen*
- mathematische Analyse adaptive Systeme
- Optimierung durch De Jong und Goldberg (1989)

Koza 1992

- begründet *genetisches Programmieren*
- Programme als LISP-Bäume

seit Tagung PPSN'90

- Verständnis als gemeinsames Gebiet