

Künstliche Intelligenz

Vorlesung 6: Evolutionäre Standardalgorithmen



BEISPIEL: PROBLEM DES HANDLUNGSREISENDEN

SIMULIERTES AUSGLÜHEN UND ZUFALLSAUFSTIEG

- 1 Bringe Städte in zufällige Reihenfolge (zufällige Rundreise)
- 2 Wähle zufällig zweimal 2 Städte, die in aktueller Rundreise aufeinander folgen (alle vier Städte verschieden), trenne Rundreise zwischen Städten jedes Paares und drehe dazwischenliegenden Teil um.
- 3 Wenn neue Rundreise besser (kürzer, billiger) als alte, ersetze alte Rundreise durch neue, sonst mit Wahrscheinlichkeit $p = e^{-\frac{\Delta Q}{kT}}$.

ΔQ Qualitätsunterschied zwischen alter und neuer Rundreise

k Spannweite der Rundreisequalitäten (ggf. schätzen, z.B. $k_t = \frac{t+1}{t} \max_{i=1}^t \Delta Q_i$, wobei ΔQ_i Qualitätsunterschied im i -ten Schritt und t der aktuelle Schritt)

T Temperaturparameter (mit Zeit fallend, z.B. $T = \frac{1}{t}$)

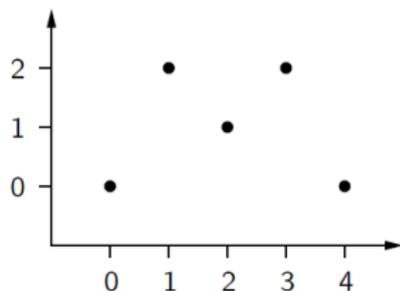
- 4 Wiederhole die Schritte 2 und 3 bis Abbruchkriterium erfüllt.



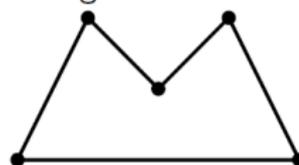
BEISPIEL: PROBLEM DES HANDLUNGSREISENDEN

SIMULIERTES AUSGLÜHEN UND ZUFALLSAUFSTIEG

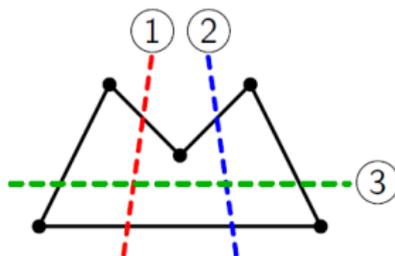
reiner Zufallsaufstieg kann in lokalem Minimum hängenbleiben:



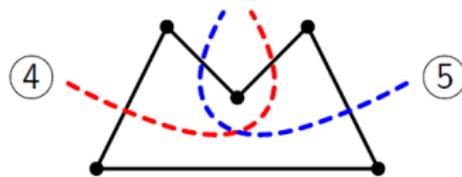
anfängliche Rundreise



Länge: $2\sqrt{2} + 2\sqrt{5} + 4 \approx 11.30$

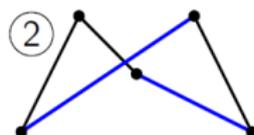
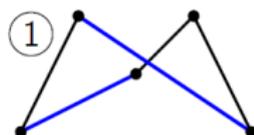


mögliche Teilungen der Rundreise

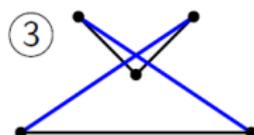


BEISPIEL: PROBLEM DES HANDLUNGSREISENDEN

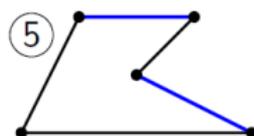
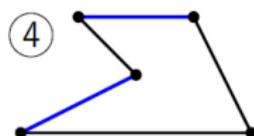
SIMULIERTES AUSGLÜHEN UND ZUFALLSAUFSTIEG



$$\sqrt{2} + 3\sqrt{5} + \sqrt{13} \quad \approx 11.73$$

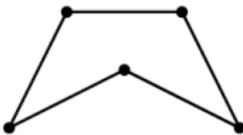


$$\sqrt{2} + 2\sqrt{13} + 4 \quad \approx 14.04$$



$$\sqrt{2} + 2\sqrt{5} + 2 + 4 \quad \approx 11.89$$

global. Optimum:



$$4\sqrt{5} + 2 \quad \approx 10.94$$

BEISPIEL: PROBLEM DES HANDLUNGSREISENDEN

SIMULIERTES AUSGLÜHEN UND ZUFALLSAUFSTIEG

- alle Modifikationen der Anfangsrundreise führen zu schlechteren Rundreisen \Rightarrow globales Optimum (von dieser Rundreise) mit reinen Zufallsaufstieg nicht erreichbar
- simulierten Ausglühen akzeptiert dagegen mitunter auch schlechtere Lösungen \Rightarrow Weg zum globalen Optimum (*aber*: keine Garantie, dass dies geschieht!)
- **beachte**: es kann von erlaubten Operationen abhängen, ob Suche in lokalem Optimum hängenbleiben kann:
 - mit weiterer Operation (Ändern der Position einer Stadt in Rundreise bzw. Entfernen von aktueller Position und Einfügen an einer anderen) kein Hängenbleiben im betrachteten Beispiel
 - auch für diese Operationenmenge: Konstruktion eines Beispiels, das in lokalem Minimum hängenbleibt



TABU SUCHE

- lokales Suchverfahren, dass bei Erzeugung neuer Individuen Geschichte berücksichtigt
- Tabu-Listen verhindern Rückkehr zu gerade betrachteten Lösungskandidaten
- Tabu-Liste = FIFO-Warteschlange fester Länge
 - Einträge sind ganze Lösungskandidaten oder Aspekte
 - Mutationen dürfen Tabu-Einträge nicht erzeugen
 - meist: FIFO-Liste mit erstrebenswerten Eigenschaften können Tabu brechen
 - auch möglich: neue beste Gesamtgüte bricht Tabu



Algorithm 7 Tabu-Suche

Input: Zielfunktion F

Output: bestes Individuum A_{best}

```
1:  $t \leftarrow 0$ 
2:  $A(t) \leftarrow$  erzeuge zufälligen Lösungskandidaten
3: bewerte  $A(t)$  durch  $F$ 
4:  $A_{\text{best}} \leftarrow A(t)$ 
5: initialisiere Tabu-Liste
6: while Terminierungsbedingung nicht erfüllt {
7:    $P \leftarrow \emptyset$ 
8:   while  $|P| < \lambda$  {
9:      $B \leftarrow$  mutiere  $A(t)$ 
10:    bewerte  $B$  durch  $F$ 
11:    if  $(A(t), B) \notin$  Tabu-Liste oder  $B.F \succ A_{\text{best}}.F$  {
12:       $P \leftarrow P \cup \{B\}$ 
13:    }
14:  }
15:   $t \leftarrow t + 1$ 
16:   $A(t) \leftarrow$  bestes Individuum aus  $P$ 
17:  if  $A(t).F \succ A_{\text{best}}.F$  {
18:     $A_{\text{best}} \leftarrow A(t)$ 
19:  }
20:  Tabu-Liste  $\leftarrow$  aktualisiere durch  $(A(t-1), A(t))$ 
21: }
22: return  $A_{\text{best}}$ 
```

MEMETISCHE ALGORITHMEN

- einerseits populationsbasierte Algorithmen
 - Vorteil: breites Durchforsten des Suchraums
 - Nachteil: langsam
- andererseits lokale Suche
 - Vorteil: schnelle Optimierung
 - Nachteil: anfällig für lokale Optima
- Memetische Algorithmen: Kombination beider Techniken
- Herkunft des Namens (Richard Dawkins): **Meme** sind Verhaltenselemente, die individuell erworben werden können (im Gegensatz zu Genen)
- Vorgehen: jedes neue Individuum wird sofort lokal optimiert
 - nur wenige Schritte oder
 - bis in lokales Optimum



MEMETISCHE ALGORITHMEN

Algorithm 8 Memetischer Algorithmus

Input: Bewertungsfunktion F

Output: bestes Individuum

- 1: $t \leftarrow 0$
- 2: $P(t) \leftarrow$ initialisiere Population der Größe μ
- 3: $P(t) \leftarrow$ LOKALE-SUCHE(F) für jedes Individuum in $P(t)$
- 4: bewerte $P(t)$ durch F
- 5: **while** Terminierungsbedingung nicht erfüllt {
- 6: $E \leftarrow$ selektiere Eltern für λ Nachkommen aus $P(t)$
- 7: $P' \leftarrow$ erzeuge Nachkommen durch Rekombination aus E
- 8: $P'' \leftarrow$ mutiere Individuen in P'
- 9: $P''' \leftarrow$ LOKALE-SUCHE(F) für jedes Individuum in P''
- 10: bewerte P''' durch F
- 11: $t \leftarrow t + 1$
- 12: $P(t) \leftarrow$ Umweltsektion auf P'''
- 13: }
- 14: **return** bestes Individuum aus $P(t)$



EIGENSCHAFTEN

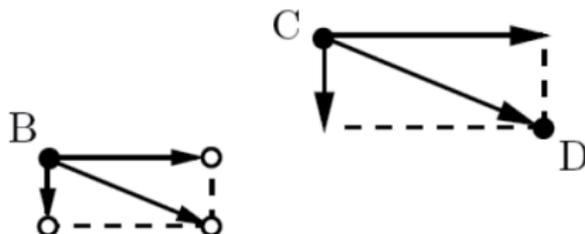
- oft stark beschleunigte Optimierung
- aber: Suchdynamik kann entscheidend eingeschränkt werden
 - Mutation bleibt oft in breiten lokalen Optima
 - Rekombination hat beschränkte Ausgangssituation
 - Teile des Suchraums sind evtl. unerreichbar



DIFFERENZIALEVOLUTION

Idee

- keine Anpassung der Schrittweite in A.S
- sondern: Relation der Individuen in der Population wird als Grundlage für Schrittweite herangezogen



DIFFERENZIALEVOLUTION

Algorithm 9 DE-Operator

Input: Individuen A, B, C, D

Output: optimiertes Individuum A'

- 1: $\text{index} \leftarrow$ wähle Zufallszahl gemäß $U(\{1, \dots, l\})$
 - 2: **for each** $i \in \{1, \dots, l\}$ {
 - 3: $u \leftarrow$ wähle Zufallszahl gemäß $U([0, 1))$
 - 4: **if** $u \leq \tau$ **or** $i = \text{index}$ {
 - 5: $A'.G_i \leftarrow B'.G_i + (C.G_i - D.G_i) \cdot \alpha$
 - 6: } **else** {
 - 7: $A'.G_i \leftarrow A.G_i$
 - 8: }
 - 9: }
 - 10: **return** A'
-



DIFFERENZIALEVOLUTION

Algorithm 10 Differentialevolution

Input: Bewertungsfunktion F

Output: bestes Individuum aus $P(t)$

```
1:  $t \leftarrow 0$ 
2:  $P(t) \leftarrow$  erzeuge Population der Größe  $\mu$ 
3: bewerte  $P(t)$  durch  $F$ 
4: while Terminierungsbedingung nicht erfüllt {
5:    $P(t+1) \leftarrow \emptyset$ 
6:   for  $i \leftarrow 1, \dots, \mu$  {
7:     do {
8:        $A, B, C, D \leftarrow$  selektiere Eltern uniform zufällig aus  $P(t)$ 
9:     } while  $A, B, C, D$  paarweise verschieden
10:     $A' \leftarrow$  DE-OPERATOR( $A, B, C, D$ )
11:    bewerte  $A'$  durch  $F$ 
12:    if  $F(A') > F(A)$  {
13:       $P(t+1) \leftarrow P(t+1) \cup \{A'\}$ 
14:    } else {
15:       $P(t+1) \leftarrow P(t+1) \cup \{A\}$ 
16:    }
17:  }
18: }
19:  $t \leftarrow t + 1$ 
20: return bestes Individuum aus  $P(t)$ 
```



SCATTER SEARCH

■ Zutaten

- Population mit Lösungskandidaten
- Variationsoperatoren
- Selektionsdruck
- zusätzlich: lokale Suche

■ Aber...

- ein deterministisches Verfahren!
- weiträumige Erforschung des Suchraums:
 - breite Initialisierung
 - systematische Erzeugung neuer Individuen



ABLAUF

- iterierter Ablauf durch zwei Phasen
 - ① Erzeugen neuer Individuen und Auswahl derjenigen, die die größte Vielfalt garantieren
 - ② Rekombination aller Paarungen der gewählten Individuen
 - ③ Selektion der Besten und Iteration bis sich nichts mehr ändert
- Beispiel: reellwertige Problemräume mit $\mathcal{G} = \Omega = \mathbb{R}^n$.



PHASE 1

- 1 Diversitätsgenerator erzeugt μ Individuen in P , zum Beispiel:
 - pro Suchraumdimension Wertebereich in vier Partitionen
 - Häufigkeit der erzeugten Individuen pro Partition merken
 - Partition invers proportional auswählen
- 2 separat: Population der besten α Individuen wird um die β Individuen aus P erweitert, die $\min_{B \in P_{best}} d(A.G, B.G)$ maximieren ($A \in P$).



PHASE 2

- 1 Teilmengengenerator wählt Individuen aus Menge der Besten aus. Beispiel: (hier sehr einfach) alle mögliche Paare
- 2 Kombinationsoperator anwenden. Beispiel: arithmetischer Crossover mit $U([-1/2, 3/2])$.
- 3 lokal optimieren
- 4 wenn alle erzeugt, dann Wahl der $\alpha + \beta$ Besten
- 5 iterieren solange sich Menge der Besten ändert
- 6 α Besten für nächste Phase 1.



SCATTER SEARCH

Algorithm 11 Scatter Search

Input: Bewertungsfunktion F

```
1:  $P_{\text{best}} = \emptyset$ ;  $P = \emptyset$ 
2: for  $t \leftarrow 1, \dots, \text{maxIter}$  {
3:   while  $|P| < \mu$  {
4:      $A \leftarrow$  erzeuge ein Individuum mit Diversitätsgenerator
5:      $A \leftarrow$  LOKALE-SUCHE( $F$ ) angewandt auf  $A$ ; bewerte  $A$  durch  $F$ 
6:     if  $A \notin P \cup P_{\text{best}}$  {
7:        $P \leftarrow P \cup \{A\}$ 
8:     }
9:   }
10:  if  $t = 1$  {
11:     $P_{\text{best}} \leftarrow$  selektiere  $\alpha$  Individuen aus  $P$  mit BESTEN-SELEKTION
12:     $P \leftarrow$  streiche Individuen aus  $P_{\text{best}}$  in  $P$ 
13:  }
14:  for  $k \leftarrow 1, \dots, \beta$  {
15:     $A \leftarrow$  dasjenige Individuum aus  $P$ , das  $\min_{B \in P_{\text{best}}} d(A, G, B, G)$  maximiert
16:     $P \leftarrow$  streiche Individuum  $A$  in  $P$ 
17:     $P_{\text{best}} \leftarrow P_{\text{best}} \cup \{A\}$ 
18:  }
19:  do {
20:     $P' \leftarrow \emptyset$ ; Mengen  $\leftarrow$  erzeuge Teilmengen von  $P_{\text{best}}$  durch Teilmengenoperator
21:    for each  $M \in$  Mengen {
22:       $A \leftarrow$  wende Kombinationsoperator auf  $M$  an
23:       $A \leftarrow$  LOKALE-SUCHE( $F$ ) angewandt auf  $A$ ; bewerte  $A$  durch  $F$ 
24:      if  $A \notin P_{\text{best}} \cup P'$  {
25:         $P' \leftarrow P' \cup \{A\}$ 
26:      }
27:    }
28:     $P_{\text{best}} \leftarrow$  selektiere  $\alpha + \beta$  Ind. aus  $P_{\text{best}} \cup P'$  mit BESTEN-SELEKTION
29:  } while  $P_{\text{best}}$  hat sich nicht geändert
30:   $P_{\text{best}} \leftarrow$  selektiere  $\alpha$  Individuen aus  $P$  mit BESTEN-SELEKTION
31: }
32: return bestes Individuum in  $P_{\text{best}}$ 
```



DIE GÜTELANDSCHAFT

- Im Wechselspiel zwischen Selektion und Mutation bestimmt die Mutation die möglichen Veränderungen, die von einer zur nächsten Generation auftreten können, während die Selektion bestimmte Schritte ausschließt oder akzeptiert.
- Gerade der erste Aspekt kann über die Notation des Nachbarschaftsgraphen gut verdeutlicht werden, der alle möglichen Mutationen als Kanten aufzeigt.



NACHBARSCHAFTSGRAPH

Definition

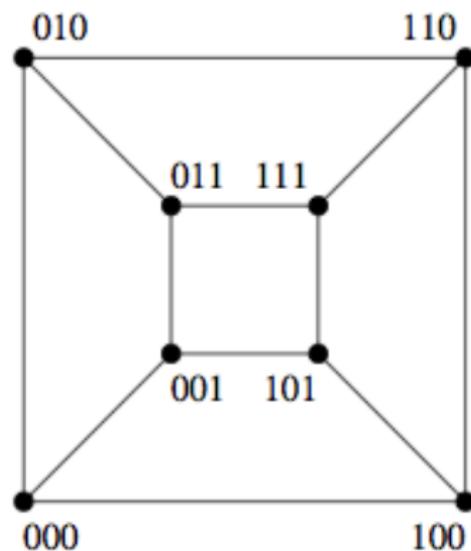
Sei $Mut^\xi: G \times Z \rightarrow G \times Z$ ein Mutationsoperator und $Z = \{\perp\}$, dann ist der *Nachbarschaftsgraph* zu Mut definiert als gerichteter Graph $G = (V, E)$ mit Knotenmenge $V = G$ und Kantenmenge

$$E = \{(A.G, B.G) \in V \times V \mid \exists \xi \in \Xi. Mut^\xi(A) = B\}.$$



BEISPIEL

Da die Mutation in unserem Beispiel symmetrisch ist, existiert für jede gerichtete Kante im Nachbarschaftsgraphen auch eine Rückkante. Daher wird in diesem und allen weiteren Bildern dieses Abschnitts der Graph ungerichtet dargestellt.

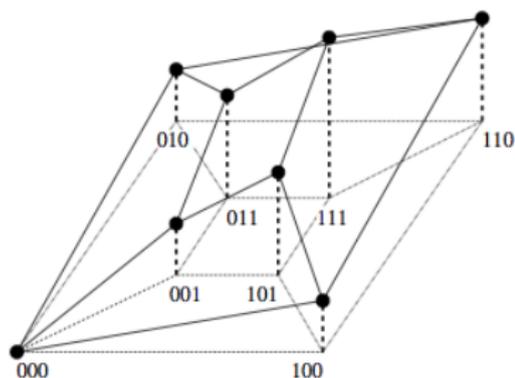


BEISPIEL

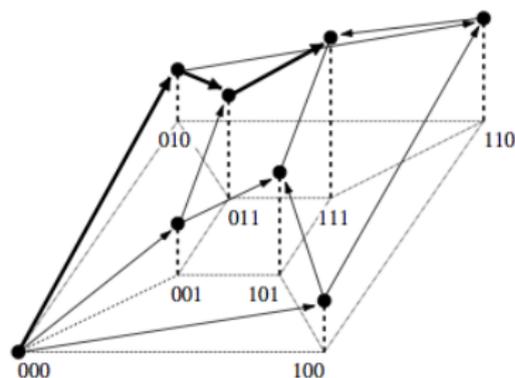
- Jede Kante entspricht einer Veränderung an einem Individuum durch den Mutationsoperator.
- Damit repräsentiert ein zufälliger Pfad im Graph den Ablauf, der durch mehrfaches, iteratives Anwenden der zufälligen Mutation entsteht.
- Im Englischen spricht man auch von einem sog. **random walk**.
- Durch die Selektion nach jeder Mutation wird der Suchprozess zielgerichtet, da keine Verschlechterung mehr möglich ist.
- Dies kann man visualisieren, indem man über der Struktur des Nachbarschaftsgraphen eine Gütelandschaft errichtet - dann sind keine abwärts führenden Mutation mehr erlaubt.



GÜTELANDSCHAFT BEISPIEL



(a) GüteLandschaft



(b) Selektion in der GüteLandschaft

Abbildung 2: Für das Einsenzählproblem mit drei Bits und die EIN-BIT-BINÄRE-MUTATION wird (a) die über dem Nachbarschaftsgraphen liegende GüteLandschaft gezeigt und (b) die Wirkung der Selektion durch die Pfeile visualisiert: nur in Pfeilrichtung kann sich der Hillclimber bewegen.

GÜTELANDSCHAFT

Definition

Eine *Gütelandschaft* (G, F) wird durch einen Nachbarschaftsgraphen $\mathcal{G} = (G, E)$ und eine induzierte Bewertungsfunktion $F: G \rightarrow \mathbb{R}$ definiert, die jedem Knoten seine Höhe in der Landschaft zuordnet. Ferner sei $w = w_1 w_2 \dots w_k \in G^+$ ein *Weg* in der Landschaft, falls für alle $i \in \{1, \dots, k-1\}$ die Kante $(w_i, w_{i+1}) \in E$ existiert.

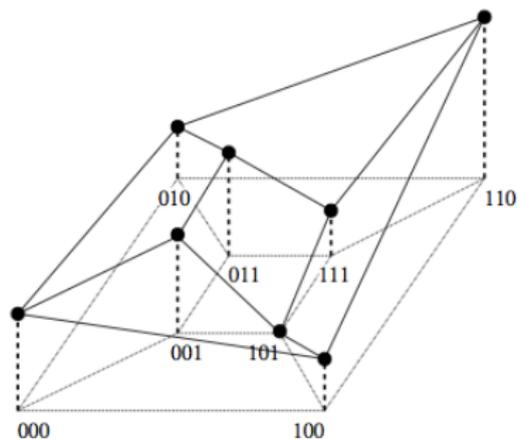


DAS PROBLEM LOKALER OPTIMA

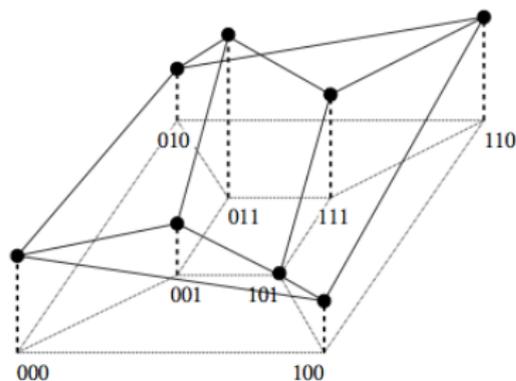
- Die bisherigen Betrachtungen sind in vielerlei Hinsicht nur ein Beispiel für den Idealfall, wie das folgende Beispiel illustriert.
- Weist man die Gütwerte den Lösungskandidaten aus Abbildung 2 in einer anderen Reihenfolge zu, erhält man beispielsweise die Gütelandschaften in Abbildung 3.
- Die zugehörigen Optimierungsprobleme fallen nicht mehr in die Klasse des Musterabgleichs.
- Der binäre Hillclimber kann nicht mehr von jedem Punkt aus das Maximum des Problems erreichen.



GÜTELANDSCHAFT BEISPIEL



(a) Plateau als lokales Optimum



(b) Unterschied zwischen Plateau und lokalem Optimum

Abbildung 3: Zwei zufällig erzeugte Gütelandschaften über dem Nachbarschaftsgraphen mit drei Bits, in denen es keinen Weg für BINÄRES-HILLCLIMBING vom Individuum 000 zum Maximum gibt. (a) zeigt ein Plateau als lokales Optimum, (b) enthält ein Plateau, das kein lokales Optimum darstellt.

LOKALES OPTIMUM, PLATEAU

Definition

Sei $Mut^\xi: G \times Z \rightarrow G \times Z$ ein Mutationsoperator, $G = (G, E)$ der zugehörige Nachbarschaftsgraph und (G, F) eine Gütelandschaft. Dann heißt ein Lösungskandidat A mit $A.G \in G$ ein

- **lokales Optimum**, falls alle möglichen Mutanten nicht besser sind $\forall \xi \in \Xi. F(A.G) \geq F(B.G)$ mit $B = Mut^\xi(A)$ und für alle Wege $w_1 (= A.G)w_2 \dots w_k$ mit $F(w_k) > F(A.G)$ gilt, dass mindestens einer der Lösungskandidaten $w_i (2 \leq i < k)$ eine schlechtere Güte hat: $F(A.G) > F(w_i)$.
- **Plateau-Punkt**, falls alle möglichen Mutanten nicht besser sind $\forall \xi \in \Xi. F(A.G) \geq F(B.G)$ mit $B = Mut^\xi(A)$ und wenigstens ein benachbarter Lösungskandidat die gleiche Güte hat $\exists \xi \in \Xi. F(A.G) = F(B.G)$ mit $B = Mut^\xi(A)$.

KODIERUNG

WÜNSCHENSWERTE EIGENSCHAFTEN EINER KODIERUNG

- Kodierung ist problemspezifisch zu wählen
- kein allg. Kochrezept, um (gute) Kodierung zu finden aber: einige Prinzipien, die beachtet werden sollten

Wünschenswerte Eigenschaften einer Kodierung:

- Darstellung ähnliche Phänotypen durch ähnliche Genotypen
- ähnliche Fitness von ähnlich kodierten Lösungskandidaten
- Abgeschlossenheit von Ω unter verwendeten evolutionären Operatoren



KODIERUNG: ERSTE WÜNSCHENSWERTE EIGENSCHAFT

Darstellung ähnliche Phänotypen durch ähnliche Genotypen

- Mutationen einzelner Gene führen zu ähnlichen Genotypen (einzelne Alleländerungen \Rightarrow kleine Änderung des Chromosoms)
- wenn Eigenschaft nicht erfüllt ist, können naheliegende Verbesserungen u.U. nicht erzeugt werden
- Konsequenz: große Änderung des Genotyps, um zu ähnlichem (und u.U. besseren) Phänotyp zu kommen

Beispiel zur Verdeutlichung:

- Optimierung einer reellen Funktion $y = f(x_1, \dots, x_n)$
- Darstellung der (reellen) Argumente durch Binärcodes
- Problem: Kodierung als Binärzahl führt zu **Hamming-Klippen**



BINÄRKODIERUNG REELLER ZAHLEN

gegeben: reelles Intervall $[a, b]$ und Kodierungsgenauigkeit ε
gesucht: Kodierungsvorschrift für $x \in [a, b]$ als Binärzahl z ,
sodass z um weniger als ε von x abweicht.

Idee: teile $[a, b]$ in gleich große Abschnitte der Länge $\leq \varepsilon$
 \Rightarrow 2^k Abschnitte mit $k = \left\lceil \log_2 \frac{b-a}{\varepsilon} \right\rceil$
kodiert durch $0, \dots, 2^k - 1$



BINÄRKODIERUNG REELLER ZAHLEN

Abschnitte: $k = \left\lceil \log_2 \frac{b-a}{\varepsilon} \right\rceil$ oder $k = \left\lceil \log_2 \frac{b-a}{2\varepsilon} \right\rceil$

Kodierung: $z = \left\lfloor \frac{x-a}{b-a}(2^k - 1) \right\rfloor$ oder $z = \left\lfloor \frac{x-a}{b-a}(2^k - 1) + \frac{1}{2} \right\rfloor$

Dekodierung: $x = a + z \cdot \frac{b-a}{2^k - 1}$

Beispiel: Intervall $[-1, 2]$, Genauigkeit $\varepsilon = 10^{-6}$, $x = 0.637197$

$$k = \left\lceil \log_2 \frac{2 - (-1)}{10^{-6}} \right\rceil = \left\lceil \log_2 3 \cdot 10^6 \right\rceil = 22$$

$$\begin{aligned} z &= \left\lfloor \frac{0.637197 - (-1)}{2 - (-1)}(2^{22} - 1) \right\rfloor = 2288966_{10} \\ &= 1000101110110101000110_2 \end{aligned}$$



HAMMING KLIPPEN

Problem:

- benachbarte Zahlen können sehr verschieden kodiert sein
- Kodierungen haben großen Hamming-Abstand (\neq verschied. Bits)
- Mutationen/Crossover überwinden „Hamming-Klappen“ schwer

Beispiel:

- Darstellung der Zahlen von 0 bis 1 durch 4-Bit-Zahlen
- also Abbildung $\frac{k}{15} \rightarrow k$
- $\frac{7}{15}$ (0111) und $\frac{8}{15}$ (1000) haben Hamming-Abstand 4



STANDBINÄRE KODIERUNG

Eine binäre Zeichenkette $A.G = A.G_1 \dots A.G_\ell \in \mathbb{B}^\ell$ repräsentiert mit *standardbinärer Kodierung* die folgende ganze Zahl

$$dec_{stdbin}(A.G) = \sum_{j=0}^{\ell-1} A.G_{\ell-j} \cdot 2^j.$$

Damit kann auch ein reellwertiges Intervall $[ug, og] \subset \mathbb{R}$ durch

$$dec_{stdbin,ug,og}(A.G) = ug + \frac{og - ug}{2^\ell - 1} \cdot dec_{stdbin}(A.G)$$

mit der Genauigkeit $\frac{og-ug}{2^\ell-1}$ dargestellt werden.



HAMMING ABSTAND

- Zwei binäre Zeichenketten $A.G, B.G \in \mathbb{B}^l$ besitzen den **Hamming-Abstand**

$$d_{ham}(A.G, B.G) = |\{i \in \mathbb{N}_0 \mid 1 \leq i \leq l \wedge A.Gi \neq B.Gi\}|.$$

- Dieses Maß gibt die Anzahl der Einzelinformationen an, die zwingend verändert werden müssen, um die Binärketten ineinander zu überführen.
- Die Zeichenketten 011 und 100, besitzen den maximal möglichen Hamming-Abstand 3: Es müssten folglich alle Bits invertiert werden, um vom enkodierten Wert 6 zum Wert 7 zu gelangen.
- Sobald ein Hamming-Abstand größer als 1 vorliegt, spricht man von einer Hamming-Klippe entsprechender Größe.
- Zerschneiden große Hamming-Klippen phänotypische Nachbarschaften, wird der Suchraum zerklüftet und dadurch eine Optimierung erschwert.



GRAY KODIERUNG

- Eine Möglichkeit, Hamming-Klippen zu vermeiden, besteht in der Wahl einer anderen Kodierung, der so genannten **Gray-Kodierung**.
- Sie besitzt die Eigenschaft, dass alle benachbarten Werte einer diskreten Suchraumdimension auf binäre Zeichenketten mit dem Hamming-Abstand 1 abgebildet werden.
- Die Gray-Kodierung lässt sich durch folgende Konversionsregel auf die standardbinäre Kodierung zurückführen:



GRAY KODIERUNG

Definition

Die *Gray-Kodierung* wird mittels der standardbinären Kodierung eingeführt. Eine standardbinär kodierte Zeichenkette $b = b_1 \dots b_l \in \mathbb{B}^l$ lässt sich durch die folgende Konversion in eine Gray-kodierte Zeichenkette $A.G = A.G_1 \dots A.G_l$ überführen ($1 \leq i \leq l$)

$$A.G_i = \begin{cases} b_i & \text{falls } i = 1 \\ b_{i-1} \oplus b_i & \text{falls } i > 1, \end{cases}$$

wobei das exklusive Oder \oplus der Addition modulo 2 entspricht. Ein Bit der standardbinär kodierte Zeichenkette lässt sich mit der folgenden Regel aus der Gray-kodierten Zeichenkette $A.G$ ableiten:

$$b_i = \bigoplus_{j=1}^i A.G_j = A.G_1 \oplus \dots \oplus A.G_i.$$

GRAY KODIERUNG

Definition

Mit dieser Transformation ergibt sich als Dekodierungsregel für eine Gray-kodierte Zeichenkette $A.G$

$$dec_{gray}(A.G) = dec_{stdbin}\left(\bigoplus_{j=1}^1 A.G_j \cdots \bigoplus_{j=1}^l A.G_j\right).$$

Mehrere Zahlen können erneut durch Konkatenation aneinander gefügt werden.



GRAY-KODES: VERMEIDUNG VON HAMMING-KLIPPEN

Lösung: Gray-Kodes

Hamming-Abstand benachbarter Zahlen = 1 Bit

binär	Gray
0000	0000
0001	0001
0010	0011
0011	0010

binär	Gray
0100	0110
0101	0111
0110	0101
0111	0100

binär	Gray
1000	1100
1001	1101
1010	1111
1011	1110

binär	Gray
1100	1010
1101	1011
1110	1001
1111	1000



GRAY KODIERUNG

Gray-Kodes sind nicht eindeutig

- jeder Kode, in dem sich Kodierungen benachbarter Zahlen nur in 1 Bit unterscheiden, heißt Gray-Kode
- Berechnung von Gray-Kodes meist aus Binärzahlkodierung



GRAY-KODES: BEISPIEL

Beispiel: Intervall $[-1, 2]$, Genauigkeit $\varepsilon = 10^{-6}$, $x = 0.637197$

$$z = \left\lfloor \frac{0.637197 - (-1)}{2 - (-1)} (2^{22} - 1) \right\rfloor = 2288966_{10}$$
$$= 1000101110110101000110_2$$

$$g = 1000101110110101000110_2$$
$$\oplus 100010111011010100011_2$$
$$= 1100111001101111100101_2$$



ZUSAMMENFASSUNG

- Lokale Optima hängen ausschließlich von der gewählten Darstellung (einschließlich der Dekodierungsfunktion) und dem Mutationsoperator ab.
- Die Anzahl der so eingeführten lokalen Optima kann als eine Maßzahl für die Angepasstheit eines Operators an das Problem gesehen werden.
- Je weniger lokale Optima von einem Operator und der Repräsentation induziert werden, desto bessere Ergebnisse können bei der Suche erwartet werden.
- Insbesondere bei einem lokalen Suchalgorithmus, der lediglich durch einen Mutations- und einen Selektionsoperator bestimmt wird, ist die Anzahl der lokalen Optima entscheidend.



KODIERUNG: ZWEITE WÜNSCHENSWERTE EIGENSCHAFT

Ähnlich kodierte Lösungskandidaten sollten ähnliche Fitness haben.

Problem der Epistasie:

- *in der Biologie:* ein Allel eines (sog. epistatischen) Gens unterdrückt Wirkung aller möglichen Allele eines/mehrerer anderer Gene
- *in evolutionären Algorithmen:*
Wechselwirkung zwischen Genen eines Chromosoms, Änderung der Fitness durch Änderung eines Gens hängt stark von Ausprägungen der anderen Gene ab



BEISPIEL: PROBLEM DES HANDLUNGSREISENDEN

Finde Rundreise mit minimalen Kosten durch n Städte.

zwei verschiedene Kodierungen der Rundreise:

1. Permutation der Städte

- besuche Stadt an k -ter Position im k -ten Schritt
- *geringe Epistasie*: z.B. Austausch zweier Städte ändert Fitness (Kosten) i.A. etwa gleich stark (lokale Tour-Änderung)

2. Angabe der Position der jeweils nächsten Stadt in Liste, aus der alle besuchten Städte gestrichen werden

- *hohe Epistasie*: Änderung eines Gens (speziell vorn liegendes) kann gesamte Rundreise ändern (globale Tour-Änderung)
- führt oft zu großen Änderungen der Fitness



ZWEITE KODIERUNG: WIRKUNG EINER MUTATION

Mutation	Chromosom	Liste noch zu besuchender Städte	Rundreise
davor	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">5</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">3</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">3</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">2</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">2</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">1</div>	1, 2, 3, 4, 5 , 6 1, 2, 3 , 4, 6 1, 2, 4 , 6 1, 2 , 6 1, 6 1	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">5</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">3</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">4</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">2</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">6</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">1</div>
danach	<div style="border: 1px solid black; padding: 2px; display: inline-block; background-color: #cccccc; margin-bottom: 2px;">1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">3</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">3</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">2</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">2</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">1</div>	1 , 2, 3, 4, 5, 6 2, 3, 4 , 5, 6 2, 2, 3 , 6 2, 2, 6 2	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">4</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">5</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">3</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">6</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 2px;">2</div>



KODIERUNG: DRITTE WÜNSCHENSWERTE EIGENSCHAFT

Der Suchraum Ω sollte (soweit möglich) unter den verwendeten evolutionären Operatoren abgeschlossen sein.

Verlassen des Suchraums ist u.U. Definitionsfrage
allgemein: **Suchraum wird verlassen**, wenn

- neues Chromosom nicht sinnvoll interpretiert/dekodiert wird
- Lösungskandidat bestimmte Anforderungen nicht erfüllt,
- Lösungskandidat durch Fitnessfunktion falsch bewertet wird

Problem der **Abstimmung** von Kodierung und EA-Operatoren:

- Verwenden kodierungsspezifischer evolutionärer Operatoren
- Einsatz von Mechanismen zur „Reparatur“ von Chromosomen
- Strafterms zur Verringerung der Fitness von Chromosomen $\notin \Omega$



VERLASSEN DES SUCHRAUMS: BEISPIEL

N-DAMEN PROBLEM

zwei verschiedene Kodierungen: Chromosom der Länge n

1. Spaltenpositionen der Damen je Zeile (Allele $0, \dots, n - 1$)

Operatoren: Ein-Punkt-Crossover, Standardmutation

Entstehung stets gültiger Vektoren von Spaltenpositionen

⇒ Suchraum wird nicht verlassen

2. Nummern der Felder (Allele $0, \dots, n^2 - 1$) der Damen

Operatoren: Ein-Punkt-Crossover, Standardmutation

Entstehung von Chromosomen mit mehreren Damen auf 1 Feld

⇒ Suchraum wird verlassen



VERLASSEN DES SUCHRAUMS: LÖSUNGSANSÄTZE

N-DAMEN PROBLEM

- **Andere Kodierung verwenden:** Kodierung 1 vermeidet Problem und Ω deutlich kleiner (wenn durchführbar, beste Variante!)
- **Kodierungsspezifische evolutionäre Operatoren:**
 - *Mutation:* Ausschließen bereits vorhandener Allele beim Zufall
 - *Crossover:* suche zuerst Feldnummern je Chromosom, die im anderen Chromosom nicht vorkommen, und wende auf verkürzten Chromosomen Ein-Punkt-Crossover an
- **Reparaturmechanismus:** finde und ersetze mehrfach auftretende Feldnummern, bis alle Feldnummern verschieden
- **Strafterm:** verringere Fitness um Anzahl der Doppel-/Mehrfachbelegungen von Feldern, ggf. multipliziert mit Gewicht



VERLASSEN DES SUCHRAUMS: AM BEISPIEL DES TSP

N-DAMEN PROBLEM

- Darstellung der Rundreise durch Permutation der Städte (Stadt an k -ter Position wird im k -ten Schritt besucht.)
- Ein-Punkt-Crossover kann Raum der Permutationen verlassen:

3	5	2	8	1	7	6	4
1	2	3	4	5	6	7	8

3	5	2	4	5	6	7	8
1	2	3	8	1	7	6	4

- **Kodierungsspezifische evolutionäre Operatoren:**
 - *Mutation*: z.B. Zweiertausch, Teilstück-Verschiebung, Inversion
 - *Crossover*: Kantenrekombination (wird später noch besprochen)
- **Reparaturmechanismus**: entferne doppelt auftretende Städte und hänge fehlenden Städte ans Ende an:

3	5	2	4	5	6	7	8	1
---	---	---	---	--------------	---	---	---	---

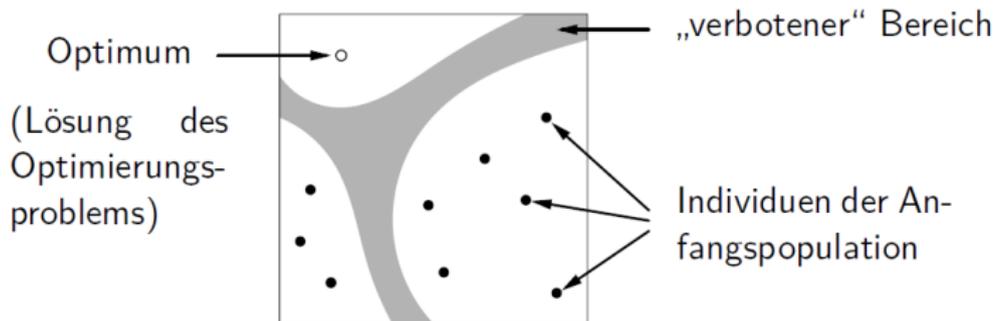
- **Strafterm**: verringere Fitness um Wert c für jede fehlende Stadt



VERLASSEN DES SUCHRAUMS

N-DAMEN PROBLEM

- falls Ω *nicht zusammenhängend*, erschwert *Reparaturmech.* Suche
- sofortiges Zurückführen „verbotener“ $x \in \Omega$ in erlaubte Bereiche



- in solchen Fällen: **Strafterm** einzuführen
- $x \in \Omega$ im „verbotenen“ Bereich wird bestraft aber nicht entfernt
- Strafterm sollte mit Zeit wachsen: unterdrückt in späteren Generationen $x \in \Omega$ in „verbotenen“ Bereichen

PRINZIPIEN DER SELEKTION

- bessere Individuen (bessere Fitness) sollen größere Chancen haben, Nachkommen zu zeugen (differentielle Reproduktion)
- **Selektionsdruck:** Stärke der Bevorzugung guter Individuen
- Wahl des Selektionsdrucks: Gegensatz von **Durchforstung des Suchraums (engl. exploration):**
 - möglichst breite Streuung der Individuen über Ω
 - möglichst große Chancen, globales Optimum zu finden

⇒ geringer Selektionsdruck wünschenswert

Ausbeutung guter Individuen (engl. exploitation):

- Anstreben des (u.U. lokalen) Optimums in Nähe guter Individuen
 - Konvergenz zum Optimum
- ⇒ hoher Selektionsdruck wünschenswert



VERGLEICH VON SELEKTIONSVERFAHREN

Vergleichen von Verfahren für erzeugten Selektionsdruck durch Maße

- **Übernahmezeit:** ist Anzahl der Generationen bis Population konvergiert
(Population heißt *konvergiert*, wenn alle Individuen identisch)
- **Selektionsintensität:** bestimmt durch Selektionsdifferenzial zwischen durchschnittlicher Güte vor und nach Selektion



SELEKTIONSINTENSITÄT

Definition (Selektionsintensität)

Sei (Ω, f, \succ) das betrachtete Optimierungsproblem und werde ein Selektionsoperator $\text{Sel}^\xi : (\mathcal{G} \times \mathcal{Z} \times \mathbb{R})^r \rightarrow (\mathcal{G} \times \mathcal{Z} \times \mathbb{R})^s$ auf eine Population P mit durchschnittlicher Güte μ_f und Standardabweichung σ_f der Gütewerte angewendet. Dann sei μ_f^{sel} die durchschnittliche Güte der Population P_{sel} und der Selektionsoperator besitzt die *Selektionsintensität*

$$I_{\text{sel}} = \begin{cases} \frac{\mu_f^{\text{sel}} - \mu_f}{\sigma_f} & \text{falls } \succ = >, \\ \frac{\mu_f - \mu_f^{\text{sel}}}{\sigma_f} & \text{sonst.} \end{cases}$$



SELEKTIONSINTENSITÄT

je größer I_{sel} , desto größer der Selektionsdruck

Beispiel:

- 10 Indiv. mit Fitness: 2.0, 2.1, 3.0, 4.0, 4.3, 4.4, 4.5, 4.9, 5.5, 6.0
- Selektion liefert Individuen mit Gütewerten: 2.0, 3.0, 4.0, 4.4, 5.5

$$\mu_f = \frac{1}{|P|} \sum_{i=1}^{|P|} A^{(i)} \cdot F \quad (\text{Mittelwert der Fitness})$$

$$\sigma_f = \sqrt{\frac{1}{|P| - 1} \sum_{i=1}^{|P|} (A^{(i)} \cdot F - \mu_f)^2} \quad (\text{Standardabweichung})$$

$$\Rightarrow \mu_f = 4.07, \quad \sigma_f = 1.27, \quad \mu_f^{sel} = 3.78, \quad I_{sel} = \frac{4.07 - 3.78}{1.27} = 0.228$$

Kritik an Selektionsintensität:

- dieses Maß setzt Standardnormalverteilung der Gütewerte voraus
- ist oft nicht der Fall bei allgemeinen Optimierungsproblemen

WAHL DES SELEKTIONSDRUCKS

- **beste Strategie:** zeitabhängiger Selektionsdruck
geringer Selektionsdruck in früheren Generationen
höherer Selektionsdruck in späteren Generationen
- d.h. zuerst gute Durchforstung des Suchraums,
dann Ausbeutung der erfolgversprechendsten Region
- Steuerung des Selektionsdrucks über Skalierung der
Fitnessfunktion oder über Parameter des Selektionsverfahrens
- wichtige **Selektionsverfahren:**
Glücksradauswahl, Rangauswahl, Turnierauswahl
- wichtige **Skalierungsmethoden:**
Anpassung der Fitnessvariation, linear dynamische Skalierung,
 σ -Skalierung



GLÜCKSRADAUSWAHL (ENGL. ROULETTE WHEEL SELECTION)

- das bekannteste Verfahren
- berechne relative Fitness der Individuen $A^{(i)}$, $1 \leq i \leq |P|$

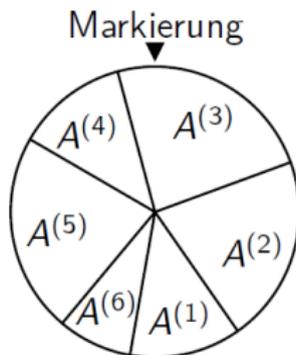
$$f_{\text{rel}}\left(A^{(i)}\right) = \frac{A^{(i)} \cdot F}{\sum_{j=1}^{|P|} A^{(j)} \cdot F}$$

und interpretiere $f_{\text{rel}}\left(A^{(i)}\right)$ als Auswahlw'keit von $A^{(i)}$
(sog. **fitnessproportionale Selektion**)

- **Beachte:** absolute Fitness $A \cdot F$ darf nicht negativ sein
(ggf. pos. Wert addieren oder neg. Werte zu Null setzen)
- **Beachte:** Fitness muss zu maximieren sein
(sonst: Selektion schlechter Individuen mit hoher W'keit)
- **Veranschaulichung:** Glücksrad mit 1 Sektor je Individuum $A^{(i)}$,
Sektorgrößen = relative Fitnesswerte $f_{\text{rel}}\left(A^{(i)}\right)$



GLÜCKSRADAUSWAHL: VERANSCHAULICHUNG



Auswahl eines Individuums:

1. drehe Glücksrad
2. wähle Indiv. des markierten Sektors

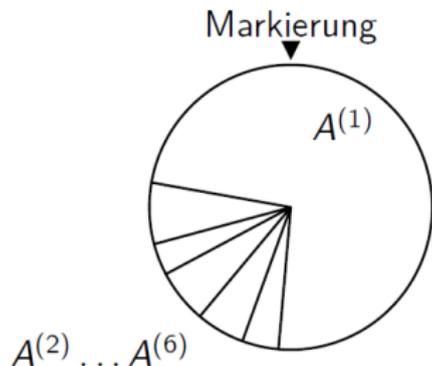
Auswahl der Zwischenpopulation:

- wiederhole Auswahl so oft, wie # Individuen in Population

technischer Nachteil: Berechnung der relativen Fitness durch Summierung aller Fitnesswerte (Normierungsfaktor)

- konstante Ausgangspopulation während Auswahl
- erschwerte Parallelisierung der Implementierung

GLÜCKSRADAUSWAHL: DOMINANZPROBLEM



- Individuum mit sehr hoher Fitness kann Auswahl **dominieren**
 - verstärkte Dominanz in Folgegenerationen aufgrund von Kopien/sehr ähnliche Individuen
- ⇒ **Crowding:** Population aus gleichen/sehr ähnlichen Individuen

- führt zum sehr schnellen Finden eines (lokalen) Optimums
- **Nachteil:** Diversität der Population geht verloren
 - Ausbeutung eines oder weniger guter Individuen
 - keine Durchforstung des Suchraums, sondern lokale Optimierung (in späten Generationen erwünscht, am Anfang unerwünscht)

GLÜCKSRADAUSWAHL: SELEKTIONSINTENSITÄT

Theorem

Bei reiner fitnessproportionaler Selektion in einer Population mit durchschnittlicher Güte μ_f und Gütevarianz σ_f^2 beträgt die Selektionsintensität

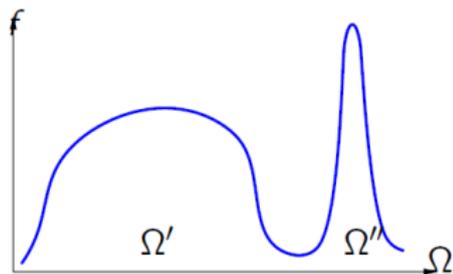
$$I_{sel} = \frac{\sigma_f}{\mu_f}.$$



FITNESSFUNKTION: PROBLEM DER VORZEITIGEN KONVERGENZ

Dominanzproblem zeigt starken Einfluss der Fitnessfunktion auf Wirkung der fitnessproportionalen Selektion

- Problem der **vorzeitigen Konvergenz**:
falls Wertebereich der zu maximierenden Funktion sehr groß
- Beispiel: anfangs im Bereich Ω'' kein Chromosom \rightarrow Population bleibt durch Selektion nahe (lokales) Maximum im Bereich Ω'



Individuen, die sich dem Bereich zw. Ω' und Ω'' nähern, haben sehr schlechte Chancen auf Nachkommen

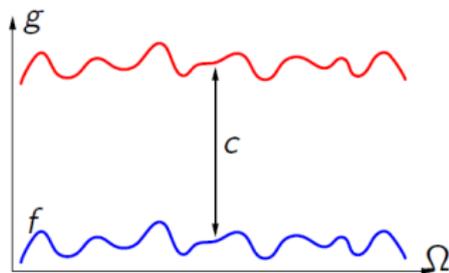
PROBLEM DES VERSCHWINDENDEN SELEKTIONSDRUCKS

Problem der **absoluten Höhe** der Fitnesswerte i.V. zu ihrer **Variation** umgekehrt: Problem des **verschwindenden Selektionsdrucks**:

- Maximierung von $f : \Omega \rightarrow \mathbb{R}$ ist äquivalent zur Maximierung von $g : \Omega \rightarrow \mathbb{R}$ mit $g(x) \equiv f(x) + c$, $c \in \mathbb{R}$

$$c \gg \sup_{x \in \Omega} f(x) \implies \forall x \in \Omega : g_{\text{rel}}(x) \approx \frac{1}{|P|} \quad (\text{Pop.-größe } |P|)$$

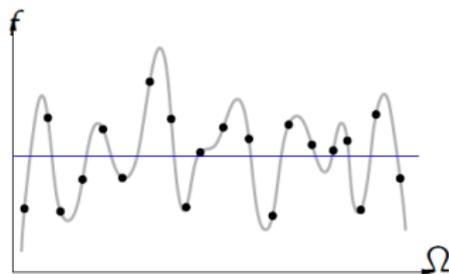
\Rightarrow (zu) geringer Selektionsdruck



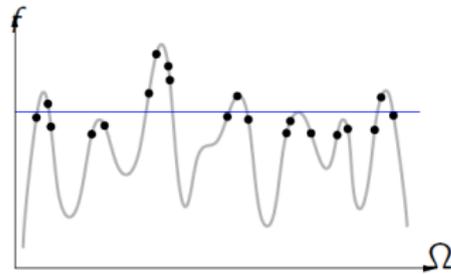
- obwohl Maxima an gleichen Stellen: mit EA unterschiedlich leicht zu finden
- mit g : nur (zu) geringe Unterschiede der f_{rel}

PROBLEM DES VERSCHWINDENDEN SELEKTIONSDRUCKS

- EA erzeugt u.U. dieses Problem selbst
- er steigert tendenziell (durchschnittliche) Fitness der Individuen
- höherer Selektionsdruck am Anfang, da Fitnesswerte zufällig
- später: geringerer Selektionsdruck (umgekehrt wäre besser)
- Beispiel: Punkte zeigen Individuen der Generation



frühe Generation



späte Generation

ANPASSUNG DER FITNESSFUNKTION

Lösungsansatz: **Skalierung der Fitness** (Verschiebung der Fitnesswerte)

linear dynamische Skalierung:

$$f_{\text{lds}}(A) = \alpha \cdot A.F - \min \left\{ A^{(i)}.F \mid P(t) = \{A^{(1)}, \dots, A^{(r)}\} \right\}, \quad \alpha > 0$$

- Idee: Minimum der Fitness der Population bei allen Individuen abziehen
- anstatt Minimum von $P(t)$ auch Minimum der letzten k Generationen
- gewöhnlich $\alpha > 1$

σ -Skalierung:

$$f_{\sigma}(A) = A.F - (\mu_f(t) - \beta \cdot \sigma_f(t)), \quad \beta > 0$$

Problem: Wahl der Parameter



ANPASSUNG DER FITNESSFUNKTION

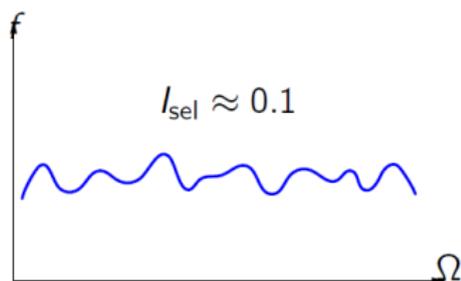
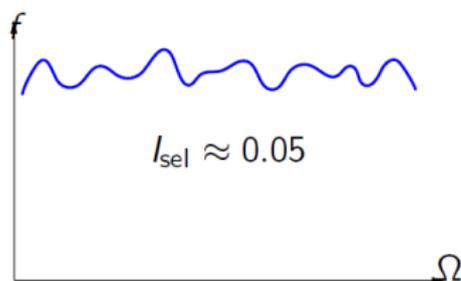
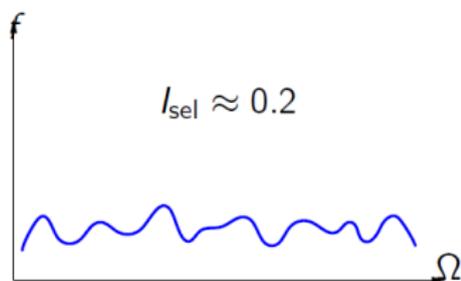
- betrachte **Variationskoeffizienten** der Fitnessfunktion

$$v = \frac{\sigma_f}{\mu_f} = \frac{\sqrt{\frac{1}{|\Omega|-1} \sum_{x' \in \Omega} \left(f(x') - \frac{1}{|\Omega|} \sum_{x \in \Omega} f(x) \right)^2}}{\frac{1}{|\Omega|} \sum_{x \in \Omega} f(x)}, \quad v(t) = \frac{\sigma_f(t)}{\mu_f(t)}$$

- empirische Feststellung: $v \approx 0.1$ liefert gutes Verhältnis von Durchforstung und Ausbeutung
 - wenn $v \neq 0.1$, dann Anpassung von f (z.B. durch Skalierung)
 - v ist nicht berechen-, sondern nur schätzbar
 - praktische Berechnungen von v : Ersetzen von Ω durch $P(t)$
 - somit: Annäherung von v durch **Selektionsintensität** $I_{sel}(t)$
- ⇒ in jeder Generation: berechne $I_{sel}(t)$ und passe f entsprechend an (σ -Skalierung mit $\beta = \frac{1}{I_{sel}^*}$, $I_{sel}^* = 0.1$)



ILLUSTRATION DER SELEKTIONSINTENSITÄT



- zu hohe I_{sel} : vorzeitige Konvergenz
- zu niedrige I_{sel} : verschwindender Selektionsdruck
- gut: $I_{sel} \approx 0.1$

ANPASSUNG DER FITNESSFUNKTION: ZEITABHÄNGIGKEIT

- bestimme f_{rel} nicht direkt aus $f(x)$, sondern aus $g(x) \equiv (f(x))^{k(t)}$
- zeitabhängige Exponent $k(t)$ steuert Selektionsdruck
- **Verfahren zur Bestimmung von $k(t)$** [Michalewicz, 1996]
(soll Selektionsintensität I_{sel} in Nähe von $I_{sel}^* \approx 0.1$ halten)

$$k(t) = \left(\frac{I_{sel}^*}{I_{sel}} \right)^{\beta_1} \left(\tan \left(\frac{t}{T+1} \cdot \frac{\pi}{2} \right) \right)^{\beta_2} \left(\frac{I_{sel}}{I_{sel}^*} \right)^{\alpha}$$

I_{sel}^* , β_1 , β_2 , α : Parameter des Verfahrens

I_{sel} : Variationskoeffizient (z.B. aus $P(t=0)$ geschätzt)

T : maximale Anzahl zu berechnender Generationen

t : aktueller Zeitschritt (Nummer der Generation)

- Empfehlung: $I_{sel}^* = 0.1$, $\beta_1 = 0.05$, $\beta_2 = 0.1$, $\alpha = 0.1$

ANPASSUNG DER FITNESSFUNKTION: BOLTZMANN-SELEKTION

- bestimme relative Fitness nicht direkt aus $f(x)$, sondern aus $g(x) \equiv \exp\left(\frac{f(x)}{kT}\right)$
- zeitabhängige **Temperatur** T steuert Selektionsdruck
- k ist Normierungskonstante
- Temperatur nimmt z.B. linear bis zu vorher festgelegten Maximalzahl an Generationen ab



VARIATION DURCH MUTATION

- Variationen (Mutationen): kleine Veränderungen in der Biologie
- Mutationsoperator: ändert möglichst wenig am Lösungskandidaten bzgl. Fitnessfunktion
- im Folgenden: Untersuchung im Zusammenspiel mit Selektion
- hier: Verhalten eines einfachen Optimierungsalgorithmus auf sehr einfachem Optimierungsproblem (Abgleich mit einem vorgegebenen Bitmuster)



ROLLEN DER MUTATION

- Mutationsoperatoren können unterschiedliche Aufgaben in evolutionären Algorithmen erfüllen.
- In den bisherigen Abschnitten hat die Mutation die Rolle des wichtigsten (weil einzigen) Suchoperators.
- Unter dieser Prämisse übernimmt sie zwei Funktionen: einerseits die **Feinabstimmung (engl. exploitation)**, um ausgehend von einem guten Lösungskandidaten das zugehörige lokale Optimum zu finden, andererseits das **stichprobenartige Erforschen (engl. exploration)** weiter entfernter Gebiete des Suchraums, um das Einzugsgebiet eines potentiell besseren lokalen Optimums zu identifizieren.



BINÄRE MUTATION

Input: Individuum A mit $A.G \in \{0, 1\}^l$

Output: Individuum B

$B \leftarrow A$

for $i \in \{1, \dots, l\}$ **{**

$u \leftarrow$ wähle zufällig gemäß $U([0, 1))$

if $u \leq p_m$ **{** /* Mutationswahrscheinlichkeit p_m */

$B.G_i \leftarrow 1 - A.G_i$

}

}

return B

Abbildung 4: alle Bits werden mit einer Wahrscheinlichkeit invertiert.



GAUSS MUTATION

- Arbeitet direkt auf der reellwertigen Zahl x .
- Addiert zu jeder Komponente des bisherigen Lösungskandidaten einen normal- bzw. Gaußverteilten Zufallswert $u_i \sim N(0, \sigma)$. Die zugehörige Dichtefunktion ist

$$\phi(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e\left(-\frac{1}{2\sigma^2} \cdot x^2\right)$$

mit der Varianz σ^2

- Durch diese Wahrscheinlichkeitsverteilung werden viele Mutationen nur kleine Veränderungen vornehmen, aber auch größere Sprünge sind möglich.



GAUSS MUTATION

- alternative reellwertige Mutation
- direkt auf den reellwertigen Zahlen
- Addition einer normalverteilten Zufallszahl auf jede Komponente

Input: Individuum A mit $A.G \in \mathbb{R}^l$

Output: Individuum B

```
for  $i \in \{1, \dots, l\}$  {  
     $u_i \leftarrow$  wähle zufällig gemäß  $N(0, \sigma)$  /* Standardabweichung  $\sigma$  */  
     $B_i \leftarrow A_i + u_i$   
     $B_i \leftarrow \max\{B_i, ug_i\}$  /* untere Wertebereichsgrenze  $ug_i$  */  
     $B_i \leftarrow \min\{B_i, og_i\}$  /* obere Wertebereichsgrenze  $og_i$  */  
}  
return  $B$ 
```



VERGLEICH DER MUTATIONSVERFAHREN

ANSATZ

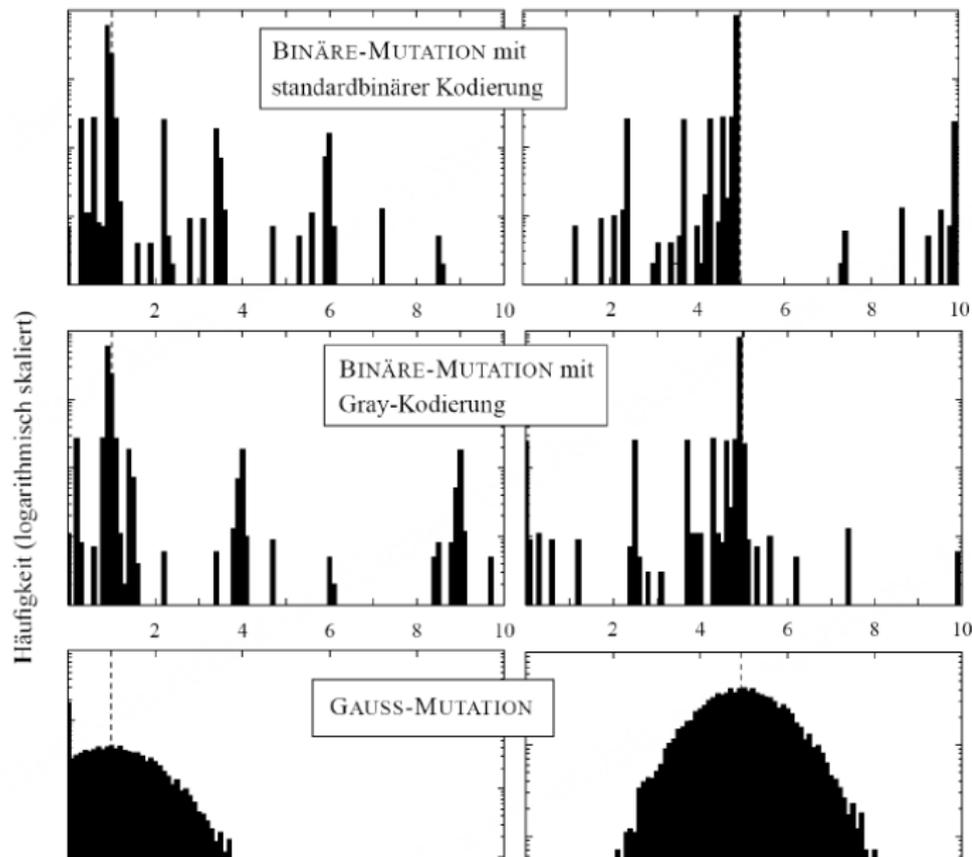
- Optimierung der einfachen Funktion

$$f_2(x) = \begin{cases} x & \text{falls } x \in [0, 10] \subset \mathbb{R} \\ \text{undefiniert} & \text{sonst} \end{cases}$$

- zwei Elternindividuen (1.0 und 4.99)
- Anwendung von drei Mutationsoperatoren auf diese beiden Individuen, jeweils 10000 Mal
- Häufigkeitsverteilungen (bei Gauß-Mutation $\sigma = 1$)



VERGLEICH DER MUTATIONSVERFAHREN



VERGLEICH DER MUTATIONSVERFAHREN

- Gauß-Mutation mit kleinem σ sehr gut für Exploitation
- mit großem σ sehr breite Erforschung
- binäre Mutation detektiert schneller interessante Regionen in Ω
- binäre Mutation eines GA hat mehrere verteilte Schwerpunkte
- Hamming-Klappen = Brüche in Häufigkeitsverteilung
- Gray-Kodierung schafft es, phänotypische Nachbarn einzubinden
- tendiert dennoch zu einer Seite des Suchraums



GENETISCHE OPERATOREN

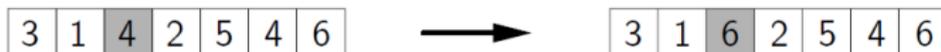
- werden auf bestimmten Teil ausgewählter Individuen (Zwischenpopulation) angewandt
- Erzeugung von Varianten und Rekombinationen bestehender Lösungskandidaten
- allgemeine Einteilung genetischer Operatoren nach Zahl der Eltern:
 - Ein-Elter-Operatoren (**Mutation**)
 - Zwei-Elter-Operatoren (**Crossover**)
 - Mehr-Elter-Operatoren
- genetische Operatoren haben bestimmte Eigenschaften (abhängig von Kodierung)
 - wenn Lösungskandidaten Permutationen sind, dann permutationserhaltende genetische Operatoren verwenden
 - allgemein: falls bestimmte Allelkombinationen unsinnig sind, sollten sie vermieden werden



STANDARDMUTATION UND ZWEIERTAUSCH

- **Standardmutation:**

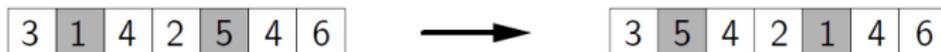
Austausch der Ausprägung eines Gens durch anderes Allel



- ggf. werden mehrere Gene mutiert (vgl. n -Damen-Problem)
- *Parameter:* Mutationswahrscheinlichkeit p_m , $0 < p_m \ll 1$ für Bitstrings der Länge l ist $p_m = 1/l$ annähernd optimal

- **Zweiertausch:**

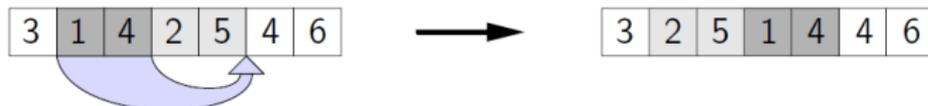
Austausch der Ausprägungen zweier Gene eines Chromosoms



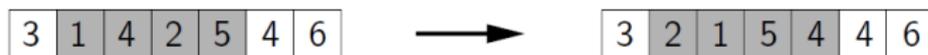
- *Voraussetzung:* gleiche Allelmengen der ausgetauschten Gene
- *Verallgemeinerung:* zyklischer Tausch von $3, 4, \dots, k$ Genen

OPERATIONEN AUF TEILSTÜCKE

- **Verschieben eines Teilstücks:**



- **Mischen/Permutieren eines Teilstücks:**



- **Umdrehen/Invertieren eines Teilstücks:**



- *Voraussetzung:* gleiche Allelmengen im betroffenen Bereich
- *Parameter:* ggf. W' keitsverteilung über Längen (und Verschiebungsweiten für Verschieben eines Teilstücks)

EIN-PUNKT- UND ZWEI-PUNKT-CROSSOVER

Ein-Punkt-Crossover

- Bestimmen eines zufälligen Schnittpunktes
- Austausch der Gensequenzen auf einer Seite des Schnittpunktes



Zwei-Punkt-Crossover

- Bestimmen zweier zufälliger Schnittpunkte
- Austausch der Gensequenzen zwischen den beiden Schnittpunkten



n-PUNKT- UND UNIFORMES CROSSOVER

n-Punkt-Crossover

- Verallgemeinerung des Ein- und Zwei-Punkt-Crossover
- Bestimmen von n zufälligen Schnittpunkten
- Abwechselndes Austauschen / Nicht-Austauschen der Gensequenzen zwischen zwei aufeinanderfolgenden Schnittpunkten

Uniformes Crossover

- für jedes Gen: bestimme ob es getauscht wird oder nicht (+: ja, -: nein, *Parameter*: W'keit p_x für Austausch)



- **Beachte:** uniformes Crossover entspricht **nicht** dem $(l - 1)$ -Punkt-Crossover! Zahl der Crossoverpunkte ist zufällig



SHUFFLE CROSSOVER

- vor Ein-Punkt-Crossover: zufälliges Mischen der Gene
- danach: Entmischen der Gene

Mischen	Crossover	Entmischen																																																																									
<table border="1"><tr><td>5</td><td>2</td><td>1</td><td>4</td><td>3</td><td>6</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>3</td><td>1</td><td>4</td><td>2</td><td>5</td><td>4</td></tr></table>	5	2	1	4	3	6	1	2	3	4	5	6	3	1	4	2	5	4	<table border="1"><tr><td>4</td><td>2</td><td>6</td><td>3</td><td>5</td><td>1</td></tr><tr><td>4</td><td>2</td><td>6</td><td>5</td><td>1</td><td>3</td></tr><tr><td>2</td><td>1</td><td>4</td><td>5</td><td>3</td><td>4</td></tr></table>	4	2	6	3	5	1	4	2	6	5	1	3	2	1	4	5	3	4	<table border="1"><tr><td>4</td><td>2</td><td>6</td><td>5</td><td>3</td><td>4</td></tr><tr><td>4</td><td>2</td><td>6</td><td>5</td><td>1</td><td>3</td></tr><tr><td>2</td><td>1</td><td>4</td><td>3</td><td>5</td><td>1</td></tr></table>	4	2	6	5	3	4	4	2	6	5	1	3	2	1	4	3	5	1	<table border="1"><tr><td>3</td><td>2</td><td>4</td><td>4</td><td>5</td><td>6</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>5</td><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	3	2	4	4	5	6	1	2	3	4	5	6	5	1	1	2	3	4
5	2	1	4	3	6																																																																						
1	2	3	4	5	6																																																																						
3	1	4	2	5	4																																																																						
4	2	6	3	5	1																																																																						
4	2	6	5	1	3																																																																						
2	1	4	5	3	4																																																																						
4	2	6	5	3	4																																																																						
4	2	6	5	1	3																																																																						
2	1	4	3	5	1																																																																						
3	2	4	4	5	6																																																																						
1	2	3	4	5	6																																																																						
5	1	1	2	3	4																																																																						

- Shuffle Crossover ist **nicht** äquivalent zum uniformen Crossover!
- jede Anzahl von Vertauschungen von Genen zwischen Chromosomen ist gleichwahrscheinlich
- uniformen Crossover: Anzahl ist binomialverteilt mit Parameter p_x
- Shuffle Crossover: eines der empfehlenswertesten Verfahren



UNIFORMES ORDNUNGSBASIERTES CROSSOVER

- ähnlich wie uniformes Crossover: entscheide für jedes Gen, ob es erhalten bleibt oder nicht
(+: ja, -: nein, *Parameter*: W'keit p_k für Erhalt)
- fülle Lücken durch fehlende Allele auf (in Reihenfolge der Vorkommen im anderen Chromosom)



- erhält **Reihenfolgeinformation**
- *alternativ*: Erhalten der „+“ bzw. „-“ markierten Gene im einen bzw. anderen Chromosom

KANTENREKOMBINATION (SPEZIELL FÜR TSP)

- Chromosom wird als Graph (Kette oder Ring) aufgefasst. Jedes Gen besitzt Kanten zu seinen Nachbarn im Chromosom
- Kanten der Graphen zweier Chromosomen werden gemischt, daher Name
- erhält Nachbarschaftsinformation

Vorgehen: 1. Aufbau einer Kantentabelle

- Liste zu jedem Allel seine Nachbarn (in beiden Eltern) (ggf. erstes und letztes Gen des Chromosoms benachbart)
- falls ein Allel in beiden Eltern gleichen Nachbarn (Seite irrelevant), dann liste diesen Nachbar nur 1x auf (aber markiert)



KANTENREKOMBINATION

Vorgehen: 2. Aufbau eines Nachkommen

- wähle erstes Allel zufällig aus einem der beiden Eltern
- lösche ausgewähltes Allel aus Kantentabelle (aus Listen der Nachbarn der Allele)
- wähle jeweils nächstes Allel aus den noch nicht gelöschten Nachbarn des vorangehenden mit folgender Priorität:
 - 1 markierte (d.h. doppelt auftretende) Nachbarn
 - 2 Nachbarn mit kürzester Nachbarschaftsliste (wobei markierte Nachbarn einfach zählen)
 - 3 zufällige Auswahl eines Nachbarn

Erzeugung des zweiten Nachkommen analog aus erstem Allel des anderen Elter (meist jedoch nicht gemacht).



BEISPIEL: KANTENREKOMBINATION

Beispiel:

A: 6 3 1 5 2 7 4

B: 3 7 2 5 6 1 4

Aufbau der Kantentabelle

Allel	Nachbarn		zusammengefasst
	in A	in B	
1	3, 5	6, 4	3, 4, 5, 6
2	5, 7	7, 5	5*, 7*
3	6, 1	4, 7	1, 4, 6, 7
4	7, 6	1, 3	1, 3, 6, 7
5	1, 2	2, 6	1, 2*, 6
6	4, 3	5, 1	1, 3, 4, 5
7	2, 4	3, 2	2*, 3, 4

- beide Chromosomen = Ring (erstes und letztes Gen benachbart): in **A** ist 4 linker Nachbar der 6, 6 ist rechter Nachbar der 4; **B** analog
- in beiden: 5, 2 und 7 stehen nebeneinander – sollte erhalten werden (siehe Markierungen)



BEISPIEL: KANTENREKOMBINATION

Aufbau eines Nachkommen

6 5 2 7 4 3 1

Allel	Nachbarn	Wahl: 6	5	2	7	4	3	1
1	3, 4, 5, 6	3, 4, 5	3, 4	3, 4	3, 4	3		
2	5*, 7*	5*, 7*	7*	7*	—	—	—	—
3	1, 4, 6, 7	1, 4, 7	1, 4, 7	1, 4, 7	1, 4	1	1	—
4	1, 3, 6, 7	1, 3, 7	1, 3, 7	1, 3, 7	1, 3	1, 3	—	—
5	1, 2*, 6	1, 2*	1, 2*	—	—	—	—	—
6	1, 3, 4, 5	1, 3, 4, 5	—	—	—	—	—	—
7	2*, 3, 4	2*, 3, 4	2*, 3, 4	3, 4	3, 4	—	—	—

- starte mit erstem Allel des Chromosoms **A** (also 6) und streiche 6 aus allen Nachbarschaftslisten (dritte Spalte)
- da unter Nachbarn der 6 (1, 3, 4, 5) die 5 kürzeste Liste hat, wird 5 für zweites Gen gewählt
- dann folgt die 2, die 7 usw.

KANTENREKOMBINATION

- Nachkomme hat meist neue Kante (vom letzten zum ersten Gen)
- kann auch angewendet werden, wenn erstes und letztes Gen nicht als benachbart gelten: Kanten werden dann nicht in Kantentabelle aufgenommen
- sind erstes und letztes Gen benachbart, dann Startallel beliebig falls nicht, dann ein am Anfang stehendes Allel
- Aufbau eines Nachkommen: es ist möglich, dass Nachbarschaftsliste des gerade ausgewählten Allels leer (Prioritäten sollen Wahrscheinlichkeit dafür gering halten; sind aber nicht perfekt)

In diesem Fall: zufällige Auswahl aus den noch übrigen Allelen



DREI- UND MEHR-ELTER-OPERATOREN

Diagonal-Crossover

- ähnlich wie 1-, 2- und n -Punkt-Crossover, aber für mehr Eltern
- bei drei Eltern: zwei Crossover-Punkte
- verschiebt Gensequenzen an Schnittstellen über Chromosomen diagonal und zyklische



- Verallgemeinerung auf > 3 Eltern:
wähle für k Eltern $k - 1$ Crossover-Punkte
- führt zu sehr guter Durchforstung des Suchraums,
besonders bei großer Elternzahl (10–15 Eltern)

CHARAKTERISIERUNG VON CROSSOVER-OP.

Ortsabhängige Verzerrung (engl. positional bias):

- falls Wahrscheinlichkeit, dass 2 Gene zusammen vererbt werden (im gleichen Chromosom bleiben, zusammen ins andere Chromosom wandern) von ihrer relativen Lage im Chromosom abhängt
- unerwünscht, weil Anordnung der Gene im Chromosom entscheidenden Einfluss auf Erfolg/Misserfolg des EA haben (bestimmte Anordnungen lassen sich schwerer erreichen)
- Beispiel: **Ein-Punkt-Crossover**
 - 2 Gene werden voneinander getrennt (gelangen in verschiedene Nachkommen), falls Crossover-Punkt zwischen sie fällt
 - je näher 2 Gene im Chromosom beieinander, desto weniger mögliche Crossover-Punkte gibt es zwischen ihnen
 - nebeneinanderliegende Gene werden mit höherer Wahrscheinlichkeit als entferntliegende in gleichen Nachkommen gelangen.



CHARAKTERISIERUNG VON CROSSOVER-OP.

Verteilungsverzerrung (engl. distributional bias):

- falls Wahrscheinlichkeit, dass best. Anzahl von Genen ausgetauscht wird, nicht für alle Anzahlen gleich
- unerwünscht, weil Teillösungen unterschiedl. Größe unterschiedl. gute Chancen haben, in nächste Generation zu gelangen
- Verteilungsverzerrung meist weniger kritisch (d.h. eher tolerierbar) als ortabhängige Verzerrung
- **Beispiel: uniformes Crossover**
 - da jedes Gen unabhängig von allen anderen mit W'keit p_x ausgetauscht, Anzahl k der ausgetauschten Gene ist binomialverteilt mit Parameter p_x :

$$P(K = k) = \binom{n}{k} p_x^k (1-p_x)^{n-k} \quad \text{mit } n \hat{=} \text{ Gesamtzahl der Gene}$$

⇒ sehr kleine und sehr große Anzahlen sind unwahrscheinlicher



INTERPOLIERENDE UND EXTRAPOLIERENDE REKOMBINATION

MOTIVATION

- bisher: nur kombinierende Operatoren für Verknüpfung mehrerer Individuen
 - Ein-Punkt-, Zwei-Punkt- und n -Punkt-Crossover
 - Uniformes (ordnungsbasiertes) Crossover
 - Shuffle Crossover
 - Kantenrekombination
 - Diagonal-Crossover
- alle stark abhängig von Diversität der Population
- erschaffen keine neuen Genbelegungen und können somit nur Teilbereiche von Ω erreichen, die Individuen der Population enthalten
- hohe Diversität einer Population ist Zeichen für sehr gute Erforschung von Ω durch kombinierende Operatoren



DIE VIELFALT IN EINER POPULATION

Sei die Population $P = \langle A^{(i)} \rangle_{1 \leq i \leq s}$ zum Genotyp $\mathcal{G} = G^\ell$ gegeben – d. h. $A^{(i)}.G \in \mathcal{G}$. Dann werden die folgenden *Maße für die Diversität* definiert. Der *mittlere Abstand* der Individuen in der Population beträgt

$$\text{Divers}_{\text{Abstand},d}(P) = \frac{1}{s \cdot (s-1)} \cdot \sum_{1 \leq i, j \leq s} d(A^{(i)}.G, A^{(j)}.G),$$

wobei $d : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$ ein beliebiges Abstandsmaß ist. Die *Shannon-Entropie* als positionorientierte Diversität für $\mathcal{G} = \mathbb{B}^\ell$ ist definiert als

$$\text{Divers}_{\text{Entropie}}(P) = \frac{1}{\ell} \cdot \sum_{k=1}^{\ell} (-\#_0(P, k) \cdot \log(\#_0(P, k)) - \#_1(P, k) \cdot \log(\#_1(P, k))),$$

$$\text{mit } \#_1(P, k) = \frac{\#\{1 \leq i \leq s \mid A^{(i)}.G_k = 1\}}{s}$$

$$\text{und } \#_0(P, k) = \frac{\#\{1 \leq i \leq s \mid A^{(i)}.G_k = 0\}}{s}.$$

Die *teilstringorientierte Diversität* ist definiert als

$$\text{Divers}_{\text{Teilstring}}(P) = \frac{s \cdot \#\left(\bigcup_{1 \leq i \leq s} \text{Teil}(A^{(i)})\right)}{\sum_{1 \leq i \leq s} \#\text{Teil}(A^{(i)})},$$

$$\text{wobei } \text{Teil}(A) = \bigcup_{1 \leq i \leq j \leq \ell} \{A.G_i \dots A.G_j\}.$$

Je größer die jeweilige Maßzahl ist, desto größer ist die Vielfalt in der Population.



BEISPIEL

Für die Population $P_1 = \langle 0001, 0011, 1111 \rangle$ gilt:

$$Divers_{\text{Abstand},d}(P_1) = \frac{1}{6} \cdot (2 \cdot 1 + 2 \cdot 2 + 2 \cdot 3) = 2,0 \quad (\text{mit } d = d_{\text{ham}})$$

$$Divers_{\text{Entropie}}(P_1) = \frac{1}{4} \cdot \left((-1 \cdot \log 1 - 0) + 3 \cdot \left(-\frac{2}{3} \cdot \log \frac{2}{3} - \frac{1}{3} \cdot \log \frac{1}{3} \right) \right) \approx 0,4774$$

$$Divers_{\text{Teilstring}}(P_1) = \frac{3 \cdot 12}{7 + 8 + 4} \approx 1,895$$

da $\text{Teil}(0001) = \{0, 1, 00, 01, 000, 001, 0001\}$

$\text{Teil}(0011) = \{0, 1, 00, 01, 11, 001, 011, 0011\}$

$\text{Teil}(1111) = \{1, 11, 111, 1111\}$.

Für die Population $P_2 = \langle 0011, 0110, 1100 \rangle$ gilt:

$$Divers_{\text{Abstand},d}(P_2) = \frac{1}{6} \cdot (8 \cdot 2) \approx 2,667 \quad (\text{mit } d = d_{\text{ham}})$$

$$Divers_{\text{Entropie}}(P_2) = \frac{1}{4} \cdot \left(4 \cdot \left(-\frac{2}{3} \cdot \log \frac{2}{3} - \frac{1}{3} \cdot \log \frac{1}{3} \right) \right) \approx 0,6365$$

$$Divers_{\text{Teilstring}}(P_2) = \frac{3 \cdot 13}{8 + 8 + 8} \approx 1,625$$

da $\text{Teil}(0011) = \{0, 1, 00, 01, 11, 001, 011, 0011\}$

$\text{Teil}(0110) = \{0, 1, 01, 11, 10, 011, 110, 0110\}$

$\text{Teil}(1100) = \{0, 1, 11, 10, 00, 110, 100, 1100\}$.

Während die Population P_2 eine höhere Diversität hinsichtlich des Hamming-Abstands und der bitweise berechneten Entropie aufweist, ist P_1 diverser bezüglich der Teilstrings, da größere Unterschiede zwischen den Teilstrings der einzelnen Individuen bestehen.



ZUSAMMENFASSUNG

- Die Diversität ist keinesfalls eindeutig.
- Vielmehr gilt wie auch schon bei den Mutationsoperatoren, dass **das betrachtete Diversitätsmaß passend zum Optimierungsproblem gewählt werden muss.**
- So könnte etwa die Entropie für eine Instanz des Musterabgleichs passend sein, da die einzelnen Bits im Genotyp völlig unabhängig voneinander sind. Aber für das Handlungsreisendenproblem wäre etwa die teilstringorientierte Diversität interessant, da damit einzelne Abschnitte der Rundtouren unabhängig von ihrer Position beschrieben werden.
- Bei der Vielfalt in einer Population ist vor allem der Extremfall kritisch, bei dem alle Individuen gleich sind und damit die Population ihre möglichen Vorteile eingebüßt hat.



KONVERGIERTE POPULATION

Definition

Eine Population $P = (A^{(i)})_{1 \leq i \leq s}$ heißt *konvergiert*, wenn alle Individuen identisch sind, d. h. für alle $1 \leq i, j \leq s$ gilt $A^{(i)}.G = A^{(j)}.G$.

Bezüglich evolutionärer Algorithmen wird der Begriff der Konvergenz mit zwei unterschiedlichen Bedeutungen gebraucht. Einerseits kann wie bei der mathematischen Definition die Annäherung der Gütewerte an ein lokales oder globales Optimum gemeint sein - dann aber immer in endlicher Zeit. Andererseits kann damit der Verlust der Vielfalt in der Population bezeichnet werden.

Eine konvergierte Population ist ein Anzeichen dafür, dass die Optimierung beendet ist. Falls das globale Optimum nicht erreicht wurde, spricht man von vorzeitiger Konvergenz.



BEISPIEL

Die Population $P_3 = \langle 1111, 1111, 1111 \rangle$ ist konvergiert. Wie man leicht sieht, erreichen die Diversitätsmaße bei dieser Situation ihre minimalen Werte.

$$Divers_{\text{Abstand},d}(P_3) = 0,0 \quad (\text{mit } d = d_{\text{ham}})$$

$$Divers_{\text{Entropic}}(P_3) = \frac{1}{4} \cdot (4 \cdot (-1 \cdot \log 1 - 0)) = 0,0$$

$$Divers_{\text{Teilstring}}(P_3) = \frac{3 \cdot 4}{4+4+4} = 1,0, \text{ da } \text{Teil}(1111) = \{1, 11, 111, 1111\}.$$



EIN VERGLEICHENDES EXPERIMENT

- Ist die Population in der Lage, das Problem der lokalen Optima zu verkleinern? Betrachte BINÄRES-HILL-CLIMBING sowie eine populationsbasierte Variante POPULATIONSBASIERTES- BINÄRES-HILLCLIMBING, bei der für jedes Elternindividuum in der Population exakt ein Kindindividuum durch die Mutation erzeugt wird.
- Die anschließende Umweltselektion BESTEN-SELEKTION reduziert die Population auf die bessere Hälfte.
- Als Optimierungsgegenstand wählen wir die Rastrigin-Funktion, eine Benchmark-Funktion, die häufig zum Vergleich von Algorithmen herangezogen wird,

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

mit $n = 2$ und $5, 12 \leq X_1, X_2 \leq 5, 12$.



POPULATIONSBASIERTES-BINÄRES-HILLCLIMBING

POPULATIONSBASIERTES-BINÄRES-HILLCLIMBING(Zielfunktion F)

- 1 $t \leftarrow 0$
- 2 $P(t) \leftarrow$ erzeuge Population mit μ (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 $\lfloor P' \leftarrow P' \circ \langle B \rangle$
- 10 $t \leftarrow t + 1$
- 11 $\lfloor P(t) \leftarrow$ Selektion aus P' mittels BESTEN-SELEKTION
- 12 **return** bestes Individuum aus $P(t)$



BESTEN-SELEKTION

```
BESTEN-SELEKTION( Güterwerte  $\langle A.F^{(i)} \rangle_{i=1,\dots,r}$  )  
1   $I \leftarrow \langle \rangle$   
2  for  $j \leftarrow 1, \dots, s$  (Anzahl der zu wählenden Individuen)  
3  do  $\lceil$   $index_j \leftarrow$  derjenige Index aus  $\{1, \dots, r\} \setminus I$  mit dem besten Güterwert  
4      $\lfloor I \leftarrow I \cup \langle index_j \rangle$   
5  return  $I$ 
```



RASTRIGIN-FUNKTION

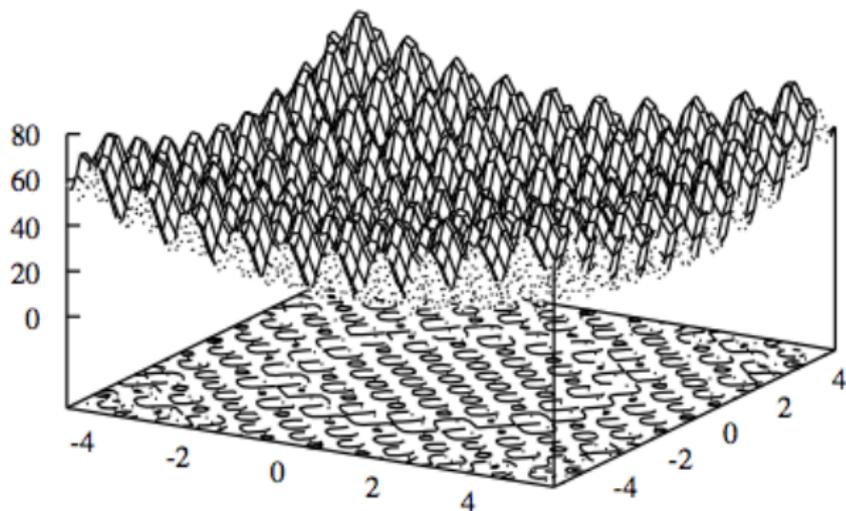


Abbildung 7: Rastrigin-Funktion für Dimension $n = 2$. Deutlich ist eine große Anzahl lokaler Minima zu erkennen. Die beiden Suchraumvariablen werden im Genotyp jeweils mit 16 Bits standardbinär kodiert.

EIN VERGLEICHENDES EXPERIMENT

- BINÄRES-HILLCLIMBING und POPULATIONSBASIERTES-BINÄRES-HILLCLIMBING wurden jeweils 100 mal auf die Rastrigin-Funktion angesetzt.
- Ersteres wurde nach 10.000 Iteration abgebrochen und zweiteres nach 200 Generationen mit einer Populationsgröße von 50 Individuen.
- So haben beide Algorithmen die gleiche Anzahl neuer Individuen bewertet.



EIN VERGLEICHENDES EXPERIMENT

- BINÄRES-HILLCLIMBING hat in 63% der Experimente das globale Optimum $f((0, 0)) = 0$ (im Rahmen der verfügbaren Genauigkeit) gefunden.
- POPULATIONSBASIERTES-BINÄRES-HILLCLIMBING hat in 76% der Experimente das Optimum gefunden.
- Bei Abbruch des Algorithmus hat die durchschnittliche Güte über alle Experimente 0,479 beim Hillclimber und 0,297 bei dem populationsbasierten Hillclimber betragen.



OPTIMIERUNG DURCH POPULATIONSBASIERTES-BINÄRES-HILLCLIMBING

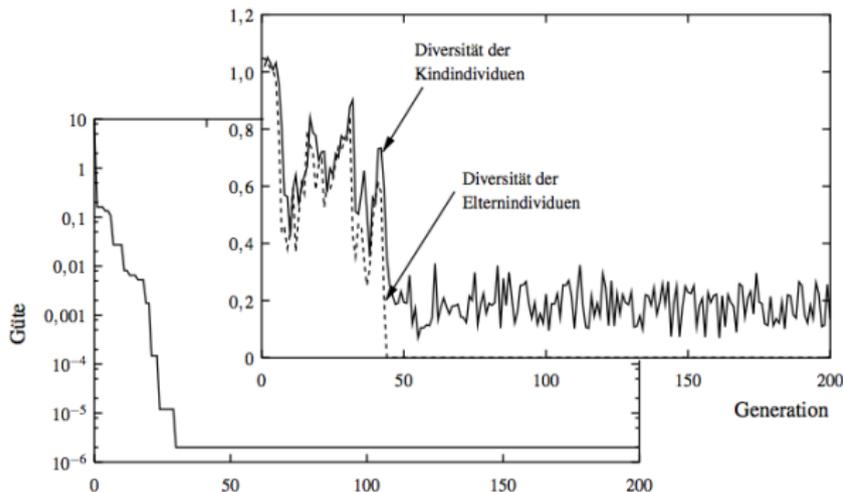


Abbildung 8: Die Optimierung mit dem populationsbasierten binären Hillclimber konvergiert zehn Generationen nach der letzten gefundenen besten Güte. Die Kindindividuen besitzen eine gewisse Grunddiversität.

OPTIMIERUNG DURCH POPULATIONSBASIERTES-BINÄRES-HILLCLIMBING

- Nach Generation 35 wird der finale Gütewert erreicht und etwa 10 Generationen später ist die Population konvergiert.
- Auch nach der Konvergenz erzeugt die Mutation einen gewissen Pegel an Grunddiversität der Kindindividuen, welche jedoch nicht mehr zu einer Verbesserung führt.



FOLGERUNGEN FÜR DIE SELEKTION

- Aus der gleichzeitigen Betrachtung mehrerer Individuen ergeben sich verschiedene Anforderungen an die Selektionsoperatoren.
- Bei einer Elternselektion sollten alle Individuen eine Chance haben, ausgewählt zu werden, da andernfalls der Aufwand für die Verwaltung einer großen Population nicht gerechtfertigt ist.
- Grundsätzlich gibt es zwei gängige Möglichkeiten, dies zu gewährleisten, nämlich
 - indem jedes Individuum Elter für genau $m > 0$ Kinder wird oder
 - indem jedes Individuum mit einer individuellen Wahrscheinlichkeit als Elter gewählt wird.



FOLGERUNGEN FÜR DIE SELEKTION

- Im ersten Fall entsteht kein Selektionsdruck, da alle Individuen gleich behandelt werden.
- Im zweiten Fall kann durch die Vergabe der Auswahlwahrscheinlichkeiten der Selektion eine Richtung gegeben werden.



FOLGERUNGEN FÜR DIE SELEKTION

- Die Umweltselektion hat die Aufgabe, aus den vorhandenen Individuen die Population der nächsten Elternindividuen zusammenzustellen.
- Dabei soll sowohl eine möglichst große Vielfalt erhalten bleiben, aber auch die tatsächlich besseren Individuen aufgenommen werden.
- Diese beiden Ziele können sich widersprechen, sodass in einigen Fällen eine reine Auswahl der besten Individuen dem Erhalt der Vielfalt nicht gerecht wird.



FOLGERUNGEN FÜR DIE SELEKTION

- Dies gilt insbesondere dann, wenn durch die Mutation und die Rekombination auch unveränderte Kopien von Individuen entstehen können, was zu einer raschen Konvergenz der Population führen kann. Auch hier gibt es zwei Ansätze, mit diesen Anforderungen umzugehen:
 - die reine Auswahl der besten Individuen und
 - die zufällige Auswahl, wobei bessere Individuen eine höhere Wahrscheinlichkeit haben und jedes Individuum nur einmal gewählt werden kann.



EIGENSCHAFTEN DER SELEKTION

Definition

Ein durch die Indexselektion $IS^\xi: \mathbb{R}^r \rightarrow \{1, \dots, r\}^s$ definierter Selektionsoperator heißt

- *deterministisch*, falls $\forall x \in \mathbb{R}^r \forall \xi, \xi' \in \Xi. IS^\xi(x) = IS^{\xi'}(x)$,
- *probabilistisch* genau dann, wenn er nicht deterministisch ist,
- *duplikatfrei*, falls
 $\forall x \in \mathbb{R}^r \forall \xi \in \Xi \forall 1 \leq i < j \leq s. (IS^\xi(x))_i \neq (IS^\xi(x))_j$.



DUPLIKATFREIHEIT

- Wird in der Regel von Operatoren der Umweltselektion verlangt, um die Diversität möglichst hoch zu halten.
- Dabei wird allerdings nur verhindert, dass ein Individuum über seinen Index mehrfach gewählt wird.
- Schon vorhandene Duplikate in der Population können mehrfach ausgewählt werden.
- Bei der Elternselektion ist dies nicht so bedeutend, da die mehrfach gewählten Individuen direkt in die Erzeugung neuer Individuen eingehen.



SELEKTIONSSTÄRKE

WIEDERHOLUNG

- Als theoretische Grundlage für den Vergleich von Selektionsmechanismen und damit auch für die Wahl eines geeigneten Selektionsmechanismus für einen Algorithmus existieren verschiedene Maße für den erzeugten Selektionsdruck.
- Ein Maß ist die Übernahmzeit, d. h. die Anzahl der Generationen bis die Population konvergiert ist.
- Ein zweites Maß, auf das im Weiteren noch näher eingegangen wird, ist die Selektionsintensität, die durch das Selektionsdifferenzial zwischen der durchschnittlichen Güte vor und nach der Selektion bestimmt wird.



SELEKTIONSINTENSITÄT

Definition

Sei $(\Omega, f, >)$ das betrachtete Optimierungsproblem und werde ein Selektionsoperator $\text{Sel}^\xi : (G \times Z \times \mathbb{R})^r \rightarrow (G \times Z \times \mathbb{R})^s$ auf eine Population P mit durchschnittlicher Güte F und Standardabweichung σ der Gütewerte angewandt. Dann sei \bar{F}_{sel} die durchschnittliche Güte der Population $\text{Sel}^\xi(P)$ und der Selektionsoperator besitzt die **Selektionsintensität**

$$\text{Intensität} = \begin{cases} \frac{\bar{F}_{\text{sel}} - \bar{F}}{\sigma} & \text{falls Maximierungsproblem} \\ \frac{\bar{F} - \bar{F}_{\text{sel}}}{\sigma} & \text{sonst.} \end{cases}$$



PROBABILISTISCHE ELTERNSELEKTION

- Soll bei der Wahl der Eltern ein Selektionsdruck erzeugt werden, kann dies durch eine zufällige Selektion geschehen, bei der die Wahlwahrscheinlichkeit proportional zur Güte ist.
- In der natürlichen Evolution wurde die Stärke eines Individuums indirekt durch die Anzahl seiner Nachkommen gemessen und als **Fitness** bezeichnet.
- Bei der proportionalen probabilistischen Selektion kann wiederum für jedes Individuum ein Wert vorgegeben werden, der annähernd bestimmt, wie groß die Fruchtbarkeit des Individuums und damit die Anzahl seiner Nachkommen ist.



PROBABILISTISCHE ELTERNSELEKTION

- Angenommen ein Maximierungsproblem liegt vor und die Fitnesswerte entsprechen den Gütewerten. Dann nutzt die fitnessproportionalen Selektion die folgende Auswahlwahrscheinlichkeit für die Individuen $A^{(i)} (1 \leq i \leq r)$:

$$Pr[A^{(i)}] = \frac{A^{(i)} \cdot F}{\sum_{k=1}^r A^{(k)} \cdot F}$$



FITNESSPROPORTIONALE-SELEKTION

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

```
1  Summe0 ← 0
2  for  $i \leftarrow 1, \dots, r$ 
3  do  $\lceil$  Fitness ← berechne Fitnesswert aus  $A^{(i)}.F$ 
4      $\lfloor$  Summe $i$  ← Summe $i-1$  + Fitness
5   $I \leftarrow \langle \rangle$ 
6  for  $i \leftarrow 1, \dots, s$  (Anzahl der zu wählenden Individuen)
7  do  $\lceil$   $j \leftarrow 1$ 
8      $u \leftarrow$  wähle Zufallszahl gemäß  $U([0, \textit{Summe}_r])$ 
9     while Summe $j$  <  $u$ 
10    do  $\lfloor$   $j \leftarrow j + 1$ 
11     $\lfloor I \leftarrow I \circ \langle j \rangle$ 
12 return  $I$ 
```



INTERPOLIERENDE OPERATOREN

- fügen Eigenschaften der Eltern zusammen, sodass die Eigenschaften des neuen Individuums zwischen denen der Eltern liegt
- dies führt zu geringerer Durchforstung von Ω
- interpolierende Rekombination konzentriert Population auf 1 Schwerpunkt
- fördert damit Feinabstimmung von sehr guten Individuen
- um Ω anfangs genügend zu erforschen: Verwenden einer stark zufallsbasierten, diversitätserhaltenden Mutation



ARITHMETISCHER-CROSSOVER

- ist Beispiel für interpolierende Rekombination
- arbeitet auf reellwertige Genotypen
- geometrisch: kann alle Punkte auf Strecke zwischen beiden Eltern erzeugen

Input: Individuen A, B mit $A.G, B.G \in \mathbb{R}^l$

Output: neues Individuum C

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



EXTRAPOLIERENDE OPERATOREN

- Versuchen gezielt Information aus mehreren Individuen abzuleiten
- Erstellen eine Prognose, wo Güteverbesserungen zu erwarten sind
- Extrapolierende Rekombination kann bisherigen Ω verlassen
- Ist einzige Art der Rekombination die Gütewerte benutzt
- Einfluss der Diversität ist hier schwer nachzuvollziehen
- Algorithmus: z.B. Arithmetisches Crossover mit $u \in U([1, 2])$



KLASSISCHER GENETISCHER-ALGORITHMUS

GENETISCHER-ALGORITHMUS(Zielfunktion F)

```
1   $t \leftarrow 0$ 
2   $P(t) \leftarrow$  erzeuge Population mit  $\mu$  (gerade Populationsgröße) Individuen
3  bewerte  $P(t)$  durch  $F$ 
4  while Terminierungsbedingung nicht erfüllt
5  do  $P' \leftarrow$  Selektion aus  $P(t)$  mittels SELEKTION-FITNESSPROPORTIONAL
6       $\{$  Es sei:  $P' = \langle A^{(1)}, \dots, A^{(\mu)} \rangle$   $\}$ 
7       $P'' \leftarrow \langle \rangle$ 
8      for  $i \leftarrow 1, \dots, \frac{\mu}{2}$ 
9          do  $u \leftarrow$  wähle Zufallszahl gemäß  $U([0, 1])$ 
10             if  $u \leq p_x$  (Rekombinationswahrscheinlichkeit)
11                 then  $B, C \leftarrow$  EIN-PUNKT-CROSSOVER( $A^{(2i-1)}, A^{(2i)}$ )
12                 else  $B \leftarrow A^{(2i-1)}$ 
13                      $C \leftarrow A^{(2i)}$ 
14                  $B \leftarrow$  BINÄRE-MUTATION( $B$ )
15                  $C \leftarrow$  BINÄRE-MUTATION( $C$ )
16                  $P'' \leftarrow P'' \circ \langle B, C \rangle$ 
17             bewerte  $P''$  durch  $F$ 
18              $t \leftarrow t + 1$ 
19              $P(t) \leftarrow P''$ 
20 return bestes Individuum aus  $P(t)$ 
```



BEISPIEL

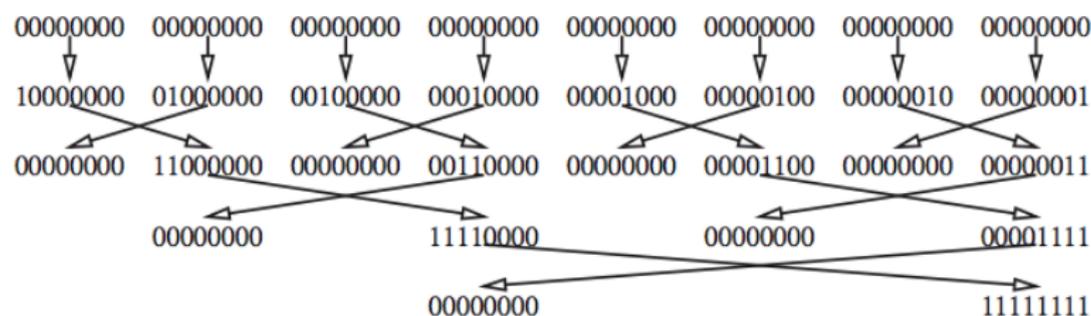
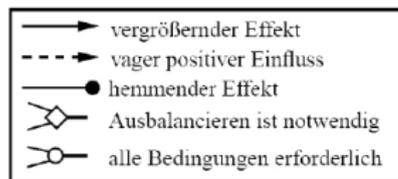
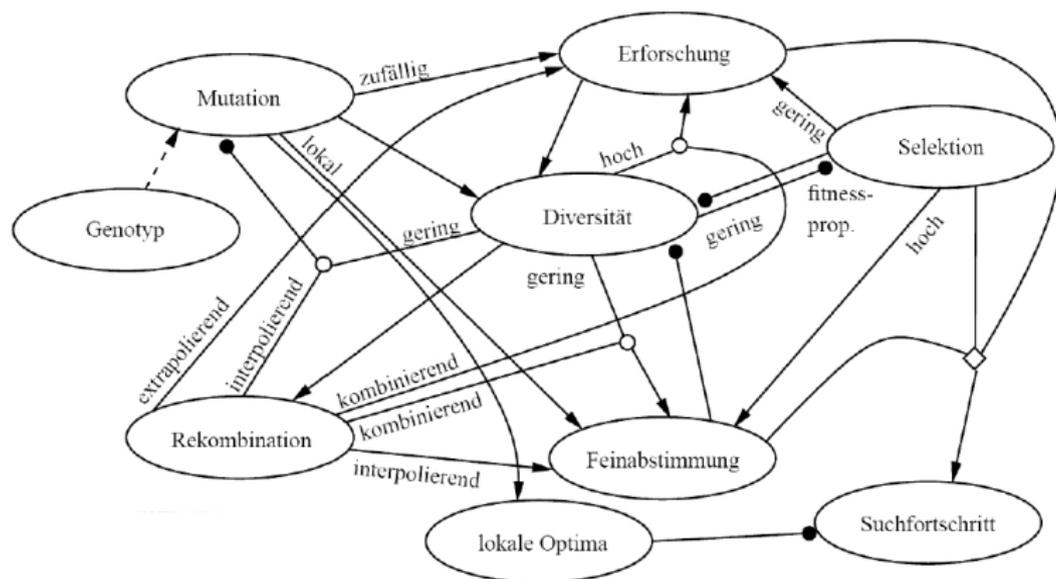


Abbildung 9: Für das Einsenzählproblem kann sich auch aus der schlechtestmöglichen Population (nur mit Nullen belegte Individuen) durch geschickte Mutationen in der ersten Generation und logarithmisch viele Iterationen mit passenden Rekombinationen das Optimum bilden.

ZUSAMMENFASSUNG



ZUSAMMENHÄNGE

Bedingung	Zielgröße	Erwarteter Effekt
Genotyp	Mutation	Nachbarschaft des Mutationsoperators wird beeinflusst
Mutation	Erforschung	zufällige Mutationen unterstützen Erforschung
Mutation	Feinabst.	gütelokale Mutationen unterstützen Feinabstimmung
Mutation	Diversität	Mutation vergrößert Diversität
Mutation	lokale Optima	gütelokale Mutationen erhalten lokale Optima des Phänotyps (zufällige Mutationen können noch mehr einführen)
Rekombination	Erforschung	extrapolierende Operatoren stärken Erforschung
Rekombination	Feinabst.	interpolierende Operatoren stören Feinabstimmung

ZUSAMMENHÄNGE

Bedingung	Zielgröße	Erwarteter Effekt
Div./Rekomb.	Mutation	geringe Diversität und interpolierende Rekombination dämpfen Ausreißer der Mutation
Diversität	Rekombination	hohe Diversität unterstützt Funktionsweise der Rekombination
Selektion	Erforschung	geringer Selektionsdruck stärkt Erforschung
Selektion	Feinabst.	hoher Selektionsdruck stärkt Feinabstimmung
Selektion	Diversität	Selektion verringert meist Diversität
Div./Rekomb.	Erforschung	kombinierende Rekombination stärkt Erforschung bei hoher Diversität
Div./Rekomb.	Feinabst.	kombinierende Rekombination stärkt Feinabstimmung bei hoher Diversität



ZUSAMMENHÄNGE

Bedingung	Zielgröße	Erwarteter Effekt
Erforschung	Diversität	erforschende Operationen erhöhen Diversität
Feinabst.	Diversität	feinabstimmende Operationen verringern Diversität
Diversität	Selektion	geringe Diversität verringert Selektionsdruck der fitnessproportionalen Selektion
lokale Optima	Suchfortschritt	viele lokale Optima hemmen Suchfortschritt
Erf./Fein./Sel.	Suchfortschritt	Ausbalancieren der drei Faktoren ist notwendig

