Evolutionäre Algorithmen: Weitere Standardalgorithmen

Karsten Weicker

HTWK Leipzig

5. Januar 2013

Überblick

1. Klassifizierende Systeme
2. Tabu-Suche
3. Memetische Algorithmen
4. Populationsbasiertes inkrementelles Lernen
5. Differentialevolution
6. Scatter Search
7. Kulturelle Algorithmen
8. Ameisenkolonien
9. Partikelschwärme

Überblick

1. Klassifizierende Systeme
2. Tabu-Suche
3. Memetische Algorithmen
4. Populationsbasiertes inkrementelles Lernen
5. Differentialevolution
6. Scatter Search
7. Kulturelle Algorithmen
8. Ameisenkolonien
9. Partikelschwärme

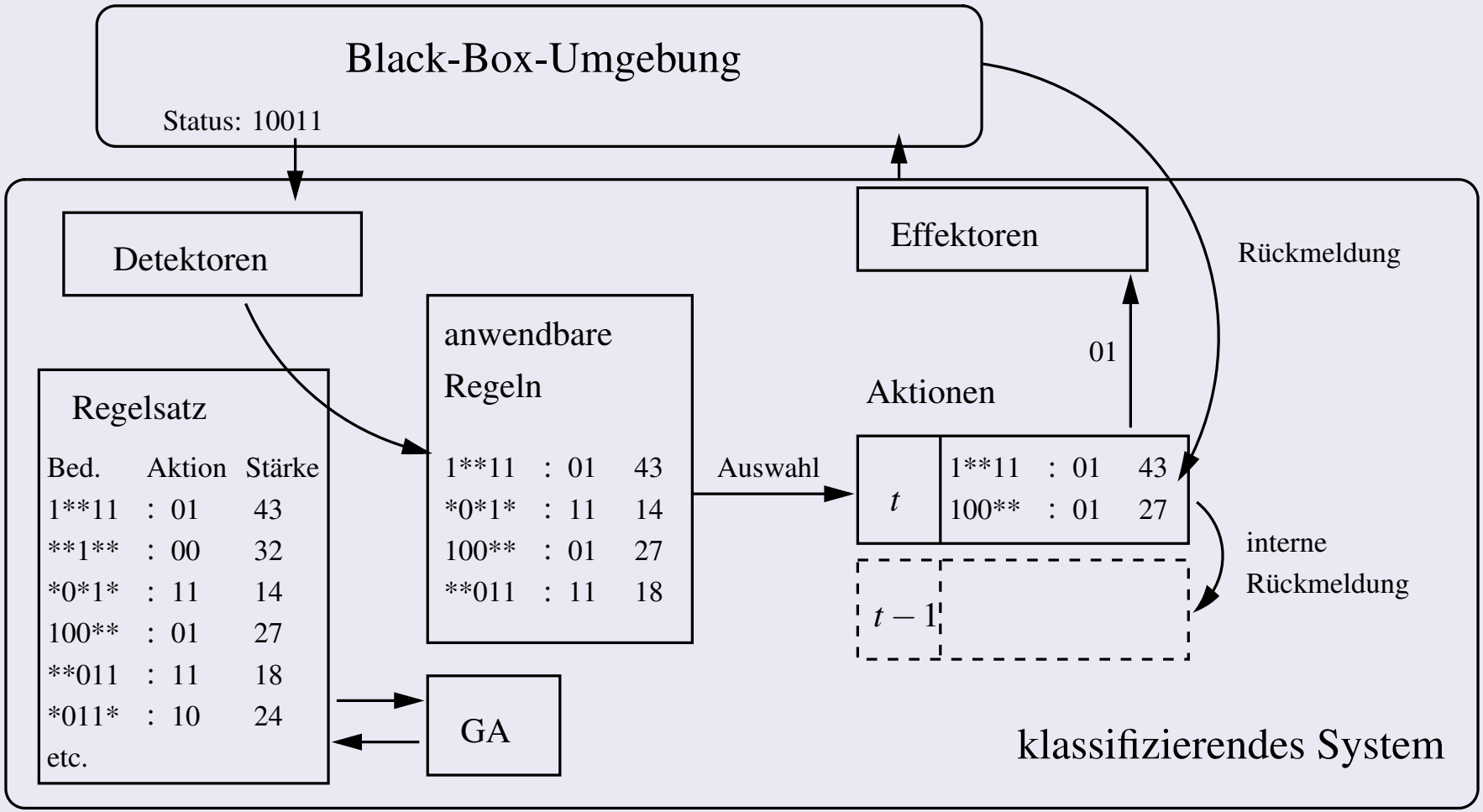
Klassifizierende Systeme

Situation

- technisches System soll geregelt werden
- Detektoren beobachten den Zustand des Systems
- Effektoren beeinflussen den Zustand
- gelegentliche kommt eine Rückmeldung, wie gut das System geregelt wird

Klassifizierende Systeme

Grundidee



Klassifizierende Systeme

Definition: Regeln

- Regel $reg = (reg.b, reg.a, reg.s) \in Bedingung \times Aktion \times \mathbb{R}$ mit
  - Bedingung $reg.b \in Bedingung = \{0, 1, *\}^l$
  - Aktion $reg.a \in Aktion = \{0, 1\}^m$
  - Stärke $reg.s \in \mathbb{R}$
- Regelsatz: $regeln \subset Bedingung \times Aktion \times \mathbb{R}$

Klassifizierende Systeme

Definition: Anwendbare Regeln

- Regel aus *regeln* ist anwendbar unter Statusmeldung $\nu \in \mathit{Status} = \{0,1\}^l$, wenn die Bedingung passt.

$$\mathit{aktiv}(\nu) = \left\{ \mathit{reg} \in \mathit{regeln} \;\middle|\; \bigwedge_{i=1}^{l} (\mathit{reg}.b_i \neq * \;\Rightarrow\; \mathit{reg}.b_i = \nu_i) \right\}.$$

Klassifizierende Systeme

Definition: Auswahlwahrscheinlichkeit

- aktive Regeln $aktiv(\nu)$ liegen vor
- mögliche Reaktionen:
  $aktion(\nu) = \{reg.a \mid reg \in aktiv(\nu)\}$
- anwendbare Regeln für $x \in aktion(\nu)$:
  $aktiv'(\nu, x) = \{reg \in aktiv \mid reg.a = x\}$
- fitnessproportionale Auswahlwahrscheinlichkeit:

$$Pr[x|\nu] = \frac{\sum_{reg \in aktiv'(\nu,x)} reg.s}{\sum_{reg \in aktiv(\nu)} reg.s}$$

Klassifizierende Systeme

Definition: Modifikation der Stärke

- ausgeführte Regeln zur Zeit t: $ausgef^{(t)}$
- Rückmeldung vom System: $rück \in \mathbb{R}$
- Modifikation von $reg \in ausgef^{(t)}$:

$$reg.s \leftarrow (1-\alpha) \cdot reg.s + \alpha \cdot \frac{rück}{\#ausgef^{(t)}}$$

- Modifikation von $reg \in aktiv(\nu) \setminus ausgef^{(t)}$:

$$reg.s \leftarrow (1-\tau) \cdot reg.s$$

Klassifizierende Systeme

Definition: Modifikation der Stärke (Forts.)

- Modifikation der letzten Regeln $reg \in ausgef^{(t-1)}$:

$$reg.s \leftarrow reg.s + \alpha \cdot \beta \cdot \frac{\sum_{reg' \in ausgef^{(t)}} reg'.s}{\#ausgef^{(t-1)}}$$

Klassifizierende Systeme

Rolle des GA

- GA erzeugt neue Regeln im Regelsatz
- steady-state (nur zwei neue Individuen)
- mehrere Lernschritte wechseln mit einer „GA-Generation“
- Auswahl proportional zu Stärkewerten
- Crossover und Mutation, neue Stärke als Durchschnitt der elterlichen Stärke
- proportional zur inversen Stärke gewählte Individuen werden ersetzt

Klassifizierende Systeme

Typische Parameter

Parameter	Wertebereich
Populationsgröße:	400–5000
Lernrate α:	0.2
Dämpfungsfaktor β:	0.71
Straffaktor τ:	0.1

Klassifizierende Systeme

Alternativen

- wesentliche bessere Mechanismen in Verfahren wie XCS
- Pittsburgh-CS (statt Michigan-CS):
  - Individuum = Regelsatz
  - Bewertung durch Simulation des zu regelnden Systems

Überblick

1 Klassifizierende Systeme

2 Tabu-Suche

3 Memetische Algorithmen

4 Populationsbasiertes inkrementelles Lernen

5 Differentialevolution

6 Scatter Search

7 Kulturelle Algorithmen

8 Ameisenkolonien

9 Partikelschwärme

Tabu-Suche

Grundlegendes

- lokales Sucheverfahren
- ähnelt dem Ablauf einer $(1, \lambda)$-Evolutionsstrategie
- bei der Erzeugung neuer Individuen wird wird die „Geschichte“ berücksichtigt
- sog. *Tabu-Listen* verhindert Rückkehr zu gerade betrachteten Lösungskandidaten

Tabu-Suche

Technik

- 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 ein Tabu brechen
- Auch möglich: neue beste Gesamtgüte bricht Tabu

Tabu-Suche

Beispiel

- Graphfärbung mit k Farben, sodass keine Kante zwischen gleich gefärbten Knoten ist
  - $\Omega = \{1, \ldots, k\}^n$
  - $$f(x) = \sum_{(v_i, v_j) \in E} \begin{cases} 1 & \text{falls } x_i = x_j \\ 0 & \text{sonst} \end{cases} \quad \text{wird minimiert}$$
  - Mutation färbt v_i von c auf d
    $\Rightarrow$ Tabu-Eintrag (i, c)

Tabu-Suche

```
TABU-SUCHE( Zielfunktion F )
 1  t ← 0
 2  A(t) ← erzeuge zufälligen Lösungskandidaten
 3  bewerte A(t) durch F
 4  bestInd ← A(t)
 5  initialisiere Tabu − Liste
 6  while  Terminierungsbedingung nicht erfüllt
 7  do ⌜ P ← ⟨⟩
 8        while #P < λ
 9        do ⌜ B ← MUTATION(A(t))
10              bewerte B durch F
11              if (A(t), B) ∉ Tabu − Liste oder B.F ≻ bestInd.F
12           ⌞ then ⌊ P ← P ∘ ⟨B⟩
13        t ← t + 1
14        A(t) ←  bestes Individuum aus P
15        if A(t).F ≻ bestInd.F
16        then ⌊ bestInd ← A(t)
17     ⌞ Tabu − Liste ← aktualisiere durch (A(t − 1), A(t))
18  return bestInd
```

Überblick

1 Klassifizierende Systeme

2 Tabu-Suche

3 Memetische Algorithmen

4 Populationsbasiertes inkrementelles Lernen

5 Differentialevolution

6 Scatter Search

7 Kulturelle Algorithmen

8 Ameisenkolonien

9 Partikelschwärme

Memetische Algorithmen

Motivation

- populationsbasierte Algorithmen
  - \+ breites Durchforsten des Suchraums
  - – langsam
- lokale Suche
  - \+ schnelle Optimierung
  - – anfällig für lokale Optima

Memetische Algorithmen

Motivation

- populationsbasierte Algorithmen
  - \+ breites Durchforsten des Suchraums
  - – langsam
- lokale Suche
  - \+ schnelle Optimierung
  - – anfällig für lokale Optima

Grundidee

- Kombination beider Techniken

Memetische Algorithmen

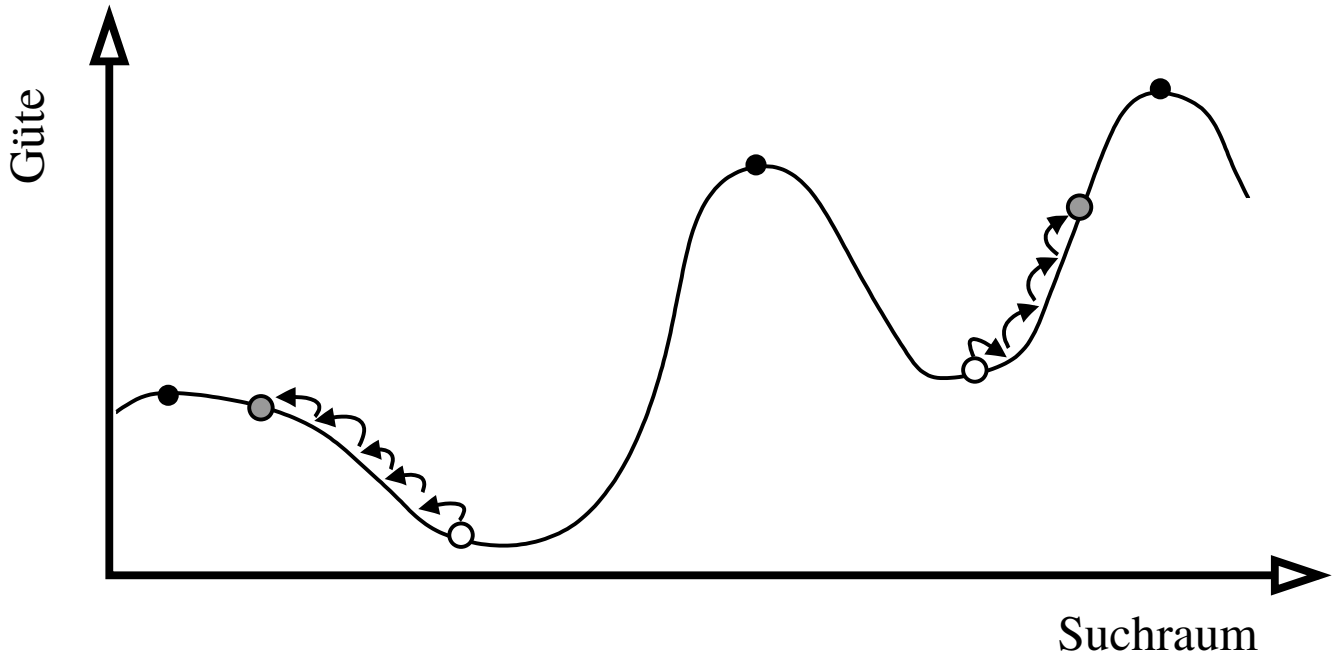
Herkunft des Namens

- Richard Dawkins: „Meme“ sind Verhaltenselemente, die individuell erworben werden können (im Gegensatz zu Genen)

Konkretes Vorgehen

- jedes neue Individuum wird sofort lokal optimiert
  - nur wenige Schritte oder
  - bis in das lokale Optimum

Memetische Algorithmen



Beispiel: SAGA

Genetischer Algorithmus mit Simulated Annealing

Memetische Algorithmen

MEMETISCHER-ALGORITHMUS(Bewertungsfunktion F)

```
 1  t ← 0
 2  P(t) ← initialisiere Population der Größe μ
 3  P(t) ← LOKALE-SUCHE(F) für jedes Individuum in P(t)
 4  bewerte P(t) durch F
 5  while  Terminierungsbedingung nicht erfüllt
 6  do ⌜ E ← selektiere Eltern für λ Nachkommen aus P(t)
 7        P' ← erzeuge Nachkommen durch Rekombination aus E
 8        P'' ← mutiere die Individuen in P'
 9        P''' ← LOKALE-SUCHE(F) für jedes Individuum in P''
10        bewerte P''' durch F
11        t ← t + 1
12     ⌞ P(t) ← Umweltselektion auf P'''
13  return bestes Individuum aus P(t)
```

Memetische Algorithmen

Eigenschaften

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

Überblick

1 Klassifizierende Systeme

2 Tabu-Suche

3 Memetische Algorithmen

4 Populationsbasiertes inkrementelles Lernen

5 Differentialevolution

6 Scatter Search

7 Kulturelle Algorithmen

8 Ameisenkolonien

9 Partikelschwärme

PBIL

Grundidee

- ein genetischer Algorithmus ohne Population
- stattdessen: nur Populationsstatistik speichern
  $\Rightarrow$ bei $\mathcal{G} = \mathbb{B}^l$ für alle l Bits die Häufigkeit der „1“
- konkrete Individuen (z.B. für die Bewertung) werden zufällig gemäß der Häufigkeiten erzeugt

PBIL

Operatoren

- Rekombination: uniformer Crossover
  $\Rightarrow$ implizit bei der Erzeugung eines Individuums
- Selektion: Wahl des besten Individuums
  $\Rightarrow$ zur Aktualisierung des Populationsstatistik
  $Prob_k^{(t)} \leftarrow B_k \cdot \alpha$ (Lernrate) $+ Prob_k^{(t-1)} \cdot (1 - \alpha)$
- Mutation: Bitflipping
  $\Rightarrow$ leichte zufällige Verschiebung der Populationsstatistik

PBIL

PBIL(Bewertungsfunktion F)

1 $t \leftarrow 0$
2 $bestInd \leftarrow$ erzeuge ein zufälliges Individuum aus $\mathcal{G} = \mathbb{B}^l$
3 $Prob^{(0)} \leftarrow (0.5, \ldots, 0.5) \quad \in [0,\ 1]^l$
4 **while** Terminierungsbedingung nicht erfüllt
5 **do** $\ulcorner P \leftarrow \langle\rangle$
6 **for** $i \leftarrow 1, \ldots, \lambda$
7 **do** $\ulcorner A \leftarrow$ erzeuge Individuum aus $\mathbb{B}^l$ gemäß $Prob^{(t)}$
8 $\llcorner P \leftarrow P \circ \langle A \rangle$
9 bewerte P durch F
10 $\langle B \rangle \leftarrow$ Selektion aus P mittels Besten-Selektion
11 **if** $F(B) \succ F(bestInd)$
12 **then** $\llbracket bestInd \leftarrow B$
13 $t \leftarrow t + 1$
14 **for each** $k \in \{1, \ldots, l\}$
15 **do** $\llbracket Prob_k^{(t)} \leftarrow B_k \cdot \alpha$ (Lernrate) $+ Prob_k^{(t-1)} \cdot (1 - \alpha)$
16 **for each** $k \in \{1, \ldots, l\}$
17 **do** $\ulcorner u \leftarrow$ wähle Zufallszahl gemäß $U((0,\ 1])$
18 **if** $u < p_m$ (Mutationswahrscheinlichkeit)
19 **then** $\ulcorner u' \leftarrow$ wähle Zufallszahl gemäß $U(\{0,\ 1\})$
20 $\llcorner \quad \llcorner \quad \llcorner Prob_k^{(t)} \leftarrow u' \cdot \beta$ (Mutationskonstante) $+ Prob_k^{(t)} \cdot (1 - \beta)$
21 **return** $bestInd$

PBIL

Probleme

- genetischer Algorithmus kann Abhängigkeit zwischen einzelnen Bits lernen
- PBIL betrachtet die einzelnen Bits isoliert

Beispiel

Population 1					Population 2			
1	1	0	0	Individuum 1	1	0	1	0
1	1	0	0	Individuum 2	0	1	1	0
0	0	1	1	Individuum 3	0	1	0	1
0	0	1	1	Individuum 4	1	0	0	1
0.5	0.5	0.5	0.5	Populationsstatistik	0.5	0.5	0.5	0.5

PBIL

Lernrate

- niedrig: betont Erforschung
- hoch: betont Feinabstimmung

Parameter	Wertebereich
Populationsgröße λ :	20–100
Lernrate α:	0.05–0.2
Mutationsrate p_m:	0.001–0.02
Mutationskonstante β:	0.05

PBIL

Alternative Verfahren

- bessere Techniken zur Schätzung der Verteilung guter Lösungskandidaten
- insbesondere: interne Abhängigkeiten modellieren – z.B: durch Bayes-Netze
- Beispiel: BOA etc.

Überblick

1 Klassifizierende Systeme

2 Tabu-Suche

3 Memetische Algorithmen

4 Populationsbasiertes inkrementelles Lernen

5 Differentialevolution

6 Scatter Search

7 Kulturelle Algorithmen

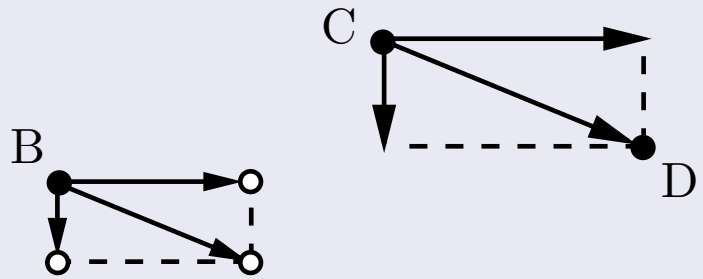
8 Ameisenkolonien

9 Partikelschwärme

Differentialevolution

Idee

- ähnlich zur Evolutionsstrategie
- keine Anpassung der Schrittweite in $A.S$
- sondern: Relation der Individuen in der Population wird als Grundlage für die Schrittweite herangezogen



Differentialevolution

DE-Operator(Individuen A, B, C, D)

1 $index \leftarrow$ wähle Zufallszahl gemäß $U(\{1, \ldots, I\})$
2 **for each** $i \in \{1, \ldots, I\}$
3 **do** $\ulcorner$ $u \leftarrow$ wähle Zufallszahl gemäß $U([0,\ 1))$
4 **if** $u \leq \tau$ (Wichtung der Rekombination) oder $i = index$
5 **then** $\llcorner$ $A'_i \leftarrow B_i + (C_i - D_i) \cdot \alpha$ (Skalierungsfaktor)
6 **else** $\llcorner$ $A'_i \leftarrow A_i$
7 **return** A'

Differentialevolution

Details

- DE-Operator: Kombination aus Rekombination und Mutation
- Selektion: ein neues Individuum ersetzt das Elternindividuum genau dann, wenn es besser ist

Parameter	Wertebereich
Populationsgröße μ:	10–100, 10·n
Wichtung der Rekombination τ:	0.7–0.9
Skalierungsfaktor α:	0.5–1.0

Differentialevolution

DIFFERENTIALEVOLUTION(Bewertungsfunktion F)

```
 1  t ← 0
 2  P(t) ← erzeuge Population der Größe μ
 3  bewerte P(t) durch F
 4  while  Terminierungsbedingung nicht erfüllt
 5  do ⌜ P(t+1) ← ⟨⟩
 6        for i ← 1, ..., μ
 7        do ⌜ repeat ⌊ A, B, C, D ← selektiere Eltern uniform zufällig aus P(t)
 8                 until A, B, C, D sind paarweise verschieden
 9                 A' ← DE-Operator(A, B, C, D)
10                 bewerte A' durch F
11                 if F(A') ⪰ F(A)
12                 then ⌊ P(t+1) ← P(t+1) ∘ ⟨A'⟩
13              ⌞ else  ⌊ P(t+1) ← P(t+1) ∘ ⟨A⟩
14        ⌞ t ← t+1
15  return bestes Individuum aus P(t)
```

Überblick

1 Klassifizierende Systeme

2 Tabu-Suche

3 Memetische Algorithmen

4 Populationsbasiertes inkrementelles Lernen

5 Differentialevolution

6 Scatter Search

7 Kulturelle Algorithmen

8 Ameisenkolonien

9 Partikelschwärme

Scatter Search

Zutaten

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

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

Scatter Search

Ablauf

- iterierter Ablauf durch zwei Phasen
  - Erzeugen neuer Individuen und Auswahl derjenigen, die die größte Vielfalt garantieren
  - Rekombinieren aller „Paarungen“ der gewählten Individuen
    Selektion der Besten und Iteration bis sich nichts mehr ändert

Beispiel

reellwertige Problemräume mit $\mathcal{G} = \Omega = \mathbb{R}^n$

Scatter Search

Phase 1

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

Scatter Search

Phase 2

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

```
SCATTER-SEARCH( Bewertungsfunktion F )
 1  bestP = ⟨⟩
 2  P ← ⟨⟩
 3  for t ← 1, ..., maxIter
 4  do ⌜ while #P < μ
 5        do ⌜ A ← erzeuge ein Individuum mit einem Diversitätsgenerator
 6           ⌞ A ← LOKALE-SUCHE(F) angewandt auf A
 7           bewerte A durch F
 8           if A ∉ P ∘ bestP
 9        ⌞ then ⌊ P ← P ∘ ⟨A⟩
10        if t = 1
11        then ⌜ bestP ← selektiere α Individuen aus P mit BESTEN-SELEKTION
12             ⌞ P ← streiche Individuen aus bestP in P
13        for k ← 1, ..., β
14        do ⌜ A ← dasjenige Individuum aus P das min_{B∈bestP} d(A.G, B.G) maximiert
15           ⌞ P ← streiche Individuum A in P
16        ⌞ bestP ← bestP ∘ ⟨A⟩
17        repeat ⌜ P' ← ⟨⟩
18               Mengen ← erzeuge Teilmengen von bestP durch einen Teilmengengenerator
19               for each M ∈ Mengen
20               do ⌜ A ← wende einen Kombinationsoperator auf M an
21                  A ← LOKALE-SUCHE(F) angewandt auf A
22                  bewerte A durch F
23                  if A ∉ bestP ∪ P'
24               ⌞ then ⌊ P' ← P' ∘ ⟨A⟩
25             ⌞ bestP ← selektiere α + β Ind. aus bestP ∘ P' mit BESTEN-SELEKTION
26        until bestP hat sich nicht geändert
27     ⌞ bestP ← selektiere α Individuen aus P mit BESTEN-SELEKTION
28  return bestes Individuum aus bestP
```

Scatter Search

Empfohlene Parameter

Parameter	Wertebereich
Populationsgröße μ:	50–150
Anzahl der besten Individuen α:	5–20
Erweiterung der besten Individuen β:	5–20

Überblick

1 Klassifizierende Systeme

2 Tabu-Suche

3 Memetische Algorithmen

4 Populationsbasiertes inkrementelles Lernen

5 Differentialevolution

6 Scatter Search

7 Kulturelle Algorithmen

8 Ameisenkolonien

9 Partikelschwärme

Kulturelle Algorithmen

Motivation

- weiterer Informationsspeicher zusätzlich zur genetischen Ebene
- Kultur bestimmt auf Werten gestütztes Verhalten
- Ebene der Kultur wird in EAs eingeführt

Kulturelle Algorithmen

Motivation

- weiterer Informationsspeicher zusätzlich zur genetischen Ebene
- Kultur bestimmt auf Werten gestütztes Verhalten
- Ebene der Kultur wird in EAs eingeführt

Kollektives kulturelles Wissen

- Speicher: Überzeugungsraum (engl. *belief space*)
- wird durch die besten Individuen einer Generation modifiziert
- situatives und normatives Wissen

Kulturelle Algorithmen

CULTURAL-ALGORITHM(Bewertungsfunktion F)

1 $t \leftarrow 0$
2 $P(t) \leftarrow$ initialisiere die Population
3 $\mathcal{BS}(t) \leftarrow$ initialisiere den Überzeugungsraum
4 bewerte $P(t)$ durch F
5 **while** Terminierungsbedingung nicht erfüllt
6 **do** $\ulcorner P' \leftarrow$ bestimme wichtige Individuen aus $P(t)$
7 $\mathcal{BS}(t+1) \leftarrow \mathcal{BS}(t)$ wird durch P' angepasst
8 $P'' \leftarrow$ erzeuge Nachkommen von $P(t)$ auf der Basis von $\mathcal{BS}(t+1)$
9 bewerte $P(t)$ durch F
10 $t \leftarrow t+1$
11 $\llcorner P(t) \leftarrow$ Selektion aus $P'(\circ P(t-1))$
12 **return** bestes Individuum aus $P(t)$

Kulturelle Algorithmen

Situatives Wissen

- für $\Omega = \mathbb{R}^n$: letzten beiden besten Individuen
- Mutation soll sich ggf. in diese Richtung orientieren

Normatives Wissen

- für $\Omega = \mathbb{R}^n$: Ober- und Untergrenze pro Dimension
- größter/kleinster Wert der 20% besten Individuen der letzten Generation ermitteln
- immer: Vergrößerung akzeptieren
- nur bei besserem Gütewert des entsprechenden „Grenzindividuums“: Verkleinerung akzeptieren

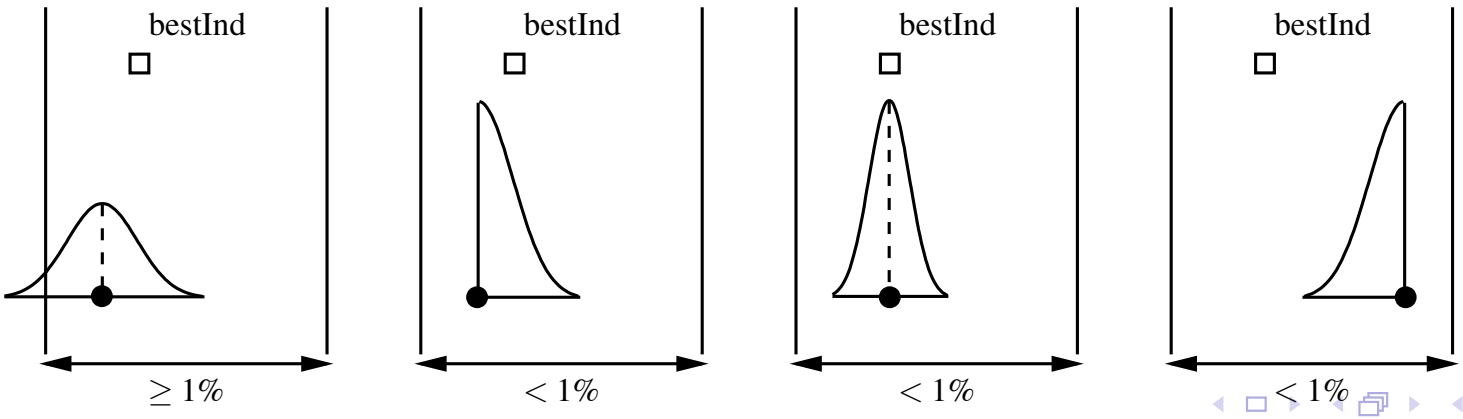
Kulturelle Algorithmen

Stabile Suchraumdimension

- wenn Bereich auf $< 1\%$ eingeschränkt
- und letzte beste Individuen innerhalb

Mutation

- falls stabil: am besten Individuum orientieren
- sonst: selbstangepasste Schrittweite



Kulturelle Algorithmen

CA-MUTATION(Individuum A)

1 $u' \leftarrow$ wähle zufällig gemäß $\mathcal{N}(0,\ 1)$
2 **for each** $i \in \{1, \ldots, l\}$
3 **do** $\ulcorner$ $u_i'' \leftarrow$ wähle zufällig gemäß $\mathcal{N}(0,\ 1)$
4 $B.S_i \leftarrow A.S_i \cdot \exp\left(\frac{1}{\sqrt{2l}} \cdot u' + \frac{1}{\sqrt{2\sqrt{l}}} \cdot u_i''\right)$
5 $u \leftarrow$ wähle zufällig gemäß $\mathcal{N}(0, B.S_i)$
6 **if** $\mathcal{BS}$ ist stabil für Dimension i
7 **then** $\ulcorner$ **switch**
8 **case** $A.G_i < BestInd.G_i : B.G_i \leftarrow A.G_i + |u|$
9 **case** $A.G_i > BestInd.G_i : B.G_i \leftarrow A.G_i - |u|$
10 $\llcorner$ **case** $A.G_i = BestInd.G_i : B.G_i \leftarrow A.G_i + \frac{u}{2}$
11 **else** $\llbracket$ $B.G_i \leftarrow A.G_i + u$
12 $\llcorner$ $B.G_i \leftarrow \max\{ug_i, \min\{og_i, B.G_i\}\}$
13 **return** B

CA für Randbedingungen

Anpassung der Mutation (1)

- modifizierte Mutation der kulturellen Algorithmen
- regelmäßig werden Bereichsgrenzen für jede Suchraumdimension abgeleitet
  (Minimum und Maximum der 20% besten gültigen Individuen bei Turnierumweltselektion)

CA für Randbedingungen

Anpassung der Mutation (2)

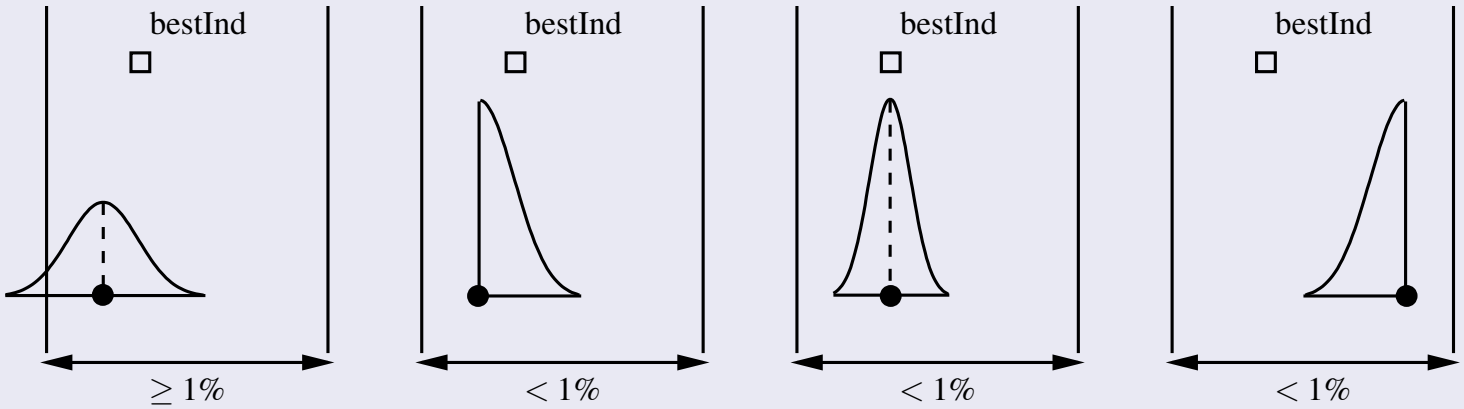
- Einteilung dieser Bereiche in s gleichlange Abschnitte
- es gibt s^n Parzellen des Suchraums
- Klassifizierung der Parzellen:

Anzahl gültige Individuen	Anzahl ungültige Individuen	
	$= 0$	> 0
$= 0$	unbekannt	ungültig
> 0	gültig	halbgültig

CA für Randbedingungen

Anpassung der Mutation (3)

- Mutation außerhalb der Parzellen:



- unbekannte, gültige und halbgültige Parzellen: sehr kleine Mutationsschrittweite
- ungültige Parzellen: Sprung in die nächstgelegene halbgültige (bzw. gültige) Parzelle

Überblick

1 Klassifizierende Systeme

2 Tabu-Suche

3 Memetische Algorithmen

4 Populationsbasiertes inkrementelles Lernen

5 Differentialevolution

6 Scatter Search

7 Kulturelle Algorithmen

8 Ameisenkolonien

9 Partikelschwärme

Ameisenkolonien

Motivation

- Insektenkolonien:
  - bewältigen komplexe Aufgabenstellungen
  - ohne zentrale Steuerung
  - mit einfacher Basiskommunikation
- konkret: kürzeste Wege bei Ameisen
  - Weg zur Futterstelle
  - Ameisen hinterlassen Duftstoff Pheromon
  - viel Pheromon $\Rightarrow$ Weg wird häufiger gewählt
  - kürzere Wege haben größere Frequenz

Ameisenkolonien

Idee

- interessant: Probleme deren Lösung ein Weg in einem Graphen ist
- Beispiel: Handlungsreisendenproblem
- μ Ameisen konstruieren jeweils eine Rundtour

Ameisenkolonien

Pheromonerzeugung

- Ameise platziert Pheromon auf Kanten invers zur Tourlänge
- vom alten Pheromonwert „verdunstet“ ein Anteil $(1 - \alpha)$
- exakt:

$$\tau_{i,j} \leftarrow \alpha \cdot \tau_{i,j} + \sum_{k=1}^{\mu} \mathit{wert}(A^{(k)}, i, j)$$

$$\text{mit } \mathit{wert}(A, i, j) = \begin{cases} \frac{1}{A.F} & \text{falls } \langle i, j \rangle \text{ in } A \text{ enthalten} \\ 0 & \text{sonst.} \end{cases}$$

Ameisenkolonien

Auswahl der nächsten Stadt

- Auswahlwahrscheinlichkeit hängt von zwei Faktoren ab
  - Pheromonmenge auf der Kante
  - inverse Entfernung der durch die Kante verbundenen Städte
- exakt:

$$Pr[v_j|v_i] = \begin{cases} \frac{\tau_{i,j} \cdot (nah_{i,j})^\beta}{\sum_{v_k \in verf\ügbar} \tau_{i,k} \cdot (nah_{i,k})^\beta} & \text{falls } v_j \in verf\ügbar \\ 0 & \text{sonst} \end{cases}.$$

- durch einen Regler Θ kann die Wahrscheinlichkeit durch deterministische Wahl des Besten ersetzt werden

AMEISENKOLONIE-TSP(Bewertungsfunktion F (TSP mit n Städten))

```
 1  t ← 0
 2  PM^(t) ← initialisiere Pheromon
 3  while Terminierungsbedingung nicht erfüllt
 4  do ⌜ for i ← 1, ..., μ (|Anzahl der Ameisen|)
 5        do ⌜ A^(i) ← ⟨1⟩ (|initialisiere neue Ameise|)
 6           aktuell ← 1
 7           for k ← 2, ..., n
 8           do ⌜ u ← wähle Zufallszahl gemäß U([0, 1))
 9              if u < θ (|Regler für Exploration|)
10              then ⌊ nächster ← wähle Knoten gemäß Pr[v_j|aktuell]
11              else ⌊ nächster ← Knoten j mit maximalem Pr[v_j|aktuell]
12              A^(i) ← A^(i) ∘ ⟨nächster⟩
13           ⌞ aktuell ← nächster
14        ⌞ bewerte A^(i) durch F
15     t ← t + 1
16     ⌞ PM^(t) ← aktualisiere PM^(t−1)
17  return beste gefundene Rundreise
```

Ameisenkolonien

Parametereinstellungen

Parameter	Wertebereich
Anzahl der Ameisen μ:	10
Gewichtung der Entfernung β:	2–6
Verdunstungsgrad α:	0.6–0.9
Explorationsregler θ:	0.2–0.9

Überblick

1 Klassifizierende Systeme

2 Tabu-Suche

3 Memetische Algorithmen

4 Populationsbasiertes inkrementelles Lernen

5 Differentialevolution

6 Scatter Search

7 Kulturelle Algorithmen

8 Ameisenkolonien

9 Partikelschwärme

Partikelschwärme

Motivation

- Modellierung sozialer Interaktion
- Verbesserung wird durch Nachahmung und Lernen von anderen Individuen erreicht (statt Selektion)
- Schwarmverhalten von Vögeln/Fischen

Konkret

- Genotyp $A.G \in \mathbb{R}^l$
- Zusatzinformationen $A.S = (\nu_1, \ldots, \nu_l, B_1, \ldots, B_l) \in \mathbb{R}^{2 \cdot l}$ mit
  - Veränderungsvektor $(\nu_1, \ldots, \nu_l)$
  - bester Punkt dieses Individuums $(B_1, \ldots, B_l)$

Partikelschwärme

Mutation

- zwei Komponenten in $A'.G_i = A.G_i + \nu'_i$
  - das Bestreben eines Individuums zu seinen Erfolgen zurückzukehren
  - eine Orientierung des Individuums an den besten Erfolgen seiner Nachbarn
- $\nu'_k = \beta \cdot \nu_k^{(i)} + \alpha_1 \cdot u_1 \cdot (B_k^{(i)} - A^{(i)}.G_k) + \alpha_2 \cdot u_2 \cdot (best_k - A^{(i)}.G_k)$
  - *best* z.B. aus $A^{(i-1)}$, $A^{(i)}$ und $A^{(i+1)}$
  - $u_1, u_2 \sim U([0,\ 1])$
  - Trägheitsfaktor β
  - Einfluss der besten Position α_1
  - Sozialer Faktor α_2

PARTIKELSCHWARM(Bewertungsfunktion F)

1 $t \leftarrow 0$
2 $P(t) \leftarrow$ initialisiere die Population der Größe μ
3 bewerte $P(t)$ durch F
4 $best \leftarrow$ Genotyp des besten Individuums in $P(t)$
5 **while** Terminierungsbedingung nicht erfüllt
6 **do** $\ulcorner P' \leftarrow \langle\rangle$
7 **for** $i \leftarrow 1, \ldots, \mu$
8 **do** $\ulcorner$ (sei $A^{(i)}.S = (\nu_1^{(i)}, \ldots, \nu_l^{(i)}, B_1^{(i)}, \ldots, B_l^{(i)})$)
9 $u_1, u_2 \leftarrow$ wähle Zufallszahlen gemäß $U([0,\ 1])$
10 **for each** $k \in \{1, \ldots, l\}$
11 **do** $\llcorner \nu'_k \leftarrow \beta \cdot \nu_k^{(i)} + \alpha_1 \cdot u_1 \cdot (B_k^{(i)} - A^{(i)}.G_k) + \alpha_2 \cdot u_2 \cdot (best_k - A^{(i)}.G_k)$
12 **if** $\|(\nu'_1, \ldots, \nu'_l)\| > MAX$ (maximale Veränderung)
13 **then** $\llcorner (\nu'_1, \ldots, \nu'_l) \leftarrow$ skaliere den Veränderungsvektor auf die Länge MAX
14 **for each** $k \in \{1, \ldots, l\}$
15 **do** $\llcorner A'.G_k \leftarrow A_k^{(i)} + \nu'_k$
16 bewerte A' durch F
17 **if** $A'.F \succeq B^{(i)}.F$
18 **then** $\llcorner P' \leftarrow P' \circ \langle (A'_1, \ldots, A'_l, \nu'_1, \ldots, \nu'_l, B_1^{(i)}, \ldots, B_l^{(i)}) \rangle$
19 $\llcorner$ **else** $\llcorner P' \leftarrow P' \circ \langle (A'_1, \ldots, A'_l, \nu'_1, \ldots, \nu'_l, A'_1, \ldots, A'_l) \rangle$
20 $t \leftarrow t + 1$
21 $P(t) \leftarrow P'$
22 $\llcorner best \leftarrow$ Genotyp des besten Individuums in $P(t)$
23 **return** bestes Individuum aus $P(t)$

Partikelschwärme

Parameterwerte

Parameter	Wertebereich
Trägheit β:	0.8–1.0
kognitiver Faktor α_1:	1.5–2.0
sozialer Faktor α_2:	1.5–2.0
maximale Veränderung MAX:	$og_i - ug_i$
Populationsgröße μ:	20–60

Überblick

1 Klassifizierende Systeme

2 Tabu-Suche

3 Memetische Algorithmen

4 Populationsbasiertes inkrementelles Lernen

5 Differentialevolution

6 Scatter Search

7 Kulturelle Algorithmen

8 Ameisenkolonien

9 Partikelschwärme

Genetischer Algorithmus

Repräsentation: klassisch: $\mathbb{B}^l$ mit fester Länge l, auch: $\mathbb{R}^l$ und $\mathcal{S}_n$, Dekodierung

Mutation: Bitflipping, gleichverteilte reellwertige Mutation, spezielle Permutationsoperatoren

Rekombination: k-Punkt- und uniformer Crossover, arithmetischer Crossover, Ordnungsrekombination

Selektion: Elternselektion, fitnessproportional oder Turnierselektion

Population: mittelgroße Populationen

Besonderheiten: theoretische Grundlage: Schema-Theorem

Evolutionsstrategie

Repräsentation: $\mathbb{R}^l$, meist gilt Genotyp = Phänotyp, zusätzliche Strategieparameter

Mutation: Gauss-Mutation, erst Mutation der Strategieparameter

Rekombination: anfangs keine, später: arithmetischer und uniformer Crossover, auch: globale Varianten, anderer Operator auf Strategieparametern

Selektion: Eltern: gleichverteilt, Kinder: Komma- bzw. Plus-Selektion

Population: eher kleine Populationen, manchmal: $\mu = 1$

Besonderheiten: Selbstadaptation für Schrittweiten

Evolutionäres Programmieren

Repräsentation: anfangs: endlicher Automat, später: $\mathbb{R}^l$ mit Strategieparametern

Mutation: Modifikation der Zustände und Übergänge in Automaten, Gauss-Mutation, gleichzeitige Mutation der Strategieparameter

Rekombination: keine

Selektion: anfangs: Plus-Selektion, später: 2-fache q-stufige Turnierselektion aus Eltern und Kindern

Population: mittelgroße Populationen

Besonderheiten: Es gilt $\mu = \lambda$, Selbstadaptation

Genetisches Programmieren

Repräsentation: Bäume, aber auch lineare Darstellungen variabler Länge

Mutation: internes Umhängen oder Modifikation von Teilbäumen

Rekombination: Austausch von Teilbäumen

Selektion: verschiedene, meist wie genetischer Algorithmus

Population: sehr große Populationen

Besonderheiten: oft nur ein Operator auf ein Individuum, spezielle Initialisierung, Methoden zur Vermeidung von Introns

lokale Suche

Repräsentation: beliebig

Mutation: beliebig

Rekombination: keine

Selektion: Verbesserungen immer, Verschlechterungen mit gewisser Wahrscheinlichkeit

Population: ein Individuum

Besonderheiten: zentrales Problem: zu frühe Konvergenz

Klassifizierendes System

Repräsentation: Regel oder Menge an Regeln, k-är kodiert

Mutation: Bitflipping

Rekombination: k-Punkt-Crossover

Selektion: fitnessproportional

Population: bei Michigan: ausreichend für Komplexität der Aufgabe

Besonderheiten: Michigan: Population ist Regelsatz, Pittsburgh: Individuum ist Regelsatz

Tabu-Suche

Repräsentation: phänotypnah

Mutation: unumkehrbare durch Tabu-Listen

Rekombination: keine

Selektion: bestes Individuum

Population: ein Elter, mehrere Kinder

Besonderheiten: bestes gefundene Individuum wird zusätzlich gespeichert

Memetischer Algorithmus

Repräsentation: beliebig

Mutation: wird mit lokaler Suche verknüpft

Rekombination: beliebig

Selektion: beliebig

Population: beliebig

Besonderheiten: beliebig

Populationsbasiertes inkrementelles Lernen

Repräsentation: $\mathbb{B}^l$

Mutation: Änderung in der Populationsstatistik

Rekombination: implizit

Selektion: bestes Kindindividuum geht in Statistik ein

Population: wird durch Populationsstatistik ersetzt

Besonderheiten: benötigte Individuum werden aus der Statistik zufällig erzeugt

Differentialevolution

Repräsentation: $\mathbb{R}^l$

Mutation: Mischoperator

Rekombination: Mischoperator

Selektion: Kind ersetzt Elter bei Verbesserung

Population: klein/mittelgroß

Besonderheiten: Mutation nutzt Populationsinformation

Scatter Search

Repräsentation: $\mathbb{R}$ und andere

Mutation: keine

Rekombination: Teilmengengenerator und Kombination

Selektion: Selektion der Besten

Population: mittelgroß

Besonderheiten: viele Varianten, deterministisches Verfahren

Kultureller Algorithmus

Repräsentation: $\mathbb{R}^l$ und andere

Mutation: nutzt Information des Überzeugungsraums

Rekombination: keine

Selektion: Umweltselektion

Population: mittelgroß

Besonderheiten: Überzeugungsraum speichert normatives und situationsbezogenes Wissen

Ameisenkolonie

Repräsentation: verschiedene

Mutation: jede Ameise konstruiert einen Lösungskandidaten

Rekombination: keine

Selektion: Güte bestimmt Einfluss auf die globale Pheromonmenge

Population: Anzahl der Ameisen pro Iterationsschritt

Besonderheiten: kein EA-Schema, globale Pheromonmengen repräsentieren Lösungskandidaten ähnlich zur Statistik in PBIL

Partikelschwarm

Repräsentation: $\mathbb{R}^l$

Mutation: basiert auf Trägheit und Orientierung an den Nachbarn

Rekombination: keine

Selektion: Orientierung am Besten (Population/eigene Historie)

Population: klein/mittelgroß

Besonderheiten: kein EA-Schema, eher: synchrones Durchkämmen des Suchraums