Evolutionäre Algorithmen

Vorlesung 7

Evolutionsstrategien
Evolutionäres Programmieren

Evolutionsstrategien (ES)

- reellwertiger Genotyp $\mathcal{G} = \mathbb{R}^l$
- keine Elternselektion (uniforme Auswahl)
- deterministische Selektion durch Verkleinern der Population
- primärer Operator: Mutation

- Techniken zur Anpassung der Mutation
- sekundär: Rekombination

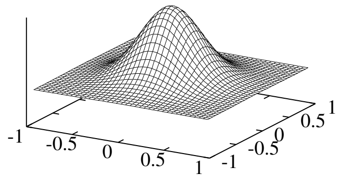
ES: Mutation

- komponentenweise Addition zu $A = (A_1, \ldots, A_l)$:

$$B_i \leftarrow A_i + u_i$$

mit $u_i \sim \mathcal{N}(0, \sigma)$

- Dichtefunktion $p_{\text{Gauss}}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}x^2}$



ES: Mutation

- keine Ober- und Untergrenze
- Erwartungswert: 0
- erwartete Schrittweite: σ
- kleinere Veränderungen haben größere Wahrscheinlichkeiten als große Veränderungen

- kritische Rolle der Standardabweichung σ

$1/5$-Regel

- ▷ Anpassung von σ
- ▷ Erfolgswahrscheinlichkeit p_s über letzte k Generationen (Prozentsatz der besseren Nachkommen)
- ▷ nahe dem gesuchten Optimum: p_s ist klein

  $\Rightarrow$ Feinabstimmung durch Verkleinerung von σ
- ▷ ist p_s groß: großes Verbesserungspotential

  $\Rightarrow$ Schrittweite σ vergrößern

$1/5$-Regel

- ▷ Anpassung alle k Generationen

$$\sigma' \leftarrow \begin{cases} \sigma/c, & \text{if } p_s > 1/5 \\ \sigma c, & \text{if } p_s < 1/5 \\ \sigma, & \text{if } p_s = 1/5 \end{cases}$$

- ▷ i.d.R.: $0.817 \leq c \leq 1$
- ▷ Problem: lokale Optima

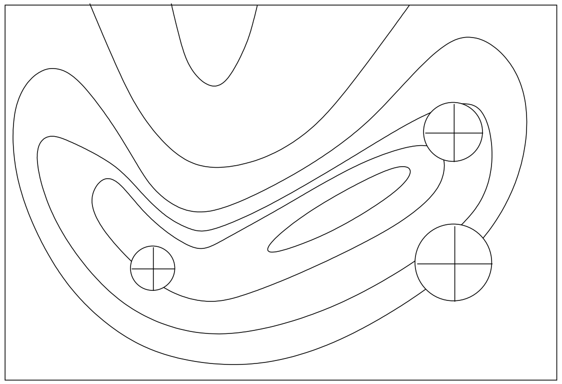
Selbst-Adaptation

- ▷ letzte Standardabweichung wird im Individuum gespeichert

  $A' = \langle A_1, \ldots, A_n, \sigma \rangle$
- ▷ dient als Ausgangswert für nächste Mutation
- ▷ 1. Schritt: $\sigma' \leftarrow \sigma e^{\frac{1}{\sqrt{l}} u}$
- ▷ 2. Schritt: $B_i \leftarrow a_i + \mathcal{N}(0, \sigma')$
- ▷ neues Individuum: $\langle B_1, \ldots, B_n, \sigma' \rangle$
- ▷ Strategieparameter σ unterliegt indirekt dem Selektionsdruck

Selbst-Adaptation

- ▷ individuelle Schrittweiten für die einzelnen Individuen



l Strategieparameter

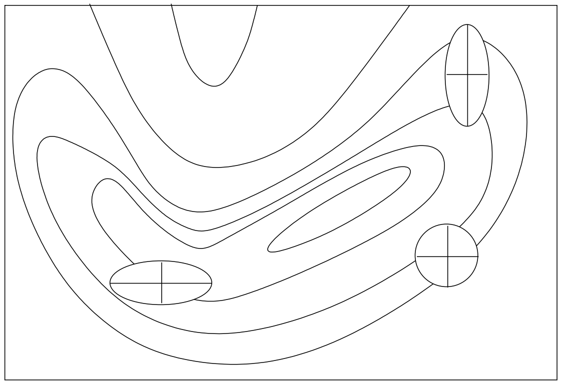
- Einführung eines Strategieparameters für jede Dimension
- $A = \langle A_1, \ldots, A_l, \sigma_1, \ldots, \sigma_l \rangle$
- $\sigma_i' \leftarrow \sigma_i e^{\frac{1}{\sqrt{2l}} u + \frac{1}{\sqrt{2\sqrt{l}}} u_i'}$

  $B_i \leftarrow A_i + \mathcal{N}(0, \sigma')$

  mit $u, u_i' \sim \mathcal{N}(0,1)$

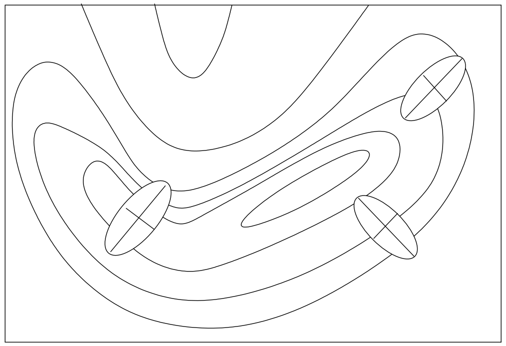
l Strategieparameter

- eine bessere Ausrichtung der Individuen im Suchraum ist möglich



Kovarianzmatrixadaptation

- zusätzlich: Kovarianzmatrix zur Berücksichtigung einer beliebigen Ausrichtung ($\frac{1}{2}l(l+1)$ Parameter)



ES: Rekombination

- keine Rekombination
- intermediär (arithmetisch mittelnd) auf zwei Eltern
- diskret (uniformer Crossover) auf zwei Eltern
- global intermediär auf kompletter Population
- global diskret auf kompletter Population

- häufig: paarweise diskret auf Objektvariablen, global intermediär auf Standardabweichungen, keine Rekombination auf Winkeln

ES: Selektion

- ▷ Komma-Selektion (μ, λ) ($\lambda > \mu$)

  λ Kinder werden auf μ beste Individuen reduziert

  meist: $\frac{\mu}{\lambda} = \delta$ mit $\frac{1}{7} \leq \delta \leq \frac{1}{5}$
- ▷ Plus-Selektion $(\mu + \lambda)$

  λ Kinder und μ Eltern werden auf μ beste Individuen reduziert

  Selbstanpassung scheint hier nicht so gut zu funktionieren

ES: Gesamtalgorithmus

```
EINGABEN: Parameter, Zielfunktion F, Rekombinationswahrscheinlichkeit p_r
t ← 0
erzeuge Population P(t)
bewerte Population P(t) mittels F
while Terminierungsbedingung nicht erfüllt do
  P' ← ⟨⟩
  for i ∈ {1, ..., λ} do
    B, B', ... ← selektiere Eltern uniform zufällig aus P(t)
    A ← rekombiniere B, B', ...
    C ← mutiere A
    bewerte C mittels F
  end for
  t ← t + 1
  (globale Anpassung von σ)
  P(t) ← Selektion aus P' (∪P(t − 1))
end while
AUSGABE: bestes Individuum aus P(t)
```


Evolutionäres Programmieren

- ▷ keine Elternselektion: jedes Eltern hat genau ein Kind
- ▷ Mutation
- ▷ keine Rekombination
- ▷ duplikatfreie Umweltselektion aus Eltern und Kindern

- ▷ ursprünglich: auf endlichen Automaten zur Zeitreihenprognose
- ▷ später: auf reellwertigen Tupeln

EP auf Automaten

- ▷ $\mathcal{G} = \mathcal{S} = \langle Q, q_0, Q \times \Sigma \to Q, Q \times \Sigma \to \Sigma \rangle$
- ▷ Mutation: Änderung eines Ausgabesymbols
- ▷ Mutation: Änderung eines Folgezustands
- ▷ Mutation: Hinzufügen eines Zustands
- ▷ Mutation: Löschen eines Zustands
- ▷ Mutation: Veränderung des Startzustands

- ▷ Plus-Selektion

Modernes EP

- Suchraum: $\mathcal{S} = \mathbb{R}^l$
- Individuen: $A = (A_1, \ldots, A_l, \sigma_1, \ldots, \sigma_l)$
- Mutation: neues Individuum $(B_1, \ldots, B_l, \sigma'_1, \ldots, \sigma'_l)$ durch

  $B_i \leftarrow A_i + \mathcal{N}(0, \sigma_i)$

  $\sigma'_i \leftarrow \sigma_i + s\mathcal{N}(0, \sigma_i)$

  (s ist Parameter für die Stärke der Anpassung)

  Falls $\sigma_i < 0$ wird stattdessen ein kleines $\epsilon > 0$ benutzt

Modernes EP: Selektion

- zweistufige q-fache Turnierselektion
- für jedes Individuum A werden q zufällige „Gegner"

  $B_1, \ldots, B_q$ gewählt
- Erfolgsrate: $s_A = \frac{1}{q} \sum_{1 \leq i \leq q} Q(A, B_i)$

  mit $Q(A,B) = \begin{cases} 1, & f(A) \succ f(B) \\ 0, & \text{sonst} \end{cases}$