

Evolutionäre Algorithmen: Vergleich von Algorithmen

Karsten Weicker

HTWK Leipzig

5. Januar 2013

Überblick

- 1 Herausforderung der evolutionären Algorithmen
- 2 Abhängigkeit der Parameter
- 3 Vergleich zweier Algorithmen

Überblick

- 1 Herausforderung der evolutionären Algorithmen
- 2 Abhängigkeit der Parameter
- 3 Vergleich zweier Algorithmen

Was ist der beste EA?

Ein Zitat der 1980er

... Later, with newfound success under their belts, these same users confidently strike out “to really make these algorithms fly,” oftentimes by introducing an odd array of programming tricks and hacks. The usual result is disappointment in the “improved” GA. Although it works better on some problems, it works worse on most.

*David E. Goldberg,
Zen and the Art of Genetic Algorithms*

Was ist der beste EA?

Absicht der Aussage

- Mit dem GA können leicht Erfolge erzielt werden.
- Verbesserungsversuche führen zur Verschlechterung auf vielen Problemen
- Der GA ist besser auf vielen Problemen

Wahrer Gehalt der Aussage

- Es gilt immer, dass sehr gute Performance auf einem Problem mit schlechten Ergebnissen auf anderen bezahlt werden!

No-Free-Lunch-Theorem

Randbedingung

- Menge $\mathcal{F}$ aller Optimierungsprobleme $F : \Omega \rightarrow \mathbb{R}$
- Menge $\mathcal{A}$ aller Optimierungsalgorithmen auf Ω

Charakterisierung der Optimierung

- Problem $F \in \mathcal{F}$ wird wie folgt bearbeitet:

$$\text{Optimierung}_{F,n} : \mathcal{A} \rightarrow \Omega^n.$$

- Für $\text{Optimierung}_{F,n}(\text{Alg}) = (y_1, \dots, y_n) \in \Omega^n$
gilt: $y_i \neq y_j$ für $i \neq j$

Vergleich zweier Algorithmen

Definition

- Seien $Alg_1, Alg_2 \in \mathcal{A}$
- Vergleich durch Leistungsmaß

$$QuAlg_{F,n}(Alg) = H_n(F(y_1), \dots, F(y_n))$$

mit beliebiger Funktion $H_n : \mathbb{R}^n \rightarrow \mathbb{R}$

Beispiele

- bester gefundener Wert $H_n \equiv \max$
- online-Performance $H_n \equiv \sum_{i=1}^n F(y_i)$

Erwartete Leistung

Motivation

- nichts ist über das bearbeitete Optimierungsproblem bekannt
- jedes $F \in \mathcal{F}$ ist gleich wahrscheinlich

Definition

$$\text{Erw} [QuAlg_{F,n}(Alg) \mid F \in \mathcal{F}] = \frac{1}{\#\mathcal{F}} \sum_{F \in \mathcal{F}} QuAlg_{F,n}(Alg).$$

No-Free-Lunch-Theorem

Aussage

- Es gilt für $Alg_1, Alg_2 \in \mathcal{A}$ und $\mathcal{F}$ bzgl. $QuAlg$:

$$\text{Erw} [QuAlg_{F,n}(Alg_1) \mid F \in \mathcal{F}] = \text{Erw} [QuAlg_{F,n}(Alg_2) \mid F \in \mathcal{F}]$$

Beweisidee

- Über Integrale als Erwartungswerte
- oder: als kombinatorische Überlegung

No-Free-Lunch-Theorem: Beweis

O.B.d.A.

- $\Omega = \{x_1, \dots, x_m\}$
- vorkommende Gütwerte $r_i \in \mathbb{R}$ ($1 \leq i \leq m$)

Folgerung

- jedes $F \in \mathcal{F}$ wird durch eine Permutation $\pi \in \mathcal{S}_m$ festgelegt
- $F(x_i) = r_{\pi(i)}$ für $1 \leq i \leq m$
- Es gibt $m!$ unterschiedliche Funktionen in $\mathcal{F}$

No-Free-Lunch-Theorem: Beweis

Ablauf einer Optimierung

- erster Punkt $y_1 = x_{j_1}$ wird völlig unabhängig von F gewählt
- die m Gütwerte können mit gleicher Wahrscheinlichkeit auftreten
- ein gefundener Wert kann zu genau $(m - 1)!$ unterschiedlichen Funktionen gehören

No-Free-Lunch-Theorem: Beweis

jeweils
eine mögliche
Funktion



x_1	x_2	x_3	x_4
1	2	3	4
1	2	4	3
1	3	2	4
1	3	4	2
1	4	2	3
1	4	3	2

x_1	x_2	x_3	x_4
2	1	3	4
2	1	4	3
2	3	1	4
2	3	4	1
2	4	1	3
2	4	3	1

x_1	x_2	x_3	x_4
3	1	2	4
3	1	4	2
3	2	1	4
3	2	4	1
3	4	1	2
3	4	2	1

x_1	x_2	x_3	x_4
4	1	2	3
4	1	3	2
4	2	1	3
4	2	3	1
4	3	1	2
4	3	2	1

No-Free-Lunch-Theorem: Beweis

Ablauf einer Optimierung (i -te Iteration)

- $i - 1$ Punkte wurden bisher gewählt
- insgesamt konnten $m \cdot \dots \cdot (m - i + 2) = \frac{m!}{(m-i+1)!}$ Gütefolgen auftreten
- jeweils $m - i + 1$ verschiedene nächste Gütewerte sind möglich
- dann jeweils: $(m - i)!$ Funktionen möglich

No-Free-Lunch-Theorem: Beweis

x_1	x_2	x_3	x_4
1	2	3	4
1	2	4	3
1	3	2	4
1	3	4	2
1	4	2	3
1	4	3	2

x_1	x_2	x_3	x_4
2	1	3	4
2	1	4	3
2	3	1	4
2	3	4	1
2	4	1	3
2	4	3	1

x_1	x_2	x_3	x_4
3	1	2	4
3	1	4	2
3	2	1	4
3	2	4	1
3	4	1	2
3	4	2	1

x_1	x_2	x_3	x_4
4	1	2	3
4	1	3	2
4	2	1	3
4	2	3	1
4	3	1	2
4	3	2	1

No-Free-Lunch-Theorem: Beweis

Algorithmus 1:

x_1	x_2	x_3	x_4
1	2	3	4
1	2	4	3
1	3	2	4
1	3	4	2
1	4	2	3
1	4	3	2

x_1	x_2	x_3	x_4
2	1	3	4
2	1	4	3
2	3	1	4
2	3	4	1
2	4	1	3
2	4	3	1

x_1	x_2	x_3	x_4
3	1	2	4
3	1	4	2
3	2	1	4
3	2	4	1
3	4	1	2
3	4	2	1

x_1	x_2	x_3	x_4
4	1	2	3
4	1	3	2
4	2	1	3
4	2	3	1
4	3	1	2
4	3	2	1

Algorithmus 2:

x_1	x_2	x_3	x_4
1	2	3	4
1	2	4	3
1	3	2	4
1	3	4	2
1	4	2	3
1	4	3	2

x_1	x_2	x_3	x_4
2	1	3	4
2	1	4	3
2	3	1	4
2	3	4	1
2	4	1	3
2	4	3	1

x_1	x_2	x_3	x_4
3	1	2	4
3	1	4	2
3	2	1	4
3	2	4	1
3	4	1	2
3	4	2	1

x_1	x_2	x_3	x_4
4	1	2	3
4	1	3	2
4	2	1	3
4	2	3	1
4	3	1	2
4	3	2	1

No-Free-Lunch-Theorem: Beweis

Abschluss des Beweises

- jede mögliche Reihenfolge an Gütewerten tritt bei genau gleich vielen Funktionen auf
- dies ist völlig unabhängig vom Vorgehen des Algorithmus
- Es folgt direkt:
$$\text{Erw} [QuAlg_{F,n}(Alg) \mid F \in \mathcal{F}]$$

resultiert in einem identischen Werte für jeden Algorithmus Alg

No-Free-Lunch-Theorem: Folgerungen

Spezialisierung eines EAs

- gilt auf $\mathcal{F}' \subset \mathcal{F}$:

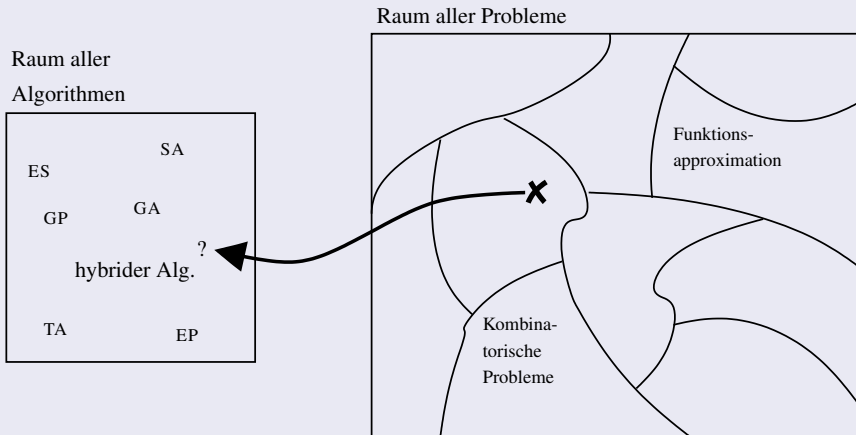
$$\text{Erw} [QuAlg_F(Alg_1) \mid F \in \mathcal{F}'] < \text{Erw} [QuAlg_F(Alg_2) \mid F \in \mathcal{F}'] ,$$

- dann folgt direkt:

$$\text{Erw} [QuAlg_F(Alg_1) \mid F \in \mathcal{F} \setminus \mathcal{F}'] > \text{Erw} [QuAlg_F(Alg_2) \mid F \in \mathcal{F} \setminus \mathcal{F}']$$

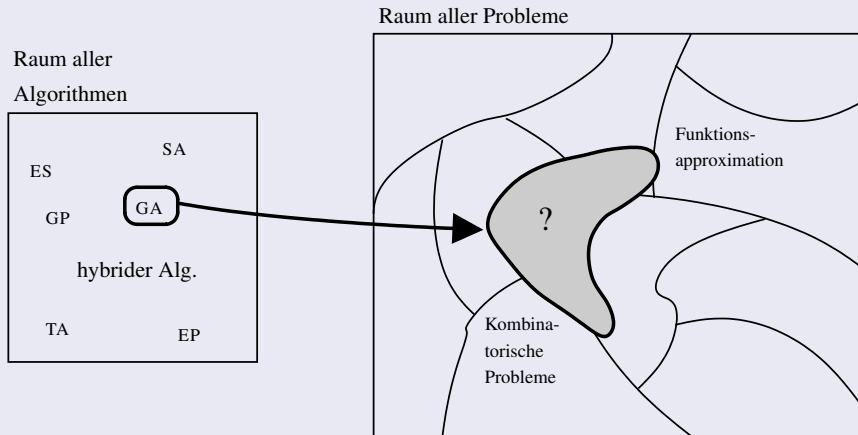
No-Free-Lunch-Theorem

Konsequenz für Anwender



No-Free-Lunch-Theorem

Konsequenz für Wissenschaftler



Überblick

- 1 Herausforderung der evolutionären Algorithmen
- 2 Abhängigkeit der Parameter
- 3 Vergleich zweier Algorithmen

Parametrierung eines EAs

Bsp.: Einsenzählproblem

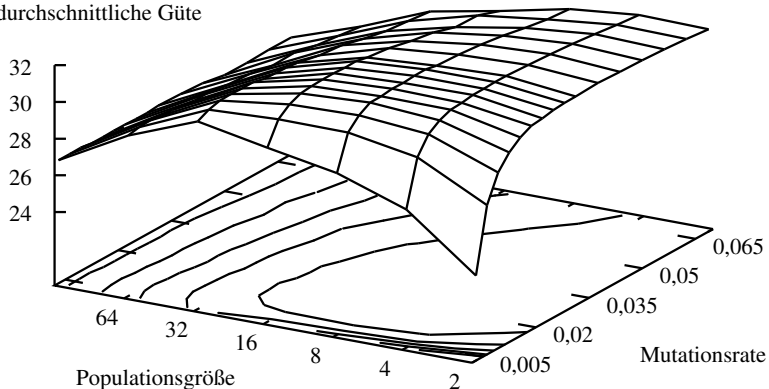
- 32 Bits
- GA mit Turnierselektion
- 2-Punkt-Crossover mit $p_x = 0,9$
- 512 Bewertungen mit F

einzustellende Parameter

- Mutationsrate: 0,005, 0,01, 0,015, ..., 0,065
- Populationsgröße: 2, 4, 8, 16, 32, 64

Parametrierung eines EAs

durchschnittliche Güte



Parametrisierung eines EAs

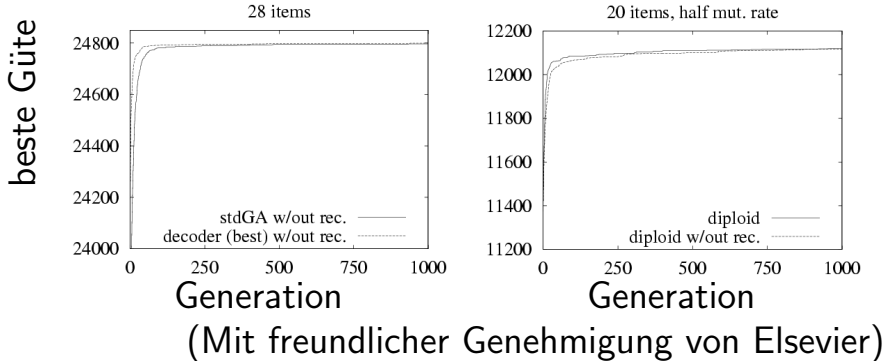
Folgerung

- separates aufeinanderfolgendes Einstellen der beiden Parameter führt i.d.R. zu suboptimalen Ergebnissen
- die verschiedenen Parameter sind stark voneinander abhängig
- es bleibt nur der direkte Vergleich zur Beurteilung von Algorithmen

Überblick

- 1 Herausforderung der evolutionären Algorithmen
- 2 Abhängigkeit der Parameter
- 3 Vergleich zweier Algorithmen

Zwei Vergleichskurven



Statistischer Hypothesentest

Ausgangsdaten

- jeweils v Experimente:

$$X_{Alg1,1}, \dots, X_{Alg1,v} \text{ und } X_{Alg2,1}, \dots, X_{Alg2,v}$$

Aggregierte Daten

- Erwartungswert: $\text{Erw}_{Alg1} = \frac{1}{v} \sum_{i=1,\dots,v} X_{Alg1,i}$
- Varianz: $\text{Var}_{Alg1} = \frac{1}{v-1} \sum_{i=1,\dots,v} (X_{Alg1,i} - \text{Erw}_{Alg1})^2$

t-Wert

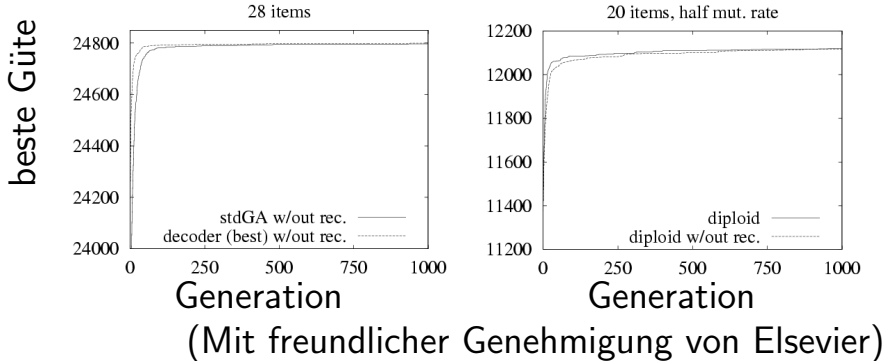
$$t = (\text{Erw}_{Alg1} - \text{Erw}_{Alg2}) / \sqrt{\frac{\text{Var}_{Alg1} + \text{Var}_{Alg2}}{v}}$$

Statistischer Hypothesentest

Signifikanter Unterschied

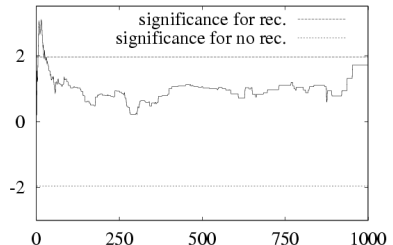
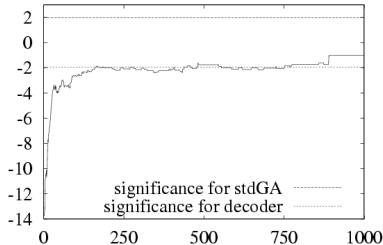
- je größer der Betrag von t , desto sicherer ist Hypothese (Gleichheit der Erwartungswerte) abzulehnen
- Fehlerwahrscheinlichkeit z.B. in Tabellen nachschlagen
- Unterschied ist *signifikant*, wenn Fehler $< 0,05$
 - bei $\nu = 10$ wenn $|t| > 1,984$
 - bei $\nu = 50$ wenn $|t| > 2,101$
- *sehr signifikant*, wenn Fehler $< 0,01$

Zwei Vergleichskurven



Zwei Vergleichskurven

t-Wert für beste Güte



Generation

Generation

(Mit freundlicher Genehmigung von Elsevier)