UNIVERSITÄT DORTMUND

Wintersemester 2005/06

Fundamente der Computational Intelligence

(Vorlesung)

Prof. Dr. Günter Rudolph

Fachbereich Informatik

Lehrstuhl für Algorithm Engineering

Inhalt

Konvergenzaussagen

Satz:

Sei $D_k = | f(x_k) - f^* |$ für $k \geq 0$ durch einen (1+1)-EA generiert,
$S^* = \{ x^* \in S : f(x^*) = f^* \}$ die Menge der optimalen Lösungen und
$P_m(x, S^*)$ die W'keit, von $x \in S$ durch eine Mutation nach S^* zu gelangen.

Wenn für jedes $x \in S \setminus S^*$ gilt $P_m(x, S^*) \geq \delta > 0$, dann $D_k \to 0$ vollständig.

Beweis:

Für den (1+1)-EA gilt: $P(x, S^*) = 1$ für $x \in S^*$ wg. Selektion des Besseren.

Es reicht also zu zeigen, dass der EA durch Mutation S^* sicher erreicht:

Erfolg in 1. Iteration: $P_m(x, S^*) \geq \delta$.
Kein Erfolg in 1. Iteration $\leq 1 - \delta$.
Kein Erfolg in k. Iteration $\leq (1- \delta)^k$.
$\Rightarrow$ Erfolg in k. Iteration $\geq 1 - (1- \delta)^k \to 1$ für $k \to \infty$.

Wg. $P\{ D_k > \varepsilon \} \leq (1- \delta)^k \to 0$ folgt Konvergenz in W'keit und da $\sum_{k=0}^{\infty} (1-\delta)^k < \infty$ folgt sogar vollständige Konvergenz. ■

Die Bedingung: $\forall x \in S \setminus S^*$ gilt $P_m(x, S^*) \geq \delta > 0$

- ist hinreichend, aber nicht notwendig und
- etwa für die globale Mutation mit Mutationsw'keit $p \in (0,1)$ erfüllt, da

  $P_m(x, y) = p^{H(x,y)} (1-p)^{n-H(x,y)} > 0$ für alle x, y in S,

  wobei n = Dimension und H(x,y) der Hamming-Abstand zwischen x und y.

Achtung:

Die gleiche Konvergenzaussage erhält man für reine Zufallssuche:

```
Wähle X0 ∈ S, setze k = 0.
repeat
  Wähle Yk zufällig gleichverteilt aus S.
  Falls f(Yk) < f(Xk) dann  Xk+1 = Yk
                      sonst Xk+1 = Xk
until Terminierung
```

Folgerung:

Solche Aussage sind von geringer Bedeutung, weil δ astronomisch klein!

Beispiel:

alle Bits müssen invertiert werden $\Rightarrow p^n = n^{-n} = \delta$

Übergang zum Optimum erfolgt mit W'keit δ (geometrische Verteilung)

$\Rightarrow$ mittlere Zeit bis zum Eintreffen des Ereignisses $1/\delta = n^n$

Sei n = 20 und wir haben einen Tera-Hertz-Rechner (10^{12} Iterationen/Sekunde).

Dann warten wir im Mittel $20^{20} / 10^{12}$ Sekunden = $2^{20} \times 10^8$ Sekunden auf Lösung,

also über 3,325 Millionen Jahre!

$\Rightarrow$ Das ist mathematisch gesehen zwar endlich, praktisch jedoch **unendlich**!

Also:

Stochastische Konvergenzaussagen (auch für Simulated Annealing etc.) sind mit Vorsicht zu genießen!

Aber:

Negative Aussage (keine Konvergenz bzw. kein sicheres Besuchen des Optimums) ist schon von Bedeutung!

Beispiel:

Sicheres Auffinden des Optimums in Zeit T mit W'keit $\gamma > 0$.

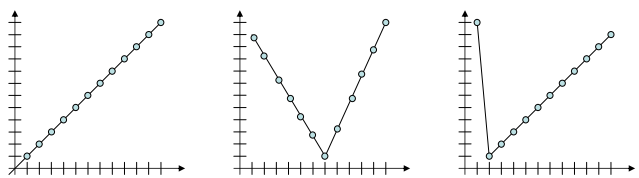
$\Rightarrow$ Multistart! W'keit des Mißerfolgs sinkt expoentiell schnell!

Allerdings spielt Größe von γ eine Rolle!

"Functions of Unication" (Goldberg/Deb 1993):

Es existiert g: $\{0,1, \ldots n\} \rightarrow \mathbb{R}$ mit f(x) = g(||x||), wobei || . || Hamming-Norm.

Beispiel: Counting Ones f(x) = || x ||



$\gamma = 1$ $\gamma = 0.5$ γ klein

NIAH
(needle-in-a-haystick)

Populationen: $(\mu + \lambda)$-EA

```
Wähle μ Individuen ∈ S
repeat
  do λ times:
    wähle zufällig ein Elter aus und
    mutiere Individuum
  end (→ λ Nachkommen)
  wähle μ beste aus μ + λ Individuen aus
until Terminierung
```

Reproduktion durch Mutation

Selektion

$\mu = \lambda = 1$ ist (1+1)-EA

Bei Populationen ist Variation nicht auf Mutation begrenzt:

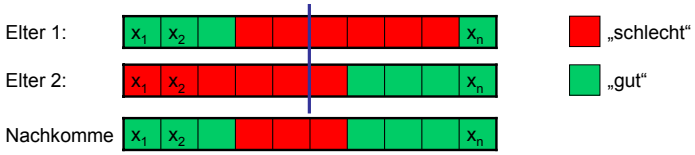
Sexuelle Reproduktion! → **Rekombination / Crossover**

Warum manchmal sinnvoll?

Annahme: Separierbares Problem $f(x) = \sum_{i=1}^{n} f_i(x_i)$

d.h. Lösung des Gesamtproblems durch Lösung kleiner Teilprobleme

⇒ Rekombination von Teillösungen



- **1-Punkt-Crossover:**
  Wähle 1 Position k ∈ { 1, 2, …, n-1 } zufällig aus.
  Nachkomme besteht aus ersten k Komponenten des 1. Elter und
  n-k letzten Komponenten des 2. Elter.

- **Multi-Point-Crossover:**
  Wähle mehrere Positionen zufällig ohne Zurücklegen aus.
  Positionen sortieren.
  Dann immer abwechselnd Komponenten übernehmen
  bis zur jeweiligen Position vom 1. oder 2. Elter.

- **Uniform Crossover:**
  Für jede Komponente Elter zufällig auswählen.
  Normalerweise gleichverteilt.

Es gibt: Verallgemeinerungen für mehr als 2 Eltern.

Zudem: Das funktioniert bei allen Produkträumen! Auch $\mathbb{R}^n$, $\mathbb{Z}^n$, …

Selektion: seien $v_1, v_2, \ldots, v_\lambda$ Fitnesswerte der Population

- **Proportionale Selektion** (Holland 1975)
  Annahme: $v_i > 0$

  W'keit, das i-te Individuum zu selektieren: $s_i = \frac{v_i}{\sum_{j=1}^{\lambda} v_j}$

  Bei λ Selektionen:
  Individuum i wird im Mittel $\lambda\, s_i$ mal gezogen

  Wenn $v_i \geq \lambda^{-1} \sum_j v_j$ („above average fitness“),
  dann wird Individuum i im Mittel mindestens 1 mal gezogen

  Allerdings: keine Garantie, dass das beste Individuum „überlebt“

Selektion: seien $v_1, v_2, \ldots, v_\lambda$ Fitnesswerte der Population

- **Stochastic Universal Sampling** (Baker 1987)

  Annahme: $v_i > 0$

  Man will erzwingen, dass etwa λ s_i Individuen gezogen werden:

  Sei c_i die Summe der Selektionsw'keiten bis Index i: $c_i = s_1 + s_2 + \ldots + s_i$

```
ziehe U ∈ (0,1) gleichverteilt, setze a = 0
for i = 1 to λ do
  while λ c_i > a + U do
    selektiere Individuum i
    a++
  endwhile
endfor
```

Das beste Individuum überlebt sicher!

Selektion: seien $v_1, v_2, \ldots, v_\lambda$ Fitnesswerte der Population

- **Tunierselektion** (Goldberg & Deb 1991)

  Man zieht q mal gleichverteilt mit W'keit $1 / \lambda$
  Das beste der q Individuen wird selektiert.

  Annahme: v_i aufsteigend sortiert (Rang 1 ist der beste)

  $Q_i = \{ i, i+1, \ldots, \lambda \}^q$ mit $|Q_i| = (\lambda - i + 1)^q$ enthält Ind. mit Rang $\geq i$,

  Q_i enthält auch Tupel ohne Index i, diese müssen wir abziehen:

  $Q_i \setminus Q_{i+1}$ mit $| Q_i \setminus Q_{i+1} | = | Q_i | - | Q_{i+1} | = (\lambda - i + 1)^q - (\lambda - i)^q$

  also: $s_i = [(\lambda - i + 1)^q - (\lambda - i)^q] / \lambda^q$ und $s_i < 1$

Selektion: seien $v_1, v_2, \ldots, v_\lambda$ Fitnesswerte der Population

- **μ beste** (Schwefel 1975)
  - **$(\mu + \lambda)$-Selektion**

    wähle μ beste Individuen aus μ Eltern und λ Nachkommen
    $\Rightarrow$ der beste Elter überlebt mit W'keit 1
  - **(μ , λ)-Selektion**

    wähle μ beste Individuen nur von den λ Nachkommen
    $\Rightarrow$ der beste Elter kann nicht überleben

    begrenzte Lebensdauer

manchmal auch „truncation selection“ genannt

Selektion: seien $v_1, v_2, \ldots, v_\lambda$ Fitnesswerte der Population

- **EP-Tunierselektion** (Fogel)

  selektieren aus μ Eltern und λ Nachkommen

  Für jedes Individuum i:

  ziehe q Individuen gleichverteilt

  jedesmal wenn i nicht schlechter, dann erhält es Bonuspunkt

  Selektiere μ Individuen mit meisten Bonuspunkten

  Man kann Fälle konstruieren, dass der beste Elter nicht überlebt!

Restriktionen / Nebenbedingungen:

- Lethalmutationen
- Straffunktionen
- Reparatur