

Organic Computing 2

Lösungsvorschlag Blatt03

Lukas Huhn Qiang Chang Victor Gerling

30. Juni 2019

Universität Augsburg

Institut für Informatik

Lehrstuhl für Organic Computing

1. Aufgabe 02

2. Aufgabe 03

3. Aufgabe 04

Aufgabe 02

Wie werden Lösungen binär kodiert?

- Anhand des Wertebereichs und der Schrittweite wird ein unsigned integer generiert.
- Dieser wird mittels Integer.toBinaryString(i) (Java Funktion) in einen Binär-String umgewandelt.
- String wird evtl. mit zusätzlichen Nullen aufgefüllt, um gewünschte Stringlänge zu erreichen.
- Beispiel: siehe Code.

Wie wird Population initialisiert?

- Jeder Wert des Eingabe-Tupels wird zufällig anhand des Wertebereichs initialisiert.

Wie werden Eltern ausgewählt?

- Sortiere Population anhand des generierten Fitnesswertes.
- Erstelle so eine Rang-basierte Wahrscheinlichkeitsverteilung
 $\Rightarrow (1 - e^{-rank}) / (sizepopulation))$
- Ziehe mittels Roulette-Wheel Algorithmus n Eltern.
- Siehe Code.

Wie wird Crossover durchgeführt?

- Mittels uniform crossover.
- Entscheide bei jedem Bit zufällig, ob die Eltern ihr Bit vertauschen sollen.
- Siehe Code.

Wie wird Mutation durchgeführt?

- Setze Threshold für Mutation (0.05).
- Generiere bei jedem Bit Zufallszahl um zu entscheiden ob es zur Mutation kommt.
- Bitflip wenn Mutation.
- Siehe Code.

Wie wird Gray-Kodierung durchgeführt?

- Encoding: $n \text{ xor } (n \gg 1)$, z.B.: $2 \text{ xor } (2 \gg 1) \Rightarrow 3 \Rightarrow (11)_2$
- Decoding: $p := n$; while($n \gg 1 \neq 0$) do $p := p \text{ xor } n$; return p ;
- Siehe Code.

Aufgabe 03

Implementieren Sie nun einen GA für die Blackboxen, der direkt auf den Fließkommazahlen-Vektoren arbeitet! Nutzen Sie dabei ...

- Alles gleich wie A2 außer Mutation.
- Für Mutation: ziehe zufällige Zahl zwischen $[-step, step]$ und addiere sie zum jeweiligen Wert hinzu!

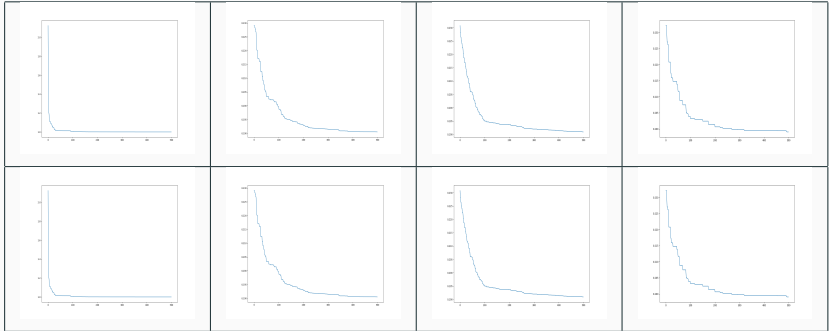
Aufgabe 04

Erwarten Sie, dass der GA mit Fließkommazahlen auf den Blackboxen besser oder schlechter funktioniert? Warum?

- Erwartung: Nein.
- Das modifizieren der Bits erlaubt eine feingranulare Anpassung der Werte.
- Eventuell langsamer um zu konvergieren, zum Ende hin jedoch potentiell präziser!

Entscheiden Sie sich bei jedem Verfahren für eine gute Parametrisierung und geben Sie diese, die Stelle des auf diese Weise beim Testen gefundenen Optimums sowie dessen Wert an!

- GA simple: Population: 600-1000, Parents: 40-80, Mutation-Thres: 0.05, step-size: 0.01-5
- GA gray: gleich
- GA float: gleich bis auf Mutation; hier step-size: 0.01-5



Oben: Naives Verfahren, GA simple, GA gray, GA float

Unten: HC, HC steepest, HC random restart, simulated annealing

⇒ bestes Verfahren: ga-simple mit 100!