



Evolutionäre Algorithmen

Übungsblatt 5, SS 2011

Abgabe: 23.05.11

Aufgabe 13 Rangbasierte Selektion - (8 Punkte)

Betrachten Sie das EA-Framework, das auf der Übungsseite verfügbar ist. Wenn Sie die Hauptklasse `Starter` ausführen, wird ein Fenster angezeigt, mit dem eine evolutionäre Optimierung eingeleitet werden kann. Rechts wird die erzielte Fitness über die Generationen t aufgetragen. Links können einige Parameter eingestellt werden, u. a. die Populationsgröße μ , die durchgeführten Generationen t_{Max} , Crossover-Wahrscheinlichkeit p_{Cross} , Mutationswahrscheinlichkeit p_{Mut} , Selektionsmethode `selM` und Selektionsparameter `selP`.

- (a) Betrachten Sie die Klasse `Selection`. Wie wirken die Selektionsarten `Dummy` und `Best`, die bereits implementiert sind? (2 Punkte)
- (b) Neben der Festlegung der Selektionswahrscheinlichkeit für ein Individuum ist das Sampling-Verfahren ausschlaggebend für die Zusammensetzung der Folgegeneration. Beschreiben Sie, wie beides in den Methoden `fitProp` und `stochasticUniversalSampling` realisiert wird. Was ist der Unterschied zur bekannten Roulette-Rad-Methode? (2 Punkte)
- (c) Fügen Sie rangbasierte Selektion mit linearer sowie nichtlinearer Skalierung hinzu, indem Sie die Methoden `linRanking` und `nonlinRanking` implementieren, welche auf *stochastic universal sampling* zurückgreifen. Gehen Sie dabei davon aus, dass der jeweilige Selektionsparameter (η^+ bzw. c) in der Variablen `selParam` der `Selection`-Instanz gesetzt ist. (4 Punkte)

Aufgabe 14 Crossover - (4 Punkte)

Gegeben sind die beiden Bitstrings 0100110110 und 1001101010. Führen Sie mit den Strings die folgenden Crossoverarten durch und berechnen Sie die Fitness beider neuer Strings bezüglich der Optimierungsprobleme f_{MiniBits} und $f_{\text{MiniBitsR}}$ (siehe unten). Alle Positionsangaben sind von links ab 1 gezählt. Crossoverpunkte bezeichnen Positionen zwischen Bits; beim Uniform-Crossover sind die Positionen der Bits selbst gegeben:

1. 1-Punkt-Crossover mit Crossoverpunkt 3
2. 2-Punkt-Crossover mit Crossoverpunkten 2 und 6
3. Uniformes Crossover: Bits Nummer 2, 3, 5 und 8 werden ausgetauscht.

Sei $b = b_1 \dots b_l$ ein Bitstring der Länge $|b| = l$, $b \in \{0; 1\}^l$ und $k = 0^d$ ein String aus d Nullen:

- $f_{\text{MiniBits}}(b) = \sum_{i=1}^l b_i$
- $f_{\text{MiniBitsR}}(b) = \max_d \{k = 0^d \text{ und } k \text{ ist Teilstring von } b\}$.

Aufgabe 15 Markov-Ketten - (8 Punkte)

Gegeben sei die Übergangsmatrix P . Darin bezeichnet p_{ij} (Zeile i , Spalte j) die Wahrscheinlichkeit, dass von Zustand z_i in Zustand z_j gewechselt wird.

P	z_1	z_2	z_3	z_4	z_5
z_1	0,2	0,1	0,4	0,1	0,2
z_2	0,0	1,0	0,0	0,0	0,0
z_3	0,4	0,0	0,3	0,0	0,3
z_4	0,0	0,0	0,0	1,0	0,0
z_5	0,4	0,3	0,1	0,1	0,1

(a) Bestimmen Sie die Wahrscheinlichkeiten für die folgenden Zustandsketten: (3 Punkte)

(i) $(x_1, \dots, x_6) = (z_1, z_3, z_3, z_5, z_1, z_2)$

(ii) $(x_1, \dots, x_6) = (z_5, z_3, z_1, z_2, z_1, z_5)$

Was fällt auf? Erklären Sie Ihre Beobachtung.

(b) Ordnen Sie P so, dass links oben eine quadratische Einheitsmatrix der absorbierenden Zustände steht. Berechnen Sie für jeden der transienten Zustände i als Initialzustand den Erwartungswert $E_i(t)$ für die Anzahl von Zuständen, die durchlaufen werden müssen, bis ein absorbierender Zustand erreicht ist. (4 Punkte)

(c) Wann besitzt ein GA absorbierende Zustände? (1 Punkt)

Allgemeiner Hinweis: Abgabe per E-Mail (andreas.jahn@uni-tuebingen.de) ist erwünscht, besonders für die Programmieraufgaben. Bitte die Klassen vorher testen, da Syntaxfehler zu Punktabzug führen. Die Klassen sollten ohne irgendwelche Extras mit dem JDK 1.6 laufen. Bitte grundsätzlich keine grafischen Oberflächen programmieren, denn dies kostet meist deutlich mehr Zeit als geplant. Wer es dennoch für unerlässlich hält, möge ausschließlich das AWT und Swing verwenden.