Anwendung evolutionärer Algorithmen Übungsblatt 3

Karsten Weicker, Prof. Volker Claus, Abgabetermin: 12.05.99, 13:15 Uhr

Denkaufgabe: Schemata

1. Sei die Fitneß eines binären Strings der Länge 4 die repräsentierte ganze Zahl (z.B. $f(0011) = 3$, $f(1111) = 15$). Was ist dann die durchschnittliche Fitneß des Schemas $1***$? Analog für die Schemata $0***$ und $**1*$?
2. Beweisen Sie, daß jeder binäre String der Länge l eine Instanz von genau 2^l verschiedenen Schemata ist.
3. Gibt es Fälle, in denen eine Population von n binären Strings der Länge l genau $n2^l$ verschiedene Schemata beinhaltet?

Programmieraufgabe: Evolutionsstrategien

Für Evolutionsstrategien stehen verschiedene Möglichkeiten zur Verfügung, die Mutation sich an den Suchraum anpassen zu lassen. Inwieweit diese Techniken eine vereinfachte Parametereinstellung mit sich bringen, ist in diesen Übungen zu testen.

Hierfür wird ein Rahmenprogramm `stdES` in C++ gestellt, welches leicht abzuändern ist. Die Quelldateien sind in der Datei `stdes.tar.gz` ist auf der WWW-Seite erhältlich. Die Datei ist mit `gunzip stdes.tar.gz` und einem anschließenden `tar xvf stdes.tar` zu entpacken. Dies erzeugt ein Verzeichnis `stdes`, in welchem sich die Dateien befinden.

Für das Programm sind die folgenden Optionen vorgesehen:

-m ⟨ganze Zahl⟩	Populationsgröße der Elternpopulation
-l ⟨ganze Zahl⟩	Populationsgröße der Kindpopulation
-g ⟨ganze Zahl⟩	Generationen
-d ⟨ganze Zahl⟩	Dimension
-s ⟨ganze Zahl⟩	Seed des Zufallszahlengenerators
-D ⟨reelle Zahl⟩	anfängliche Varianz der Mutation

-a ⟨ganze Zahl⟩	Art der Anpassung: 0 – Anpassung gemäß der 1/5-Regel 1 – Selbstanpassung mit uniformer Schrittweite 2 – Selbstanpassung mit separaten Schrittweiten
-F ⟨ganze Zahl⟩	Fitneßfunktion: 0 – Ackley-Funktion 1 – Sphären-Funktion 2 – eine weitere Funktion

Es liefert als Ergebnis das File `fitness.dat`, welches für die verschiedenen Generationen die beste Fitneß (2. Spalte), die mittlere Fitneß (3. Spalte) und die schlechteste Fitneß (4. Spalte) enthält.

Aufgaben

1. Im Programm wurde die separate Schrittweitenanpassung ausgelassen. Implementieren Sie diese in den Dateien `ind_separate.H` und `ind_separate.C`.

2. Ebenso sind nur die ersten beiden Fitneßfunktionen implementiert. Wählen Sie aus den hinten angegebenen Funktionen eine aus und implementieren Sie diese in der Datei `your_ff.C`.

3. Testen Sie ähnlich zu der Aufgabe des letzten Übungsblattes verschiedene Einstellungen für verschiedene Fitneßfunktionen. Variieren Sie dabei die anfängliche Schrittweite Delta für alle drei Anpassungsarten.

   Für die 30-dimensionalen Funktionen Sphäre und Ackley sowie die selbstgewählte Fitneßfunktion sind für jede Einstellung jeweils 10 verschiedene Läufe mit 10 verschiedenen Seeds für den Zufallszahlengenerator durchzuführen. Jedes Experiment läuft über 200 Generationen. Protokollieren Sie dabei aus den 10 Läufen die beste und die mittlere Fitneß jeweils in den Generationen 100 und 200.

   Über die WWW-Seite ist wieder ein Beispiel-Skript `computeaverage.tcl` erhältlich, welches für eine Parameterkombination die Experimente für verschiedene Zufalls-Seeds startet und den Durchschnitt für die beste, mittlere und schlechteste Fitneß berechnet und in der Datei `avg_fitness.dat` ablegt.

4. Interpretieren Sie die Resultate. Was sind gute Einstellungen für das Delta? Zeigt sich eine Überlegenheit einer speziellen Art der Anpassung?

Mögliche Fitneßfunktionen:

1. Fletcher and Powell:

$$
\begin{aligned}
f(x) &= \sum_{i=1}^{n} (A_i - B_i)^2 \\
A_i &= \sum_{j=1}^{n} (a_{ij} sin\alpha_j + b_{ij} cos\alpha_j) \\
B_i &= \sum_{j=1}^{n} (a_{ij} sinx_j + b_{ij} cosx_j)
\end{aligned}
$$

   mit $a_{ij}, b_{ij} \in [-100, 100], \alpha_j, x_j \in [-\pi, \pi]$.

2. Shekel-10:

$$
f(x) = -\sum_{i=1}^{m} \frac{1}{c_i + \sum_{j=1}^{n} (x_i - A_{ij})^2}
$$

   mit $n = 4$, $m = 10$, $0 \leq x_i \leq 10$ und

i	A_{ij}				c_i
	1	2	3	4	
1	4	4	4	4	0.1
2	1	1	1	1	0.2
3	8	8	8	8	0.2
4	6	6	6	6	0.4
5	3	7	3	7	0.4
6	2	9	2	9	0.6
7	5	5	3	3	0.3
8	8	1	8	1	0.7
9	6	2	6	2	0.5
10	7	3.6	7	3.6	0.5

3. Verrauschte Quartic function:

$$
f(x) = \sum_{i=1}^{n} i x_i^4 + N(0,1)
$$

   mit $n = 30$, $x_i \in [-1.28, 1.28]$ und $N(0,1)$ normalverteilte Zufallsvariable.

4. Kowalik:

$$f(x) \quad = \quad \sum_{i=1}^{11} \left(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right)^2$$

mit $n = 4$, $x_i \in [-5.0, 5.0]$ und

i	a_i	b_i^{-1}
1	0.1957	0.25
2	0.1947	0.5
3	0.1735	1
4	0.1600	2
5	0.0844	4
6	0.0627	6
7	0.0456	8
8	0.0342	10
9	0.0323	12
10	0.0235	14
11	0.0246	16

5. Claus-Hotz-Funktion: Diese Funktion ist auf Permutationen definiert. Wie kann ein reellwertiges Tupel auf eine Permutation dekodiert werden?

$$f(p) \quad = \quad -2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{\|p_j - p_i\|}{j - i}$$

wobei p eine Permutation ist.