



## Evolutionäre Algorithmen

### Übungsblatt 7, SS 2011

Abgabe: 07.06.11

#### Aufgabe 19 PBIL: Population-based incremental learning - (12 Punkte)

- (a) Implementieren Sie den PBIL-Algorithmus wie im Skript angegeben. Überlegen Sie dazu eine sinnvolle Representation für die Individuen als auch den Wahrscheinlichkeitsvektor  $\vec{P}(t)$ . Testen Sie Ihre Implementierung anhand des Mini-Bits-Problems, das Ihnen bereits aus Aufgabe 12 bekannt ist. (6 Punkte)
- (b) Vergleichen Sie die Konvergenzgeschwindigkeit Ihrer PBIL-Implementation mit dem Hill-Climber und der Monte-Carlo-Suche. Setzen Sie dabei für alle Verfahren sowohl die Populationgröße als auch die Problemgröße auf 50 bei einer 10 000-fachen Evaluierung. Die Lernrate  $\eta$  betrage 0,04 und  $\mu$  sei 10. (3 Punkte)
- (c) Wie müssen die Einstellungen des Hill-Climbers verändert werden, damit dieser deutlich schneller konvergiert als das PBIL-Verfahren? Begründen Sie Ihre Antwort. (1 Punkt)
- (d) Experimentieren Sie mit den PBIL-Parametern  $\eta$ ,  $\mu$  und finden Sie eine Konfiguration, die eine deutlich bessere Konvergenzgeschwindigkeit erzielt. Wie können Sie sich dieses Verhalten erklären? (2 Punkte)

#### Aufgabe 20 Bucket-Brigade-Algorithmus - (8 Punkte)

Führen Sie den Bucket-Brigade-Algorithmus für folgendes Classifier-System über  $t = 5$  Schritte aus (analog dem Beispiel im Skript), wobei die Stärke  $S_i(0)$  aller Classifier zu Beginn 1000 beträgt. Weiterhin sei  $C_{\text{bid}} = 0,1$  und die Bestrafung unproduktiver Regeln deaktiviert ( $C_{\text{tax}} = 0,0$ ). Die Initialnachricht sei 10101. Die Classifier seien:

- 1) 10#01 : 01101
- 2) 0110# : 00101
- 3) 11##1 : 11011
- 4) 101## : 00010
- 5) ##101 : 11101.

Ein Classifier bietet in dieser Aufgabe proportional zu seiner Stärke  $B_i = C_{\text{bid}} S_i$  ohne Berücksichtigung der Spezifität. Jeder Classifier sendet seine Nachricht, wenn ein Match mit seiner Bedingung auftritt (nicht nur der höchstbietende Classifier). Nur in diesem Fall zahlt der Classifier auch sein Gebot  $B_i(t)$ . Der Reward für die Regel  $j$ , welche die Nachricht gesendet hatte, die zum Match der Bedingung des Classifiers  $i$  führte, berechnet sich zu:

$$\forall j \in \text{Enable}(i) \quad R_{ij}(t) = \frac{1}{|\{j | j \in \text{Enable}(i)\}|} B_i(t),$$

wobei der endgültige Reward die Summe aller Erträge über die Classifier  $i$  folgendermaßen berechnet wird:  $R_j(t) = \sum_i R_{ij}(t)$ . Zu jedem Zeitschritt ist neben der Liste der neuen Nachrichten die Stärke des jeweiligen Classifiers  $S_i(t)$ , sein Gebot  $B_i(t)$  und sein Reward  $R_i(t)$  anzugeben. Stellen Sie Ihre Ergebnisse in Tabellenform dar. Bei Geboten mit Nachkommastellen sollten Sie zur Vereinfachung auf ganze Zahlen runden.

**Allgemeiner Hinweis:** Abgabe per E-Mail ([andreas.jahn@uni-tuebingen.de](mailto: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.