

9. Übungsblatt

besprochen am 20., 22. und 23.06.2016

Aufgabe 1 Genetische Programmierung: Terminal- und Funktionssymbole

Geben Sie Terminal- und Funktionssymbole \mathcal{T} und \mathcal{F} an, um folgende Probleme sinnvoll durch eine genetische Programmierung zu lösen:

- die Klassifikation von Hunden als Terrier (20-40 cm groß), Kleinpudel (35-45 cm groß), Zwergpudel (28-35 cm groß), oder deutsche Schäferhunde (55-65 cm groß),
- die Herleitung des dritten Keplerschen Gesetzes: „Die Quadrate der Umlaufzeiten T_1 und T_2 zweier Planeten um die Sonne sind proportional zu den dritten Potenzen der großen Halbachsen a_1 und a_2 ihrer Bahnen um die Sonne.“

Aufgabe 2 Genetische Programmierung: Kreuzen/Crossover

Geben Sie einen Algorithmus an, der aus zwei gegebenen (Eltern-)Lisp/Scheme-Ausdrücken (also symbolischen Ausdrücken in Präfixnotation) durch Crossover einen neuen Lisp/Scheme-Ausdruck erzeugt! Geben Sie bei dieser Aufgabe den Algorithmus in Lisp oder Scheme an, wenn Sie eine dieser Sprachen beherrschen. Verwenden Sie ansonsten Pseudocode. (Hinweis: Zerlegen Sie das Problem in zwei Teilprobleme:

- die zufällige Auswahl eines Teilausdrucks und
- das Einfügen eines Ausdrucks in einen anderen an einer zufällig gewählten Stelle.)

Wie kann man mit dieser Funktion bzw. einer Teilfunktion gleichzeitig die Mutationsoperation bereitstellen?

Aufgabe 3 n -Damen-Problem

In der Vorlesung wurde das auf der Vorlesungsseite verfügbare Programm `qga.c` besprochen, welches das n -Damen-Problem mithilfe eines evolutionären Algorithmus löst, sowie das Programm `queens.c`, welches das n -Damen-Problem mit Backtracking löst. Experimentieren Sie etwas mit diesen Programmen und beantworten Sie die folgenden Fragen:

- Kann man (bei diesem Problem) eher auf das Crossover oder auf die Mutation verzichten? Wie erklären Sie sich ihre Beobachtungen?
Hinweise: Das Crossover kann man ausschalten, indem der Bruchteil der dieser Operation unterworfenen Individuen durch Angabe von `-f0` auf 0 gesetzt wird. Zum Ausschalten der Mutation setzt man die Mutationswahrscheinlichkeit durch Angabe von `-m0` auf 0.
- Ist der Defaultwert der Mutationswahrscheinlichkeit (0.1 bzw. 10%) optimal? Welche Werte führen zu besseren Ergebnissen?
- Vergleichen Sie die Laufzeiten der beiden Programme für $n = 10, 20$ und 30 ! Verwenden Sie für die Mutationswahrscheinlichkeit einen der Werte, die sich in Teilaufgabe b) als gut herausgestellt haben. Erhöhen Sie ggf. die Anzahl der zu berechnenden Generationen (mit `-gAnzahl`).

Aufgabe 4 Packprobleme

Gegeben seien n Objekte, die in k Behälter verpackt werden sollen. Die Größen der Objekte und der Behälter seien bekannt. Vereinfachend nehmen wir an, dass die Abmessungen der Objekte/Behälter keine Rolle spielen, d.h., dass ein Objekt in einen Behälter gepackt werden kann, wenn nur der verbleibende Restraum größer ist als das Objekt.

- a) Wie kann man dieses Problem durch einen evolutionären Algorithmus lösen? Welche (anderen) Möglichkeiten der Kodierung von Lösungskandidaten fallen Ihnen ein? Welche Vor- und Nachteile haben diese Kodierungen?
- b) Wie kann man den Ansatz so verallgemeinern, dass nach einer Lösung gesucht wird, die möglichst wenige Behälter verwendet?
- c) Wie kann man das Packproblem durch genetische Programmierung lösen? Gehen Sie speziell auf die Terminal- und Funktionssymbole ein.