

Praktika 2

Aufgabe 1

Kodierung

[1, 3, 2, 4]

Selektion

[3, 1, 4, 2]

Rekombination

[1, 3, | 2, 4] + [3, 1, | 4, 2] = [4, 2, 1, 3]

Mutation

[4, 2, 1, 3] → [5, 2, 1, 3]

Zweite Möglichkeit : [1, 3, 2, 4] → [3, 1, 4, 2] Das erste Individuum wechseln zum 2. Spielfeld da der 2. Spielfeld richtig ist

Ist der Lösung gültig : [5, 2, 1, 3] → Ja / [3, 1, 4, 2] → Ja

Fitnessfunktion:

Die Fitness bewertet, wie viele Konflikte zwischen den Königinnen auftreten.

Zwei Königinnen stehen im Konflikt, wenn sie entweder in derselben Zeile stehen oder sich auf derselben Diagonale befinden.

Einfärben von Landkarten

Kodierung: Array [A, B, C, D, E, F]

Farben: [Rot,Blau,Grün]

Rekombination [A,B,C,D,E,F,Rot,Blau,Grün]

Mutation: [A Rot , B Blau , C Grün , D Rot , F Blau , E Blau]

Fitnessfunktion:

Ich zähle, wie viele Konflikte in der Färbung vorkommen.

Ein Konflikt liegt vor, wenn zwei benachbarte Länder die gleiche Farbe haben.

Aufgabe 2

LandKarte

Populationsgröße	Mutationsrate	Durchschnittliche Konflikte	Beste Fitness
20	0.01	1.2	0.85
50	0.05	0.5	0.94
100	0.10	0.0	1.00

8queen

Populationsgröße	Mutationsrate	Durchschnittliche Konflikte	Beste Fitness
20	0.01	7.1	0.86
50	0.05	2.3	0.96

100	0.10	0.1	1.00
-----	------	-----	------

Aufgabe 3

Where's Waldo

Bei „Where's Waldo“ hat Randal Olson geschaut, an welchen Stellen Waldo in den Büchern vorkommt. Diese Punkte hat er als Koordinaten gespeichert. Dann hat er mit einem **genetischen Algorithmus** versucht, eine **möglichst kurze Suchroute** zu finden, die alle diese möglichen Stellen abläuft.

Jede mögliche Reihenfolge der Punkte ist dabei eine Lösung. Die **Fitness** ist die **Gesamtlänge des Weges**: Je kürzer der Weg, desto besser. Die besten Lösungen werden immer wieder leicht verändert, bis eine sehr gute Suchreihenfolge entsteht.

Das Ergebnis zeigt eine einfache Strategie:

Erst unten links suchen, dann oben rechts, danach unten rechts.

Dort taucht Waldo am häufigsten auf.

American Fuzzy

American Fuzzy Lop (AFL) ist ein Programm, das automatisch nach Fehlern in Software sucht. Es gibt dem Programm immer wieder leicht veränderte Eingabedateien und beobachtet, ob das Programm dadurch **neue Stellen im Code** ausführt oder sogar abstürzt.

AFL benutzt dabei **genetische Algorithmen**:

Die Eingabedateien gelten als „Individuen“. Wenn eine Eingabe eine neue Stelle im Code erreicht, wird sie als **gut** bewertet und für neue Varianten weiter benutzt. Schlechte Eingaben werden verworfen.

Die **Fitness** ergibt sich also daraus, wie **viel neuer Programmcode** durch eine Eingabe ausgeführt wird – je **mehr neue Codewege**, desto besser.

AFL ist praktisch, weil es:

ohne große Einstellungen funktioniert,

sehr schnell ist,

echte Fehler in realer Software findet.

Es wird vor allem benutzt, um **Sicherheitslücken** (Crashes, Speicherfehler) in Programmen aufzudecken.

Evolution Simulator

Im **Evolution Simulator** gibt es viele kleine Kreaturen. Sie sollen sich bewegen und möglichst **weit kommen**.

Die, die **weiter laufen**, sind **besser**. Von diesen guten Kreaturen werden **neue** gemacht, die ihnen ähnlich sind. Dabei werden kleine Änderungen gemacht, damit sie vielleicht **noch besser** werden.

Das passiert **immer wieder** über viele Generationen. So lernen die Kreaturen langsam, **besser zu laufen**.