



Evolutionäre Algorithmen

Prof. Dr. H. A. Kestler, Dr. J. Kraus , Dr. J. Schwab, Dr. P. Thiam
SoSe 2023

Übungsblatt 8

Abgabe bis 26.07.2023, 24 Uhr

Die Übungsaufgaben sollen mit der Skript-Sprache R durchgeführt werden. Reichen Sie Ihre Lösungen bis zum Abgabetermin auf Moodle ein.

1. Aufgabe (30): Genetic Programming (GP)

In dieser Aufgabe sollen Sie mittels Genetic Programming (GP) einen Agenten `agent(mat)` generieren, der sich durch ein Labyrinth bewegen kann. Ein Beispiel-Code ist in `labyrinth.R` zu finden. Der Agent erhält Zugriff auf sein Nahumfeld `mat` (3×3 Matrix) und soll die nächste Bewegungsrichtung ('l', 'r', 'o', 'u') bestimmen. Die einzelnen Felder der Matrix `mat` können die Werte 'S' = Start, 'E' = Ende, 'W' = Wand und 'F' = Frei annehmen. Beispiel-Labyrinth sind in `mazes01.RData` - `mazes05.RData` zu finden (Abbildung 1).

Ihr GP Algorithmus soll folgende Symbole verwenden

$$F = \{ \begin{array}{l} \text{if.S(pos, IF, ELSE)}, \\ \text{if.E(pos, IF, ELSE)}, \\ \text{if.W(pos, IF, ELSE)}, \\ \text{if.F(pos, IF, ELSE)} \end{array} \}$$

$$T = \{ \text{'l', 'r', 'o', 'u'} \}.$$

Hierbei gilt $\text{pos} \in T$, $\text{IF, ELSE} \in T \cup F$.

Die Funktional-Symbole F können wie folgt interpretiert werden (Bsp.: `if.W(pos, IF, ELSE)`):

Wenn an der benachbarten Position `pos` ein 'W' zu finden ist, führe IF aus ansonsten ELSE.

Nutzen Sie die Fitness-Funktion aus `labyrinth.R`:

```
fitness <- function(agent, labyrinth, k=1000).
```

Sie zählt die Schritte, die der Agent `agent` von 'S' nach 'E' braucht (max k=1000 Schritte).

Erzeugen Sie einen GP mit 100 Individuen und 1000 Generationen. Implementieren Sie eine Mutations- und eine Rekombinationsstrategie. Visualisieren Sie Ihren Agenten mit Hilfe von `plot.gesamt()`, ebenfalls in `labyrinth.R`.

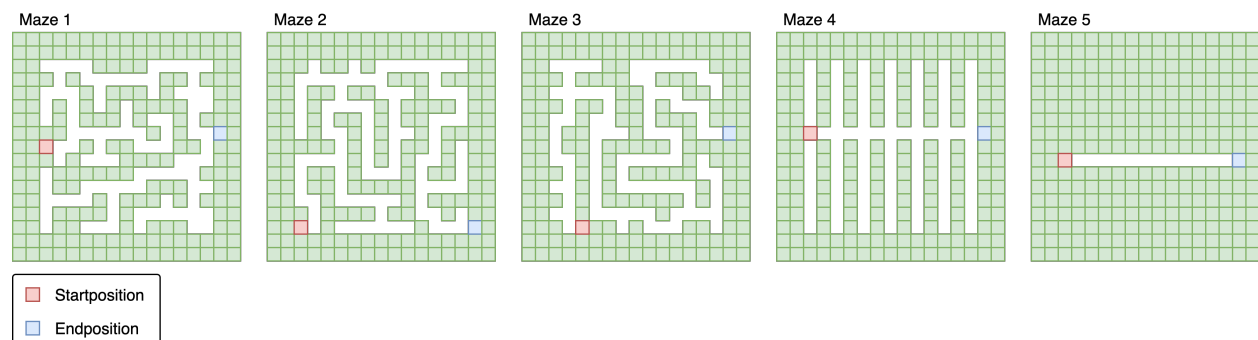


Abbildung 1: Labyrinth