

Die Hausaufgaben müssen einzeln bearbeitet und abgegeben werden. Geben Sie die schriftlich zu bearbeitenden Aufgaben als pdf-Dateien ab.

Abgabefrist ist der 3.12.2023 – 23:59 Uhr

Abgabe

Wir benutzen für die Abgabe der Hausaufgaben Git. Jedes Repository ist nur für den Studierenden selbst sowie für die Betreuer und Korrektoren sichtbar.

Für die Hausaufgabe benötigen Sie **kein** neues Repository. Es wird das Repository benutzt, das für Übungsblatt 2 angelegt wurde.

Damit das Repository im Laufe der Vorlesung übersichtlich bleibt und wir sinnvoll korrigieren können, achtet bitte auf das Folgende:

- Der finale Commit für ein Übungsblatt muss mit einem Tag oder einer klaren Commit-Message versehen sein, der/die diesen Commit eindeutig als die finale Abgabe erkennen lässt.
- Erstellt **pro Übungsblatt** einen Ordner, in dem die Files liegen, die kein Code sind (Textfiles, Plots, ...).
- Benennen Sie Ihre Datei, in der Sie die Aufgabe 3 lösen (Maximierung der Rastrigin-Funktion), als `rastrigin.py`.

Nicht oder zu spät gepushte (Teil-)Abgaben werden mit 0 Punkten bewertet!

Aufgabe 1 – Kodierung (20P)

Bei dem [Behälterproblem](#) geht es darum, Objekte verschiedener Größen auf möglichst wenige Behälter zu verteilen, ohne dass ein Behälter überläuft.

Kodieren Sie dieses Problem, indem Sie Lösungskandidaten als (i) Bitstrings, (ii) Permutationen, (iii) Graphen und (iv) Vektoren natürlicher Zahlen repräsentieren. Erklären Sie jeweils

1. die Kodierungs- und die Dekodierungsfunktion;
2. welche Vor- oder Nachteile Sie für diese Kodierung in Bezug auf die Anwendung eines evolutionären Algorithmus sehen.

Aufgabe 2 – Erweiterung Implementierung SGA (30P)

Erweitern Sie Ihre Implementierung des Simple Genetic Algorithm von Blatt 2 um die folgenden Funktionen:

1. Führen Sie ein Archiv ein, in dem die beste bisher gefundene Lösung über den gesamten Algorithmus hinweg gespeichert wird, und geben Sie diese am Ende des Algorithmus zusätzlich aus.
2. Fügen Sie Ihrem Simple Genetic Algorithm einfache logging-Funktionalitäten hinzu. Es soll eine Datei entstehen, in der für jede Iteration die folgenden Informationen stehen:
 - a) das beste Individuum mit seiner Fitness;
 - b) die durchschnittliche Fitness der Population;
 - c) die Laufzeit der Iteration.

Die Ausgabe darf in eine json-, xml-, txt- oder csv-Datei erfolgen.

Aufgabe 3 – Verwendung SGA (20P)

In dieser Aufgabe benutzen wir den *Simple Genetic Algorithm*, um eine Funktion mit zwei natürlichen Zahlen als Parameter zu maximieren. Wir betrachten die **Rastrigin-Funktion**, die häufig als Testproblem genutzt wird, um die Performanz von Optimierungsalgorithmen zu bewerten. Die 2-dimensionale Rastrigin-Funktion ist definiert als

$$f(x_1, x_2) = 20 + \sum_{i=1}^2 [x_i^2 - 10 \cos(2\pi x_i)].$$

Verwenden Sie Ihre Implementierung des SGA, um die Rastrigin-Funktion auf dem Gebiet $[0, 1023] \times [0, 1023] \subseteq \mathbb{N}^2$ zu maximieren und dabei die Kodierung als Binärzahl bzw. per Gray-Code miteinander zu vergleichen. Gehen Sie in den folgenden Schritten vor.

1. Implementieren Sie eine Auswertung der Fitness für eine Kodierung Tupel natürlicher Zahlen als Binärzahl bzw. im Gray-Code. Definieren Sie also die Rastrigin-Funktion und implementieren Sie eine Dekodierung von Tupeln von Binärzahlen bzw. Zahlen im Gray-Code in ein Tupel natürlicher Zahlen.
2. Führen Sie die folgenden Experimente je einmal mit einer Crossoverwahrscheinlichkeit von 0 (kein Crossover) und einmal mit einer von 0.4 durch (wir wollen auch die Auswirkungen des Crossover-Operators beobachten):

Lassen Sie mit den gleichen Werten für die anderen relevanten Parameter (Mutationswahrscheinlichkeit, Populationsgröße, Anzahl an Iterationen) Ihren SGA **je zehnmal** für die beiden unterstützten Kodierungen (Binärzahl bzw. Gray-Code) laufen.

3. Berechnen Sie für alle vier Set-ups für jede Iteration a) den Durchschnitt (Median) über die 10 Durchläufe über die durchschnittliche Fitness (Median) und b) den Durchschnitt (Median) über die Fitness der besten Lösung. Plotten Sie jeweils die Entwicklung der durchschnittlichen Fitness gegen die Anzahl der Iterationen.

4. Bewerten Sie Ihr durchgeführtes Experiment:

- a) Beschreiben Sie (ohne Wertung!) Ihre Beobachtungen: Wie unterscheidet sich das Verhalten des SGA für die beiden Kodierungen und mit oder ohne Anwendung des Crossoveroperators?
- b) Ziehen Sie Schlussfolgerungen aus Ihren Beobachtungen: Welches Set-up war vorteilhaft?