

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

Abgabefrist ist der 28.01.2024 – 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, ...).

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

Aufgabe 1 – Implementierung Terminationskriterium (10P)

Implementieren Sie das folgende Terminationskriterium: Ein evolutionärer Algorithmus soll terminieren, wenn sich die *Fitness der besten gefundenen Lösung* über T Iterationen hinweg nicht um mehr als einen Faktor θ verbessert hat; T und $1 < \theta$ sollen hierbei Parameter des Algorithmus werden. Ist T_0 die aktuelle Iteration, x_{\max}^i die beste Lösung der Population nach der i -ten Iteration und f die betrachtete Fitnessfunktion, so soll der Algorithmus also terminieren, sobald

$$f(x_{\max}^{T_0}) \leq \theta \cdot f(x_{\max}^i)$$

für alle i zwischen $T_0 - T$ und $T_0 - 1$ gilt.

Sie dürfen die Möglichkeit, dieses Terminationskriterium zu benutzen, als Parameter in Ihren SGA integrieren oder, ausgehend von ihrem SGA, einen neuen evolutionären Algorithmus implementieren, der das Terminationskriterium nutzt.

Aufgabe 2 – Implementierung Selektionskriterien (20P)

Implementieren Sie die folgenden Selektionsmechanismen:

1. Implementieren Sie die linear-skalierte rangbasierte Berechnung von Selektionswahrscheinlichkeiten (Folie 227 f.) als Alternative zur fitness proportionate selection. Der zugehörige Parameter s soll ein Parameter des evolutionären Algorithmus werden.

2. Implementieren Sie *Stochastic Universal Sampling (SUS)* (Folie 231 f.) als Alternative, um Individuen gemäß einer gegebenen Selektionswahrscheinlichkeit zu sampeln.
3. Implementieren Sie Tournament Selection als Alternative zur Berechnung des mating pools (parent selection). Der Parameter k (Turniergröße) soll Parameter des evolutionären Algorithmus werden. Der Parameter λ (Anzahl der zu wählenden Elemente) kann u. U. erschlossen werden, abhängig davon, in welchen evolutionären Algorithmus Sie Tournament Selection einbauen (siehe unten).
4. Implementieren Sie Plus- **und** Komma-Selektion (Folie 241 f.) als Alternativen bei der survivor selection; die Anzahl der pro Iteration zu erzeugenden Elemente (λ) soll ein Parameter des evolutionären Algorithmus werden.

Sie dürfen die verschiedenen Implementierungen auf unterschiedliche (sinnvoll benannte!) evolutionäre Algorithmen aufteilen oder einen evolutionären Algorithmus implementieren, der über Parameter steuerbar ist. Rahmenbedingungen (vergleiche die praktischen Anwendungen in der nächsten Aufgabe):

- Jede der hier implementierten Selektionsvarianten soll gemeinsam mit dem in Aufgabe 1 implementierten Terminationskriterium nutzbar sein (Konvergenz der Fitness).
- Ihre Algorithmen, die Selektionswahrscheinlichkeiten verwenden (fitness proportionate oder rangbasiert), sollen jeweils SUS als Sampling-Methode nutzen können.
- Komma- und Plus-Selektion als Methoden zur survivor selection sollen Tournament Selection als Methode der parent selection kombinierbar sein.

Aufgabe 3 – Experimente (20P)

Wählen Sie eins der Optimierungsprobleme, die wir auf den bisherigen Übungsblättern bearbeitet haben, aus und optimieren es mit unterschiedlich konfigurierten evolutionären Algorithmen erneut. Terminationsbedingung soll jeweils sein, dass es für 100 Iterationen keine Verbesserung um den Faktor θ mehr gab (wählen Sie hierbei θ passend zum gewählten Optimierungsproblem). Mutations-, Crossoverwahrscheinlichkeit und Populationsgröße sollen in allen Experimenten gleich gewählt sein (wählen Sie hier Werte, die in Ihren bisherigen Experimenten vielversprechend waren). Betrachten Sie die folgenden vier Varianten:

1. Variation im Samplingprozess im Vergleich zum SGA:
 - parent selection: Berechnung der Selektionswahrscheinlichkeiten fitness proportionate, aber Samplingvorgang per SUS
 - survivor selection: generational model des SGA (stimmt mit (μ, μ) überein)
2. Verwendung von rangbasierter Selektion im SGA:
 - parent selection: linear-skalierte rangbasierte Berechnung der Selektionswahrscheinlichkeit (mit $s = 2$), Samplingvorgang per SUS
 - survivor selection: generational model des SGA (stimmt mit (μ, μ) überein)
3. Verwendung von Turnierselektion im SGA:
 - parent selection: Turnierselektion, wobei die Turniergröße ca. einem Zehntel der Populationsgröße entsprechen soll

- survivor selection: generational model des SGA (stimmt mit (μ, μ) überein)

4. Verwendung von Plus-Selektion

- parent selection: Turnierselektion, wobei die Turniergröße ca. einem Zehntel der Populationsgröße entsprechen soll
- survivor selection: Plus-Selektion $(\mu + \lambda)$, wobei $\frac{\lambda}{\mu} \geq 3$ gelten soll

Führen Sie die Optimierung wieder in jedem Set-Up zehnmal durch und plotten Sie die durchschnittliche Entwicklung der Fitness der besten Lösung und der durchschnittlichen Fitness. Diskutieren Sie knapp (und für alle Experimente gesammelt), ihre Beobachtungen und vergleichen Sie die Ergebnisse der Experimente.