Evolucijsko računanje

Vaje 2025/2026

1 Optimizacijski problemi in naključno iskanje

1.1 Testni optimizacijski problemi

V prvem delu naloge boste implementirali nekaj preprostih testnih optimizacijskih problemov. Pripravite razred Problem, ki bo hranil število dimenzij (**d**), spodnjo in zgornjo mejo (lowerBound in upperBound) ter ime (name). Razred naj ima abstraktno metodo evaluate, ki za podan x izračuna fitnes. Dodajte tudi metodo generateRandomSolution, ki generira naključnega posameznika. Implementirajte naslednje minimizacijske testne probleme:

- Sphere $L_b = \{-100 \text{ if } i \text{ is even}, -10 \text{ if } i \text{ is odd}\}_{i=1}^{d}$ $U_b = \{100 \text{ if } i \text{ is even}, 10 \text{ if } i \text{ is odd}\}_{i=1}^{n}$
- Ackley $L_b = -32.768$ $U_b = 32.768$ for all dimensions
- Griewank $L_b = -600$ $U_b = 600$ for all dimensions
- Rastrigin $L_b = -5.12$ $U_b = 5.12$ for all dimensions
- Schwefel26 $L_b = -500$ $U_b = 500$ for all dimensions
- Rosenbrock $L_b = -5$ $U_b = 10$ for all dimensions
- Trid $L_b = -d^2$ $U_b = d^2$ for all dimensions
- Bukin $L_b = [-15, -3]$ $U_b = [-5, 3]$
- Carrom Table $L_b = -10$ $U_b = 10$ for all dimensions
- Styblinski-Tang $L_b = -5$ $U_b = 5$ for all dimensions, global minimum: $f(x^*) = -39.16616570377142d$, at $x^* = (-2.90353401818596, \ldots, -2.90353401818596)$
- Levy $L_b = -10$ $U_b = 10$ for all dimensions
- Michalewicz $L_b = 0$ $U_b = \pi$ for all dimensions

Za vsak problem uporabite spodnje in zgornje meje, ki so navedene tukaj, in ne tistih, ki so podane na povezanih virih!

Seznam uporabljenih spletnih strani za dodatno pomoč pri implementaciji problemov:
Virtual Library of Simulation Experiments: Test Functions and Datasets
Infinity77 Global Optimization Test Functions
Al-Roomi Benchmark Functions for Unconstrained Optimization

Za preverjanje pravilnosti implementacije problemov boste napisali Unit teste. Za vsak problem preverite, ali se fitnes za globalni optimum izračuna pravilno. Če problem omogoča nastavljanje poljubnega števila dimenzij, napišite teste za 2, 5 in 10 dimenzij. Na primer, za problem Sphere, če pokličemo metodo evaluate s pozicijo globalnega optimuma $x^* = (0, \ldots, 0)$, mora metoda vrniti $f(x^*) = 0$. Pri izračunu fitnes funkcij lahko pride do majhnih zaokrožitvenih napak, ki lahko vplivajo na rezultate pri primerjavi rešitev. Da zagotovite natančnost in pravilnost pri primerjavah, nastavite toleranco za primerjavo neposredno v Unit testih ali ročno izračunajte absolutno razliko. Natančnost naj bo nastavljena na 1e-7.

1.2 Naključno iskanje

Implementirajte algoritem naključnega iskanja (Random Search). Dodajte razrede Solution, Algorithm in RandomSearch. Razred Solution naj hrani pozicijo in fitnes rešitve. Razred Algorithm naj vsebuje eno abstraktno metodo execute, ki prejme objekt tipa Problem in zaustavitveni pogoj maxFes (maksimalno število ovrednotenj), ter vrne objekt tipa Solution. Razred RandomSearch naj deduje iz razreda Algorithm.

Psevdokod algoritma Random Search:

Algorithm 1 Pseudocode for Random Search

Input: $Problem$: containing the fitness function to optimize
Input: $maxFes$: maximum number of evaluations
Output: $bestSolution$: the best solution found
Initialize $bestSolution \leftarrow \emptyset$
for $i = 1$ to $maxFes$ **do**
  $candidate_i \leftarrow$ generate random solution from Problem
  if $candidate_i.fitness$ is better than $bestSolution.fitness$ **then**
    $bestSolution \leftarrow candidate_i$
  end if
end for
  return $bestSolution$

Za analizo uspešnosti algoritma je potrebno izvesti statistiko rezultatov. Ustvarite ločen razred StatisticsUtility, ki bo vseboval metode za: iskanje najboljše rešitve, izračun povprečja in izračun standardnega odklona. Algoritem RandomSearch zaženite 100-krat na vsakem testnem problemu in zabeležite najboljšo rešitev vsakega zagona, pri čemer nastavite parameter maxFes na **3000 * d**. Če je pri določenem problemu možno nastaviti več dimenzij, poženite algoritem za dimenzije 2, 5 in 10. Z uporabo razreda StatisticsUtility izpišite za vsak problem najboljšo rešitev (minimalna vrednost), povprečje, in standardni odklon testni problem in dimenzijo.

Primer izpisa rezultatov za problem Sphere z dimenzijo 2:

```
Problem: Sphere, Dimensions: 2 Min: 5.08e-57, Average: 0.045, Std: 0.012
```

* Naloga je **obvezna** in vredna **15 točk**.