

## Genetický algoritmus – Hľadanie globálneho minima Genetic Algorithm

### Obsah

<b>Úloha 2: Genetický algoritmus pre 10-D optimalizáciu</b>	<b>2</b>
Cieľ úlohy . . . . .	2
Postup riešenia . . . . .	2
<b>Odrovzdanie a hodnotenie</b>	<b>4</b>
Dokumentácia výsledkov (PDF) . . . . .	4
Obhajoba riešenia . . . . .	4
<b>Odporeúčané nástroje a technická realizácia</b>	<b>5</b>
Matlab . . . . .	5
Python . . . . .	5
<b>Teoretické poznámky</b>	<b>6</b>
Selektívny tlak a diverzita populácie . . . . .	6
Elitizmus a „výber najlepšieho“ . . . . .	6
Poznámka k terminológii výberu . . . . .	6

## Úloha 2: Genetický algoritmus pre 10-D optimalizáciu (4b + 1b)

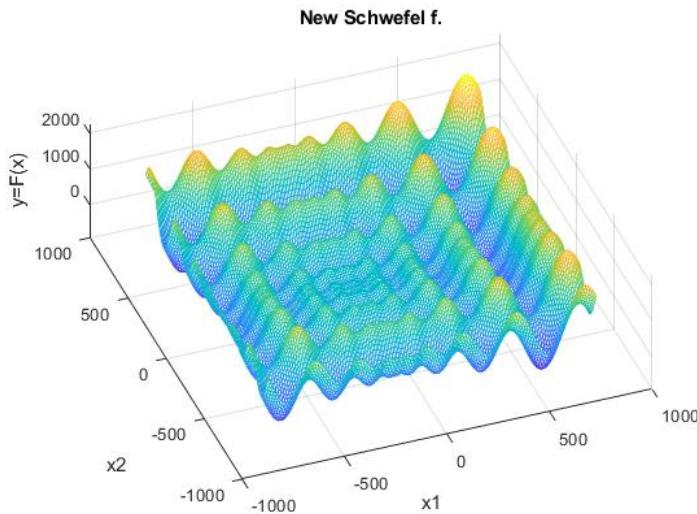
### Cieľ úlohy

Najdite globálne minimum Novej Schwefelovej funkcie pre 10 premenných (`testfn3c`, 10-D úloha) pomocou genetického algoritmu.

Jedinec (chromozóm) reprezentuje vektor:

$$X = [x_1, x_2, x_3, \dots, x_{10}]$$

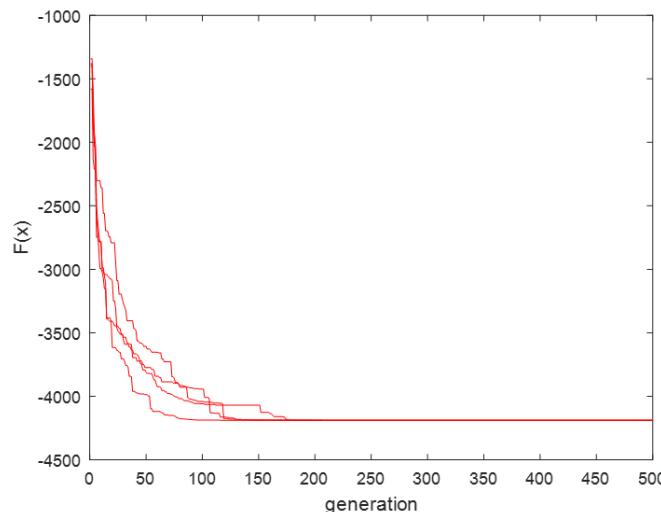
pričom optimalizácia prebieha nad všetkými 10 premennými súčasne.



Obr. 1: Ilustrácia zadania pre 10-D úlohu: optimalizovaný vektor  $X = [x_1, \dots, x_{10}]$ .

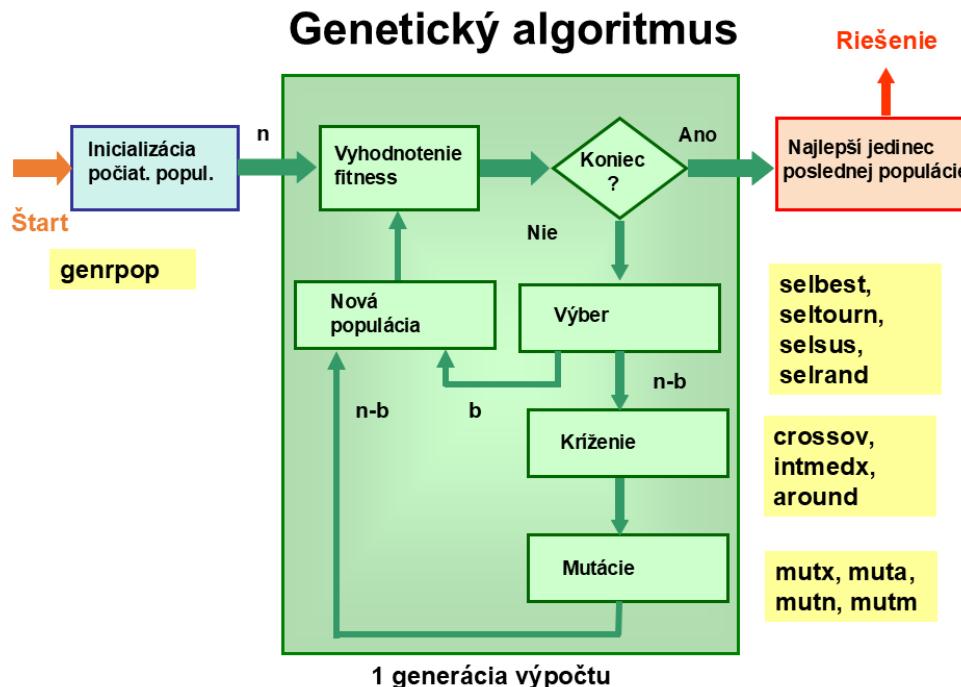
### Postup riešenia

1. Pozrite si prednášku o evolučných a genetických algoritnoch (GA). Ako implementačný základ použite odporúčaný manuál k toolboxu `genetic`.
2. Implementujte vlastný riadiaci program (hlavnú slučku GA), ktorý bude realizovať iteráciu cez generácie. Jednotlivé genetické operácie (napr. selekcia, kríženie, mutácie a súvisiace pomocné funkcie) použite z dodaného `Genetic Toolbox` (resp. `genetic_all.py`).
3. Spustite GA a vykreslite priebeh fitness funkcie v závislosti od počtu generácií. Vypíšte súradnice optimálneho jedinca (gény chromozómu) a jeho zodpovedajúcu hodnotu fitness. Teoretická hodnota globálneho minima je približne  $f_{\min} \approx -792.72 \cdot n$  (pre  $n = 10$  teda približne  $-7927.2$ ). Vzhľadom na stochastický charakter algoritmu sa za postačujúci výsledok považuje, ak aspoň v jednom zo spustení dosiahne fitness hodnotu  $-7920$  alebo nižšiu.
4. GA spustite viackrát, všetky priebehy vykreslite do jedného prehľadného grafu a výsledky porovnajte, urobte z nich stručný záver.
5. **Experimentovanie s parametrami:** Vyskúšajte zmeniť vybrané parametre genetických operácií (napr. pravdepodobnosť mutácie, pravdepodobnosť kríženia, veľkosť populácie, počet generácií alebo spôsob výberu). V správe uveďte, ktoré parametre ste menili, a popíšte pozorovaný vplyv na kvalitu riešenia a priebeh konvergencie. Osobitne sledujte správanie algoritmu pri nevhodne zvolenej (príliš nízkej alebo príliš vysokej) miere mutácie.
6. Výsledky aj program archivujte pre potreby prezentácie.



Obr. 2: Ilustračný príklad: graf evolúcie fitness pre viac spustení.

**Schéma genetického algoritmu.** Pre lepšiu orientáciu v implementácii je nižšie uvedená schéma základného cyklu genetického algoritmu (populácia → vyhodnotenie → selekcia → kríženie → mutácia → nová generácia). Pri obhajobe je potrebné vedieť stručne vysvetliť, ako tieto kroky realizuje vaša implementácia a aké parametre ste použili.



Obr. 3: Schéma genetického algoritmu (GA)

### Bonusová úloha (+1 bod)

Otestujte váš algoritmus na zložitejších problémoch. Pre zisk bonusového bodu musíte úspešne vyriešiť a do vašej dokumentácie (PDF) spracovať **obe** nasledujúce podúlohy:

- a) Nájdenie minima **Novej Schwefelovej funkcie** rozšírenej na **100 premenných**.
  - b) Nájdenie minima funkcie **Eggholder** pre **10 premenných** (funkciu eggholder nájdete v **Genetic Toolbox**).
- 

### Odovzdanie a hodnotenie

Hodnotenie zadania prebieha formou osobnej konzultácie. Zadanie sa odovzdáva spravidla **na začiatku cvičenia** po odprezentovaní nasledujúceho zadania.

Počas semestra môže študent nahradíť najviac **dve** cvičenia. V takom prípade je možné zadanie odovzdať aj na cvičení, na ktorom študent nahrádza. V ostatných prípadoch sa zadanie odovzdáva vždy **svojmu cvičiacemu**.

### Dokumentácia výsledkov (PDF)

Pripavte si stručnú správu (napr. MS Word / PDF), ktorá obsahuje grafy z viacerých spusťení. Dokument slúži ako podklad pre diskusiu, aby nebolo nutné čakať na opakovane výpočty počas obhajoby zadania.

Pre toto zadanie uveďte v správe aj: (1) graf priebehu fitness pre viac behov v jednom obrázku, (2) použité nastavenia GA (napr. zvolené parametre a krátke popis experimentovania), (3) vypísané výsledné riešenie (najlepší jedinec) a jeho fitness, (4) krátky komentár k pozorovaným rozdielom medzi behmi.

### Obhajoba riešenia

Súčasťou hodnotenia je demonštrácia funkčného kódu a vysvetlenie implementácie. Body sa udelenujú najmä za porozumenie princípu algoritmu a schopnosť zdôvodniť dosiahnuté výsledky, nie iba za spustiteľný program.

## Odporučané nástroje a technická realizácia

Zadanie je možné vypracovať v prostredí **Matlab** (dostupná licencia STU) alebo v jazyku **Python**.

Ako alternatívu bez inštalácie je možné použiť **MATLAB Online** (<https://matlab.mathworks.com/>). Pre Python odporúčame **Google Colab** (<https://colab.research.google.com/>).

### Matlab

Použite dodané súbory a odporúčané príkazy pre vizualizáciu podľa pokynov na cvičení.

### Python

Ak sa rozhodnete pre Python, odporúčame využiť knižnice NumPy a Matplotlib. Pre interaktívne zobrazenie (napr. zobrazenie hodnoty pri ukázaní na bod v grafe) môžete použiť plotly alebo doplnok `mplcursors` pre Matplotlib.

Oficiálne repozitáre predmetu:

- **Organizácia (prehľad):** <https://github.com/STU-FEI-OUI>
- **Zadania a kódy pre GA blok:** <https://github.com/STU-FEI-OUI/UMINT-GA>
- **Genetic Toolbox:** <https://github.com/STU-FEI-OUI/Genetic-toolbox>

## Teoretické poznámky

### Selektívny tlak a diverzita populácie

**Selektívny tlak** vyjadruje, ako silno výber zvýhodňuje lepšie jedince. Vyšší selektívny tlak často vedie k **rýchlejšej konvergencii**, t. j. populácia sa rýchlo zjednotí okolo jedného riešenia, čo zároveň znamená **vyšie riziko stagnácie v lokálnom minime** (predčasná konvergencia).

**Diverzita** popisuje rôznorodosť jedincov, jej pokles typicky znižuje schopnosť algoritmu preskúmať alternatívne oblasti vyhľadávacieho priestoru.

Pri nastavovaní GA je vhodné vyhnúť sa extrémom. Príliš silný selekčný tlak môže spôsobiť tzv. „Habsburský efekt“: populácia sa rýchlo „príbuzensky“ zjednotí (kopíruje sa veľa veľmi podobných jedincov), diverzita klesne a algoritmus môže uviaznuť v lokálnom minime. Naopak, príliš vysoká miera mutácie vedie k „černobyľskému efektu“ – riešenia sa náhodne „rozrysú“ a zlepšovanie sa správa skôr ako náhodné hľadanie. Pri veľmi vysokých hodnotách mutácie sa to už správa takmer ako čiastočné generovanie novej populácie.

### Elitizmus a „výber najlepšieho“

**Elitizmus** znamená, že určitý počet najlepších jedincov sa **prenáša do ďalšej generácie bez zmeny**. **Výber najlepšieho jedinca** je len výber konkrétneho jedinca (napr. na vypísanie výsledku alebo ako rodiča) a sám o sebe neznamená automatický nezmenený prenos do ďalšej generácie.

### Poznámka k terminológii výberu

V anglickej literatúre sa pod pojmom **roulette wheel selection** zvyčajne myslí výber, kde je pravdepodobnosť zvolenia jedinca určená podľa **fitness** a úspešnejší jedinec má väčšiu šancu na výber, pretože zabera väčšiu časť „kolesa“. Niekoľko sa však v materiáloch môžete stretnúť aj s „ruletou“, kde majú všetky jedince **rovnaké výseky** a výber je potom čisto náhodný so zhodnou pravdepodobnosťou pre každého jedinca.

**Rank selection** je iný typ výberu a nie je to isté ako vyššie uvedený fitness-vážený „ruletový“ výber, pretože pravdepodobnosť výberu odvodzuje od **poradia** jedincov (ranku) a nie priamo z ich fitness. (zdroj)