

Genetická optimalizace

(18HA protokol)

Aleksej Gaj*

05/09/2023

Obsah

1	Zadání	2
2	Genetická optimalizace - konkrétní implementace	2
3	Zvolená úloha - Rastriginova funkce ve 2D	3
4	Výsledky a porovnání	3
5	Závěry a otevřené otázky	6
6	Přílohy	7

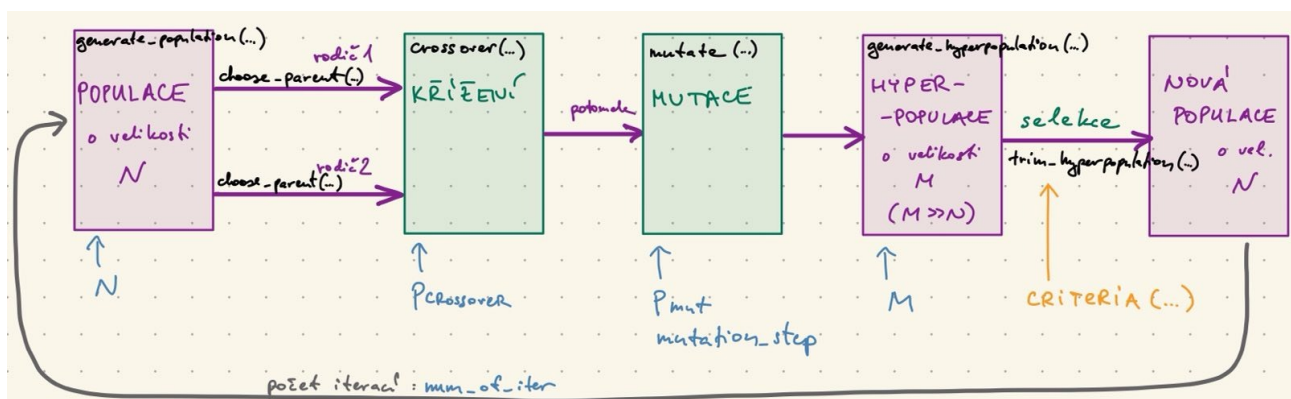
*email: aleksejalex@gmail.com

1 Zadání

- implementovat základní podobu genetické optimalizace (GO) v jazyce Python
- implementovat Rastiginovu funkci ve 2D vč. vizualizace
- pomocí GO najít minimum Rastriginovy funkce na zadaném 2D intervalu, postup vizualizovat

2 Genetická optimalizace - konkrétní implementace

Následující schéma stručně odráží strukturu programu. Cílem je extremalizovat zadanou funkci (v této práci byla zvolena minimalizace funkce na symetrickém 2D intervalu)



Obrázek 1: Schéma implementace. Černě jsou názvy funkcí, modře volitelné parametry programu.

Implementace GO je v souboru `mygo.py`¹. Obsahuje následující metody:

- `generate_random_vector` - vygeneruje vektor, představující jedince (element populace), tj. n-tice čísel mezi dvěma zadanými hodnotami, jejíž prvek je uniformně náhodně zvolen. Lze zvolit typ `bool`, `int` nebo `float`
- `sort_by_criteria` - tato funkce utřídí populaci vzestupně nebo sestupně (posle toho, zde je třeba hledat minimum nebo maximum)
- `generate_population` - vygeneruje celou populaci (vektor vektorů), pomocí funkce `generate_random_vector`
- `choose_parent` - funkce, která z populace náhodně vybere jednoho rodiče
- `crossover` - funkce, která s pravděpodobností p_{mut} zkříží dva rodiče
- `mutate` - funkce s pravděpodobností $p_{crossover}$ mutuje n -tý prvek vektoru (elementu populace)
- `generate_hyperpopulation` - vygeneruje tzv. *hyperpopulaci*
- `trim_hyperpopulation` - utříděnou hyperpopulaci ořízne tak, aby počet elementů odpovídal zadané hodnotě

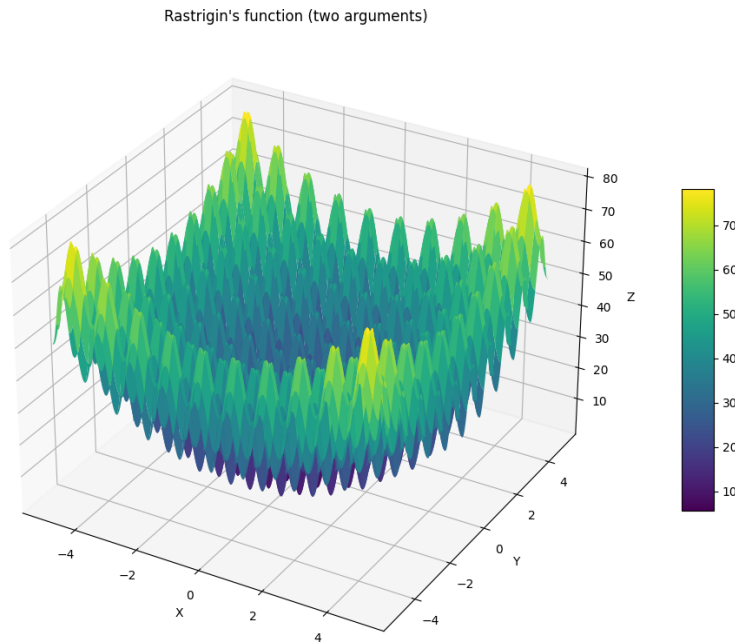
¹Veškeré zdrojové kódy a tato dokumentace jsou dostupné na <https://github.com/aleksejalex/GenOpt>.

3 Zvolená úloha - Rastriginova funkce ve 2D

Nejčastější využití Rastriginovy funkce je testování optimalizačních algoritmů. Její funkční předpis pro dvě proměnné má tvar:

$$\forall x, y \in \mathbb{R}^2 : f(x, y) = 20 + x^2 - 10 \cos(2\pi x) + y^2 - 10 \cos(2\pi y) \quad (1)$$

a zde je vykreslena:



Obrázek 2: Rastriginova funkce pro dva argumenty na intervalu $[-5, 5] \times [-5, 5]$

Funkce (výpočet i vykreslování tvaru funkce) je implementována v souboru `rastrigin_fction.py`.

4 Výsledky a porovnání

Program se stejným nastavením parametrů byl spouštěn po sobě 1000krát nezávisle, nalezené hodnoty minima Rastriginovy funkce byly zaznamenány. Po řadě experimentů se ukázali vhodné volby parametrů uvedené v Tabulkách níže. V příloze je několik tabulek pro jiné parametry. Vhodnou volbou kroku mutace (`mut_step`) a přiměřenou velikostí populace lze počet iterací dostat na “rozumnou” hodnotu.

Ukázalo se, že velikost kroku mutace je zcela klíčová, zatímco pravděpodobnost křížení v podstatě nehraje roli. Dále je vidět (viz tabulky v příloze), že velká populace na malém počtu kroků je neefektivní, stejně tak malá populace při relativně velkém počtu kroků.

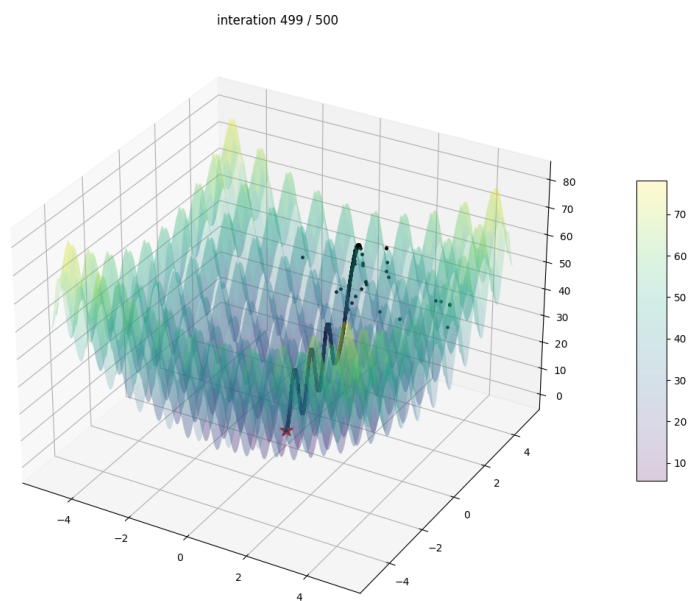
	N	n_iter	p_mut	p_cross	mut_step	min
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	7.000000	500.000000	0.800000	0.500000	0.010000	1.558124
std	0.000000	0.000000	0.000000	0.000000	0.000000	4.801539
min	7.000000	500.000000	0.800000	0.500000	0.010000	0.009546
25%	7.000000	500.000000	0.800000	0.500000	0.010000	0.009919
50%	7.000000	500.000000	0.800000	0.500000	0.010000	0.009919
75%	7.000000	500.000000	0.800000	0.500000	0.010000	0.011080
max	7.000000	500.000000	0.800000	0.500000	0.010000	25.899127

Tabulka 1: Základní statistická analýza 1000 běhů GO s uvedenými parametry

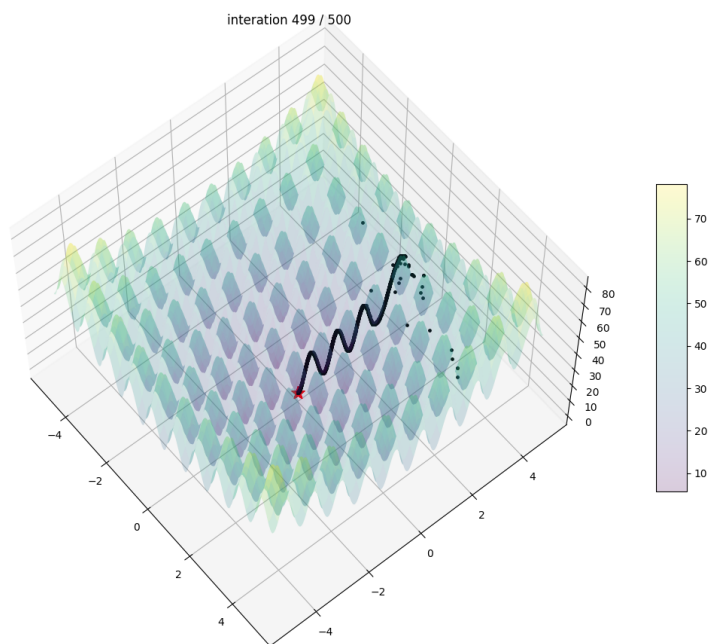
	N	n_iter	p_mut	p_cross	mut_step	min
0	7	500	0.800000	0.500000	0.010000	0.009919
1	7	500	0.800000	0.500000	0.010000	0.009919
2	7	500	0.800000	0.500000	0.010000	0.009919
3	7	500	0.800000	0.500000	0.010000	19.789577
4	7	500	0.800000	0.500000	0.010000	0.009919
5	7	500	0.800000	0.500000	0.010000	0.009919
6	7	500	0.800000	0.500000	0.010000	0.009919

Tabulka 2: Hodnoty prvních 7 minim z 1000 běhů GO s uvedenými parametry

Zrovna zde je vidět, že občas pro některé hodnoty parametrů (malý počet kroků nebo příliš malé velikosti populace) optimalizace “zabloudí”, tj. nalezená hodnota minima je výrazně odlišná od ostatních hodnot.



Obrázek 3: Vizualizace průběhu GO. Velikost populace je 7, počet iterací 500. Černé tečky představují jednotlivé populace, červená hvězdička označuje nalezené minimum.



Obrázek 4: Vizualizace průběhu GO, jiný pohled.

5 Závěry a otevřené otázky

Lze uzavřít, že program zadání, sformulované výše, splnil.

Pro:

- jednoduchý zdrojový kód
- vlastní implementace (jen minimální závislost na existujících knihovnách)
- dost rychlá (z hlediska absolutního času)
- přehledná vizualizace, snadný export obrázku

Contra:

- řada parametrů (velikost populace, pravděpodobnosti mutace a křížení, počet kroků simulace, aj.) volena v programu napevno (automatická volba - vylepšení do budoucna)
- vykreslování je nestabilní a neefektivní (bylo by vhodné použít jinou knihovnu, která má ovšem mnohem komplikovanější ovládání)

V další verzi by stálo za to rozmyslet “chytré” zastavování, aby se zbytečně neprováděli iterace, které už významně nezlepší výsledek. Například zastavit, když několik po sobě nalezených extrémů jsou blízko u sebe a zároveň menší (větší) než cokoliv dosud nalezeného.

Další efektivní vylepšení by byl adaptivní krok mutace (v aktuální verzi je nastaven napevno). Případně obě vylepšení zkombinovat: ve chvíli, kdy bude podmínka zastavení splněna, výrazně zjemnit krok a udělat několik iterací.

Z možných technických vylepšení se nabízí již zmíněné použití jiné knihovny na vykreslování nebo úprava implementace do objektové podoby.

6 Přílohy

	N	n_iter	p_mut	p_cross	mut_step	min
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	7.000000	500.000000	0.800000	0.500000	0.300000	8.289295
std	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
min	7.000000	500.000000	0.800000	0.500000	0.300000	8.289295
25%	7.000000	500.000000	0.800000	0.500000	0.300000	8.289295
50%	7.000000	500.000000	0.800000	0.500000	0.300000	8.289295
75%	7.000000	500.000000	0.800000	0.500000	0.300000	8.289295
max	7.000000	500.000000	0.800000	0.500000	0.300000	8.289295

Tabulka 3: Základní statistická analýza 1000 běhů GO s uvedenými parametry, **velký krok mutace**

	N	n_iter	p_mut	p_cross	mut_step	min
0	7	500	0.800000	0.500000	0.300000	8.289295
1	7	500	0.800000	0.500000	0.300000	8.289295
2	7	500	0.800000	0.500000	0.300000	8.289295
3	7	500	0.800000	0.500000	0.300000	8.289295
4	7	500	0.800000	0.500000	0.300000	8.289295
5	7	500	0.800000	0.500000	0.300000	8.289295
6	7	500	0.800000	0.500000	0.300000	8.289295

Tabulka 4: Hodnoty prvních 7 minim z 1000 běhů GO s uvedenými parametry, **velký krok mutace**

	N	n_iter	p_mut	p_cross	mut_step	min
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	2.000000	500.000000	0.800000	0.500000	0.010000	15.192047
std	0.000000	0.000000	0.000000	0.000000	0.000000	9.393377
min	2.000000	500.000000	0.800000	0.500000	0.010000	0.006566
25%	2.000000	500.000000	0.800000	0.500000	0.010000	7.526809
50%	2.000000	500.000000	0.800000	0.500000	0.010000	15.721875
75%	2.000000	500.000000	0.800000	0.500000	0.010000	21.852262
max	2.000000	500.000000	0.800000	0.500000	0.010000	47.973598

Tabulka 5: Základní statistická analýza 1000 běhů GO s uvedenými parametry, **malá populace**

	N	n_iter	p_mut	p_cross	mut_step	min
0	2	500	0.800000	0.500000	0.010000	32.151734
1	2	500	0.800000	0.500000	0.010000	11.031633
2	2	500	0.800000	0.500000	0.010000	17.700297
3	2	500	0.800000	0.500000	0.010000	21.525223
4	2	500	0.800000	0.500000	0.010000	9.252460
5	2	500	0.800000	0.500000	0.010000	2.612960
6	2	500	0.800000	0.500000	0.010000	22.229268

Tabulka 6: Hodnoty prvních 7 minim z 1000 běhů GO s uvedenými parametry, **malá populace**

	N	n_iter	p_mut	p_cross	mut_step	min
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	30.000000	100.000000	0.800000	0.500000	0.010000	21.434498
std	0.000000	0.000000	0.000000	0.000000	0.000000	9.882711
min	30.000000	100.000000	0.800000	0.500000	0.010000	1.788724
25%	30.000000	100.000000	0.800000	0.500000	0.010000	13.729086
50%	30.000000	100.000000	0.800000	0.500000	0.010000	21.879403
75%	30.000000	100.000000	0.800000	0.500000	0.010000	28.242316
max	30.000000	100.000000	0.800000	0.500000	0.010000	46.111063

Tabulka 7: Základní statistická analýza 1000 běhů GO s uvedenými parametry, **velká populace, málo kroků**

	N	n_iter	p_mut	p_cross	mut_step	min
0	30	100	0.800000	0.500000	0.010000	17.341157
1	30	100	0.800000	0.500000	0.010000	23.849684
2	30	100	0.800000	0.500000	0.010000	23.666716
3	30	100	0.800000	0.500000	0.010000	17.075429
4	30	100	0.800000	0.500000	0.010000	25.201969
5	30	100	0.800000	0.500000	0.010000	28.809787
6	30	100	0.800000	0.500000	0.010000	31.226162

Tabulka 8: Hodnoty prvních 7 minim z 1000 běhů GO s uvedenými parametry, **velká populace, málo kroků**

	N	n_iter	p_mut	p_cross	mut_step	min
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	7.000000	500.000000	0.800000	0.000000	0.010000	1.221888
std	0.000000	0.000000	0.000000	0.000000	0.000000	4.237270
min	7.000000	500.000000	0.800000	0.000000	0.010000	0.005773
25%	7.000000	500.000000	0.800000	0.000000	0.010000	0.009919
50%	7.000000	500.000000	0.800000	0.000000	0.010000	0.009919
75%	7.000000	500.000000	0.800000	0.000000	0.010000	0.010506
max	7.000000	500.000000	0.800000	0.000000	0.010000	36.683016

Tabulka 9: Základní statistická analýza 1000 běhů GO s uvedenými parametry, **bez křížení**

	N	n_iter	p_mut	p_cross	mut_step	min
0	7	500	0.800000	0.000000	0.010000	0.009919
1	7	500	0.800000	0.000000	0.010000	0.009919
2	7	500	0.800000	0.000000	0.010000	0.009546
3	7	500	0.800000	0.000000	0.010000	0.009919
4	7	500	0.800000	0.000000	0.010000	0.009919
5	7	500	0.800000	0.000000	0.010000	0.009919
6	7	500	0.800000	0.000000	0.010000	0.009919

Tabulka 10: Hodnoty prvních 7 minim z 1000 běhů GO s uvedenými parametry, **bez křížení**