

# Protokol

## 1. Výběr Testovacích Funkcí

Z libovolných zdrojů bylo vybráno 25 testovacích funkcí, které splňují následující kritéria:

- Multimodální
- Spojité na celém rozsahu
- Ohodnotitelné v reálném čase
- Bez fraktálního charakteru nebo jiných „patologií“

A Literature Survey of Benchmark Functions For Global Optimization Problems.pdf

Odkaz:

[https://github.com/thieu1995/opfunu/blob/master/docs/source/\\_static/refs/name\\_based/A%20Literature%20Survey%20of%20Benchmark%20Functions%20For%20Global%20Optimization%20Problems.pdf](https://github.com/thieu1995/opfunu/blob/master/docs/source/_static/refs/name_based/A%20Literature%20Survey%20of%20Benchmark%20Functions%20For%20Global%20Optimization%20Problems.pdf)

## A Literature Survey of Benchmark Functions For Global Optimization Problems

Momin Jamil<sup>\*†</sup>, Xin-She Yang<sup>‡</sup> \*Blekinge Institute of Technology  
SE-37179, Karlskrona, Sweden

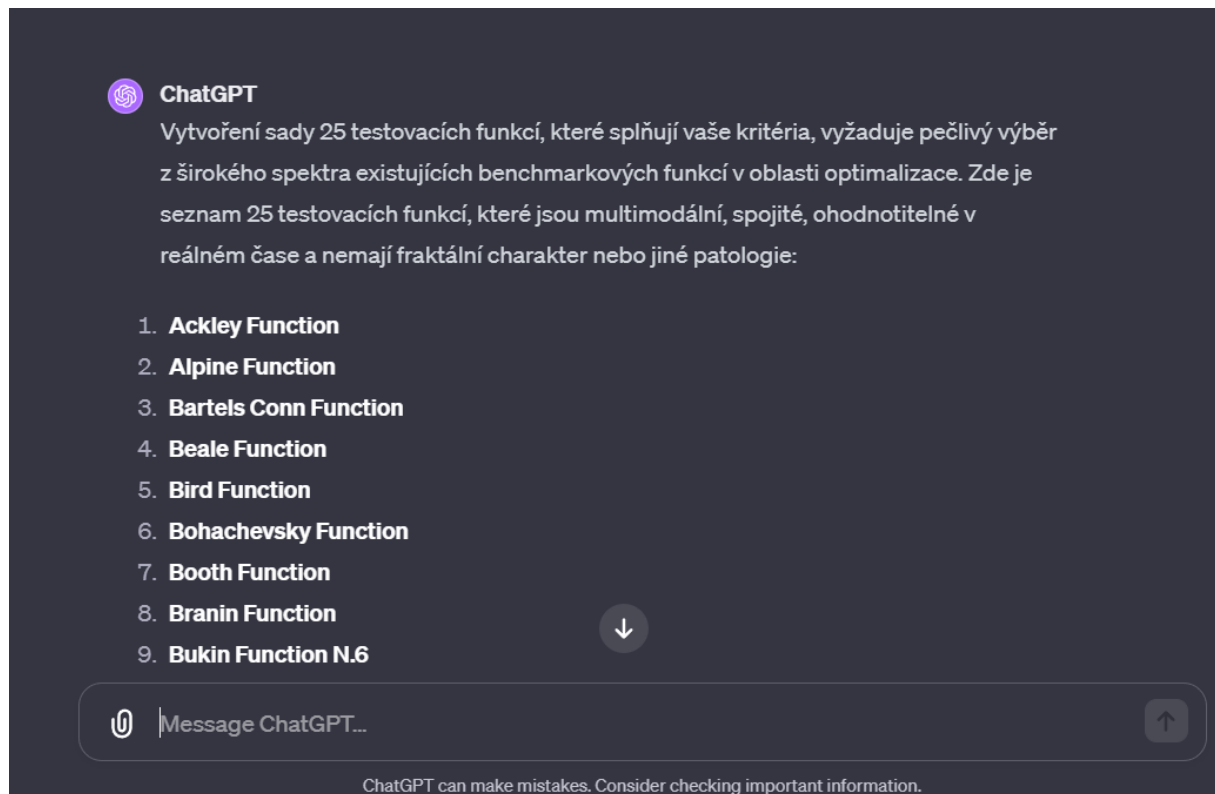
<sup>†</sup>Harman International, Cooperate Division  
Becker-Goering Str. 16, D-76307 Karlsbad, Germany  
E-mail: momin.jamil@harman.com

<sup>‡</sup>Middlesex University  
School of Science and Technology  
Hendon Campus, London NW4 4BT, UK  
E-mail: xin-she.yang@middlesex.ac.uk

### Citation details:

Momin Jamil and Xin-She Yang, A literature survey of benchmark functions for global optimization problems, *Int. Journal of Mathematical Modelling and Numerical Optimisation*, Vol. 4, No. 2, pp. 150–194 (2013).  
DOI: 10.1504/IJMMNO.2013.055204

Test functions are important to validate and compare the performance of optimization algorithms. There have been many test or benchmark functions reported in the literature; however, there is no standard list or set of benchmark functions. Ideally, test functions should have diverse properties so that can be truly useful to test new algorithms in an unbiased way. For this purpose, we have reviewed and compiled a rich set of 175 benchmark functions for unconstrained optimization problems with diverse properties in terms



## Snížení počtu iterací

Za účelem včasného odevzdání a nutného výkonu při plnění dalších povinností jsem byl nucen snížit počet iterací. Také můj hardware není na takové úlohy stavěn viz. foto

```
System Manufacturer: Gigabyte Technology Co., Ltd.
System Model: Z97X-Gaming 3
System Type: x64-based PC
Processor(s): 1 Processor(s) Installed.
[01]: Intel64 Family 6 Model 60 Stepping 3 GenuineIntel ~3301 Mhz
BIOS Version: American Megatrends Inc. F8d, 03.03.2016
Windows Directory: C:\Windows
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume1
System Locale: cs;Čeština
Input Locale: cs;Čeština
Time Zone: (UTC+01:00) Praha, Bratislava, Budapešť, Bělehrad, Lublaň
Total Physical Memory: 16 220 MB
Available Physical Memory: 7 844 MB
```

## 2. Testované Algoritmy

Otestovány byly následující optimalizační algoritmy:

DE rand/1/bin

DE best/1/bin

PSO (Particle Swarm Optimization)

SOMA all-to-one

SOMA all-to-all

Při testování byla zajištěna kontrola hranic prohledávaného prostoru metodou odrazu (reflection).

### 3. Nastavení Algoritmů

Nastavení algoritmů bylo následující:

Dimenze (D): 2, 10, 30

Velikost populace NP: 10 (pro 2D), 20 (pro 10D), 50 (pro 30D)

Function evaluations FEs: 2000\*D

Kontrola hranic: Odraz (reflection)

Opakování: 30

DE rand/1/bin

$F = 0.8$

$CR = 0.9$

DE best/1/bin

$F = 0.5$

$CR = 0.9$

PSO

$c1, c2 = 1.49618$

$w = 0.7298$

SOMA all-to-one

PathLength = 3

StepSize = 0.11

PRT = 0.7

SOMA all-to-all

PathLength = 3

StepSize = 0.11

PRT = 0.7

#### 4. Hodnocení Výkonu Algoritmů

Pro každou testovací funkci bylo určeno pořadí algoritmů. Výsledky byly vyhodnoceny na statistickou významnost pomocí Friedman rank testu.

jak mám udělat friedman rank test



### ChatGPT

Friedmanův rank test je ne-parametrický statistický test používaný pro srovnání více než dvou vzorků, které jsou nezávislé a měřené v rámci stejných podmínek. V Pythonu můžete tento test provést pomocí knihovny SciPy. Předpokládejme, že máte výsledky z více algoritmů pro různé testovací funkce a chcete zjistit, zda existují statisticky významné rozdíly mezi těmito algoritmy.

Nejprve budete potřebovat data uspořádaná tak, že každý řádek reprezentuje jednu testovací funkci a každý sloupec reprezentuje výsledky jednoho algoritmu. Poté můžete použít Friedmanův test z SciPy:

```
python Copy code  
  
import numpy as np  
from scipy.stats import friedmanchisquare  
  
# Příklad dat: výsledky tří algoritmů (A, B, C) pro 5 různých testovacích  
data_A = [1, 2, 3, 2, 4]  
data_B = [2, 3, 2, 4, 5]  
data_C = [3, 2, 1, 3, 3]  
  
# Provedení Friedmanova testu  
stat, p = friedmanchisquare(data_A, data_B, data_C)  
  
print('Statistika:', stat)  
print('p-hodnota:', p)  
  
# Vyhodnocení významnosti  
alpha = 0.05 # obvyklá hladina významnosti  
if p > alpha:  
    print('Neexistují žádné statisticky významné rozdíly mezi algoritmy')  
else:  
    print('Existují statisticky významné rozdíly mezi algoritmy.')
```

Pamatujte, že tento test předpokládá, že data nejsou normálně rozdělená, což je důvod, proč se používá ne-parametrický test. Hodnota `p` udává pravděpodobnost, že rozdíly mezi skupinami jsou náhodné. Nižší hodnota `p` znamená, že rozdíly mezi

## 5. Analýza Výsledků

Statistika: 2.2105263157894752

p-hodnota: 0.33112373295101094

Neexistují žádné statisticky významné rozdíly mezi algoritmy.