# *Mesterséges Intelligenciák alapjai 2023-24 I. félév II. beadandó*

*Genetikus algoritmusok*

*Fazekas Levente Áron*

*2023/24 I. félév*

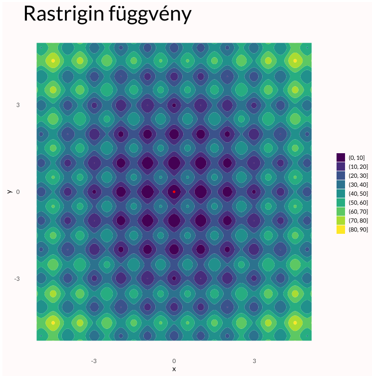
## *Folytonos függvények*

Adottak az alábbi függvények:

### *Rastrigin*

$$f(X) = An + \sum_{i=1}^{n} \left[x_i^2 - A\cos(2\pi x_i)\right]$$
$$A = 10,\ -5,12 \leq x_i \leq 5,12,\ n = 2$$

Globális minimum: $f(0,0) = 0$.

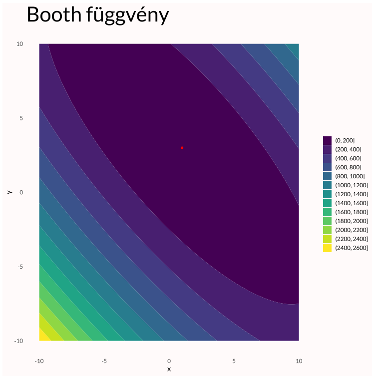



1. ábra. Rastrigin függvény

### *Booth*

$$f(x,y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$
$$-10 \leq x, y \leq 10$$

Globális minimum: $f(1,3) = 0$.

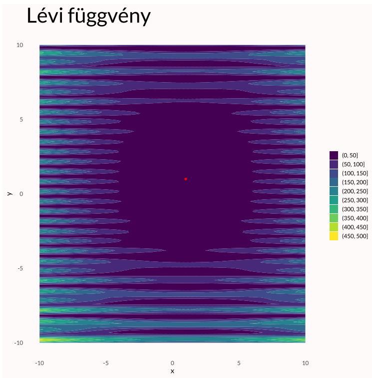



2. ábra. Booth függvény

### *Lévi*

$$f(x,y) = \sin^2 3\pi x + (x-1)^2\left(1 + \sin^2 3\pi y\right) + (y-1)^2(1 + \sin^2 2\pi y)$$
$$-10 \leq x, y \leq 10$$

Globális minimum: $f(1,1) = 0$




3. ábra. Lévi függvény

### *Feladat*

Írjon olyan genetikus algoritmust, amely ezeket a függvényeket minimalizálja! Futtassa le az algoritmust 5, 10, 20, 50, 100 generációval, 5, 10, 20, 50, 100 kromoszómával, relatív, rangsor-szerinti, diverzitás és fitnessz rangsor-szerinti túlélési valószínűségekkel! A maximális lépésméretek: 0.1, 0.2, 0.5, 1, 1.5, 2. Vizsgálja meg az elitizmus hatásait is!

Készítsen táblázatot a talált eredményekről, a futási időkről! Vizsgálja meg, hogy az algoritmus hogyan "konvergál", azaz az egyes generációk során hogyan változnak az eredmények (legjobb, legrosszabb, szórás)!

Más leállási feltételek bevezetésével csökkenthető a futási idő, míg a végeredmény azonos?

Vizsgálja meg a mutációk és rekombinációk számának vagy valószínűségének hatását!

Vizsgáljon meg több különböző kiválasztási módszert (Rulettkerék, Versenyeztetés)!

A **Rastrigin** függvény $n$ dimenziós. Vizsgálja meg milyen hatása van annak, ha a dimenziók számát növeljük: 3, 4, 5, 10, 100. A több dimenzió hosszabb kromoszómákat eredményez, ezek keresztezéséhez használjon több különböző módszert: egypontos keresztezés, kétpontos keresztezés, többpontos keresztezés, uniform keresztezés, path relinking.

| G | K | L | Val. fv. | Elit | Célfv. | Start | Eredmény | $t$ |
|---|---|---|---|---|---|---|---|---|
| 100 | 100 | 1.0 | F + D rank | Igen | Rastrigin | $f(-2.5,-2.5) = 52.5$ | $f(1,3) = 0$ | 0.323mp |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 5 | 5 | 0.1 | Relatív | Nem | Rastrigin | $f(-2.5,-2.5) = 52.5$ | $f(-2,-2.3) = 22.38$ | 0.001mp |

1. táblázat. Példatáblázat

## *Traveling Salesman Problem (TSP)*

Az utazó ügynök probléma a klasszikus optimalizálási problémák közé tartozik. Feladatunk, hogy olyan sorrendjét válasszuk meg az érintett városoknak, hogy a megtett távolságot minimalizáljuk. Fontos, hogy a megoldásunknak sorrendnek kell maradnia, azaz $n$ város esetén (0, 1, 2, ..., n-1) indexek sorrendje kell legyen mindig.

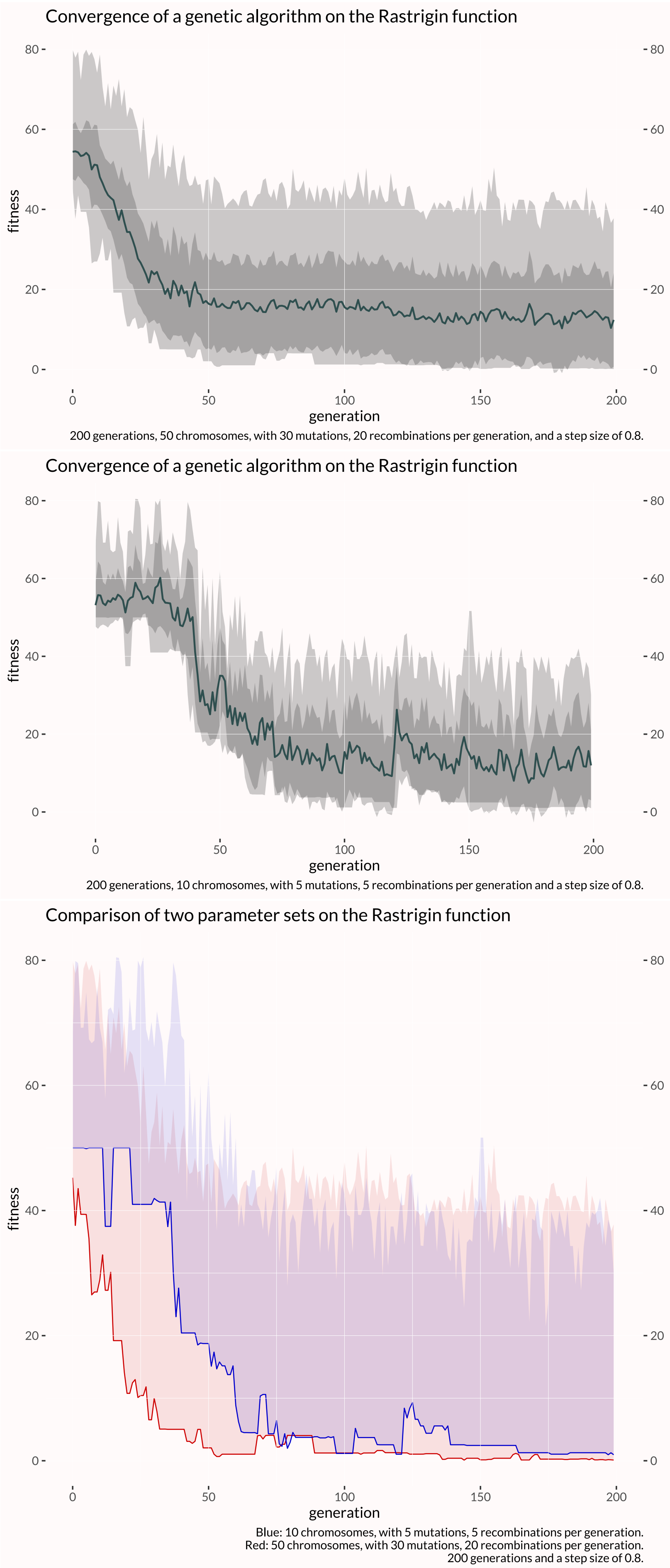
A célfüggvény egyszerű python implementációja:

```
def fitness(distances, s):
    dist = 0
    prev = s[0]
    for i in s:
        dist += distances[(prev, i)]
        prev = i
    dist += distances[(s[-1], s[0])]
    return dist
```

ahol a `distances` argumentum a városok közti távolságokat tartalmazó dictionary `{(város1, város2): távolság}`, és `s` a sorrend. Visszatérési értéke a megtett távolság, ennek a **minimalizálása** a célfüggvény.

Az eddig tanult technikák egy részét alkalmazza erre a problémára! Vizsgálja meg, hogy az egyes paraméterek, megoldások hogyan befolyásolják az eredményt és a futási időt.

Néhány "benchmark" feladat: Taillard feladatok (link)




4. ábra. Két paraméterkészlet konvergenciája