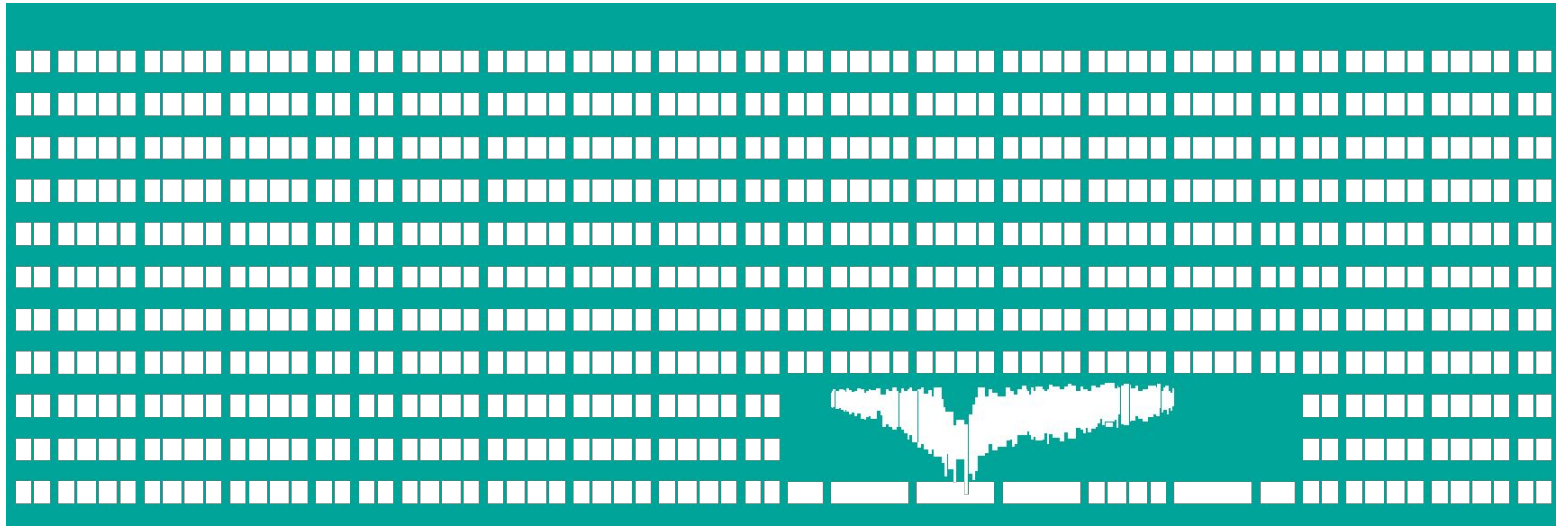
VŠB TECHNICKÁ UNIVERZITA OSTRAVA

VSB TECHNICAL UNIVERSITY OF OSTRAVA



www.vsb.cz

VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Biologically inspired algorithms
Exercise 5

Ing. Lenka Skanderová, Ph. D.

Content

- Differential Evolution

Differential Evolution – Control Parameters

- NP … number of individuals
- G_{maxim}… number of generation cycles
- F … mutation constant (also denoted as scaling factor)
- CR … crossover range

Differential Evolution – Individual and Population

- Consider the objective function to be Sphere function $F(\vec{x}_i) = \sum_{j=1}^{D} x_{i,j}^2 , i = 1, \dots, NP$
  - i … index of individual within a population
  - j … index of a parameter of an individual
  - Example:
    - $NP = 4$
    - $F = 0.5$
    - $CR = 0.5$
    - $G_{maxim} = 10$
- DE was developed to work with real numbers

Individuals (solutions) representation

	$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	$\vec{x}_4$
	3.4	2.0	5.1	0.5
	2.5	4.5	3.3	0.7
	1.0	1.0	2.4	1.2
	5.0	6.0	1.8	1.1
	2.2	5.0	4.1	0.0
$f(\vec{x}_i)$	48.65	86.25	62.71	3.39

Differential Evolution – Pseudocode – Main Loop

```
pop = Generate NP random individuals (you can use the class Solution mentioned in Exercise 1)
g = 0

while g < g_maxim :
  new_pop = deepcopy(pop) # new generation
  for each i, x in enumerate(pop): # x is also denoted as a target vector
    r1, r2, r3 = select random indices(from 0 to NP-1) such that r1!=r2!=r3!=i
    v = (x_r1.params - x_r2.params)*F + x_r3.params # mutation vector. TAKE CARE FOR BOUNDARIES!
    u = np.zeros(dimension) # trial vector
    j_rnd = np.random.randint(0, dimension)

    for j in range(dimension):
     if np.random.uniform() < CR or j == j_rnd:
       u[j] = v[j] # at least 1 parameter should be from a mutation vector v
     else:
       u[j] = x_i.params[j]

    f_u = Evaluate trial vector u

    if f_u is better or equals to f_x_i: # We always accept a solution with the same fitness as a target vector
     new_x = Solution(dimension, lower_bound, upper_bound)
          new_x.params = u
          new_x.f = f_u
    pop = new_pop
  g += 1
```

Task

- Implement DE/rand/1/bin
- Use DE to find out the optimal solution of the optimization functions
  - $NP = 20$
  - $F = 0.5$
  - $CR = 0.5$
  - $G_{maxim} = 50$
- Visualize the process of search (in 3D)
- Inspiration: Figures 1, 2, and 3

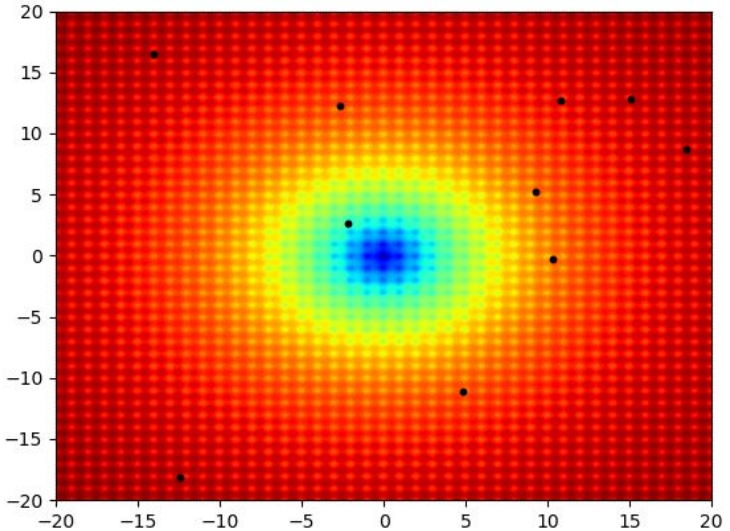


Figure 1

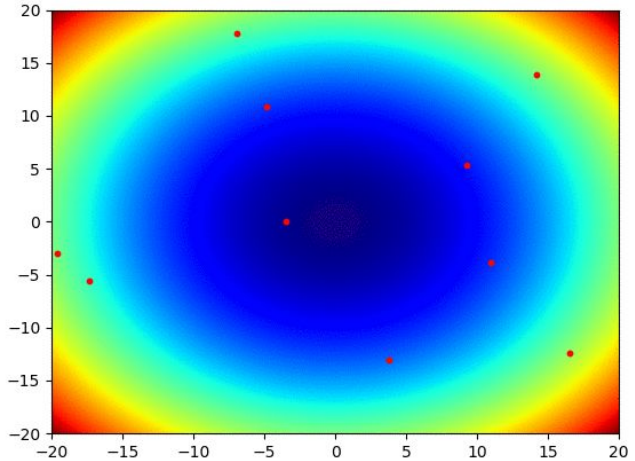


Figure 2

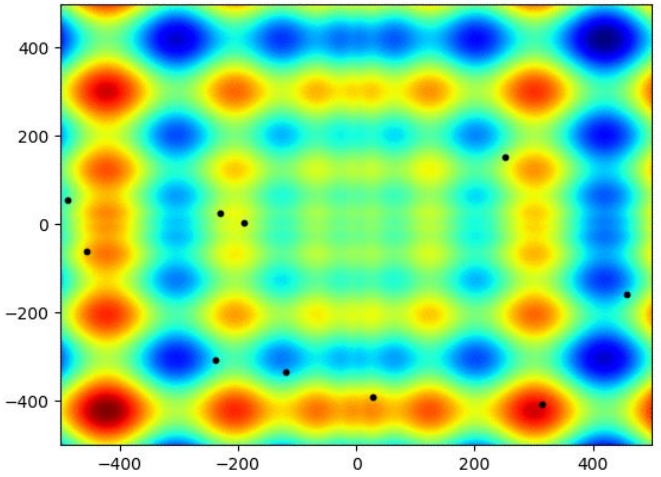


Figure 3

Thank you for your attention

Ing. Lenka Skanderová, Ph.D.

EA 407
+420 597 325 967

lenka.skanderova@vsb.cz

homel.vsb.cz/~ska206