INPUT

- f: objective function
- N_c: number of chromosomes
- N_g: maximum number of allowed generations
- p_s: the selection rate of the algorithm, with $p_s \leq 1$
- p_m: the mutation rate of the algorithm, with $p_m \leq 1$
- k: the generation counter
- a: uniformly distributed random numbers, in $\in [-0.5, 1.5]$

OUTPUT

- x_{best}, f_{best}

INITIALIZATION

- $k \leftarrow 0$

main pseudocode

```
01  while k < N_g do // termination check
02      for each g_i, i ∈ {1..N_c} do
03          f_i ← f(g_i)
04      endfor
05      Sort chromosomes by increasing fitness: N_b ← (1 − p_s)xN_c
06      Select parents w, z randomly among the best N_b chromosomes
07      Draw a_i ∈ U[−0.5, 1.5]
08          z_i ← a_i z_i + (1 − a_i)w_i
09          w_i ← a_i w_i + (1 − a_i)z_i
10      for each g_i, i ∈ {N_b + 1..N_c} do
11          Replace g_i with z_i or w_i
12      endfor
13      for each g_i, i ∈ {1..N_c} do
14          for each gene, j ∈ {1....n } do
15              Draw rU[0, 1]
16                  if r ≤ p_m then
17                      Mutate gene j of g_i
18                  endif
19          endfor
20      end for
21      for each g_i, i ∈ {1..N_c} do
22          f_i ← f(g_i)
23          Draw ri ∈ U[0, 1]
24          if (f_i < f_best) then
25              xbest ← gi, fbest ← fi
26          endif
27      endfor
28      k ← k + 1
29  endwhile
30  return x_best, f_best
```