

for iten in range (max): $a \leftarrow 2 - (iteration / max) + 2$ for it o to nun-wolves: fitness (fitness - function (wolves [i]) if fitness < alpha - fitness to gamma copy beta cutoff & fitness to gamma copy alpha entoff wolf & fitness to beta alpha - wolf < wolves [i]. copy() alpha fitnes - fitnes elif fitnese \ beta fitnese:

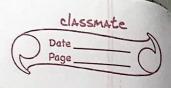
copy beta wolf & fitness to gamma

beta-wolf \ wolver[i].copy() beta-fitness - fitness elif fitness < gamma-fitness:

gamma-wolf < wolves [i]. copy()

gamma-fitness < fitness if alpha-fitness < best-fitness : best-fitness - alpha-fitness for it o to num -wolves: for je o to dim: n/ & 22 fandons floats

A1 ← 2* a* n1 - a C1 <- 2*92 n-alpha ← abs (ci*alpha - wolf [j]
-wolles [i,j]) XI - alpha - wolf [j] - AI* D-alpha



New 21 & 22 sandom floats A2 ← 2*a * 21 - a AC2 ← 2*22

D-beta - abs (c2 beta - wolf [j])
- wohes [i, j])

X2 (beta - wolf [j] - A2* D- bète

New 91892 random floats A3←2*a*r1-a C3←2*12

D-gamma = abs (C3*gamma-wolfj)
-wolves[ij])

X3 - gamma - wolf [j]-43 * D-gamme

wolves [i, j] (x1+x2+x3) | 3 wolves [i, j] cdisp (wdves [i, j] search-space [j, o], search-space [ji])

Input:

search-space = array ([-5,5], [-5,5])
num - wolves = 10

max-ites = 1000 - 100 % 10 10

Orable en obs (c1 alpha - w 1) (C1

10 0 4A - []]] - A/4 0 21