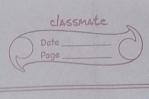
s. Gaey Wolf. 2211194 Algorithm. Interalize the population of neglices (poestions) aandensly neithin the seasch space Defino the maximum number of iterations (T) and population size(N) Define the fitness function to evaluate solutions Evaluate the fitness of each wolf in the population Identify the alpha (best solution), betal second-best and dolfa (the od-best) nochles For t=1 to Time ATT For each welf i in the popularion. For each dimension d A1= 2 \* a \* aard () - a (1=8 \* rand() Dealpha = ITI\* Xalpha [d]-x:2[d] x,=x=alpha[d]-A, +Dalpha A & = 2\* a\* nand()-a Ca= 2x nand Cl D\_bota = (2 \* x bota[d] -x o[d] X 2=x betald ] - A 8 + D beta 1 A3= 2\* a\* gand ()-@ 63=8+ sand() D delta= C3 x x delta[d]-xi[d] X3 = X-delta[d] - A3\* D\_delta Xild J= (x1+X2+x3) 13 End For C1=g-(8#+ /+.) Opedate alpha, beta and delta vedves based. on fitness End-for.

imposit numpy as no Paogram. det grey nodfroptimizer Cfi tress function, nummydres, num dimensions, max i terations, bounds Loncier bound upper bound = bounds hoolves = np. Random uniform chances bound, upor bound, I nem polices num dimensione alpha, beta, delta = None, None, None: fitness = np. acreay (I fi the s function (never) for next Stortedindices = fitnes. angust () alpha, beta, delta = noolves [sorted indices [:3]] for iteration in sange (max iterations) for i in range ( mum nahous). foer d in range (num démensions): ol, od=np. soundom. sound(), np. sandom. A1=0\* a\* 81-a C1=2 + 82. Dalpha: abs (1 \* alpha ta I- nochrati, ) XI=alphald]-Ad+D\_beta 7 + 82 = np. gandom a gand (), np. gandoman A2=2+a+81-a C2 = 2 + 02 O\_beta = abs (Co\* beta [d] - wolves [e,d] X2 - beta [d] - A2\* 0 beta



71, 82=np. gardon gard (), np. grandom gard) 43 = 24 a x 21-a A3 = 24 a x 21-a (3=d+ 82. Delta = abs (C3\* del tatd ] - reduce [1, 0] X3=dolta [d]-A3#D delta wolves [i", d] = (x) + x2+x3)/3 nool was tid in pocking chooleds tid, lower bound,
hopper bound) fitness = np. aaray ([fitness function (world) for hooly in nochels]) Sorted indices = fitness. aagsort ()
alpha, beta, delta = welves ( sorted indices [:3]) a=3- (3\* Etenation/maxituations) best solution = alpha
best fitness function (alpha)
actuary best solution, best fitness if name == "-main":

def fitness function(n)

action np. sum (n+nd)

num not mes = 30 num dimensions = 5 max ; teartions = 100 1 5 unds= (-10, 16.)

best solution, best fitness = grey wolf optimizer ( fernels function, num healus, num danencions max\_iterations, bounds)

