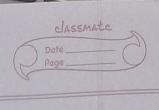
Ant colony Optimization. 0010/24 Algorithm and City gnétialize phenomene ralnes ti, je EI, no zij / Zo repeat for each ant 2 G & 1, or no no morle to stourting city is ES for ant 2 i-sio while Stodo semple current city ferom selection sets > 3/2:3 choose next city s'in roug notte passability Les Zeh & Min update solution alletion Tili) j move to new city i & j and volvile fertilize solution rector 7, (i) > is end for for each cohition I, LEE, -, ng do calculate townlength f(Ty ) = din (i) end for for all (i, j) do eraporate phonomone zij = (1-8). Zij end for determine beet colution of iteration nt against,) if The better than current beet T\*, i.e.f(T+)<f(T\*)

then set T\* L> T+ for all (i, j) Ext do reinforce zij'ts: Zij+1/9.

for all (i, j) Ent do neinface Zij > Zij + 1/2 until condition por termination met Program ... Emport numpy as no det instralize phenomenests (n, faut):
netarn np. fall ((n,n), but) det calculate distance (city, 1, aty2); networn up. lenalg. norm(cety) - city2) de j calculate probabilité os Cohenomones, distances, alpha b eta, résited, current city). n = len (phonomones)

probabilities = np. zonos(n)

r i in a for j in Range (n) it i not in quisited. probabilities [ ] = cpheromones [aurenteity] \*
alpha) \* (c) / distances [current city] )\*\* aeturn probabilities/probabilites. Sum () det aco spícities, m, alpha, beta jaho, Q, sterations n = len (cities) tand = 1/(n\* np. mean ([calculate distance cuttes[e])
cities [j]) for i in range (n) for j in
sange (n) 7) sange (n) ]) presidences = gnétaliza phenomones (n, tour 0 best tous = None



best four length = float ('ing')

for in sange (m):

visited = set()

consent city inp. random . sandint (0, n) visited. add consent city

tour append (warent wity)

probabilities = calculate probabilities (
phenomenes, distances, applia, beta)

next city = np. 20mdom, Chall (2ange (n), P = perdoabile ties

tour append (next city)

tour append (tour [0])

toner length = sum (distances / toner [E], toner [+T] for

all town lengths append (towa length)

delta fau = 6/tour length for i in range (len 1 tour) - 1);

phenomenes [tous [i], tous [i+1]] == delta tau
phenomenes [tous [i+1], tous [i]]+=delta tau

phenomonos \*= (1- The

if mintour length = min (all tous longths)

best town length = min tour length
best tour = all tours Inp. aagmin (all tour Lengthe) return best Jone, boot tous length. at name = "monain": cêties orp acray ([[0,0],[1,3],[4,3],[6,1]) for (2 in cities 3 for el 2in cities 3) appha=1 beta 2 2h020.5 Exercitions=106 best for , best tour length - aco top Cities, m, Alpha, bela, 2ho, & iteacricons)
print ("Best Pona", best tona") perent (" Best Town length; 1, best town length) Best Pars: To, np. int Col(1), np. int 64(2), np. int (64(2)) Best Tour Long th: 15,073