from random import \*

from math import \*

from sys import \*

from time import clock

import pdb

def calc\_h(board):

x = board[1]

y = board[2]

return -(20 + x\*\*2 - 10\*cos(2\*pi\*x) + y\*\*2 - 10\*cos(2\*pi\*y))

def main():

output = open('output\_GA.txt', 'w') # output data to txt file

sum = 0

for i in range(1000):

start = clock()

best\_h = -10000

bg = 0

n = 20

board = []

for x in range(n):#2n boards

board.append([])#represents a board

board[x].append(0)

board[x].append(random()\*10.24-5.12)#random initialization

board[x].append(random()\*10.24-5.12)

board[x][0] = calc\_h(board[x])#h-value for random initialization

#print board

generations = 0

while True:#generations < 100000: #generations

#print generations,board[0][0],board[-1][0]

board.sort()

if board[-1][0] > best\_h:

bg = generations

best\_h = board[-1][0]

best\_x = board[-1][1]

best\_y = board[-1][2]

mutation = (1/(1+exp(-(generations-bg))) - 0.5)

generations+=1

#print generations, board[0][0], board[-1][0], mutation, best\_x, best\_y

board = board[len(board)/3:len(board)] #gets rid of the worst third

while len(board) < n:#crossover and mutation

#CROSSOVER

temp1 = board[randint(0,len(board)-1)]

temp2 = board[randint(0,len(board)-1)]

newboard = []

newboard.append(1)

newboard.append(temp1[1])

newboard.append(temp2[2])

#newboard[0]=calc\_h(newboard)\*-1

#MUTATION

if(random() < mutation):

newboard[1] = random()\*10.24-5.12

if(random() < mutation):

newboard[2] = random()\*10.24-5.12

newboard[0] = calc\_h(newboard)

board.append(newboard)

board.sort()

if abs(board[-1][0]) <= 0.001:

end = clock()-start

output.write('1 ' + str(-1\*board[-1][0]) + ' ' + str(end) + '\n')

print (-1\*board[-1][0]), end

sum += end

break

#mutation,best\_x,best\_y,board[-1][0], bg

print 'Total Run Time:', sum

output.close()

if \_\_name\_\_ == "\_\_main\_\_": main()