<https://repl.it/repls/WeakDeadSites>

<https://repl.it/repls/FuchsiaAttachedParticles>

<https://arxiv.org/pdf/0709.1020.pdf>

import random

import math

import numpy

import matplotlib as mpl

import matplotlib.pyplot as plt

population = []

for j in range (12):

temp = []

temp.append(500)

for i in range(98):

temp.append(random.randint(0,500))

temp.append(100)

population.append(temp)

def fitness(population):

times = []

for i in range(12):

time = 0

v1 = 0

for j in range(99):

dy = abs(population[i][j+1]-population[i][j])

try:

time += (math.sqrt(25+dy^2))/(v1+math.sqrt(v1+2\*9.8\*dy))

except:

time += 1000000000000000

v1 = math.sqrt(v1+2\*9.8\*dy)

times.append(time)

print(times)

return population[times.index(min(times))]

def offspring(parent):

population = []

for i in range(12):

temp = []

temp.append(500)

for j in range(98):

y = int(abs(numpy.random.normal(0.0, 83)+parent[j+1]))

if y >= 500:

temp.append(500)

elif y <= 100:

temp.append(100)

else:

temp.append(y)

temp.append(100)

population.append(temp)

return population

for i in range(1000):

population = offspring(fitness(population))

dxlist = []

for i in range(100):

dxlist.append(5\*(i)+100)

plt.plot(dxlist, fitness(population))

plt.savefig("bruh.png")

import random

import math

import numpy

import matplotlib as mpl

import matplotlib.pyplot as plt

population = []

for j in range (12):

temp = []

for i in range(50):

temp.append(random.uniform(0,20))

population.append(temp)

def fitness(population):

times = []

for i in range(12):

time = 0

v1 = 0

for j in range(50-1):

dy = population[i][j+1]-population[i][j]

try:

time += (math.sqrt((10/50) \*\* 2 + dy \*\* 2))/(v1+math.sqrt(v1+2\*9.8\*population[i][j]))

except ZeroDivisionError:

time += 1000000000000000

v1 = math.sqrt(v1+2\*9.8\*population[i][j])

times.append(time)

print(times)

return population[times.index(min(times))]

def offspring(parent):

population = []

for i in range(12):

temp = []

temp.append(20)

for j in range(50-2):

y = numpy.random.normal(0.0, 0.01)+parent[j+1]

if y >= 20:

temp.append(20)

elif y <= 0:

temp.append(0)

else:

temp.append(y)

temp.append(10)

population.append(temp)

return population

dxlist = []

for i in range(50):

dxlist.append(i\*10/50\*2)

for i in range(20000):

population = offspring(fitness(population))

plt.plot(dxlist, fitness(population))

plt.savefig("bruh.png")

print(population)