

```

CHROMOSOME_SIZE=10
CHROMOSOME_COUNT=20
EPOCH_COUNT=1
import random
units=[[random.randrange(0,2) for i in range(0,CHROMOSOME_SIZE)] for j in
range(0,CHROMOSOME_COUNT)]
def f1(x):
    return (x-2)**2
def f2(x):
    return (x-4)**2

def to_int(unit):
    rez=0.0
    for u in unit:
        rez=rez*2+u
    return rez/(2**len(unit))*6
def get_rang(unit,units):
    rez=0
    unit_f1=f1(to_int(unit))
    unit_f2=f2(to_int(unit))
    for concurent_unit in units:
        concurent_f1=f1(to_int(concurent_unit))
        concurent_f2=f2(to_int(concurent_unit))
        if unit_f1>concurent_f1 and unit_f2>concurent_f2:
            rez-=1
        #if unit_f1<concurent_f1 or unit_f2<concurent_f2:
        #    rez+=1
    return rez

def compare(a,b):
    if a[0]<b[0]:
        return 1
    if a[0]>b[0]:
        return -1
    if a[0]==b[0]:
        return 0

def cross(parA,parB):
    childA=[]
    childB=[]
    for i in range(0,CHROMOSOME_SIZE):
        if random.random()<0.5:
            childA.append(parA[i])
            childB.append(parB[i])
        else:
            childA.append(parB[i])
            childB.append(parA[i])
    i1=random.randrange(0,CHROMOSOME_SIZE)
    childA[i1]=1-childA[i1]
    i2=random.randrange(0,CHROMOSOME_SIZE)
    childB[i2]=1-childB[i2]

```

```

return childA,childB

for i in range(0,3):
    rang_units=[]
    for unit in units:
        rang_units.append([get_rang(unit,units),unit])

    rang_units.sort(compare)
    for unit in rang_units:
        print unit[0],unit[1],to_int(unit[1])
    betterPart=rang_units[0:CHROMOSOME_COUNT/2]

    new_list=[]
    for i in range(0,CHROMOSOME_COUNT/2):
        parentA=random.choice(betterPart)[1]
        parentB=random.choice(betterPart)[1]
        while to_int(parentA)==to_int(parentB):
            parentA=random.choice(betterPart)[1]
            parentB=random.choice(betterPart)[1]
        childA,childB=cross(parentA,parentB)

    pretend=[childA,childB,parentA,parentB]
    random.shuffle(pretend)
    rang_childs=[]
    for unit in pretend:
        rang_childs.append([get_rang(unit,units),unit])
    rang_childs.sort(compare)
    new_list.append(rang_childs[0][1])
    new_list.append(rang_childs[1][1])
    units=new_list
    random.shuffle(units)
    print("="*10)

```