

Simulation der Evolution

Die Idee ist aus dem Biologie-Unterricht bekannt: Gute Eigenschaften setzen sich durch, andere sterben aus.

Stichworte:

Anfangspopulation der Individuen, Genpool

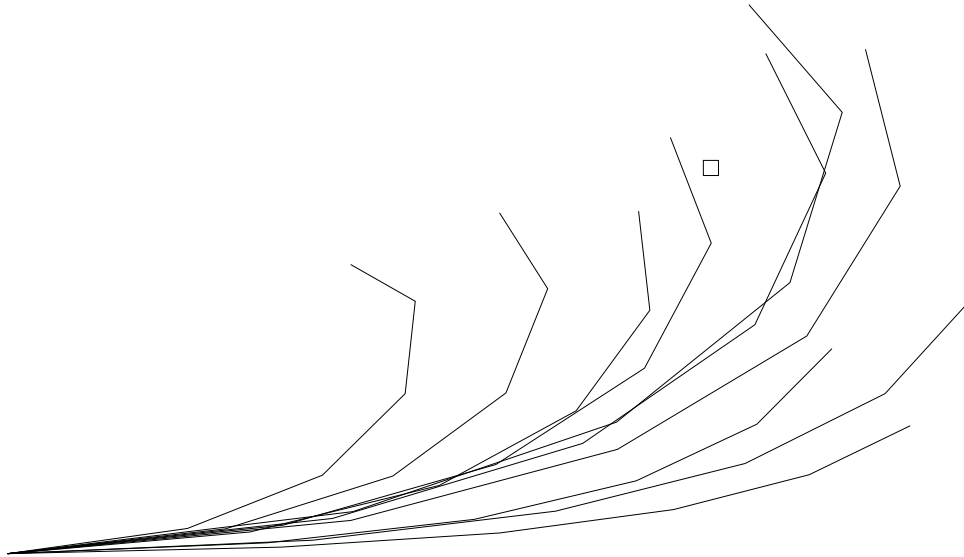
Fitnessfunktion (Bewertung für Individuen)

Selektion, Mutation, Rekombination (Mischen von Erbgut)

Folgegeneration

Individuum mit bester Fitness

Evolutionäre Algorithmen für Optimierungen komplexer Problemstellungen



Jeder Streckenzug ist der Versuch eines Individuums, dem quadratischen Ziel möglichst nahe zu kommen. Start und Ziel sind fest vorgegeben. Jedes Individuum macht 5 Schritte und kann die erste Schrittlänge und den ersten Winkel bestimmen, das Weitere ist durch $\text{Streckenlänge}(n + 1) = 0,8 \cdot \text{Streckenlänge}(n)$ und $\text{Winkel}(n + 1) = 1,7 \cdot \text{Winkel}(n)$ festgelegt (Turtle-Graphik).

(Diese einschränkenden Bewegungs-Bedingungen könnten vielfach variiert werden.)

Versuche zunächst durch einfaches Probieren, dich der Problemstellung zu nähern (Turtle-Graphik).

Programm

```
from tkinter import *
from math import *
from random import *
from turtle import *
from copy import *
import time

zx=160      # Position des Ziels
zy=100
anzahl=8   # Individuen

class Individuum:
    def __init__(self,s,w):
        self.sAnfang=s
        self.wAnfang=w
        self.s=s
        self.w=w

    def move(self,sichtbar):
        penup()
        setpos(-300,-150)
        if sichtbar==1:
            pendown()

        setheading(0)
        self.s=self.sAnfang
        self.w=self.wAnfang
        for n in range(5):
            lt(self.w)
            fd(self.s)
            self.s=0.8*self.s
            self.w=1.7*self.w

        self.d=sqrt( (xcor()- (zx+5))**2+(ycor()- (zy+5))**2 ) # Abstand
        if sichtbar==1:
            write(round(self.d)) # Endpunkt/Ziel
                                # bestimmt die Fitness
```

Programm

```
def ziel():
    penup()
    setpos(zx,zy)
    pendown()
    setheading(0)
    for n in range(4):
        fd(10)
        lt(90)

speed(-1)
ht()

L=[]
for k in range (0,anzahl):
    I=individuum(randint(100,230),8*random())           # Anfangsstreckenlänge
    L.append(I)                                       Anfangswinkel werden zufällig bestimmt

for n in range (0,anzahl):
    L[n].move(0)
L=sorted(L, key=lambda individuum: individuum.d)    # siehe OOP

while L[0].d>10:
    clear()
    ziel()
    for n in range (1,anzahl):
        L[n]=individuum((L[0].sAnfang+L[n].sAnfang)/2+randint(-10,10),
                        (L[0].wAnfang+L[n].wAnfang)/2+2*random()-1)
        L[n].move(0)    # d wird ermittelt

L=sorted(L, key=lambda individuum: individuum.d)

for n in range (0,anzahl):
    color(0.,0.,0.9)
    L[n].move(1)
time.sleep(0.5)
```

Realisiere eigene Ideen zu Selektion, Mutation, Rekombination.