Optimalizáló algoritmusok
Sztochasztikus optimalizálás

May 2, 2023

Genetikus algoritmus

(J. Holland, 1975)

Célfüggvény = ,,fitnesz függvény”

A minimumhely közelítése vektorok egy halmazával (populáció).
1 egyed = 1 vektor (vektor ← kromoszóma)

Minden lépésben egy új populáció meghatározása a jelenlegi generáció egyedeiből, 3 ,,genetikai” művelettel:

- kiválasztás
- mutáció
- keresztezés

Az új populáció kialakításában résztvevő egyedeket (szülők) a fitnesz értékük alapján választjuk ki a jelen populációból.

A 3 genetikai művelet

- kiválasztás: a jelen populáció legjobb fitnesz értékkel rendelkező egyedei (elit egyedek) változtatás nélkül bekerülnek az új populációba
- mutáció: a kiválasztott egyed kromoszómájában egy véletlen változtatás (pl. a kiválasztott vektorhoz hozzáadunk egy normális eloszlásból származó vektort). Cél: a diverzitás növelése, a keresési tér hatékonyabb feltérképezése.
- keresztezés: két kiválasztott egyed kromoszómáinak kombinációjaként áll elő az új egyed (pl. az új vektor minden egyes koordinátáját véletlenszerűen választjuk a két szülő megfelelő koordinátái közül). Cél: az alapjellemzők felismerése, a konvergencia erősítése.

Általában az új populáció lényegesen nagyobb hányada keletkezik keresztezéssel, mint mutációval.

Kiinduló populáció: véletlenszerűen

A populációk egyedszáma állandó.

Az új populáció generálásánál egy egyedet többször is kiválaszthatunk szülőnek.

Példa leállási feltételekre:

- a fitnesz érték alapján
  - a populációban a legjobb fitnesz érték egy adott küszöb alá kerül
  - a populáció átlagos fitnesz értéke egy adott értéknél kevesebbet változik
- az iterációk száma alapján
  - a generált populációk száma meghalad egy adott értéket

A szülők kiválasztása

- A jelen populáció minden egyedének meghatározzuk a fitnesz értékét, ezeket skálázzuk.
- Meghatározzuk a különböző genetikai műveletekkel keletkező egyedek arányát. (Tipikus értékek: elit egyedek 5%, keresztezéssel keletkező egyedek 80%)
- A keresztezéshez és mutációhoz valamilyen véletlen algoritmussal választjuk a szülőket, a kiválasztás valószínűsége arányos a fitnesz értékükkel. Pl.
  - Egy szakaszt felosztunk annyi részre amennyi egyed van, a részek hossza az egyedek a fitnesz értékével arányos. Adott lépésközzel végigmegyünk a szakaszon (a kezdőpontot véletlenszerűen választva, a lépéshossznál kisebb értéknek), minden lépés végén kiválasztjuk a szakasznak megfelelő egyedet.
  - Rulett módszer: a rulett kereket felosztjuk az egyedek fitnesz értékével arányos módon, ,,pörgetünk'', kiválasztjuk az adott számhoz tartozó egyedet.

Mutáció

A kiválasztott egyed génjeit véletlenszerűen módosítjuk. Pl.

- Gauss-mutáció: a vektor koordinátáit egy 0 várható értékű, adott szórású normális eloszlásból származó véletlen számmal módosítjuk. A szórás függ a kiinduló populációtól, és csökken a generációk számának növekedésével. Pl. a k-adik lépésben:

$$\sigma_k = \sigma_{k-1} \left(1 - c \cdot \frac{k}{N}\right),$$

  ahol N a generációk maximális száma, $c > 0$ paraméter. A szórás lehet ugyanaz minden koordináta esetén, vagy koordinátánként változó.
- Egyenletes mutáció: kiválasztjuk a koordináták egy részét, ezeket egy adott valószínűség szerint módosítjuk egy (koordinátánként változó) intervallumból egyenletes eloszlással választott értékre.

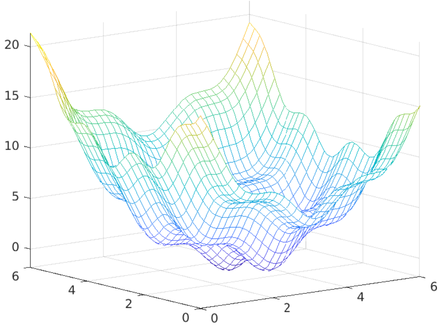
Keresztezés

Az új egyed génjeit 2 szülő génjeiből állítjuk elő valamilyen véletlen algoritmus alapján. Pl. (n változós esetben)

- minden gén esetén véletlenszerűen kiválasztjuk az egyik szülő ugyanilyen pozíción álló génjét
- kiválasztunk egy k egész számot 1 és n között, az első k gént az első, a maradék géneket a második szülőtől vesszük
- kiválasztunk egy k és egy m egész számot 1 és n között ($k < m$), a k-nál kisebb és az m-nél nagyobb sorszámú géneket az első, a többit a második szülőtől vesszük
- az adott gén a két szülő ugyanilyen pozíción álló génjeinek súlyozott összege
- az adott gén a két szülő ugyanilyen pozíción álló génjeinek konvex lineáris kombinációja, közelebb a jobb fitnesz értékű szülőhöz

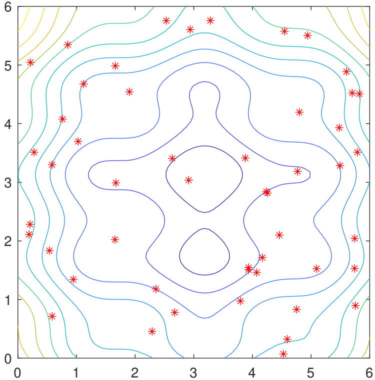
Példa

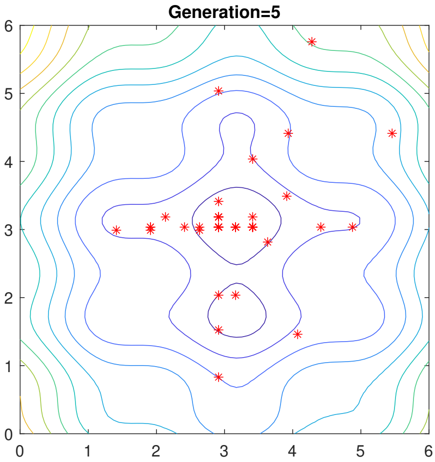
$$f(x) = (x_1 - 3.14)^2 + (x_2 - 2.72)^2 + \sin(3x_1 + 1.41) + \sin(4x_2 - 1.73)$$

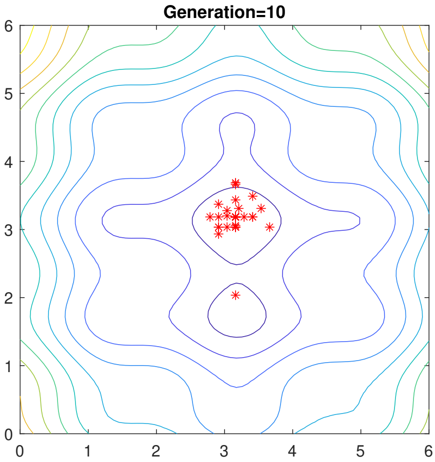


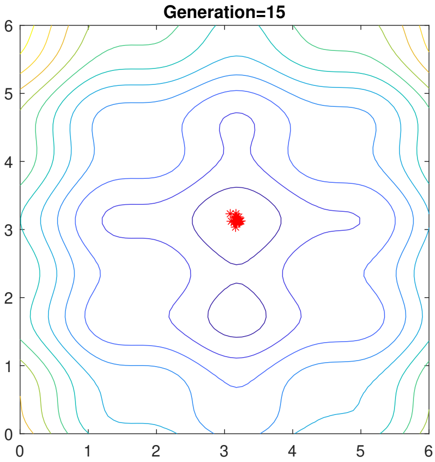
$x^* = [3.1852, 3.1298]$

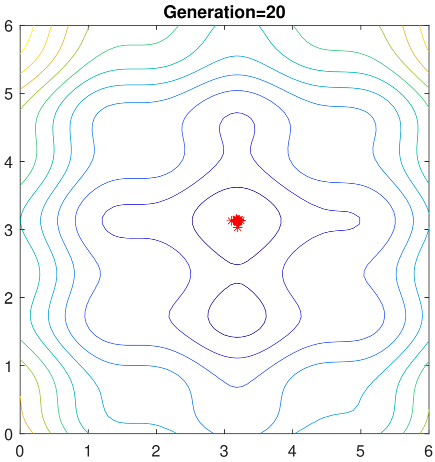
A kiinduló populáció (50 egyed)

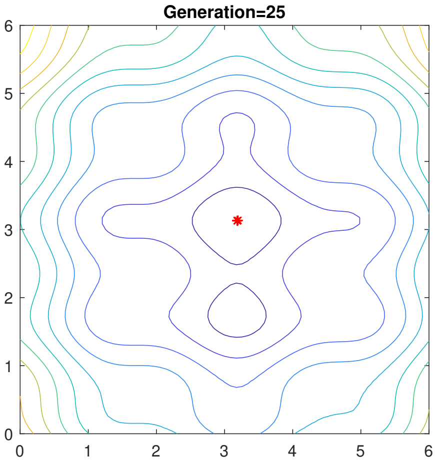



Generation=5
0
1
2
3
4
5
6
0
1
2
3
4
5
6


Generation=10
6
5
4
3
2
1
0
0
1
2
3
4
5
6


Generation=15
0
1
2
3
4
5
6
0
1
2
3
4
5
6


Generation=20
6
5
4
3
2
1
0
0
1
2
3
4
5
6


Generation=25
6
5
4
3
2
1
0
0
1
2
3
4
5
6

$$f(x) = (x_1 - 3.14)^2 + (x_2 - 2.72)^2 + \sin(3x_1 + 1.41) + \sin(4x_2 - 1.73)$$

Leállás 61 generáció után, $x = [3.1852, 3.1298]$
Függvénykiértékelések száma: 2920

Az algoritmus néhány paramétere:

egyedszám: 50,
3 elit egyed, 37 keresztezéssel, 10 mutációval keletkező egyed
a keresés a $[0, 6]^2$ tartományra szűkítve
a keresztezésnél véletlenszerűen választ a szülők génjeiből
a mutációnál Gauss-mutáció

Particle swarm optimization (PSO)

(Kennedy, Eberhart, 1995)

Hal- és madárrajok mozgása és egymás közötti információ megosztás alapján.

A minimumhelyet vektorok egy halmazával (raj) közelítjük.

x_i^t: a t-edik pillanatban az i-edik egyed elhelyezkedése
v_i^t: a t-edik pillanatban az i-edik egyed sebessége
$x_i^{t+1} = x_i^t + v_i^t$: a $(t+1)$-edik pillanatban az elhelyezkedése

Az új sebességvektor (v_i^{t+1}) függ

- az egyed előző sebességvektorától
- az egyed által eddig meglátogatott legjobb helytől
- a szomszédai által eddig meglátogatott legjobb helytől

Az algoritmus indítása

- a raj mérete (S, Matlab: $\min\{100, 10 \cdot nvars\}$)
- az egyedek elhelyezkedése (véletlenszerűen egy tartományon belül)
- az egyedek sebessége (véletlenszerűen egy tartományon belül)
- a szomszédok száma (N), kezdetben $N = N_{\min}$ (pl. 2).
- a tehetetlenség (w, az előző sebességvektor súlya a sebességvektor frissítésénél)
- a „sajátsúly" (y_1, a súly az egyed által eddig meglátogatott legjobb pont figyelembevételéhez a sebességvektor frissítésénél)
- a „közösségi súly" (y_2, a súly az egyed szomszédai által eddig meglátogatott legjobb pont figyelembevételéhez a sebességvektor frissítésénél)

p_i: az i-edik egyed által eddig meglátogatott legjobb pont
d: a raj által eddig meglátogatott legjobb pont
b: a célfüggvény értéke d-ben
c: számláló (sikertelen lépések, kiinduló érték 0)

Az algoritmus

Legyen a célfüggvény f.

Adott i esetén a helyzet és a sebességvektor frissítése:

- véletlenszerűen kiválasztunk N (az i-ediktől különböző) egyedet a rajból
- $v_i := w \cdot v_i + y_1 u_1(p_i - x_i) + y_2 u_2(g - x_i)$, ahol
  - g az N kiválasztott egyed által eddig meglátogatott legjobb pont
  - u_1, u_2 egyenletes eloszlásból véletlen vektorok
- $x_i := x_i + v_i$
- ha $f(x_i) < f(p_i)$, akkor $p_i := x_i$

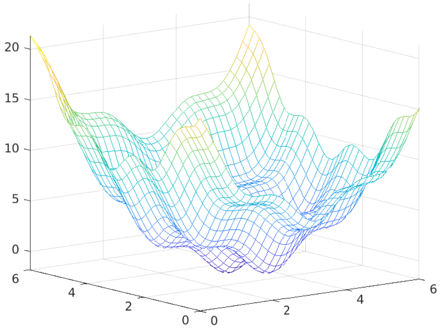
Ha minden egyed helyét és sebességvektorát frissítettük, akkor a paraméterek frissítése:

Frissítsük b (az eddigi legjobb függvényérték) és d értékét.

- Ha b értéke csökkent, akkor
  - $c := \max\{0, c - 1\}$
  - $N := N_{\min}$
  - ha $c < 2$, akkor $w := 2w$ (w-re van egy felső korlát!)
  - ha $c > 5$, akkor $w = w/2$
- Ha b értéke nem csökkent, akkor
  - $c := c + 1$
  - $N := \min\{N + N_{\min}, S\}$

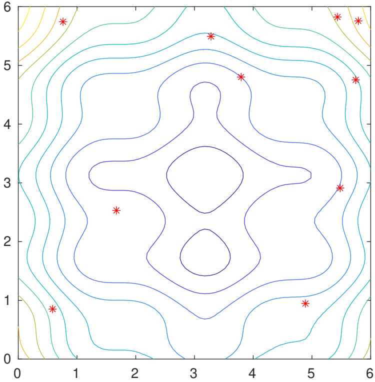
Példa

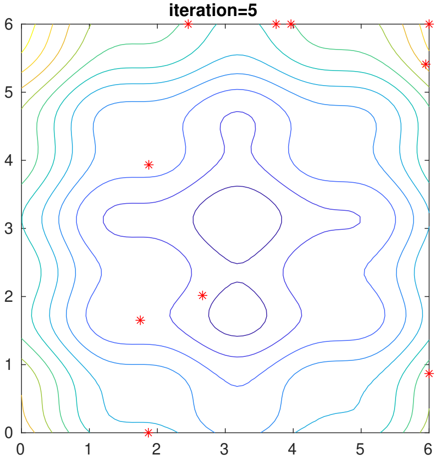
$$f(x) = (x_1 - 3.14)^2 + (x_2 - 2.72)^2 + \sin(3x_1 + 1.41) + \sin(4x_2 - 1.73)$$

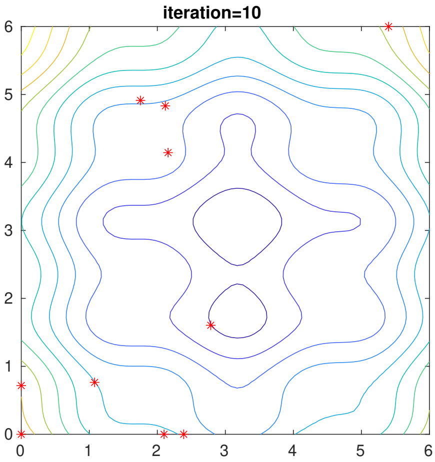


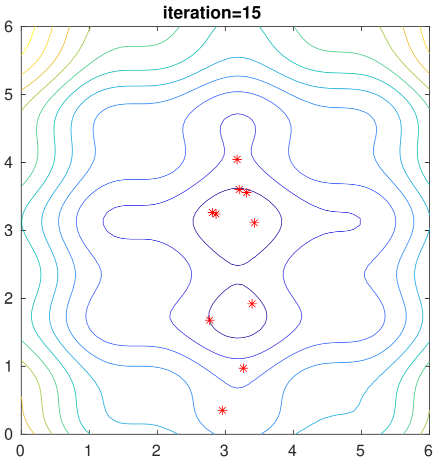
$x^* = [3.1852, 3.1298]$

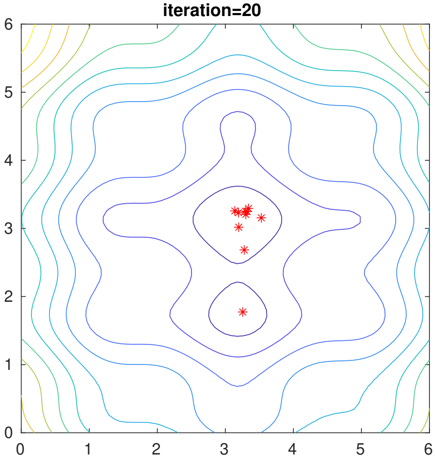
A kiinduló populáció (10 egyed)

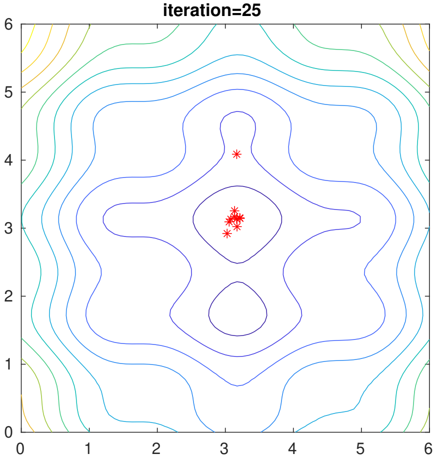


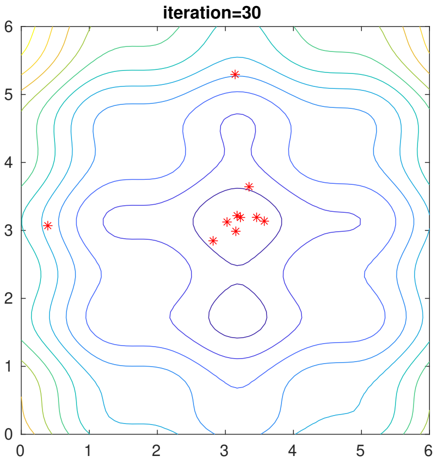

iteration=5
0
1
2
3
4
5
6

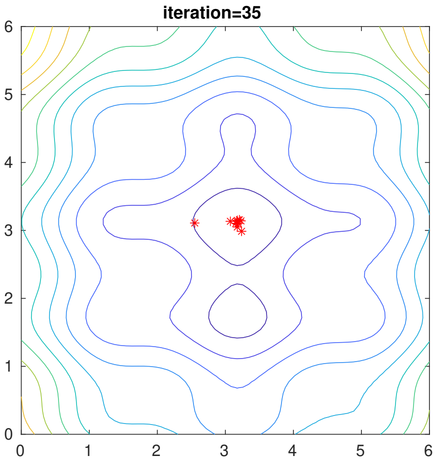

iteration=10
6
5
4
3
2
1
0
0
1
2
3
4
5
6

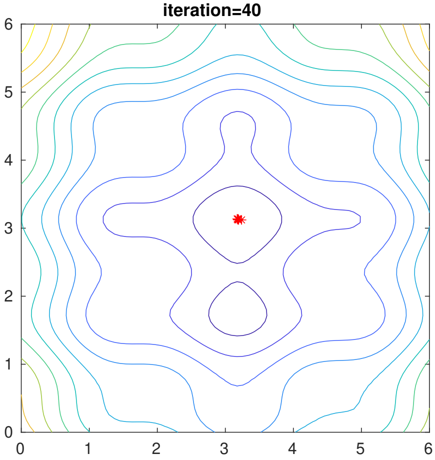

iteration=15
0
1
2
3
4
5
6
0
1
2
3
4
5
6

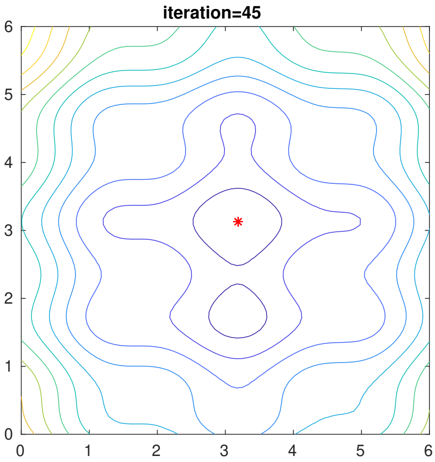

iteration=20
0
1
2
3
4
5
6


iteration=25
6
5
4
3
2
1
0
0
1
2
3
4
5
6


iteration=30
6
5
4
3
2
1
0
0
1
2
3
4
5
6


iteration=35
0
1
2
3
4
5
6
0
1
2
3
4
5
6


iteration=40
6
5
4
3
2
1
0
0
1
2
3
4
5
6


iteration=45
0
1
2
3
4
5
6
0
1
2
3
4
5
6

$$f(x) = (x_1 - 3.14)^2 + (x_2 - 2.72)^2 + \sin(3x_1 + 1.41) + \sin(4x_2 - 1.73)$$

A keresés a $[0,6]^2$ tartományra szűkítve.

egyedszám	10	15	20
iteráció	71	62	45
fv kiértékelések	720	945	920

Mindhárom esetben a minimumhely közelítése: $x = [3.1852, 3.1298]$