Evolúciós algoritmusok 5. előadás

Evolúciós stratégiák

Valós értékű reprezentáció

Bizonyos optimalizálási feladatokban a megoldások kódolására a természetes választás a valós számok. Ilyen feladatok például:

- ha a változók fizikai mennyiségeket jelölnek (súly, hossz, szög)
- egy neurális hálóban a súlyok optimalizálása
- bizonyos típusú függvények szélsőértékkeresése

Ilyenkor a megoldások genotípusa egy $(x_1, \ldots, x_k) \in \mathbb{R}^k$ vektor.

Szimulált hűtés

A genetikus algoritmusok ősének tekinthető egyik első függvényoptimalizáló eljárás a szimulált hűtés módszere volt, amely a kihűlés utánzásával igyekezett egy jó, véletlenen alapuló lokális minimumkeresést megvalósítani egy n változós függvényre.

A működési elve a következő: kiindul egy véletlen keresési pontból, majd generál néhány új potenciális pontot egy olyan eloszlás szerint, amely az eljárás kezdetén nagy valószínűséggel ad távolabbi pontokat is (a várható érték végig 0 marad).

Szimulált hűtés

Az új pontok közül kiválasztja a legjobbat, azaz ahol a célfüggvény értéke a legkisebb. Ha nincs jobb a kiindulási pontnál, akkor véletlenszerűen, de nem egyenletesen választ egyet, majd ez lesz az új kiindulási pont. Ezen lépéseket iterálja.

Az algoritmus futása alatt az új pontokat adó eloszlás paraméterét úgy változtatjuk, hogy az új pontok várhatóan egyre közelebb lesznek a kiindulási ponthoz, ezzel érjük el a konvergenciát.

Új pontok generálása

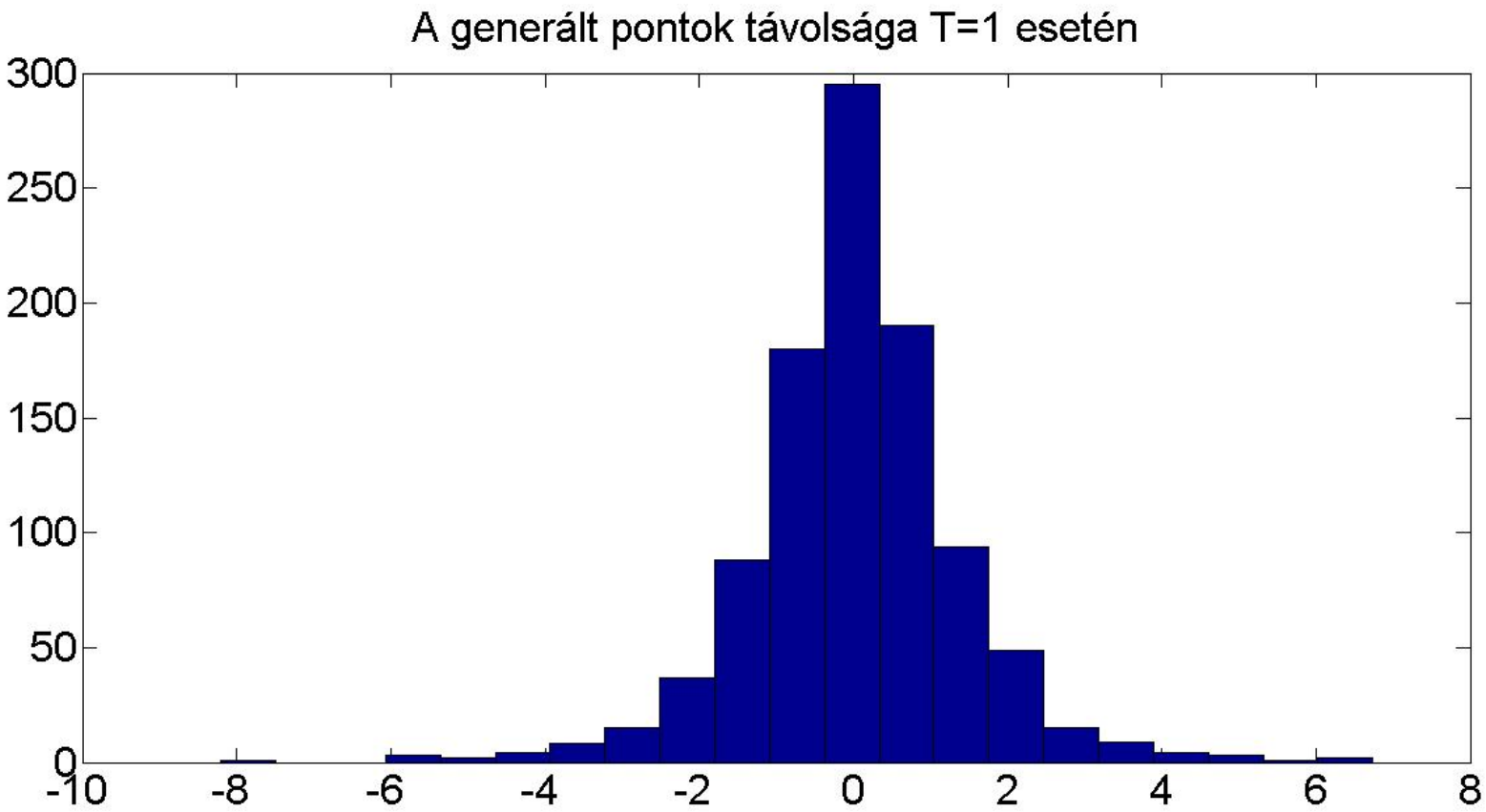
Az új pontokat jelöljük $u^1, u^2, \ldots u^m$-mel. Ekkor egy új pont koordinátáit a kiindulási x pont koordinátáiból az

$$u_i^j = x_i + T \cdot (\ln(rand1) - \ln(rand2))$$

módon nyerjük, ahol rand1 és rand2 egyenletes eloszlású véletlen számok a $[0, 1]$-en.

A sűrűségfüggvényéből kiszámítható, hogy $1 - exp\left(\frac{-k}{T}\right)$ eséllyel esik az új pont minden koordinátája az $(x_i - k, x_i + k)$ intervallumba.

A T paraméter értéke fog folyamatosan csökkenni, ezzel érjük el, hogy egyre közelebb legyenek a pontok a kiindulási ponthoz.


A generált pontok távolsága T=1 esetén
300
250
200
150
100
50
0
-10
-8
-6
-4
-2
0
2
4
6
8

Új pont kiválasztása

Ha van az új pontok között egy olyan, ahol a függvényérték kisebb mint a kiindulási pontban, akkor őt választjuk, illetve ezek között a legjobbat.

Ha viszont nincs ilyen pont, akkor legyen a j-edik ponthoz rendelt érték

$$g(j) = \exp\left(\frac{f(x) - f(u_j)}{T}\right),$$

majd $g(j)$-kből valószínűségi eloszlást készítünk normálással, és ezen eloszlás szerint választunk új kiindulási pontot (mint a rulettkerék-módszernél).

Ha az új pontbeli függvényérték közel van a minimumhoz, akkor $g(j)$ nagyobb lesz, így érjük el, hogy a jobbnak tűnő irányoknak több esélye legyen.

Paraméterek

Érdemes T-n olyannak választani, hogy a kiindulási pont és a valódi optimum távolságának nagyságrendjébe essen.

A hőmérsékletet valahány iterációnként (például 10) az addigi konstans-szorosára csökkentjük, itt 0.9 körül van a standard választás. Túl gyors csökkentés esetén lokális optimumba ragadhatunk, túl lassú esetén nem várhatunk konvergenciát.

Az új pontok számát sem érdemes túl magasra venni. Ez egyrészt lassítja az algoritmust, másrészt növeli a lokális minimumba ragadás esélyét.

Összevetés a genetikus algoritmusokkal

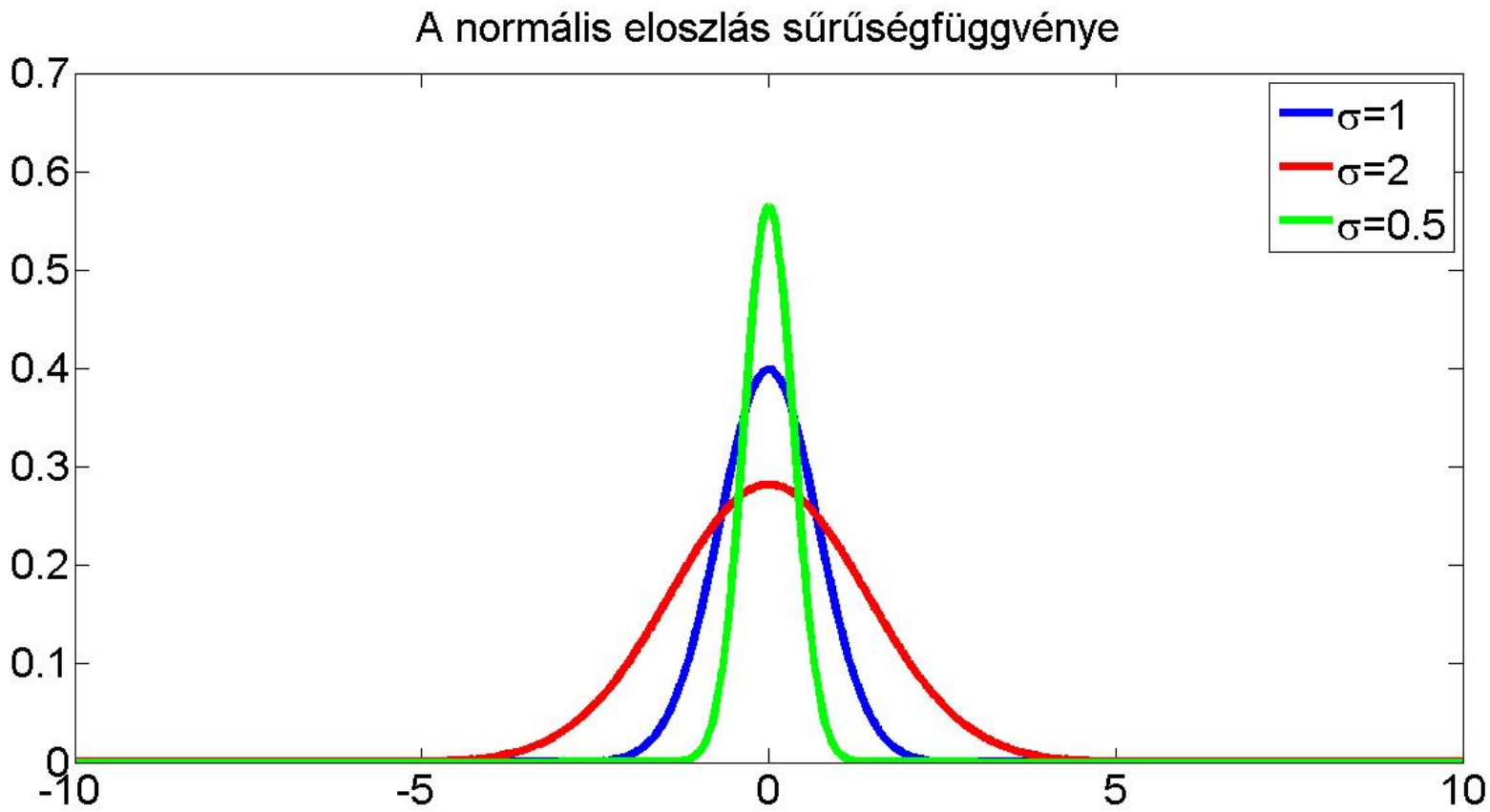
A szimulált hűtésben megjelenik a a genetikus algoritmusok néhány alapötlete: a megoldás módosításával ('mutációval') kerülünk közel az optimumhoz.

A több megoldás szimultán kezelése viszont hiányzik, és a paramétereket 'vakon' kell változtatnunk a futás közben.

Rechenberg algoritmusa

Az egyik első genetikus jellegű algoritmus folytonos optimalizálási feladatra Rechenberg nevéhez fűződik. Ez több vonásában is különbözik a szimulált hűtéstől

- csak egy új pontot hoz létre.
- az új pont koordinátáit úgy kapja, hogy a régi ponthoz egy 0 várható értékű, σ szórású normális eloszlású véletlen számot ad hozzá.
- ha az új pont jobb, akkor ő lesz a kiindulási pont, ha nem, akkor marad a régi (elitista)
- a szórást adaptívan változtatja


A normális eloszlás sűrűségfüggvénye
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-10
-5
0
5
10
σ=1
σ=2
σ=0.5

Rechenberg 1/5-ös szabálya

Minél nagyobb σ értéke, várhatóan annál nagyobbakat lép az algoritmus.

Ezért a kezdetben nagyobb σ a célszerű, hogy felderítsük a keresési teret, később, amikor már közel vagyunk a megoldáshoz kicsi σ, hogy pontosan tudjuk közelíteni a megoldás értékét.

Ha sok sikeres lépés van, akkor jó irányba tartunk. Érdemes ekkor σ értékét növelni, hogy gyorsabban legyen a konvergencia.

De honnan tudjuk, hogy már közel vagyunk a jó megoldáshoz? Onnan, hogy nem egykönnyen tudunk lépni: szinte minden lépésnél növekedne a függvény, ilyenkor kell σ értékét csökkenteni.

Rechenberg 1/5-ös szabálya

Legyen $0.817 < c < 1$ előre rögzített (ez az adaptáció feltétele). Néhány (mondjuk k) iterációnként számoljuk meg, hogy hányszor sikerült jobb függvényértéket találunk, legyen ezek gyakorisága p. Ezután k iterációnként változtassuk meg σ értékét a következőképpen:

$$\sigma = \begin{cases} \frac{\sigma}{c} & p > \frac{1}{5} \\ \sigma \cdot c & p < \frac{1}{5} \\ \sigma & p = \frac{1}{5} \end{cases}$$

Ez az algoritmus alkalmazkodik az optimalizálandó függvényhez, nem nekünk kell a paramétereket vakon változtatni.

Valós értékű reprezentáció

Olyan problémákat szeretnénk megoldani, melyek természetes kódolásához valós (lebegőpontos) számokat használunk, így egy genotípus egy $\mathbb{R}^n$ vektor lenne, ahol n a feladat dimenziója.

Ezt a vektort még kiegészítjük, így a genotípus két részből fog állni: egy $\mathbf{x}$ vektorból, ami a megoldást kódolja, és egy $\boldsymbol{\sigma}$ vektorból, ami az evolúciós stratégia paramétereit.

Ezeket egybefűzzük, azaz magának az egyednek lesz tulajdonsága, hogy a mutáció mekkorát lép, és így a paramétereket is maga az algoritmus optimalizálja. Ezzel fogjuk megvalósítani az önadaptációt.

Mutáció

A mutáció folytonos esetben a megoldás koordinátáinak megváltoztatását jelenti, tipikusan az összeset egyszerre.

A leggyakrabban használt mutáció operátor a 0 várható értékű, σ szórású, normális eloszlású véletlen szám hozzáadása a koordinátákhoz, ennek sűrűségfüggvénye:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2\sigma^2}}$$

Ha semmiképpen sem akarunk kilépni egy korlátos intervallumból az adott koordinátában, akkor célszerű az adott intervallumon egyenletes eloszlást választani.

Ha olyan eloszlást szeretnénk, amelynél nagy valószínűséggel lép nagyot, akkor a Cauchy eloszlást használhatjuk, mert ez kevésbé van a 0 körülre koncentrálva (nincs is várható értéke).

Egyenletes lépésköz

Egyenletes lépésköz esetén generálunk n darab véletlen normális eloszlású σ szórású számot, majd ezeket egyesével hozzáadjuk a koordinátákhoz, azaz a genotípus $(x_1, \ldots, x_n, \sigma)$.

Jelöljük $N(0, \sigma)$-val a 0 várható értékű, σ szórású normális eloszlású valószínűségi változót, ekkor

$$\sigma' = \sigma \mathrm{e}^{\tau N(0,1)}, \qquad \sigma' < \varepsilon_0 \rightarrow \sigma' = \varepsilon_0,$$

és

$$x_i' = x_i + N(0, \sigma)$$

Kétszer is kiértékeljük a megoldást: először, hogy mennyire jó minőségű az x vektor (amikor a szülőket választjuk ki), másrészt, hogy mennyire jó minőségű utódokat tud létrehozni, ezért fontos, hogy először σ értékét változtassuk, csak utána x-ét.

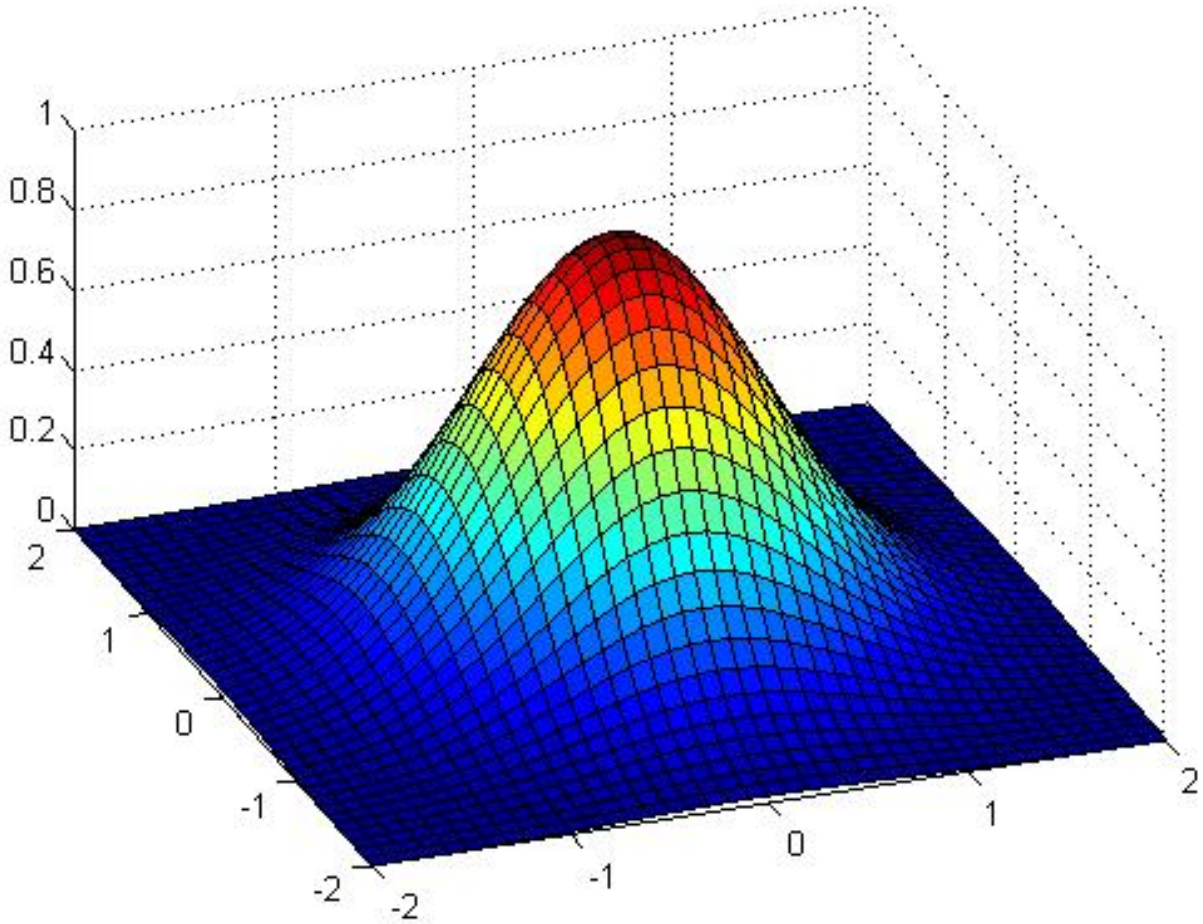
Lognormális eloszlás

Az $\mathrm{e}^{\tau N(0,1)}$ véletlen szám eloszlását lognormálisnak nevezik. Azért ezt használjuk a szórás mutációjához, mert:

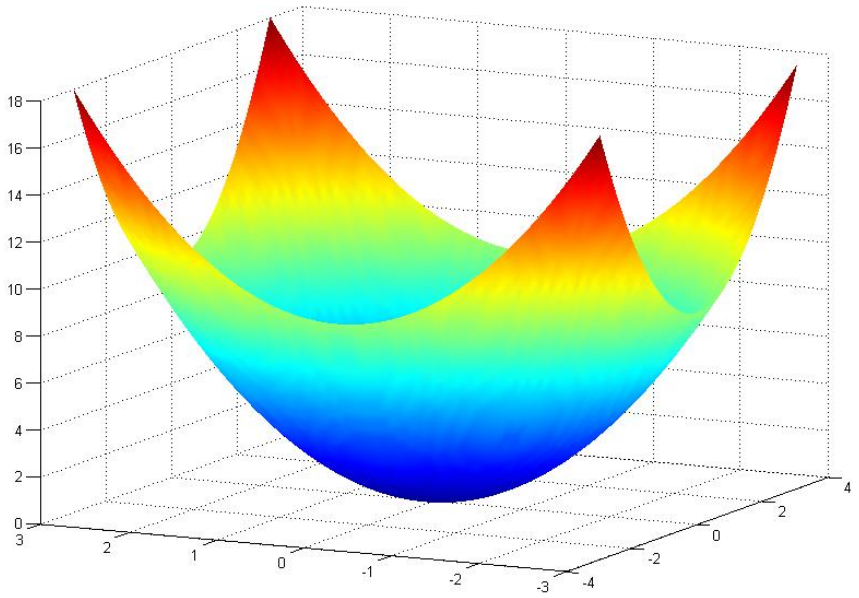
- többnyire a kisebb változtatások a megfelelőek
- a medián (0.5-ös percentilis) 1, azaz éppolyan valószínű, hogy növeli a szórást, mint az, hogy csökkenti
- a mutációnak hosszú távon semlegesnek kell lennie, és ennek a várható értéke 1

Itt a τ az eljárás paramétere, amelyet 'tanulási sebességnek' neveznek. Ezt $\tau \sim \frac{1}{\sqrt{n}}$-nek érdemes választani.

Korrelálatlan normálisok 2 dimenzióban, azonos szórás

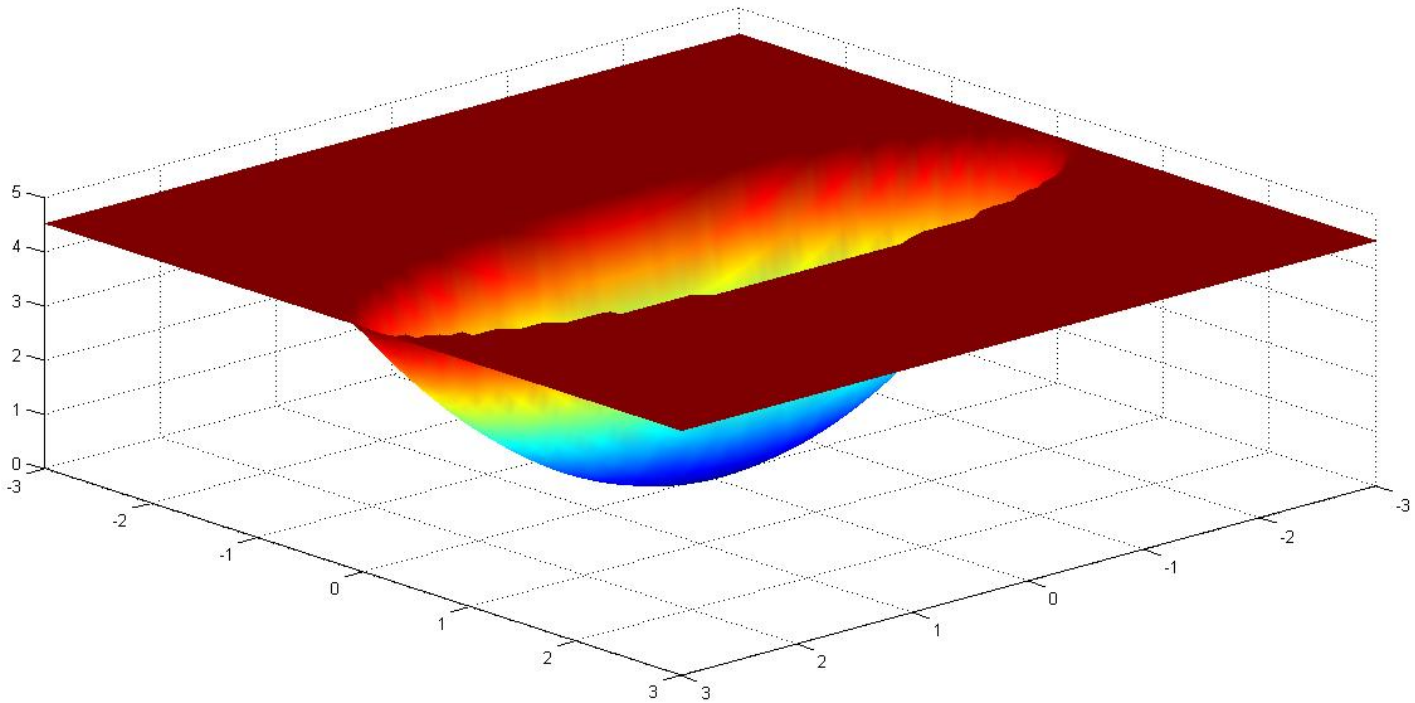


2 dimenziós probléma



Ez egy $a\xi^2 + b\eta^2$ típusú függvény. Tudjuk, hogy egy deriválható függvény a lokális szélsőérték közelében ilyen alakú, hiszen a gradiens 0. Ezen konkrét függvénynél $a \sim b$, és $a, b > 0$.

2 dimenziós probléma



Itt viszont $a >> b > 0$.

Korrelálatlan mutáció

Annak érdekében, hogy megfelelő nagyságú lépést tudjunk tenni minden irányba, megváltoztatjuk a genotípust. Most $(x_1, , \ldots, x_n, \sigma_1, \ldots, \sigma_n)$, alakú lesz. Azaz az algoritmus koordinátánként tanulja a megfelelő távolságot. Ilyenkor

$$\sigma_i' = \sigma_i \mathrm{e}^{\tau N(0,1) + \tau' N_i(0,1)}, \qquad \sigma' < \varepsilon_0 \rightarrow \sigma' = \varepsilon_0,$$

és

$$x_i' = x_i + N_i(0, \sigma_i)$$

Itt az i index azt jelenti, hogy koordinátánként külön generáljuk a véletlen számot.

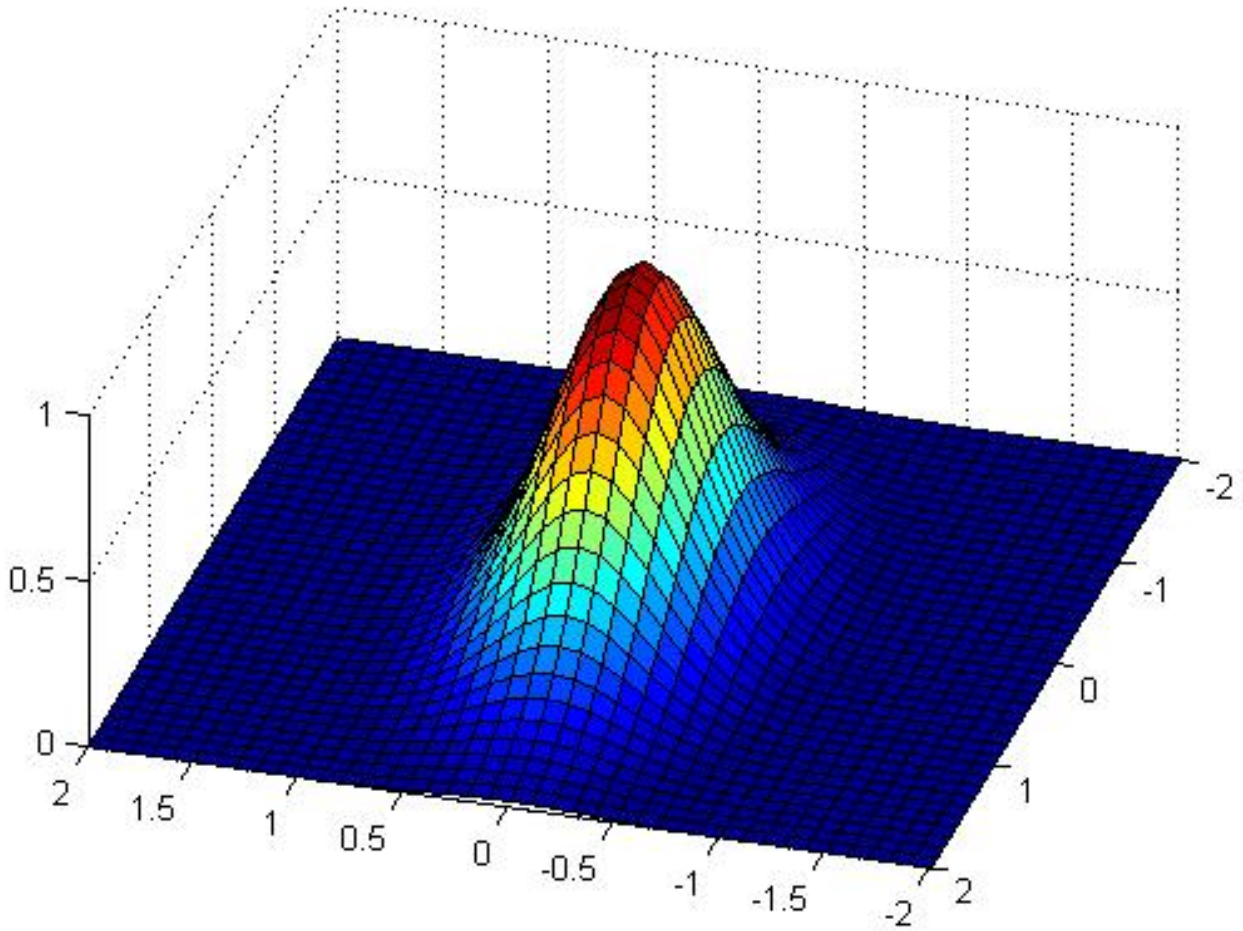
σ mutációja

Mivel két független normális eloszlású szám összege is normális eloszlású (a várható értékek összeadódnak, a szórásokból $\sqrt{\sigma_1^2 + \sigma_2^2}$ lesz), így az új mutáció szorzója is egy 1 várható értékű lognormális.

A mutáció esetén van egy alap: $\tau N(0,1)$, amely koordinátafüggetlen (azt méri, hogy hol tartunk az eljárásban), és egy koordináta-függő: $\tau' N_i(0,1)$, amely lehetővé teszi, hogy koordinátánként eltérő lépéstávolságot használjunk.

A standard választás a $\tau \sim \frac{1}{\sqrt{2n}}$, és $\tau' \sim \frac{1}{\sqrt{2\sqrt{n}}}$.

Korrelálatlan normálisok 2 dimenzióban, különböző szórások



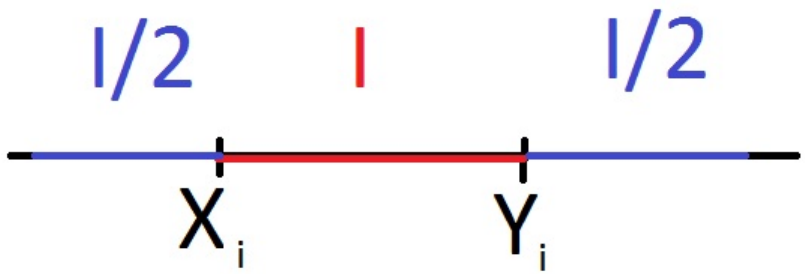
Keresztezés operátorok

Háromféle keresztezést különböztetünk meg

- diszkrét keresztezés, amikor az új allél megegyezik az egyik szülő alléljával. Ezt érdemes használni a genotípus megoldást tartalmazó részén, hogy fenntartsuk a populáció diverzitását.
- számtani keresztezés: legyen α 0 és 1 közötti szám, akkor az egyik utód $\alpha x + (1 - \alpha)y$, a másik $(1 - \alpha)x + \alpha y$. Ilyen típust érdemes használni a genotípus paramétereket tartalmazó részén.
- keverés: az előbbi α konstans helyett egy véletlen szám lesz.

Keverés

α legyen $2u - 0.5$, ahol u egy $[0, 1]$-en egyenletes véletlen szám. Ilyenkor $1/2$ eséllyel esik az utód a két szülő közé, és egyébként egyenletes lesz az $[1.5x_i - y_i, 1.5y_i - 0.5x_i]$ intervallumon. Ezzel fenntarthatjuk a diverzitást, így ez az operátor alkalmas a diszkrét keresztezés helyettesítésére.



A keresztezés globális, azaz egy utódnak kettőnél több szülője is lehet (akár koordinátánként különböző).

Túlélők kiválasztása

Az eddigi algoritmusok pontosan annyit utódot készítettek, mint ahány szülő volt, azaz a populáció elemszáma konstans volt.

Az evolúciós stratégiák tipikusan jóval több utódot (legyen a számuk λ) hoznak létre, majd ezek közül kiválasztják a legjobbakat (mondjuk μ darabot). A szelekciós nyomást ezen két szám hányadosa határozza meg, amelyet $\frac{1}{4}$ és $\frac{1}{7}$ között szokás választani. Ezt a módszert nevezik (λ, μ) kiválasztásnak.

Ha a szülők közül is választhatok utódot azt $(\lambda + \mu)$ kiválasztásnak nevezzük, de ilyenkor valószínűbb, hogy beleragad egy lokális optimumba. Ezen kiválasztás esetén a paraméterek is nehezen adaptálódnak, mert egy rossz lépéstávú (azaz rossz utódokat generáló), de jó függvényértékű megoldást az algoritmus folyton kiválaszt.

Az elitizmus a fenti probléma miatt folytonos feladatoknál nem javasolt.