Evolúciós számítások (EC) a mesterséges intelligencia része.

EC = Gyűjtőnév, olyan algoritmusokat takar, melyek **populációban fejlesztik a megoldásaikat**.

Populáció egyedekből áll. Van egy kezdeti populáció. Valamilyen modell alapján képezünk újabb egyedeket, azaz utódokat. Ezek valamely feltétel teljesülése esetén bekerülnek a populációba. Ez az eljárás ismétlődik egy adott feltételig. A legjobb egyed a végső populációban lesz a feladat megoldása.

EC módszereinek osztályozása:

1. **Biológiai evolúció ösztönözte módszerek**. Jellegzetesség: természetes szelekció, öröklési folyamat. Kronológiailag a legelső EC eljárások: genetikus algoritmus (GA), evolúciós stratégiai (ES), evolúciós programozás (EP), genetikus programozás (GP). Ezek mind evolúciós algoritmusok (EA). Gyakran szokták az EA fogalmát az EC szinonimájaként emlegetni. Pedig nem az!
2. **Biológiai eljárás, viselkedés ösztönözte módszerek**: emberi immunrendszer működése, vagy raj-intelligencia motiválja (madarak, hangyák, méhek).
3. **Matematikai modelleket alkalmazó módszerek**. Ez a polip esete (János bácsi állatkertben).

Egyed: a feladat összes lehetséges megoldását tartalmazó keresési tér egy eleme.

Egyed reprezentációja: az adatstruktúra, mellyel reprezentáljuk az elemet. Legtöbbször ez egy vektor.

Populáció: adott számú egyedet tartalmazó halmaz melyet elkülönítve tárolunk

Fitneszfüggvény: a teljes keresési térben értelmezett függvény, mely megadja egy megoldás (egyed) jóságát.

Szelekció: egy kiválasztási eljárás, amely egyedeket választ ki a populációból. Ezek az egyedek lesznek a szülők. Szelekció vonatkozhat az utódokra is, lásd: Spárta.

Rekombináció: evolúciós kereső művelet, a szülők által meghatározott keresési térben újabb elemeket, utódokat keres. Egyetlen szülő rekombinációja **klónozást** jelent.

Mutáció: evolúciós kereső művelet, utód környezetében újabb utódo(ka)t keres.

Visszahelyezés: kiválasztási eljárás, mely meghatározza, melyik utódok kerülnek vissza a populációba.

Generáció: egy időben létező egyedek a populációban.

Generációs ciklus: …

```
t=0
P(t) kezdeti populáció generálása
Fitness(P(t)) kiértékelése
Repeat
   t++; P(t)=ø
   Szelekció(P(t-1))
   Kereső műveletek alkalmazása (rekombináció, mutáció)
   Fitness(utódok) kiértékelése
   Visszahelyezés(P(t))
Until megállási feltétel teljesül
```

Elvileg mikor kell/érdemes EC-t alkalmazni? Amikor nincs más; amikor a másik eljárás sokáig tart; amikor valami újat akarunk csinálni…

Bármely EA alkalmazása esetén három alapvető részfeladatot kell megvizsgálni:

1. Hogyan ábrázolható az egyed. Pl bináris vagy valós vektor.
2. Milyen rekombinációt vagy mutációt alkalmazzunk?
3. Milyen szelekció és visszahelyező művelet jöhet szóba?

A fitness függvényt adja a feladat célfüggvénye, vagy abból képezzük.

Egyedek ábrázolási formája

1. Valós (vagy egész) vektor: $E=(x_1, x_2, ..., x_n)$, mindegyik x_i egy-egy tulajdonsághoz hozzárendelt változó. Ezek általában egy-egy véges intervallumban vannak értelmezve. A kereső műveletnek kell arra vigyázni, hogy ne lépjünk ki az intervallumból.
2. Permutáció – utazó ügynök feladat, ütemezési feladatok, stb. $E=(\pi_1, \pi_2, ..., \pi_n)$
3. Kombinálni is lehet az első kettőt: $E=(x_1, x_2, ..., x_n, \pi_1, \pi_2, ..., \pi_n)$
4. Bináris vektor: 1110101001001110001100000110011001...

Szelekció

Populáció átlagos minőségét hivatott javítani. Fitneszfüggvény méri az egyedek jóságát. Jobb egyedeket nagyobb valószínűséggel választunk ki szülőnek. Szelekció alapja az a megfigyelés, hogy **a szülő és az utód fitnesz értékei között korreláció mutatható ki**. Nem csak a szelekció, hanem a kereső műveletek is javítják a populációt. Ez a két művelet egyensúlyban kell legyen, mivel a populáció változatosságát is fenn kell tartani.

Sokféle szelekció létezik, matematikai és biológiai hátterű egyaránt.

Tulajdonságok:

1. Szelekciós intenzitás (*selection intensity*): Int=(M*-M)/σ*, ahol * az újat jelenti, M pedig az átlagos fitnesz értéket. Megmutatja, milyen mértékben vannak a legjobb egyedek kiválasztva.
2. Változatosság elvesztése (*loss of diversity*): a populáció azon egyedeinek D aránya, amelyeket nem választott ki a szelekció. Kis érték esetén csökken a korai konvergencia veszélye, mely valamely helyi szélsőértékhez vezet.
3. Szelekciós variancia (*selection variance*): $V=((\sigma^*)/(\sigma))^2$. A konvergencia sebességét lehet vele analizálni.

Rulett szelekció = fitneszarányos szelekció (fitness proportional selection)

$p(E_i)=f(E_i)/SUM(f(E_j)), j=1...n$ fitneszfüggvény nem lehet negatív

Mindezt megismételjük μ alkalommal, így kapunk μ elemű szülőcsoportot.

Sztochasztikus univerzális mintavétel (stochastic universal sampling, SUS) by Baker 1987

Az előzőhöz képest minimalizálja a duplikációk számát. Egyszerre választ a rulett keréken mü darab egyenletes eloszlású mutató szerint.

Versengő szelekció (tournament selection)

Nem a fitness értéke, hanem csak a sorrendje számít. Előbb választ egy véletlenszerű csoportot *tour* db egyeddel (*tour* egy paraméter), majd ezek közül a legnagyobb fitnesz értékű lesz a kiválasztott. Mindezt megismételjük mü alkalommal.

Csonkolásos szelekció (truncation selection)

Kiválasztjuk a populáció T-ed részét, a legjobb fitness értékűeket. Ezután ezek közül bármelyik lehet szülő, azonos eséllyel. Ez nem egy biológiai szelekció, ez mesterséges eljárás.

Lineáris sorrend alapú szelekció (linear ranking selection)

Fitnesz sorrendet alakítunk ki 1...n értékeket kapnak, 1 a legrosszabb, n a legjobb.

Pi = (2-SP)/mü + 2i(SP-1)/(mü(mü-1)), ahol SP in [1,2] a szelekciós nyomás

Rekombináció

A kiválasztott szülők állománya a populáció egy része (vagy egésze), amely tartalmazhat ismétlődéseket is. A rekombináció **több szülő felhasználásával képez utódot**. A rekombiáció a szülők környezetében képezi az utódot, a szülők által kijelölt keresési tér valamely pontja lesz az utód.

Általában 2 szülőből állítunk elő egy vagy két utódot. Cél: újabb, jobb megoldások összeállítása tulajdonságok öröklésével.

Diszkrét rekombináció (vagy uniform)

Többféle változó típus esetén alkalmazható. Hiperkockát képezünk a szülőkből: a két szülő a testátló mentén helyezkedik el, a kölkök pedig a további csúcsok közül választhatók ki.

$(x_1, x_2, ..., x_n)$, $(y_1, y_2, ..., y_n)$, $(u_1, u_2, ..., u_n)$, akkor $u_i=a_i x_i + (1-a_i) y_i$ ahol a_i vagy 0 vagy 1

Véges számú lehetőség van.

Létezik egy verzió több mint 2 szülőre. Ez esetben is mindegyik tulajdonság öröklődik egyik véletlenszerűen kiválasztott szülőtől. Új érték nem keletkezik egyik tulajdonságnál sem.

Egész és valós típusú változók rekombinációja

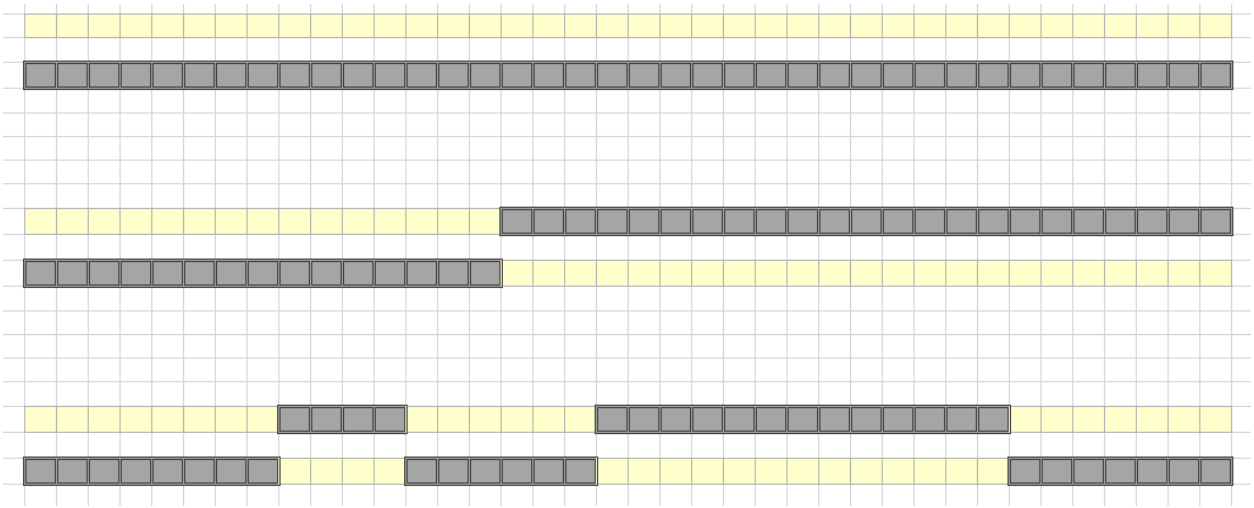
Köztes rekombináció (intermediate recombination). A keresési teret szintén a hiperkocka határozza meg, de az utódok értéke különbözhet a szülő értékeitől: $u_i=a_i x_i + (1-a_i) y_i$ ahol $-h<=a_i<=1+h$... h=0.5 szokott lenni.

Ha több szülő van, minden változóra választunk kettőt, véletlenszerűen.

Lineáris rekombináció: $u_i=a x_i + (1-a) y_i$ ahol $-h<=a<=1+h$. Egy egyednél minden változóra ugyanaz az a értéke.

Bináris stringek rekombinációja: Rekombináció vagy keresztezés (*crossover*). String hossza L.

Egypontos keresztezés. Választunk egy véletlen keresztezési pontot. Két utódot kapunk.



Többpontos keresztezés. Véletlen keresztezési pontotak növekvő sorrendbe rendezzük és rendre alkalmazzuk.

Uniform keresztezés: ugyanaz, mint a diszkrét rekombináció. Azaz minden bit kiválasztható egyik vagy másik szülőtől.

Keverő keresztezés (shuffle crossover): mindkét szülőben azonos módon összekeverjük (permutáljuk) a biteket. Ezek után Egypontos keresztezés jön, majd a biteket visszarendezzük az utódokban (inverz permutáció).

Permutációk rekombinációja

Kétféle megvalósítás:

1. Szülők permutációiból minél több, az adott pozíción található érték átörökítése
2. Szülők permutációiból minél több szomszédos érték átörökítése

Uniform sorrend alapú rekombináció:

Két szülőből két utód lesz, a szülők relatív sorrendjét örökíti át. Előbb egy véletlen bitmaszkot készít, amely minden pozíción lehet 0 vagy 1.

12345 43521	10010 01101	10040 03501	13542 23541
Szülők	Bitmaszk	Köztes	végső

Bitmaszk megmondja, hogy melyik értékeket örököljük direktben a szülőtől. Utána a maradék értékeket kitöltjük úgy, hogy minél több egyezés legyen a másik szülővel. A maradék véletlenszerűen lesz kiosztva. Balról jobbra prioritás van.

Edge rekombináció:

Két szülőből egy utódot képez. Szomszédosságot vizsgál. Készít egy edge táblát: többszörös előfordulást negatív előjel jelzi.

Utód képzés lépései: véletlen elemmel kezdi. Majd választja a negatívat, vagy ha nincs akkor amelyiknek a folytatási listája rövidebb. Ha elakadunk, véletlenszerűen választunk a maradékból és folytatjuk.

12345 43521	1 2 3 4 5	-2, 5, 4 -1, 3, 5 2, -4, 5 -3, 5, 1 4,1,3,2	-2, 5, 4 3, 5 2, -4, 5 -3, 5 4,3,2	5, 4 3, 5 -4, 5 -3, 5 4,3	4 3 -4 -3 4,3	
szülők	Edge		1	12	125	12534

Mutáció

Finom közelítéseket a mutáció valósítja meg. A mutáció az utód közvetlen környezetében keres jobb megoldásokat, pl az x pont ε szomszédossági környezete az S térben. Nhd(x, ε). Egy vagy több változó. Az ε értéke időben csökkenhet. Eleinte segít a feltérképezésben, majd az egyre kisebb érték segít a finomhangolásban.

Valós és egész típusú változók mutációja

Szinte ugyanaz, csak a lépés nagysága különbözik.

Schlierkamp-Voosen és Mühlenbein (1996): $z_i = x_i \pm range_i * \delta$, ahol a + és – esélye 0.5; $range_i$ a szomszédossági környezet szélessége; $\delta = 2^{-k}\alpha$, ahol $0 <= \alpha <= 1$. A k értéke időben növekszik. Paraméter még a mutáció valószínűsége.

Másik verzió $z_j = x_j \pm range_i * \delta * \gamma^{|i-j|}$ ahol $0.1 < \gamma < 0.7$

Bináris típusú változók mutációja. 1/n vagy valami más valószínűséggel invetáljuk a bitet.

Permutációk mutációja

Beszúrás: kiválaszt kettőt, az egyiket elköltözteti a másik mellé, a kettő közötti tólódik eggyel előre

Csere: kiválaszt kettőt és felcseréli őket

Inverz: kiválaszt kettőt és fordított sorrendbe helyezi a közötte levőket

Scramble: kiválaszt egy részhalmazt és véletlen sorrendbe rendezi őket

Hibrid reprezentációnál

Bináris alakba hozzuk, mutáció, majd vissza.

Visszahelyezés (*reinsertion*)

Utódképzési ráta (generation gap): utódok száma hányszorosa a populáció méretének

Visszahelyezési ráta (*reinsertion rate*): populáció egyedeinek hányadrészét cseréljük le utódokkal

Ha UR=VR=1, minden egyed lecserélődik, minden egyed csak egy generációt él, így nem garantált a legjobb korábbi egyed megőrzése.

Ha VR=1, akkor minden egyed csak egy generációt él, így nem garantált a legjobb korábbi egyed megőrzése.

Ha UR<=VR, akkor minden utódot visszahelyezünk. Ha VR<1, akkor ki kell választani, hogy melyik egyedet cseréljük le. Általában **elit szelekció**t szoktak alkalmazni.

Ha UR>VR, akkor az utódok közül is válogatunk.

Steady-state módszerek: csak egy-két utódot képezünk, UR és VR nagyon kicsi, visszahelyezésnél véletlen egyeddel, elit szelekciós egyeddel, vagy a szülővel szokták az utódot összehasonlítani.

Az EA ciklus kialakítása

EA algoritmus nagyon sokféle létezik, mégis a ciklus hasonló. Általános jellemzőit rendszerezni lehet.

Fontos paraméterek: rekombináció valószínűsége p_r, mutáció valószínűsége: p_m.

Néhány plusz tevékenység: (1) stratégiai paraméterek megadása; (2) kezdeti populáció létrehozása; (3) megállási feltétel megnevezése. Fitneszfüggvény értékek kiszámolása is egy szükséges művelet.

Általában ezek a stratégiai paraméterek:

- Populáció mérete azaz az egyedek száma
- A rekombináció alkalmazásának p_r valószínűsége
- A mutáció alkalmazásának p_m valószínűsége
- Utódképzési ráta UR értéke
- Visszahelyezési ráta VR értéke

Egyes műveleteknek vannak még másodlagos paraméterei is, láthattuk.

Kezdő populáció kialakítása

- Véletlenszerű, amikor semmit nem tudunk előre
- Korábbi eredmények módosított felhasználása: korábbi eredmény vagy szakértői vélemény; egyenletes vagy normális eloszlás alapján generált
- Többszöri újraindítás (restart): viszonylag jó megoldások megőrzése, további véletlen egyedekkel

Megállási feltétel (kombinálható)

- Maximális generációszám (ciklus)
- Maximális futási idő
- Adott idő alatt nem javul a megoldás minősége
- Hasonlók az egyedek
- Előre adott érték megközelítése
- A populáció minősége megfelelő, ennek mérőszámai
    - Célfüggvény értékének standard szórása az aktuális generációban
    - Átlag és legjobb érték eltérése
    - Legjobb és legrosszabb eltérése

Eddig az EA-ról beszéltünk általánosan. Mint korábban láthattuk, az EA négy alosztályból áll. Ezek közötti különbségre fogunk a továbbiakban koncentrálni.

- Darwin (1859) – A fajok eredete: tulajdonságok öröklése, szelekció, stb. Darwin-díj.
- Watson doktor DNS, ivaros és ivartalan szaporodás. Az ivartalan nem rekombinál csak mutál.
- EA: absztrakt módon utánozza a biológiai folyamatokat, leegyszerűsítve
- Genotípus – fenotípus absztrakció
- „A genotípus a génekben tárolt genetikai információk összessége, amely a környezeti viszonyokkal kölcsönhatásban meghatározza a szervezet külső megjelenését, a fenotípusát".
- „Ugyanaz a genotípus különböző környezetben különböző fenotípus formájában realizálódhat" (Staub 1987)
- Gének nem izoláltan, hanem egymással való kölcsönhatásban fejtik ki hatásukat. Nem létezik gén -> fenotípus, vagy fenotípus -> gén megfeleltetés.
- Minden EA módszer el kell döntse, hogy genotípussal dolgozik, vagy fenotípussal.
- Genotípus: a kromoszómát bitsorozatban tároljuk, a megoldásokat melyeket fenotípus formájában szeretnénk látni, dekódolnunk kell.
    - Egyrészt csak közelítő értéket kapunk
    - Könnyen alkalmazható univerzális módszer
- Három univerzális módszer + a polip
- Genetikus algoritmus (GA):
    - génsorozatnak tekinti az egyedeket
    - ivaros szaporodás öröklési mechanizmusát alkalmazza
    - génmutáció ritka
    - sztochasztikusan választja a szülőket, utódok a szülők helyére lépnek (vagy nem)
- Evolúciós stratégia (ES) – tulajdonságok alapján
    - Fenotípus szerinti reprezentáció, [0,1] valós számok
    - Rekombináció kisebb súlyú, mint a GA esetében. A szülők domináns tulajdonságainak átadása.
    - Fő művelet a mutáció: normál eloszlású véletlen számokkal módosít minden tulajdonságot.
    - Visszahelyezés: determinisztikus, mindig a legjobb egyedek maradnak.
- Evolúciós programozás (Nissen 1997) – viselkedési formák megtalálása
    - Viselkedés szintjén a rekombináció nem értelmezhető
    - Minden egyedet mutál
    - Sztochasztikusan választ új generációt, versenyeztetéssel

Ezek is csak módszercsoportok. Van lehetőség kombinálni őket.

Genetikus algoritmus

- Holland '60-as évek, Bagley (1967) adott nevet neki.
- Az egyed egy k hosszúságú bitstring
- Egyed tulajdonságai: (x_1, x_2, ..., x_n), mindegyiknek egy-egy rész felel meg a bitstringben.
- GA: egy bit egy gén, annak az értéke egy allél

- Kétféle kódolás: standard bináris vagy Gray kódolás
- Bináris: oda-vissza kódolás egyszerű
- Gray kód: $g_1 = a_1$ és $g_z=a_{z-1}$ XOR a_z vissza pedig a_z=XOR g_k, k=1...z

0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000

- Fitnesz számításhoz egyenként dekódoljuk, kiszámoljuk a fitnesz értéket, majd azt transzformáljuk, leggyakrabban így: $ø(E_i) = a\ f(E_i) + b$. Minden így kapott érték legyen pozitív és a legjobb értékek legyenek a legnagyobbak.
- Szelekció: rulett-szelekció $ø(E_i)$-vel arányos valószínűségek szerint. Egyesével választunk μ db szülőt
- Rekombináció: ez a fő művelet, GA-ban keresztezésnek nevezik.
    - Rekombináció valószínűsége: $p_r = 0.9$ (default)
    - Szülők párosítva: μ /2 szülőpár, egy- vagy többpontos keresztezés
- Mutáció egy háttérművelet: mutáció az egyed minden egyes bitjénél $p_m = 0.01$ valószínűséggel
- Visszahelyezés: UR=VR=1, minden korábbi eredmény elvész.
- GA ciklus
    - P(0) populációban az egyedek száma μ, tipikus értéke 30 és 500 között
    - Minden bit 0.5 valószínűséggel nullás vagy egyes
    - Leállási feltétel: maximum generációszám vagy maximum futási idő szokott lenni
    - Stratégiai paraméterek: μ, bitszting hossza L,
    - Változók száma n, értéktartományaik $[b_i,c_i]$
    - Rekombináció és mutáció valószínűsége
- Folytonos GA – fenotípus, ES, EP? WTF?
- Steady-state GA
    - Csak 2 szülő kerül szelektálódik, lineáris sorrend alapján
    - Duplikáció ellenőrzése utódoknál
    - Leggyengébb egyedek helyére kerülnek be az utódok