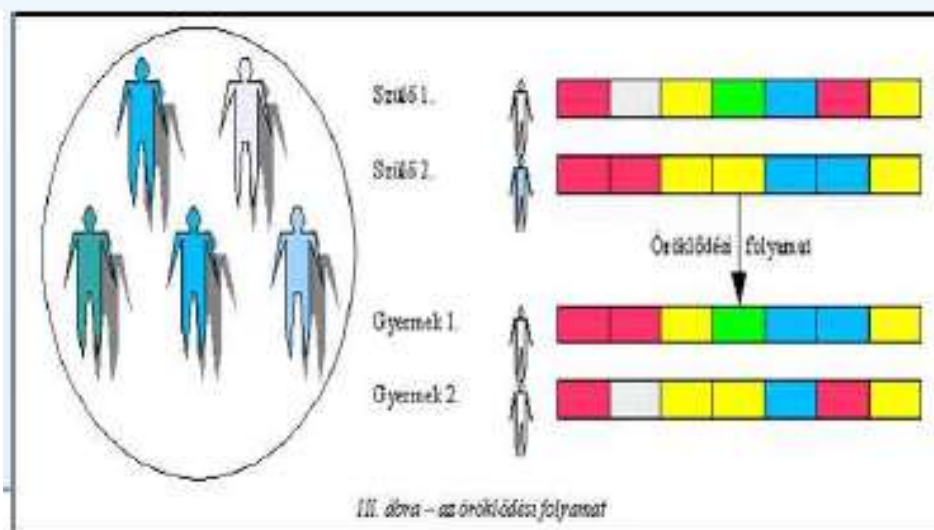


Intelligens megoldások a
döntéshozásban
Genetikus algoritmusok

Bevezetés

- 1950-60-as évek biológiai evolúció - mérnöki problémák - optimalizálási feladatok
- Darwini evolúciós elmélet
- Genetika
- Evolúciós módszerek
- Genetikus algoritmusok - John Holland (1975)
- Többpontos, párhuzamos keresés - robosztusság



Optimalizációs feladatok

- az ellenállás meghatározása a mért áramerősség és feszültség segítségével
- a sebesség számítása a mért időből és a megtett távolságból
- a napi beosztások megtervezése
- két város között az optimális út megkeresése
- adott erőforrások mellett az eredmény maximalizálása
- adott gazdasági cél mellett a ráfordítás minimalizálása
- menükészítés

Optimalizálási feladatok

Az optimalizálási feladatok során egy adott halmazon (keresési tér, S) definiált függvény (fitness függvény, f) maximumhelyét (vagy minimumhelyét) keressük.

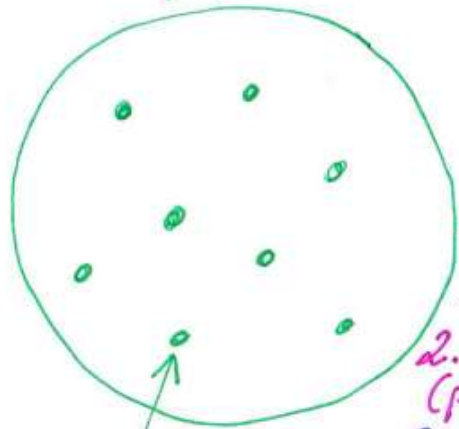
Vannak hagyományos módszerek:

- *hegymászó módszer* (gradiens módszer) véletlen pontot választunk a keresési térben,
- *szimulált lágyítás* (szimulált lehűtés) véletlenszerűen választjuk meg a lépés irányát a keresési térben

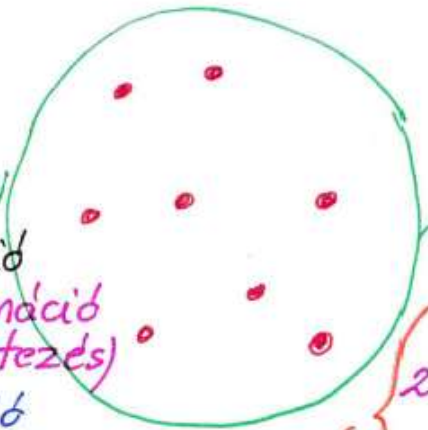
Evolúciós algoritmusok:

- evolúciós stratégia
- evolúciós programozás
- genetikus algoritmusok
- genetikus programozás

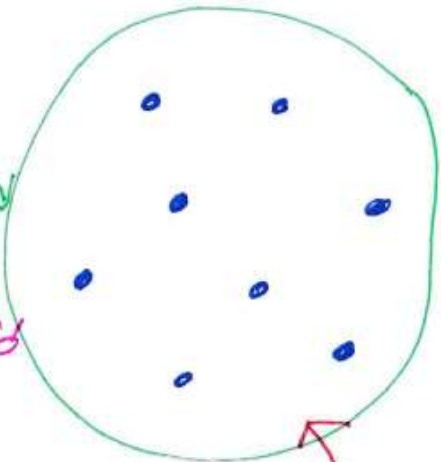
0. generáció
populáció



1. generáció



n. generáció



1. szelekció
2. rekombináció
(pl. keresztezés)
3. mutáció

1. szelekció
2. rekombináció
3. mutáció

egyed = a feladat egy
lehetőleges meg-
oldása

∀ egyednek van rátermelt-
sége, fitness értéke

az egyedeken műveleteket
hajtanak végre

közel optima-
lis megoldások
halmaza

Általános evolúciós algoritmus pszeudó-kódja

- $t := 0$ {kezdeti idő beállítása}
- $initpopulacio P_t$ {kezdeti populáció létrehozása}
- $fitnessszamit P_t$ {fitnessértékek kiszámítása}
- **while** amíg nincs kész **do**
- $P'_t := szulokivalasztas P_t$ {szülők választása}
- $keresztez P'_t$ {a szülők génjeinek keresztezése}
- $mutacio P'_t$ {véletlen mutáció}
- $fitnessszamit P'_t$ {az új fitnessz kiszámítása}
- $P_{t+1} := tulelo(P_t, P'_t)$ {az új populációba kerülnek az egyedek}
- $t := t + 1$
- **end while**

Genetikus algoritmusok

- 1975 John Holland
- a megoldásokat nem az eredeti feladatnak megfelelő formában tárolja - *kromoszóma*
- a műveleteket a kromoszómákon hajtjuk végre
- szelekció
- rekombináció
- mutáció
- egyedek fitnessértéke

GA jellemzői

- több pontos keresést valósítanak meg
- flexibilisek
- robosztusak
- biztosítják, hogy elfogadható időn belül elfogadhatóan jó megoldást találjunk
- a problémának nem egy, hanem több különböző, közel optimális megoldását nyújthatja, amelyek közül a felhasználó kiválaszthatja a neki leginkább megfelelőt

Az egyedek ábrázolási formája

- **Bináris vektor**

Genotípus formát jelent

Jelölje az (x_1, x_2, \dots, x_n) valós (egész) vektor az egyed tulajdonságait.

Bináris ábrázolásban az egyed egy sztringként jelenik meg:

$x_1, x_2, \dots, x_i, \dots, x_n$

...00|11010111|01...

az x_i változó kódolt értékei

Szelekció – rulett szelekció

- Fitnessz arányos szelekció, amely az egyedeket fitnessz értékük abszolút értékének arányában választja ki a szelekciós állományból.
- Visszatevéses művelet
- Egy egyed kiválasztását a szelekciós valószínűség határozza meg:

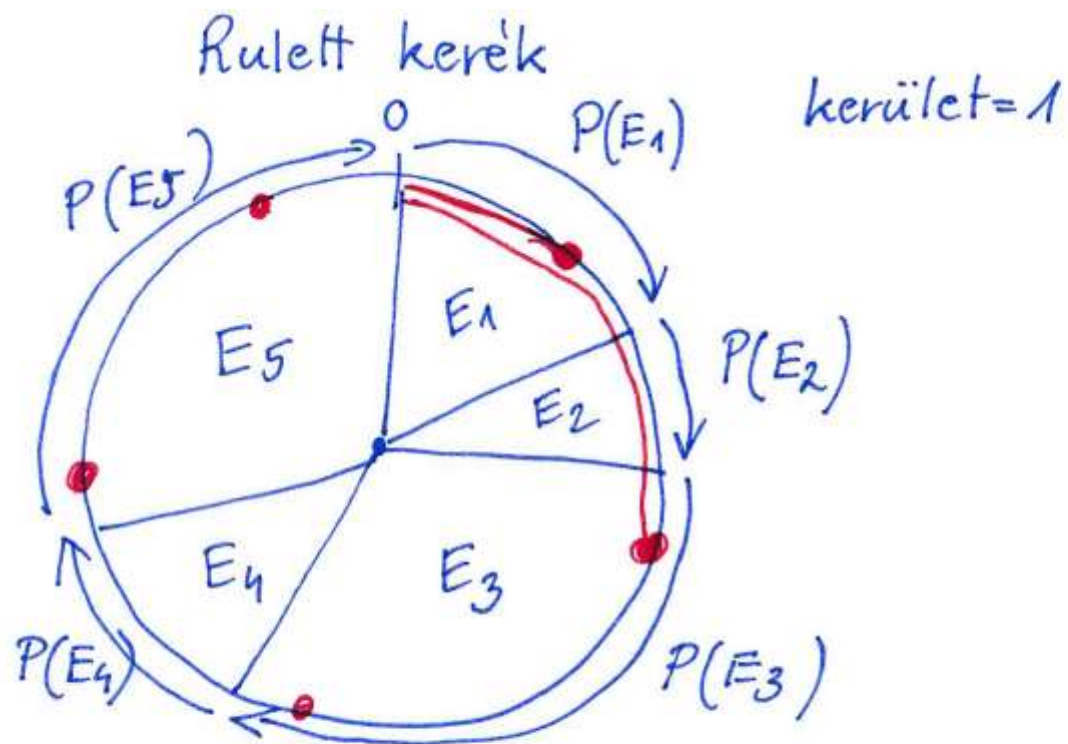
$$P(E_i) = \frac{f(E_i)}{\sum_{j=1}^n f(E_j)}$$

f a fitnessz függvény, E_i ($i = 1, \dots, n$) az egyedek

Rulett szelekció

- veszünk egy rulettet
- feleltessünk meg minden E_i egyednek valamely kiindulási pontból folyamatosan egy-egy körszeletet
- generálunk egy $[0, 1]$ -beli véletlen számot, a véletlen számot ívhossznak tekintjük
- azt az egyedet választjuk, amelynek körszeletében az ív végződik
- egy μ elemű szelekciós halmazból a választást μ -ször kell megismételni, amíg kialakul a szülők állománya

A kiválasztott egyedek közt $\mu * p(E_i)$ ($i = 1, \dots, n$) várható számú másolata lesz az E_i egyednek.



A körszelet \wedge hossza feleljen meg
 a szektor
 az egyes szelekciók valószínűségének.

Versengő szelekció

- Az egyedek fitness értékeinek sorrendjét használja fel.
- Nem fog növekedni az egyed duplikációk száma.
- Több egyedből a legjobb fitness értékű egyedet választja ki. (Biológiai szelekciót modellezi.)

Lépések:

1. **Input:** A szelekciós állomány E_i elemei és $f(E_i)$ fitness értékei ($i = 1, \dots, n$), $tour$ paraméter
2. **Output:** A populáció a szelekció után (szülők állománya): E'_i ($i = 1, \dots, n$)
3. **for** $i = 1$ **to** μ **do**
4. **for** $k = 1$ **to** $tour$ **do**
5. válasszunk egy $j \in \{1, \dots, n\}$ indexet véletlenszerűen
6. $T_k = E_j$
7. **od**
8. $E'_i = T_j$ ha $f(T_j) = \max(f(T_1), \dots, f(T_{tour}))$
9. **od**

A kiválasztott egyedek közt $\mu * p(E_i)$ ($i = 1, \dots, n$) várható számú másolata lesz az E_i egyednek.

Versengő szelekció (valds életből származik)

tour = 3

E_1 E_2 E_3 E_4 E_5 E_6
értéke: 2 7 4 8 6 9
fitness: 0,2 0,4 0,8 0,5 0,7 0,1

$i = 1$

$k=1$	$j=3$	$T_1 = E_3 = 4$	$\boxed{0,8}$	} $\rightarrow \max$
$k=2$	$j=5$	$T_2 = E_5 = 6$	0,7	
$k=3$	$j=1$	$T_3 = E_1 = 2$	0,2	

$E_1' = T_1 = E_3 = 4$

\uparrow
véletlenszerűen választjuk

$i = 2$

$k=1$	$j=2$	$T_1 = E_2 = 7$	0,4	} $\rightarrow \max$
$k=2$	$j=5$	$T_2 = E_5 = 6$	0,7	
$k=3$	$j=3$	$T_3 = E_3 = 4$	$\boxed{0,8}$	

$E_2' = T_3 = E_3 = 4$

•
•
•

$i = 6$ -ig

Rekombináció

- szelekció után keletkező szülő állomány
 - tartalmazhat ismétlődéseket
 - részben vagy egészben azonos lehet a populációval
- kettő-több szülő felhasználásával képez utódot (jellemzően kettőből egyet vagy kettőt)
- célja: a szülőkből minél jobb, újabb megoldások összeállítása az átvett, "örökölt" tulajdonságok alapján
- formáját befolyásolja a változók típusa és a problémák sajátosságai
- $P_r \approx 0,7$

Bináris sztringek rekombinációja

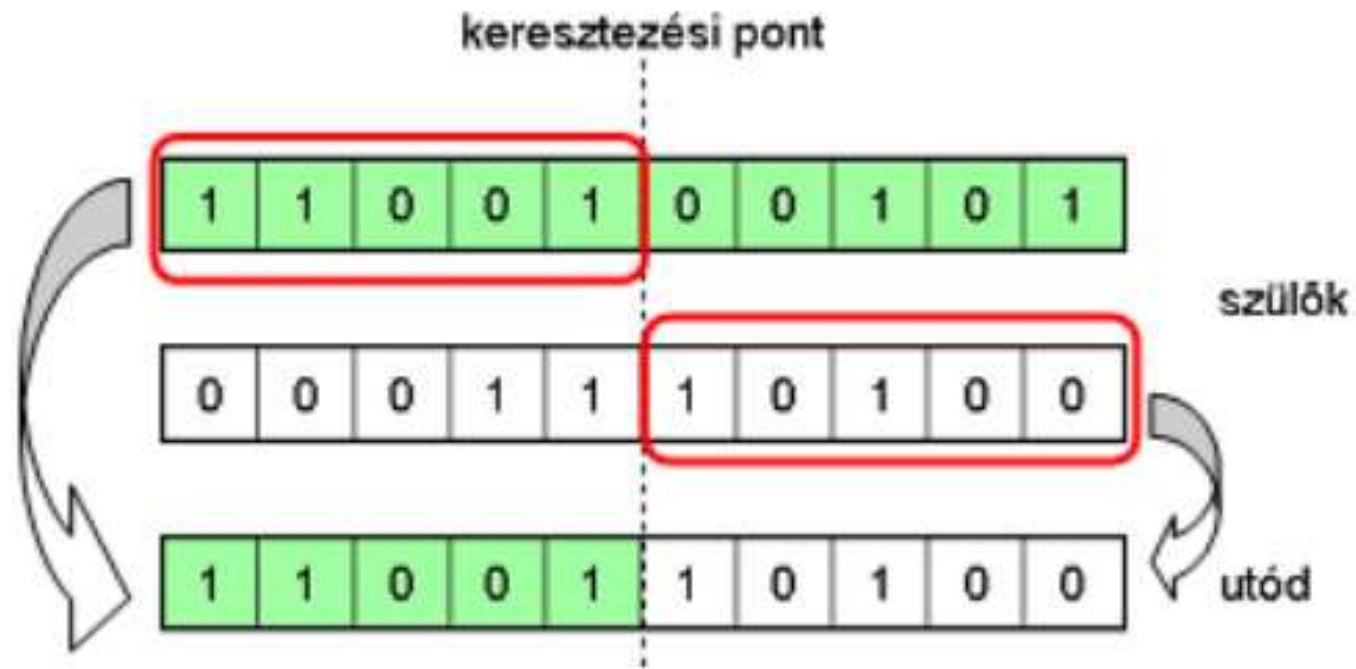
- **Egypontos keresztezés**

- két szülőből két utód
- véletlenszerűen választunk keresztezési pontot az $\{1, 2, \dots, L - 1\}$ pozíciók közül

- **Többpontos keresztezés**

- két szülőből két utód
- n számú keresztezési pontot választunk
- a kapott keresztezési pontokat növekvő sorrendbe rendezzük, majd a megfelelő, egymás után következő keresztezési pontok közti bitsorozatokot rendre más-más szülőtől választjuk

Egypontos keresztezés



Mutáció

A rekombináció nem alkalmas finom közelítések megvalósítására.

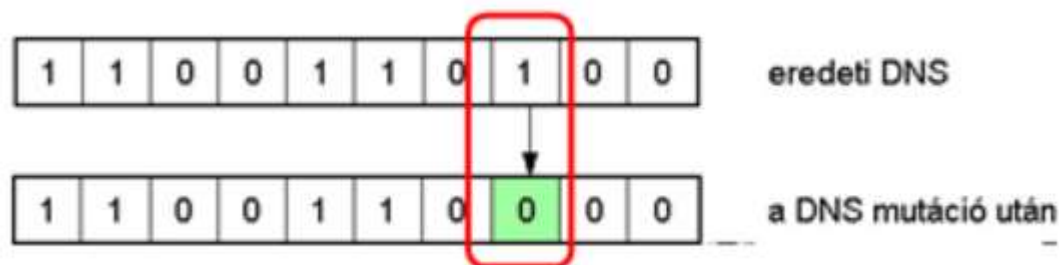
A mutáció az utód közvetlen környezetében keres jobb megoldásokat.

$$P_m \approx 0,01$$

Bináris típusú változók mutációja az egyed egy bitsorozat, melynek egyes bitjeit mutációval változtatjuk

a mutáció egy x_i változónál a következő:

$$z_i = \begin{cases} x_i & , ha \ Rnd > P_m \\ 1 - x_i & , ha \ Rnd \leq P_m \end{cases}$$



GA előkészítése

- **stratégiai paraméterek megadása**
 - populáció mérete
 - a rekombináció alkalmazásának P_r valószínűsége
 - a mutáció alkalmazásának P_m valószínűsége
 - az utódképzési ráta értéke
 - a visszahelyezési ráta értéke
- **kezdő populáció kialakítása**
 - véletlenszerű előállítás
 - előző feladat eredményeinek felhasználása
 - korábbi eredmények módosított felhasználása

GA előkészítése

megállási feltétel

- maximális generációszám elérése
- maximális futási idő elérése
- adott idő alatt nem javul a megoldás minősége
- hasonlóak az egyedek
- előre adott érték megközelítése

fitness kiértékelés

- A problémák legtöbbszörénél ismerünk egy **célfüggvényt**, amely az értelmezési tartomány, azaz a keresési tér pontjaihoz egy valós számot vagy vektort rendel.
- Az esetek többségében a fitnessfüggvényt azonosnak választjuk a célfüggvénnyel.
- Sokszor nincs célfüggvényünk és a fitnessfüggvény megfogalmazása a probléma megoldásának egyik fontos kulcseleme.
- Konkrét formája függ a reprezentációtól.

Alkalmazási területek a döntéstámogatáshoz kapcsolódóan

Automata sebességváltó

- A feladat az, hogy optimális időpontokban kapcsoljunk egy-egy fokozattal előre gyorsításkor.
- Azaz úgy váltsunk sebességet, hogy pl. 100 km/óra-ra a lehető legrövidebb idő alatt gyorsuljunk fel.
- Az optimalizálandó függvény formája igen bonyolult – hiszen az adott fokozatok nyomatékgörbéi erősen függenek attól, hogy mekkora sebességnél történt a kapcsolás.

Utazó ügynök probléma

- Pl. 50 várost kell a lehető legrövidebb útvonalon bejárni.
- Ha a biztos optimimális megoldást keressük, akkor $49!$ lehetőséget kellene végig számolni.
- GA-val közel optimális megoldást kaphatunk.

Kiterjesztett változat

- Egy olajtársaság több telephelyről hogyan juttattja el az üzemanyagot a kutakhoz.
- Nem egy „ügynök” van, hanem több, mivel több olaj szállító kamion áll rendelkezésre.
- A feladat az, hogy a telephelyekről a lehető legalacsonyabb költséggel szállítsák az üzemanyagot a kutakhoz.

Mobiltelefon-társaság

- A kommunikációhoz szükséges rádió adó-vevő központokat milyen sűrűn és hová kell elhelyezni.
- Ha egy körzet nincs lefedve, akkor onnan nem lehet telefonálni, ha pedig nincs elég sűrűn lefedve – megfelelő kapacitással -, akkor csúcsidőben nem lehet telefonálni (túlterhelés).

Szimulációs modellek

- Pl. milyen legyen az egyes nyersanyagok kitermelésének a mértéke annak érdekében, hogy meghatározott idő alatt a GDP legalább 1 százalékponttal növekedjen és a környezet minősége ne romoljon.
- Szimulációk optimalizálása tipikusan a genetikus algoritmus számára kitalált feladat.

Előrejelzési modell

- A feladat az, hogy a bank által üzemeltetett ATM bankjegy automatákban optimális szinten legyen az adott bankjegyek mennyisége.
- Az optimális azt jelenti, hogy ne legyen nagyon sok, mert ennek napi kamata költségként jelentkezik a banknál.
- De ne is legyen nagyon kevés, mert a kifogyó automata erősen rontja az adott bank imázsát.
- Továbbá, ha túl gyakran kell feltölteni az automatát, akkor a szállítási költség lesz nagyon magas.
- Az optimalitás itt több szempontot érinthet, ugyanis nem biztos, hogy a legalacsonyabb üzemeltetést kell választani akkor, ha a véletlen hatások miatt ez nagyon kockázatos (pl. gyakori kifogyás).