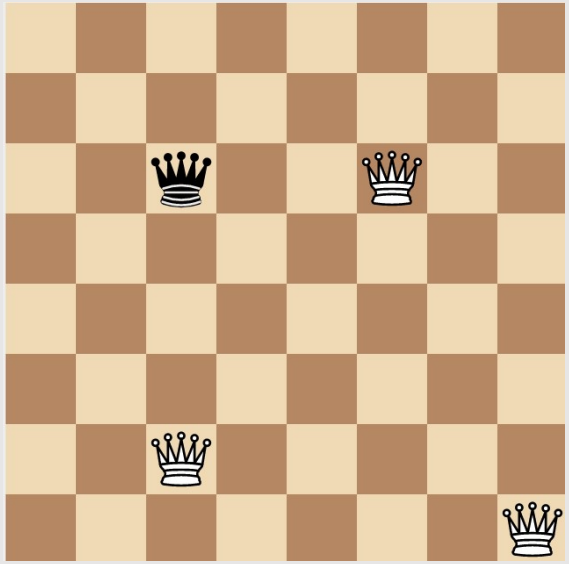
Genetikus algoritmusok – alkalmazási példák

A 8 királynő probléma megoldása

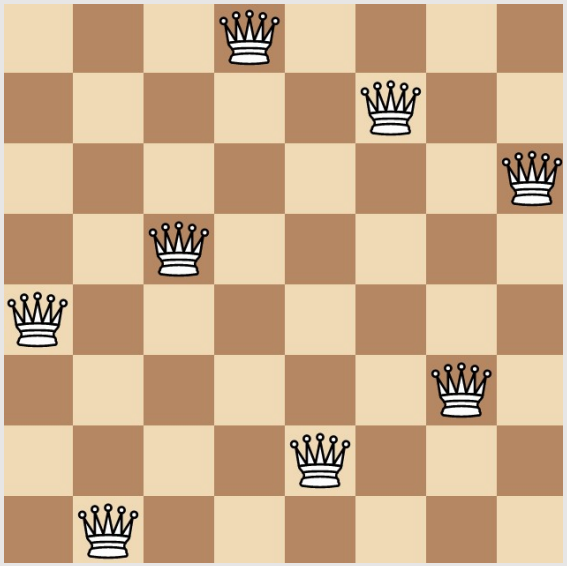
Probléma leírása

- A feladat 8 db királynő (vezér) bábú elhelyezése a sakktáblán úgy, hogy azok ne üssék egymást
- Két vezér akkor üti egymást, ha azonos sorban, oszlopban vagy átlóban helyezednek el
- Például az alábbi sakktáblán mindhárom fehér királynő ütésben van a fekete királynővel



Probléma leírása

- A problémát felírva minimalizációs feladattá, célunk, hogy minél kevesebb egymást ütő királynő-pár legyen a táblán
- Példa egy optimális megoldásra:

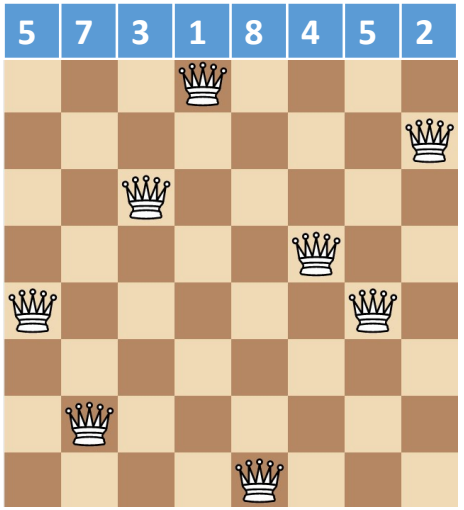


Probléma leírása

- Amennyiben a királynők elhelyezésének összes lehetséges kombinációját vesszük, $\binom{64}{8} \approx 4{,}42 \times 10^9$ esetet kell számba vennünk, melyek közül mindössze 92 elrendezés felel meg
- Azonban csökkenthetjük a kiindulási halmazt, mivel tudjuk, hogy egy adott sorban illetve oszlopban csak 1-1 királynő helyezkedhet el

Egyedek kódolása

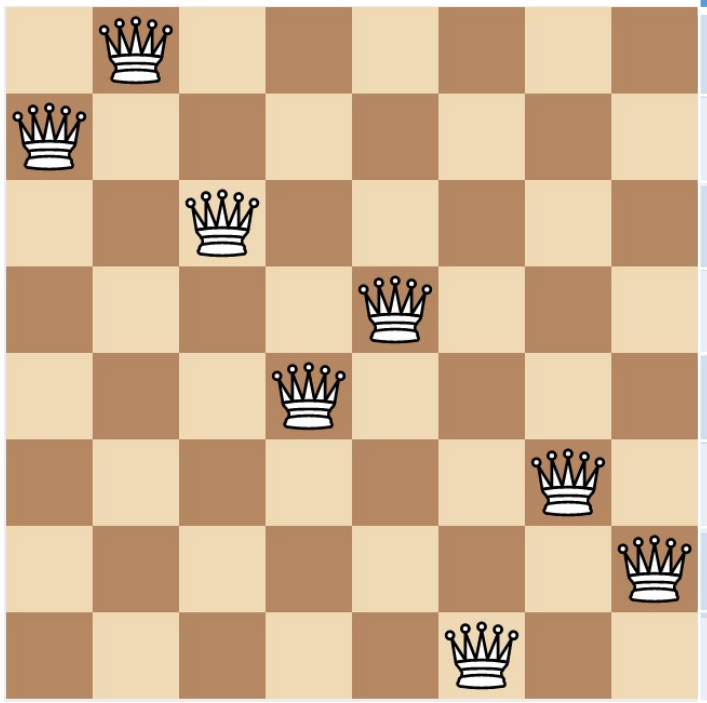
- Egy lehetséges elrendezés = egyed
- Kódolás: minden szám azt jelzi, hogy a hozzá tartozó oszlop hányadik sorában található bábú
- Mivel minden sorban csak egy bábú lehet, a kódban szereplő számok nem ismétlődhetnek → az egyedek permutációként ábrázolhatók
- Így a lehetséges elrendezések száma 8!=40320 esetre csökken



Egyedek fitneszértéke

- 8 db királynő → legfeljebb 7 másikkal lehet ütésben
- Egy tábla fitneszértéke:
  - Mivel szeretnénk maximalizációs problémát felírni, a lehetséges 8×7=56–os értéket választjuk a maximális fitnesznek
  - Minden bábúra megnézzük, hogy hány másik bábút üt, és ezt az értéket levonjuk a kiinduló 56-os értékből

Egyedek fitneszértéke



Kiindulási érték: 56
-4
-2
-0
-4
-2
-3
-3
-0
=38

Egyedek generálása

- 0. generáció elkészítése

3	6	7	4	1	8	2	5
1	5	7	3	9	1	8	4
5	6	4	2	3	8	1	7
3	8	5	6	2	7	1	4

N db (A populáció méretét a felhasználó határozza meg)

- Véletlenszerű egyedgenerálás
- Példa egy 8 elemű populációra:

```
[6 2 5 4 3 8 1 7]
[2 1 4 3 5 6 7 8]
[2 5 3 6 8 1 4 7]
[1 5 3 4 6 7 2 8]
[2 1 8 5 3 4 6 7]
[2 8 1 6 3 7 5 4]
[3 6 1 4 7 5 8 2]
[6 5 4 3 7 8 2 1]
```

Genetikus algoritmus

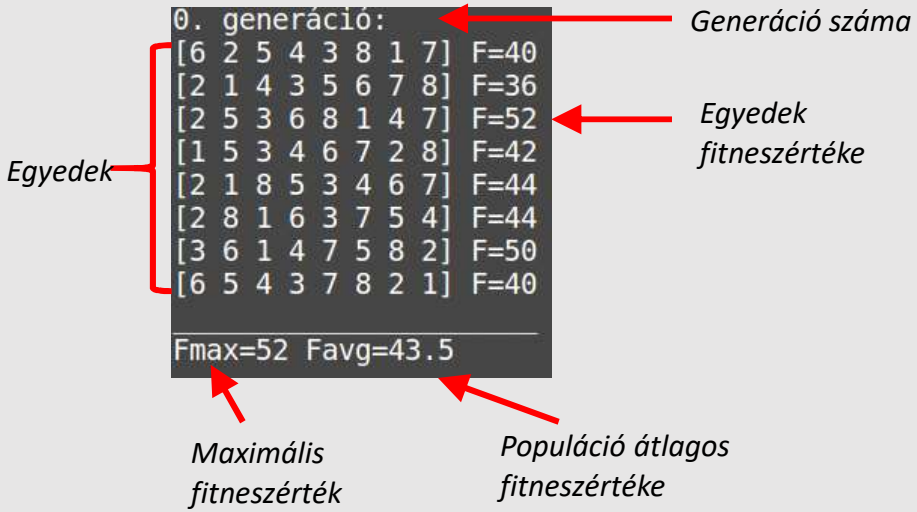
- Ciklus, melynek kilépési feltételei:
  - érje el a legújabb generáció egy eleme a kívánt 56-os fitnesz-értéket, vagy
  - leállás túl sok generáció elkészítése után, ha nincs megfelelő egyed
- Lemásolja az előző generációt, majd a másolaton futtatja a *genetikus algoritmus* függvényt, a felhasználó által megadott P_r rekombinációs és P_m mutációs valószínűséggel

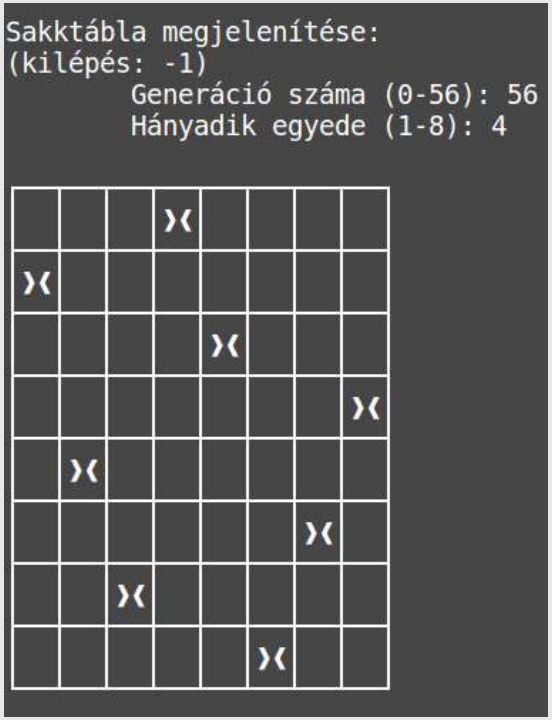
Genetikus algoritmus

1. lépés: versengő szelekció
   - 8 egyed esetén 3-at választ ki véletlenszerűen, és ezek közül a legjobb fitneszű kerül kiválasztásra → 8-szor ismételve
2. lépés: uniform sorrend alapú rekombináció
   - P_r valószínűséget figyelembe véve
   - A populáció két véletlenszerűen kiválasztott egyedén végrehajtva
3. lépés: csere típusú mutáció
   - Minden egyed P_m valószínűséggel kerül módosításra
   - Két véletlenszerű indexét kiválasztjuk, ezek kerülnek felcserélésre
4. lépés: fitnesz-értékek kiszámítása
   - Amennyiben sikerült elérni a maximális értéket, az algoritmus befejezi a működést

Megjelenítés

- Miután az algoritmus végzett, lehetőség lehet bármely generáció bármelyik egyedének megtekintésére, egyszerű táblázatos formában:





Genetikus algoritmusok

Órarend készítési probléma

Sok feltételt kell figyelembe venni:

- Minden tanárnak *x* számú órája van bizonyos osztályokban.
- Nincs olyan osztály és tanár, akinek egyszerre két órája van.
- Minden tanárnak egy héten lehetőleg legyen egy szabad napja.
- Az osztályoknak csak a nap elején vagy végén lehet lyukas órájuk.
- Ha egy tanárnak több órája van egy nap egy osztállyal, akkor lehetőleg azok egymás után legyenek.
- Egy tanárnak lehetőleg ne legyen sok lyukas órája.
- stb.

Halmazos reprezentáció

A legkisebb **adategység a halmaz**, egy olyan struktúra, amely tetszőleges számú osztályt, tanárt és tantermet tartalmaz.
Pl. két osztályhoz egy tanárt rendelünk (például összevont testnevelés óra esetén) a két osztályt és az egy tanárt felvesszük a halmazba.
Csak **egy dimenzió**ban dolgozunk, amely nem más mint az idő.
Az időtengelyen lévő időrésekbe, melyek a lehetséges órákat jelentik-kell halmazainkat elhelyezni.
Egy időrésbe több halmaz is kerülhet, itt ügyelni kell arra, hogy ne legyen ütközés, ne kerüljön egy időrésbe két olyan halmaz, melyben közös tanár vagy közös osztály szerepel.

Halmazos reprezentáció

Az $o_{i,j}$, $t_{i,j}$, $te_{i,j}$ értékekkel az i-edik halmazban szereplő osztályok, tanárok, termek sorszámát jelöljük, j pedig a halmazon belüli sorszámok indexe.

1. nap			2. nap			…
1. óra	2. óra	…	…	…	…	…
Halmaz_1: $osztály_{0_{1,1}}, osztály_{0_{1,2}}, \ldots$ $\text{tan}\acute{a}r_{t_{1,1}}, \text{tan}\acute{a}r_{t_{1,2}}, \ldots$ $terem_{te_{1,1}}, terem_{te_{1,2}}, \ldots$	Halmaz_{N+1}: $osztály_{0_{N+1,1}}, osztály_{0_{N+1,2}}, \ldots$ $\text{tan}\acute{a}r_{t_{N+1,1}}, \text{tan}\acute{a}r_{t_{N+1,2}}, \ldots$ $terem_{te_{N+1,1}}, terem_{te_{N+1,2}}, \ldots$					
Halmaz_2	Halmaz_{N+2}					
…	…	…				
Halmaz_N	Halmaz_{2N}					

Halmazos reprezentáció

Függőleges és vízszintes linearizálás

1. időrés	2. időrés	...	k. időrés
Halmaz 1	Halmaz N+1		Halmaz (k-1)*N+1
Halmaz 2	Halmaz N+2		Halmaz (k-1*N+2
...	...		
Halmaz N	Halmaz 2N		Halmaz k*N

Függőleges:

Halmaz 1	Halmaz 2	...	Halmaz N	Halmaz N+1	...	Halmaz k*N

Vízszintes:

Halmaz 1	Halmaz N+1	...	Halmaz (k-1)*N+1	Halmaz 2	...	Halmaz k*N

Egy órarend jóságának elemzése

Kemény kötések

- Tanárütközésről akkor beszélünk, amikor egy tanárnak egyidejűleg több osztály számára kéne órát tartania.
- Teremhiány jelentkezhet akkor, ha egy osztály számára nem jut tanterem, ahol az órát megtudnák tartani. (korlátozott erőforrás)
- Tanárok kemény ráérése. A tanároknak iskolán kívüli feladatai is lehetnek, melyeket kötelező ellátni. Így például előfordulhat, hogy egy tanárnak nem lehet órája a pénteki napon.

Lágy kötések

- Lyukasóra. Általános iskolában nem megengedhető, középiskolában is nehezen tolerálható, ha valamelyik osztály órarendjében lyukasóra van.
- Nulladik óra. A tanulók számára megterhelő a nulladik vagy esetleg a 7. 8. óra.
- Többszörös óra. Egy osztálynak, ne legyen egy tárgyból egy nap több órája. Például heti két fizika óra esetén ne essen ez a két óra egy napra.
- Héten belüli egyenletes óraeloszlás. A napi óraszámok között lehetőleg kis eltérés legyen az egyenletes terhelés érdekében.
- Tanárok lágy ráérése. Előfordulhat, hogy egy tanár, más intézményben is tanít. Ilyenkor kérést fogalmaz meg, hogy a hét melyik napján, milyen időben ne osszák be.

Egy lehetséges büntető pontozás

Osztályütközés: 1000
Tanárütközés: 1000
Teremhiány: 1000
Kemény tanárráérés: 500
Többszörös órák: 100
Napközi lyukasóra: 100
Első nap eleji lyukasóra: 50
Napi óraszám egyenetlensége: 50
Heti egyenletesség: 20
Napon belüli szakadozottság: 20
Lágy tanárráérés: 10

A **fitnesz függvény** pl. számolja a büntető pontok összegét.

Cél a fitnesz függvény értékének **minimalizálás**a.

Genetikus algoritmusok

Menügenerálási rendszer

A kidolgozandó intelligens algoritmussal szemben az alábbi követelmények merülhetnek fel

- Legyen tekintettel a páciens energia (kalória) igényére,
- A táplálkozási normák szerinti helyes táplálék összetételére,
- Biztosítsa a változatosságot,
- Tegye lehetővé az interaktív korrekciókat,
- Legyen tekintettel a páciens egyes speciális megkötésére (a nem kedvelt és/vagy problémát okozó komponensek minimalizálására),
- Legyen tekintettel orvosi megkötésekre (pl. diabetes vagy más orvosilag megkülönböztetendő – a kardiovaszkuláris problémához gyakran társuló – betegségek esetén),
- Esetleg a szezonális beszerzési lehetőségekre és a beszerzési költségekre.

Feladatok lehetnek

- Egy webes felületű rendszer a felhasználó által kitöltött kérdések alapján diagnózist állít fel
- A szerver paraméterként megkapja a felhasználó táplálkozással kapcsolatos igényeit:
  - allergiás a tejtermékekre,
  - nem szereti a sültburgonyát,
  - a sertéshúst és az uborkát,
  - továbbá diétás javaslatot szeretne készíttetni.
- Vele egy időben egy orvos konfigurálhatja a weben keresztül a javaslatait. Például egy célcsoport számára kér javaslatot, melyben a szénhidrát tartalmát alulról és felülről, a zsírtartalom értékét csak felülről korlátozza.
- A szerver ezeket az adatokat feldolgozza és a diagnózis, illetve a korlátok segítségével, a dietetikus szakértők által feltöltött adatbázisból pontos korlátokat ad a tápanyag-összetevőkre, és meghatározza azoknak az optimális mennyiségét.

Mire van szükségünk

- a megoldást meghatározó paraméterek halmaza,
- a paraméterek lehetséges értékeinek a halmaza,
- a megoldás jóságát a paraméter-értékek által hordozott információ, és a paraméter-értékek egymással való összefüggése határozza meg.

  (az általános és a felhasználó specifikus értékek tükrében ítéljük meg)

- Pl. ebéd esetén: leves, feltét, köret, kiegészítő és ivólé attribútumok részhalmaza.

  E véges számú attribútumnak kell értéket adnunk. Ezen értékek tápanyag-tanácsadás esetén lehetnek a receptkönyveinkben megtalálható receptjeink.

A probléma megoldásához különböző típusú (reggeli, tízórai, ebéd, uzsonna, vacsora) étkezéseket kell generálnunk

- Ebédek generálása esetén S a lehetséges megoldások, azaz az elképzelhető ebédek halmaza,
- A az attribútumok halmaza (köret, feltét, leves, ivólé, kiegészítő), tehát $n_A=5$.
- Jelölje a_1, a_2, a_3, a_4, a_5 rendre a köret, feltét, leves, ivólé, kiegészítő attribútumokat.
- Ha felépítünk egy adattörzset, akkor pl. $a_1=150$, $a_2=210$, $a_3=123$, $a_4=10$, $a_5=254$.
- A fent definiált osztályba tartozó problémák megoldásainak száma:

$$|S| = \prod_{i=1}^{n_A} a_i$$

- Az ebéd generálás probléma paramétereit behelyettesítve a képletbe azt kapjuk, hogy 9,84 milliárd megoldása lehetséges a feladatnak.
- A számítógépek jelenlegi kapacitása mellett ennyi lehetőség vizsgálata racionális idő alatt elvégezhető, de ha interaktív webes tanácsadás a célunk, akkor minél gyorsabb generálást kell elérnünk.

Javaslatok a genetikus algoritmus használatához

- A kromoszómában (egyedben) az információt direkt érték kódolással tároljuk.
- Étkezéseket generáló genetikus algoritmusnál a gyakorlatban ez azt jelenti, hogy az egyes gének sztring típusú változók, melyek egyértelműen meghatározzák az adatbázisban tárolt receptet (étel megnevezését).
- Ebéd esetén a kromoszóma 5 sztringből áll, melyek mindegyike egy ételt (receptet) reprezentál.
- Pl.
  - Gyümölcsleves
  - Párolt rizs
  - Párizsi szelet
  - Káposzta saláta
  - Rostos almalé

Rekombináció - keresztezés

A direkt érték kódoláshoz tartozó keresztezési operátor működése:

Gyümölcsleves	Húsleves	Gyümölcsleves	Húsleves
Párolt rízs	Nokedli	Párolt rizs	Nokedli
Párizsi szelet	Sertéspörkölt	Párizsi szelet	Sertéspörkölt
Káposzta saláta	Savanyú uborka	**Savanyú uborka**	Káposzta saláta
Rostos almalé	Rostos baracklé	**Rostos baracklé**	Rostos ivólé

A keresztezési folyamat a keresztezési pont megválasztásával kezdődik.

A keresztezési pontot a legegyszerűbb esetben véletlenszerűen választjuk meg.

A kromoszómák (egyedek) paraméter-értékeit felcseréljük a keresztezési ponttól kezdődően.

Mutáció

- A mutáció egy paraméter értékét változtatja meg véletlenszerűen.

Pl. Paradicsom leves helyett Zöldborsó krémleves

- Ki lehet majd próbálni, hogy a keresztezés és a mutáció valószínűségét mekkorának érdemes választani! (80%, illetve 10-40% között várható a legjobb eredmény?!)

Fitneszfüggvény

- A megoldáshoz olyan fitnesz függvényt kell használnunk, mely lehetőséget ad a korlátokon kívül eső tápanyagösszetevő-értékkel rendelkező javaslatok elvetésére, a helyes értékkel rendelkező egyedekhez pedig jó osztályzatot rendel, továbbá a nem harmonizáló étkezéseket kiszűri.
- A GA minden egyes iterációjában az osztályozó függvény jósági értéket rendel a kromoszómákhoz.

- Pl. ebéd étkezés tervezése esetén korlátozhatjuk az abban megtalálható húsmennyiséget, és egy alacsonyabb szintű megoldás, a só mennyiségét is.
- Hasonlóan, heti étkezés tervezése esetén a fitnesz függvény osztályozhat a heti terv sótartalmának, a levesek vagy a hideg vacsorák számának függvényében.

Szabálybázis

- Az osztályozó függvény táplálkozás-szakértői szempontból értékes működése érdekében szabályalapú pontozást használhatunk.
- Egy adatbázisban tárolt szabálytörzs szabályait az osztályozó függvény minden értékelésnél illeszti a kvantitatív szempontból már értékelt egyedre, és végrehajtja a szabályban tárolt utasítást.
- Az étkezésekre vonatkozó szabálybázis egy sora a következőképpen nézhet ki:

Köret	Feltét	Ivólé	Leves	Kiegészítő	Utasítás	Paraméter
Tetszőleges	Fehér hús	Paradicsomos ivólé	Paradicsomleves	Tetszőleges	Ront	50%

Amennyiben a szabálybázisban ez a sor szerepel és a fitnesz függvény olyan egyedet osztályoz, amelyre igaz, hogy paradicsomos ivólét és paradicsomlevest tartalmaz és a feltét a lehetséges értékek halmazszerű tárolásában a fehér hús halmazba esik, akkor végrehajtja az utasítást, jelen esetben felére rontja az egyed pontszámát.

A szabálybázis további finomítási lehetőségei

- Gyakran kerülhetünk olyan helyzetbe, amikor a tápanyag-összetevők attribútumainak értékére vonatkozó korlátok teljesülnek, de ízben vagy összetevőben nem harmonizálnak egymással az étkezés részei.
- Ennek elkerülésére is szabálybázis vezethető be. Ennek használatával nagymértékben személyre szabható rendszert kaphatunk, melyben lehetőség van csoportok, külön felhasználók részére szabályokat felvenni.
  - Ezek lehetnek a csoport egészségügyi problémákból adódó megszorításai, vagy a felhasználó által összeállított, testreszabott ízvilág.
  - A nem összeférhető receptösszetevőket nem tiltjuk, hanem bizonyos százalékban rontjuk a fitnesz értéküket.

    Pl. a felhasználó megadja, hogy a főzelékek és a gyümölcsös joghurt együttes fogyasztása során allergiás tünetek lépnek fel nála.

A szabálybázis további finomítási lehetőségei

- A változatos receptek generálásához szükséges szabályok is dinamikus formában adhatók meg.

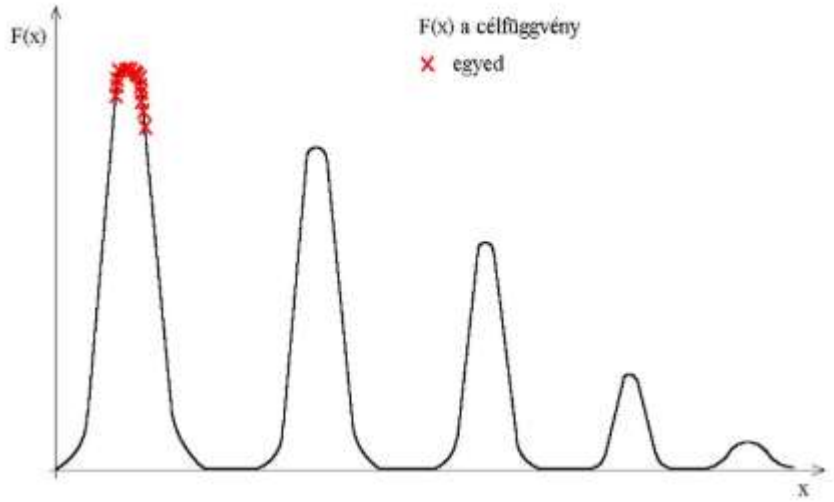
  pl. a felhasználó két hete már fogyasztott túrós rétest, ezért rontsuk az ilyen recepteket tartalmazó kromoszómák fitnesz-értékét.

- Így a szabálybázis a táplákozási-szakértők által meghatározott statikus szabályokból és a generálás során létrejövő dinamikus szabályokból áll.

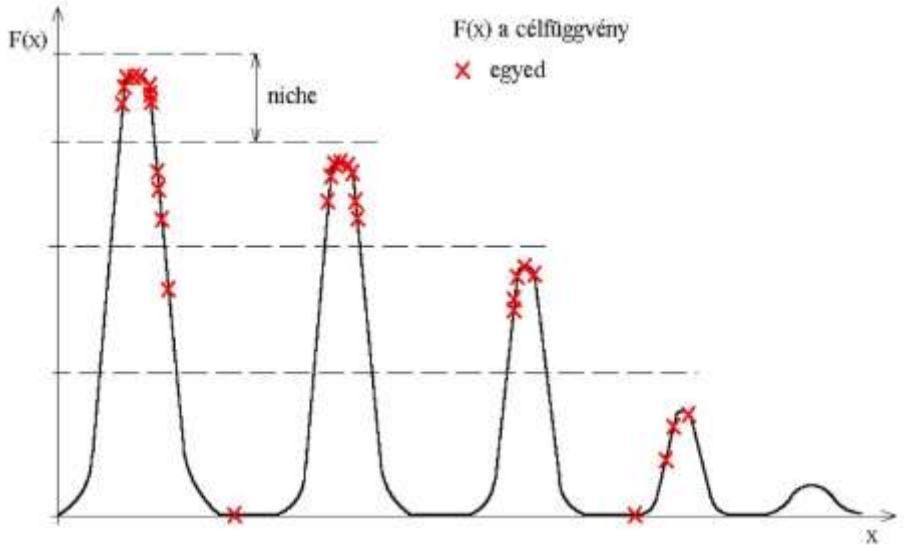
Újfajta megközelítések
Életfeltételeket alkalmazó módszerek (Niching methods)

- Megfigyelhetjük, hogy a természetben az egyes fajok különböző életkörülményekhez alkalmazkodnak, így a többi fajtól elkülönülve, őket kiszorítva használják fel az erőforrásokat.
- Ez a kiszorítás nem feltétlenül térbeli jelentésű, az egyes fajok specializálódhatnak egy erőforrás kihasználására vagy problémamegoldó módszer alkalmazására.
- Ezeket az egymástól elhatárolt, az egyedek csoportjai által elfoglalt életterületeket nevezzük *niche*-nek.
- A genetikus algoritmusok területén alkalmazva a niche-ket elérhetjük, hogy az algoritmus a lokális optimumokhoz tartozó megoldásokat is megtalálja.
- Az eljárás egyik megvalósítási módja a fitnesz megosztás, amely az egyes egyedek között elosztja egy adott terület fitness értékét. Így limitálja az egyedek számát a megoldások környezetében, ezáltal nem tud az összes egyed a globális szélsőérték köré csoportosulni.

- Egyszerű genetikus algoritmus



- Niche-éket alkalmazó algoritmus

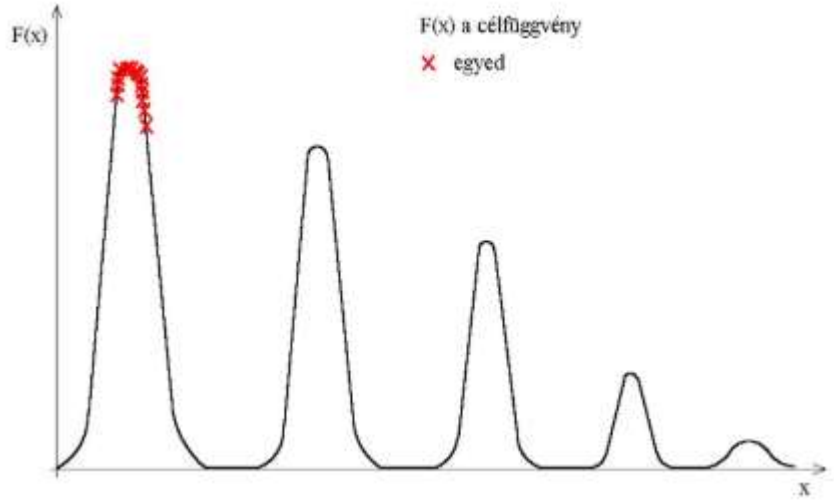


Fajkialakító módszerek (Speciation methods)

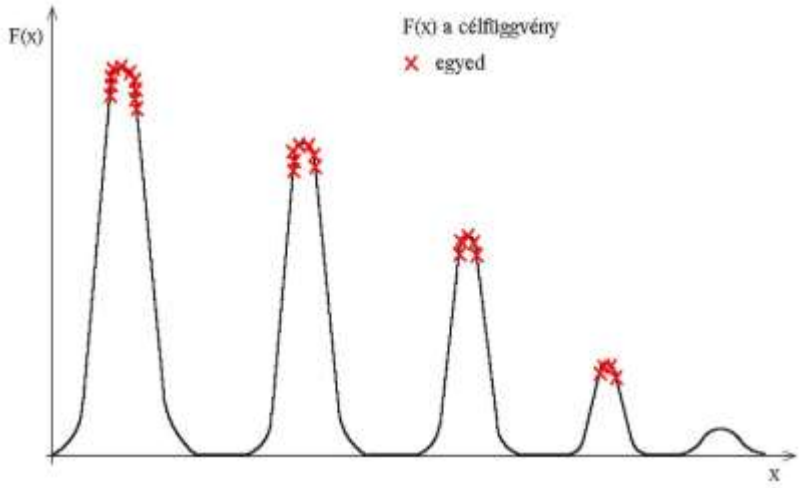
- A módszer ötlete abból a megfigyelésből ered, hogy a niche-ket alkalmazó eljárások nem tudnak egyenlő mértékben koncentrálni az egyes lokális optimumokra, és egyenlő hatékonysággal keresni az összes szélsőértéket.
- Ennek az a magyarázata, hogy az utódok létrehozásához az algoritmus különböző niche-hez tartozó egyedeket is választhat, így mindkét lokális szélsőérték szempontjából rossz egyed jönhet létre.
- A fajkialakító módszerek ezt a problémát úgy küszöbölik ki, hogy csak az egymáshoz hasonló, közeli egyedek hozhatnak létre utódokat.
- Ha a hasonlóság feltételét megfelelően határozzuk meg, akkor a keletkező egyed is ugyanazon optimum közelében marad.

- Az életfeltételeket alkalmazó és a fajkialakító modelleket a multikritériumú optimalizálási esetekné l is alkalmazzák.
- Ezek a feladatok egyidejűleg több célfüggvény szerinti optimalizálást követelnek meg.

- Egyszerű genetikus algoritmus

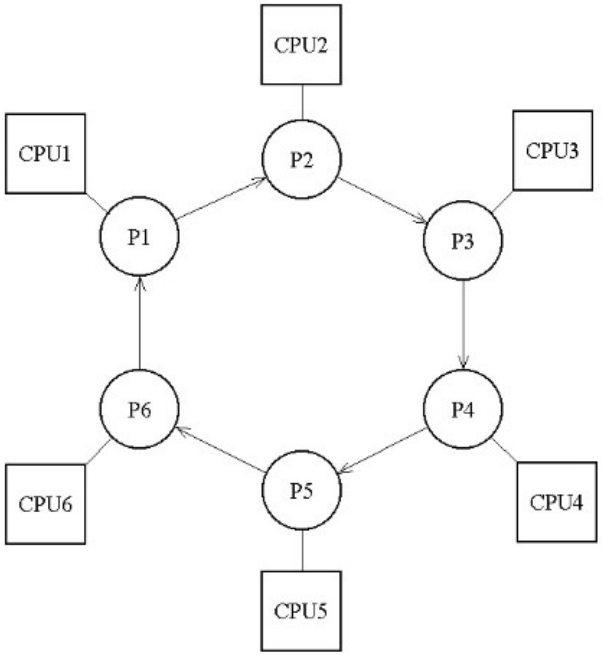


- Fajkialakító algoritmus



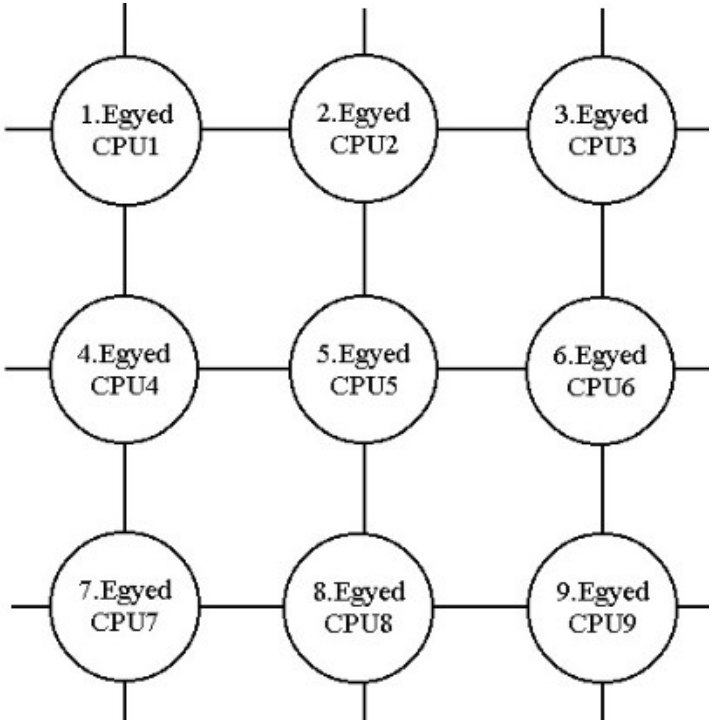
Sziget- vagy áttelepedési modellek (Island or Migration modells)

- Ezek a modellek a párhuzamos genetikus algoritmusok területén alkalmazhatók. A populáció felosztása az eljárás sebességének növelését célozza.
- A szigetmodellek a populációt több alpopulációra bontják, és ezek között úgy létesítenek kapcsolatot, hogy bizonyos időközönként az egyedek meghatározott csoportját kicserélik köztük. Az alpopulációkon egymástól függetlenül, általában külön mikroprocesszorok végzik a számításokat.
- Migrációs modell:



Diffúziós vagy celluláris modellek

- Mivel az egyedek létrehozása és célfüggvényük kiszámítása egymástól független, ezért párhuzamosan is történhet. A diffúziós modellek esetében az egyes egyedeken végzett műveleteket külön mikroprocesszorok végzik, így még az előzőnél is nagyobb sebességnövekedés érhető el.
- Diffúziós modell:



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/órára a lehető legrövidebb idő alatt gyorsuljunk fel.
- Az optimizálandó függvény formája igen bonyolult – hiszen az adott fokozatok nyomatékgörbéi erősen függnek 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).