

Evolúció algoritmusok

Evolúció, mint kereső rendszer

- ❑ A problémátérnek egyszerre több **egyedét** (a problémára adható lehetséges válaszokat) tároljuk az ún. **populációban**.
- ❑ Egy többnyire véletlen populációból indulunk ki, amelyet **lépésről lépésre javítjuk** azért, hogy megjelenjen benne egy célegyed vagy egy összességében jó populációhoz jussunk.
- ❑ Az egyedeket egy ún. **rátermettségi függvény** segítségével hasonlítjuk össze. Minden lépésben a kevésbé rátermett egyedek egy részét a rátermettebbekhez hasonló egyedekre cseréljük le. Ez a változtatás visszavonhatatlan. Ez tehát egy **nem-módosítható stratégiájú keresés**.

Evolúciós operátorok és a terminálási feltétel

- ❑ *Szelekció*: Kiválasztunk néhány (lehetőleg rátermett) egyedet szülőnek.
- ❑ *Rekombináció (keresztezés)*: Szülőkből utódok készülnek úgy, hogy a szülők tulajdonságait örököljék az utódok.
- ❑ *Mutáció*: Az utódok tulajdonságait kismértékben módosítjuk.
- ❑ *Visszahelyezés*: Új populációt alakítunk ki az utódokból és a régi populációból.
- ❑ *Terminálási feltétel*:
 - ha a célegyed megjelenik a populációban
 - ha a populáció egyesített rátermettségi függvény értéke egy ideje nem változik.

ADAT := *kezdeti érték*

while \neg *terminálási feltétel*(ADAT) **loop**

 SELECT SZ FROM *alkalmazható szabályok*

 ADAT := SZ(ADAT)

endloop

Evolúció alapalgorithmusa

Procedure EA

populáció := *kezdeti populáció*

while *terminálási feltétel nem igaz* **loop**

szülők := *szelekció*(*populáció*)

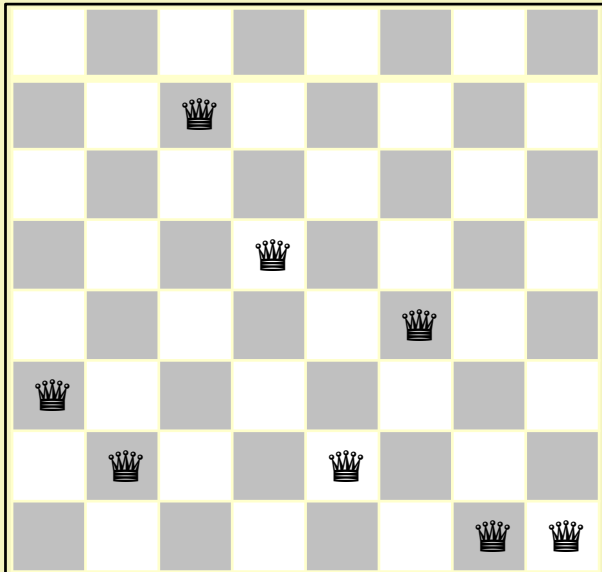
utódok := *rekombináció*(*szülők*)

utódok := *mutáció*(*utódok*)

populáció := *visszahelyezés*(*populáció*, *utódok*)

endloop

n-királynő probléma 1.

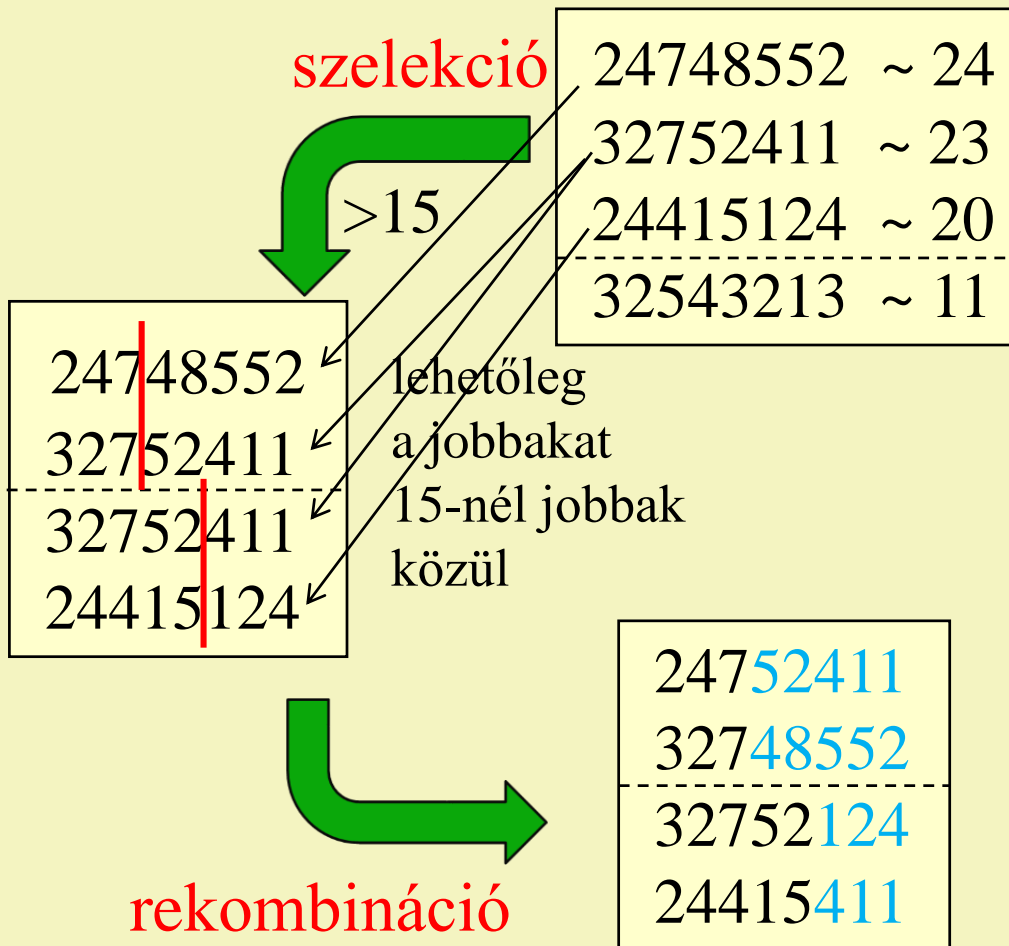


3	2	7	5	2	4	1	1
---	---	---	---	---	---	---	---

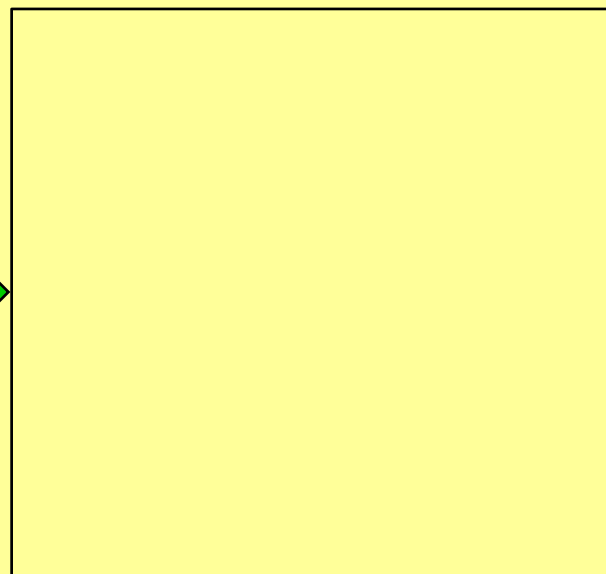
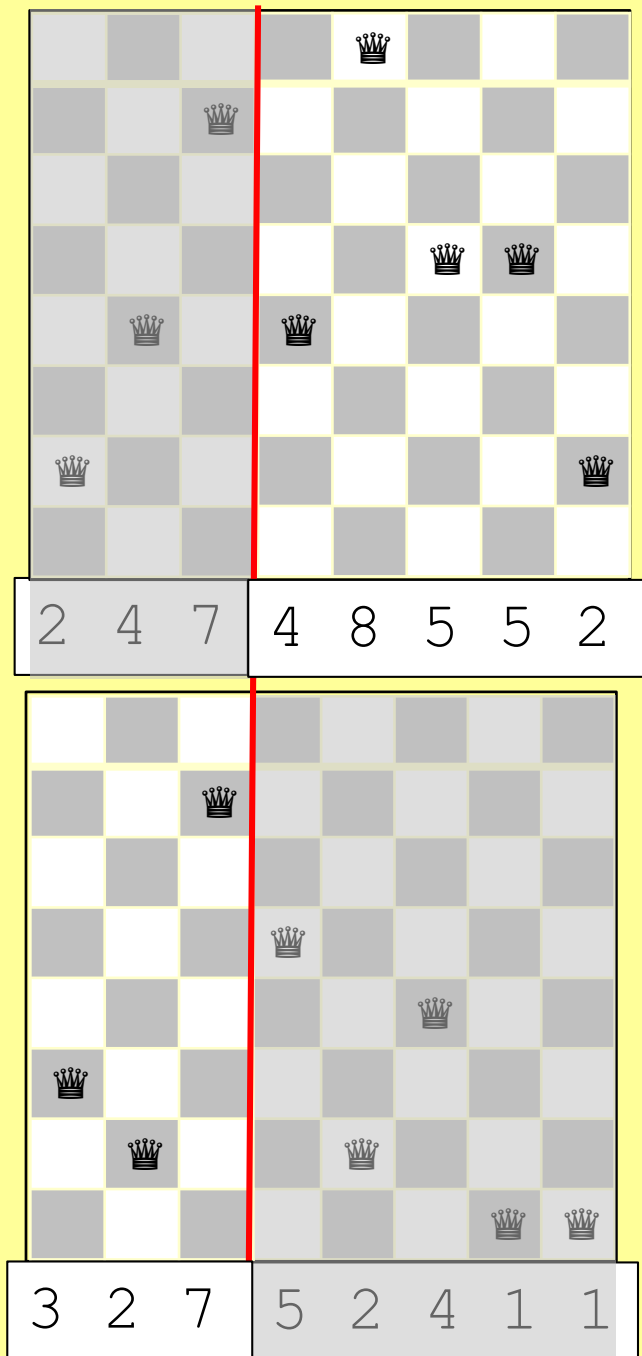
rátermettségi érték: 23

- Egyed: a királynők olyan elrendezése, ahol minden oszlop pontosan egy királynőt tartalmaz
- Reprezentáció: oszloponként a királynők sorpozícióit tartalmazó sorozat
- Rátermettségi függvény: ütésben nem levő királynő párok száma

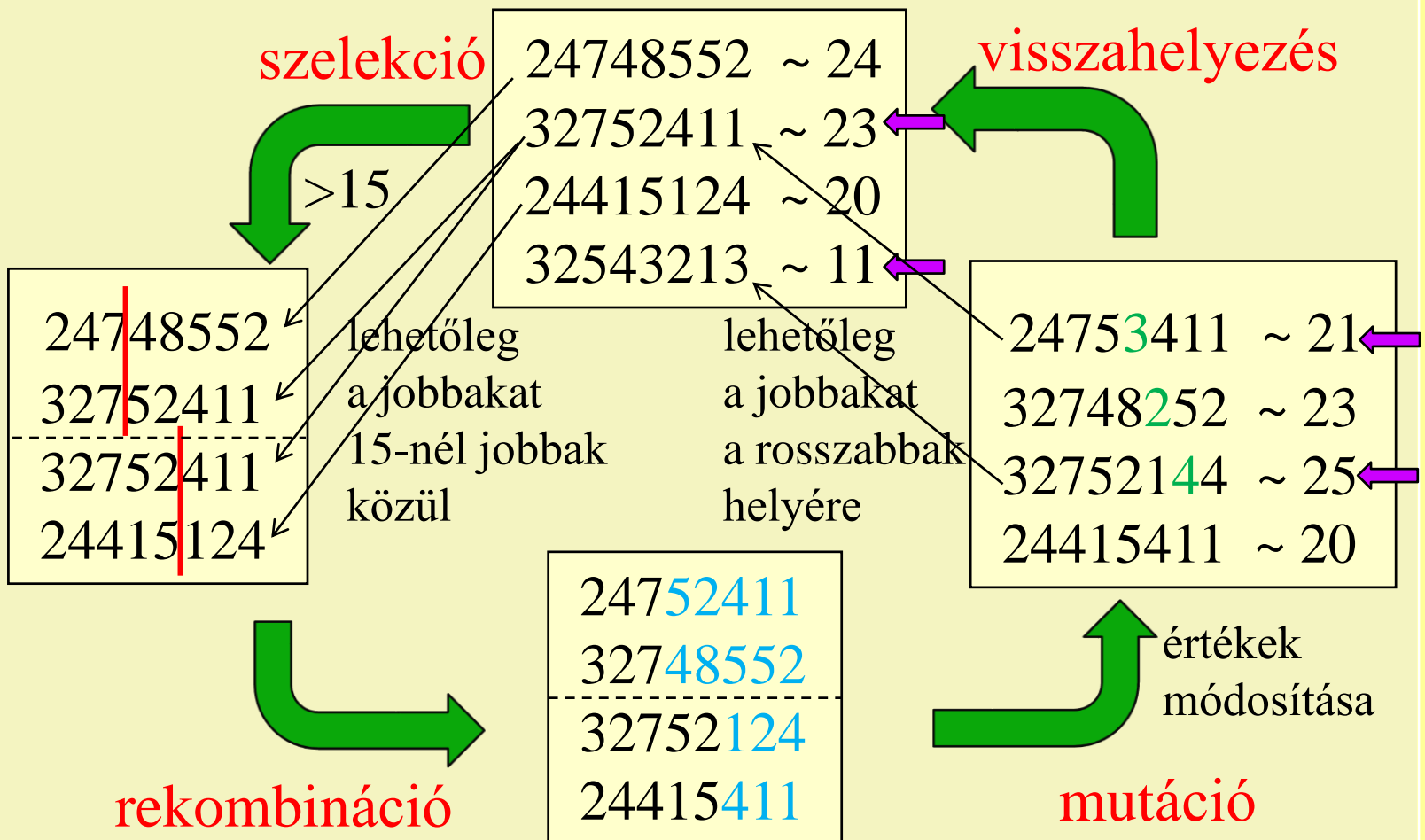
Evolúciós ciklus



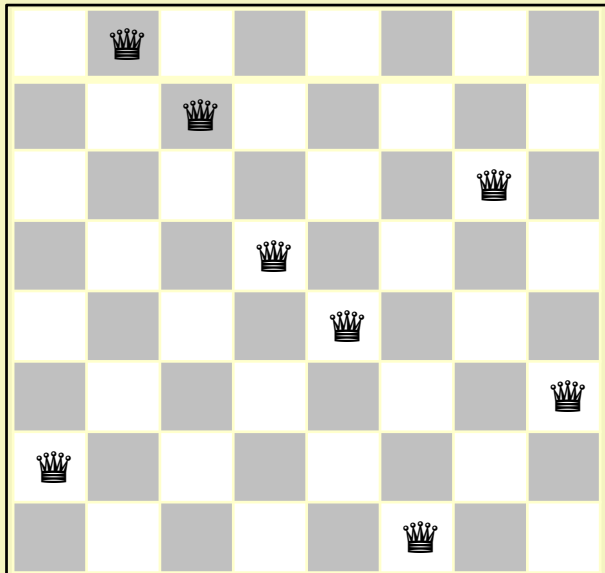
Keresztezés



Evolúciós ciklus



n-királynő probléma 2.

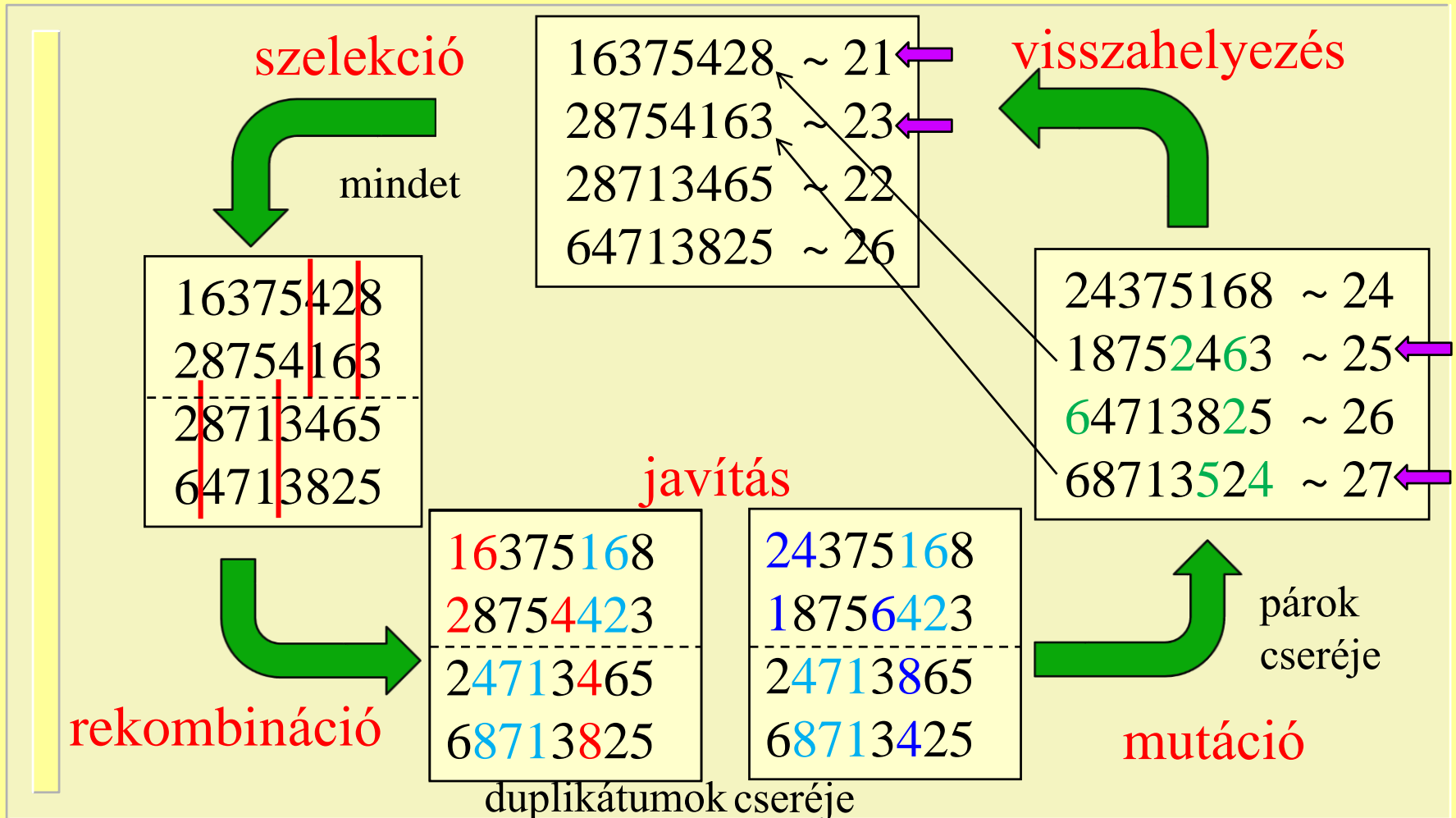


2 8 7 5 4 1 6 3

rátermettségi érték: 23

- Egyed: a királynők olyan elrendezése, ahol minden sor és oszlop pontosan egy királynőt tartalmaz
- Reprezentáció:
oszloponként a királynők sorpozícióit tartalmazó permutáció
Rátermettségi függvény: ütésben nem levő királynő párok száma

Evolúciós ciklus



Kielégíthetőségi probléma (SAT)

Adott egy n változós Boolean formula KNF alakban. A változók milyen igazság kiértékelése mellett lesz formula igaz?

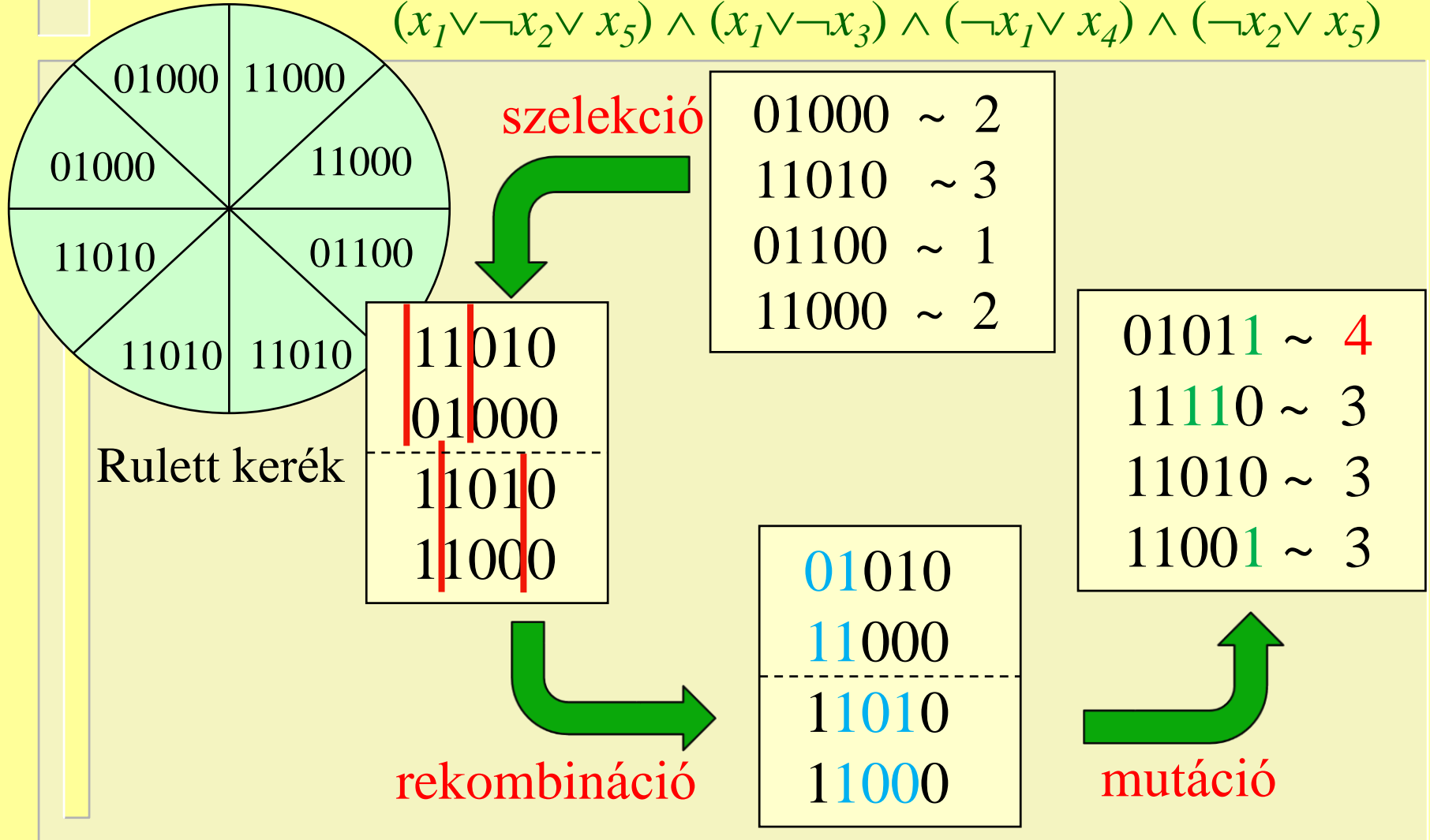
E.g.: $(x_1 \vee \neg x_2 \vee x_5) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_4) \wedge (\neg x_2 \vee x_5)$

egy megoldás: $x_1 = true, x_2 = false, x_3 = false, x_4 = true, x_5 = true$

- ❑ Egyed: egy lehetséges igazság kiértékelés
- ❑ Reprezentáció: logikai érték (bitek) sorozata
- ❑ Rátermettségi függvény: Az adott formula igazra értékelt klózainak száma

Evolúciós ciklus

$$(x_1 \vee \neg x_2 \vee x_5) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_4) \wedge (\neg x_2 \vee x_5)$$

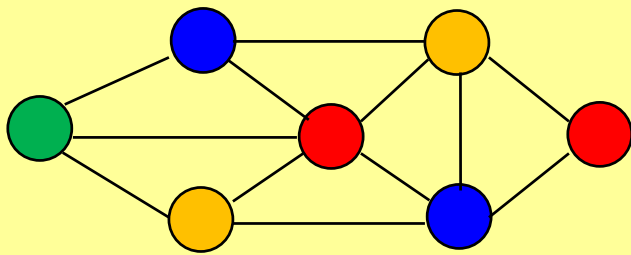


Evolúciós algoritmus tervezése

- ❑ problémátér egyedeinek reprezentációja: **kódolás**
- ❑ **rátermettségi függvény** (fitness függvény)
 - kapcsolat a kódolással és a céllal
- ❑ **evolúciós operátorok**
 - szelekció, rekombináció, mutáció, visszahelyezés
- ❑ **kezdő populáció, megállási feltétel** (cél)
- ❑ **stratégiai paraméterek**
 - populáció mérete, mutáció valószínűsége, utódképzési ráta, visszahelyezési ráta, stb.

Kódolás

- ❑ Egy egyedet egy **jelsorozattal** (kromoszómával) kódolunk. A jelsorozatnak ki kell elégítenie a **kód-invariánst**.
- ❑ Az egyedeket az őket reprezentáló kódjukon keresztül változtatjuk meg. Egy jel vagy jelcsoport, azaz a gén írja le az egyed egy tulajdonságát (attribútum-érték párját).
 - Sokszor egy génnek a kódsorozatban elfoglalt pozíciója (lókusza) jelöli ki a gén által leírt attribútumot, amelynek értéke maga a gén (allél). A kód ekkor tulajdonságonként **feldarabolható**: egy rövid kódszakasz megváltoztatása kis mértékben változtat az egyeden.
- ❑ Gyakori megoldások:
 - **Vektor**: valós vagy egész számok rögzített hosszú tömbje
 - **Bináris kód**: bitek rögzített hosszú tömbje
 - Véges sok elem **permutációja**



Gráf színezési probléma

Adott egy véges egyszerű gráf, amelynek a csúcsait négy szín felhasználásával kell úgy kiszínezni, hogy a szomszédos csúcsok eltérő színűek legyenek.

Direkt kódolás



1. 2. 3. 4. 5. 6. 7.

Az $x[i]$ az i -dik csúcs színe.

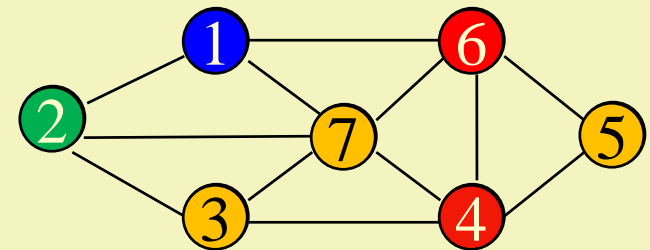
f a jó élek száma.

Indirekt kódolás

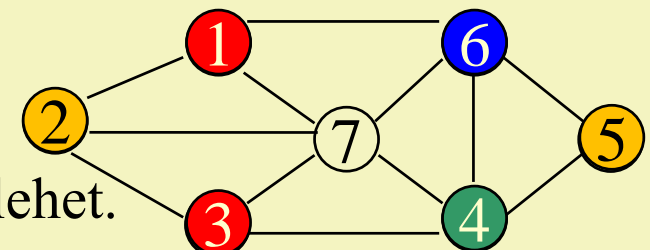


Az i -dik lépésben az $x[i]$ -dik csúcsot színezzük ki a lehető legvilágosabb színnel a szomszédjaihoz igazodva, ha lehet.

f a kiszínezett csúcsok száma



$f = 10$



$f = 6$

A kő-papír-olló játék stratégiája

Alakítsunk ki jó stratégiát egy kő-papír-olló világbajnokságra!

- Olyan függvényre van szükségünk, amelyik a korábbi csaták kimenetele alapján javaslatot tesz a soron következő lépésünkre.

- Például két korábbi csata alapján:

Előzmény: *Én:* **K** **P** *Javaslat:* **K**
 Ő: **O** **O**

- Ez még nem a teljes stratégia, mert nem csak a fenti előzményre, hanem az összes lehetséges előzményre kell soron következő lépést javasolni.

Kódolás

Egy stratégia (egyed) kódja: $\{0,1,2\}^{0..80}$

Az összes lehetséges stratégia száma: 3^{81}

Jelek

K ~ 0

P ~ 1

O ~ 2

Előzmény (ÉnŐÉnŐ)

KKKK ~ 0000 ~ 0

KKKP ~ 0001 ~ 1

KKKO ~ 0002 ~ 2

KKPK ~ 0010 ~ 3

...

OOOP ~ 2221 ~ 79

OOOO ~ 2222 ~ 80

Válasz

P ~ 1

O ~ 2

K ~ 0

P ~ 1

...

O ~ 2

K ~ 0

A stratégia: 1201 ... 20

Rátermettség kiértékelése

Stratégia: 1201 ... 20

Minta:

Játékos: 0002221222001000

Ellenfél: 0102112220101011

Jelek

K ~ 0

P ~ 1

O ~ 2

<i>Eset</i>	<i>→</i>	<i>Javaslat</i>	<i>Ellenfél</i>	<i>Érték</i>
0001	→	2	0	vereség -1
0001	→	2	1	győzelem +1
0100	→	1	1	döntetlen 0
...				
2221	→	2	1	győzelem +1
2222	→	0	0	döntetlen 0

Szelekció

- **Célja:** a rátermett egyedek kiválasztása úgy, hogy a rosszabbak kiválasztása is kapjon esélyt.
 - **Rátermettség arányos** (rulett kerék algoritmus): minél jobb a rátermettségi függvényértéke egy elemnek, annál nagyobb valószínűséggel választja ki
 - **Rangsorolásos:** rátermetség alapján sorba rendezett egyedek közül a kisebb sorszámúakat nagyobb valószínűséggel választja ki
 - **Versengő:** véletlenül kiválasztott egyedcsoportok (pl. párok) legjobb egyedét választja ki.
 - **Csonkolásos** v. **selejtezős:** a rátermetség szerint legjobb (adott küszöbérték feletti) valahány egyedből véletlenszerűen választ néhányat.

Rekombináció

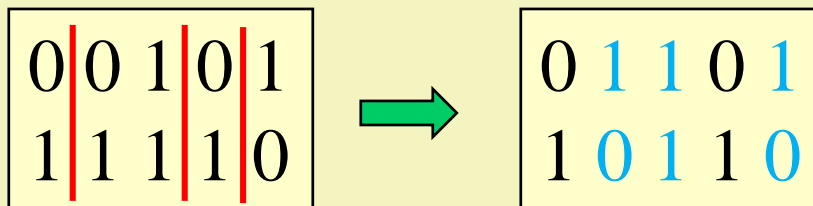
- A feladata az, hogy adott szülő-egyedekből olyan utódokat hozzon létre, amelyek a szüleik tulajdonságait "öröklik".
 - **Keresztezés:** véletlen kiválasztott pozíción jelcsoportok (gének) vagy jelek cseréje
 - **Rekombináció:** a szülő egyedek megfelelő jeleinek kombinálásával kapjuk az utód megfelelő jelét

Ügyelni kell a kód-invariáns megtartására: vizsgálni kell, hogy az új kód értelmes lesz-e (permutáció)

Keresztezés

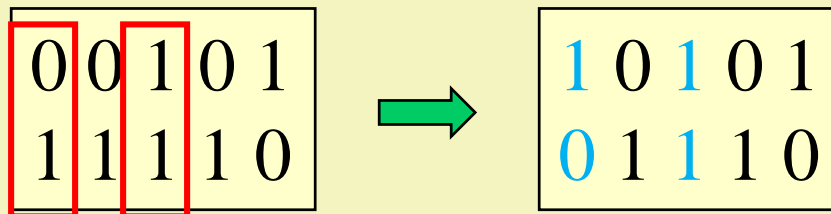
❑ Egy- illetve többpontos keresztezés

- Kódszakaszokat cserélünk



❑ Egyenletes keresztezés

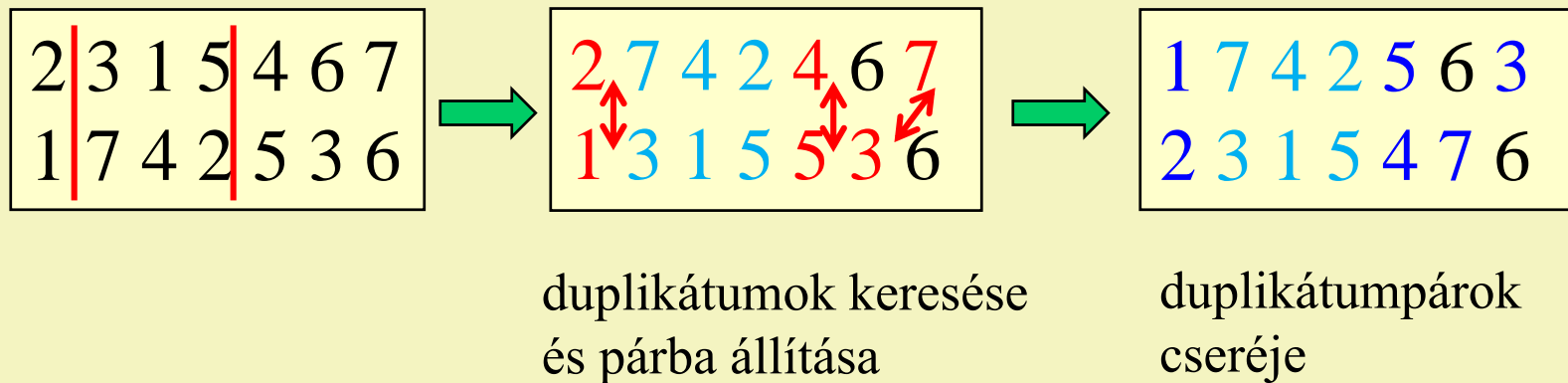
- Jeleket cserélünk



Permutációk keresztezése 1.

□ Parciálisan illesztett keresztezés

- Egy szakasz cseréje után párba állítja és kicseréli azokat a szakaszon kívüli elemeket, amelyek megsértik a permutáció tulajdonságát.

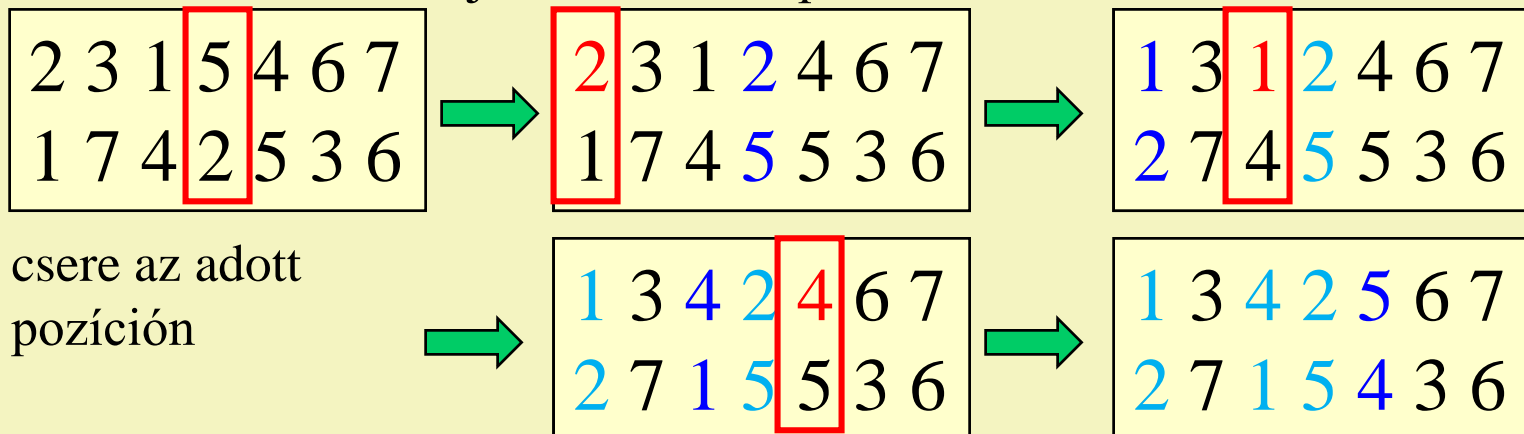


Permutációk keresztezése 2.

□ Ciklikus keresztezés

1. Választ egy véletlen $i \in [1..length]$ -t
2. $a_i \leftrightarrow b_i$
3. Keres olyan $j \in [1..length]$ -t ($j \neq i$), amelyre $a_j = a_i$,
4. Ha nem talál, akkor vége, különben $i := j$
5. goto 2.

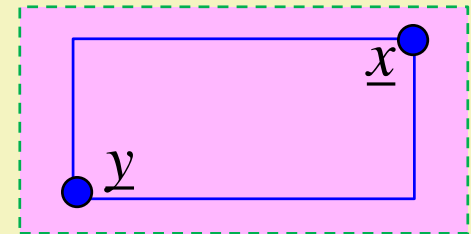
duplikátum keresése a felső utódban,
majd csere azon a pozíción is



Rekombináció vektorokra

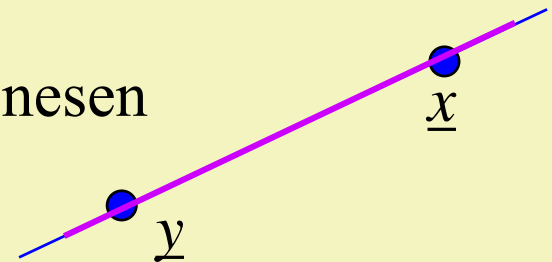
□ Köztes rekombináció

- A szülők (\underline{x} , \underline{y}) által kifeszített hipertégla környezetében lesz az utód (\underline{u}).
- $\forall i=1 \dots n : u_i = a_i x_i + (1-a_i)y_i \quad a_i \in [-h, 1+h]$ véletlen



□ Lineáris rekombináció

- A szülők (\underline{x} , \underline{y}) által kifeszített egyenesen a szülők környezetében vagy a szülők között lesz az utód (\underline{u}).
- $\forall i=1 \dots n : u_i = a x_i + (1-a)y_i \quad a \in [-h, 1+h]$ véletlen



Mutáció

- ❑ A mutáció egy egyed (utód) **kis mértékű véletlen változtatását** végzi.
- ❑ Valós tömbbel való kódolásnál kis p valószínűséggel:
 - $\forall i=1 \dots n : z_i = x_i \pm domain_i \cdot p$
- ❑ Bináris tömbbel való kódolásnál kis p valószínűséggel:
 - $\forall i=1 \dots n : z_i = 1 - x_i$ if $random[0..1] < p$
- ❑ Permutáció esetén
 - egy jelpár cseréje
 - egy kódszakaszon a jelek ciklikus léptetése vagy megfordítása vagy átrendezése.

Visszahelyezés

- A visszahelyezés a populációnak az utódokkal történő frissítése: Kiválasztja a populációnak a lecserélendő egyedeit, és azok helyére a kiválasztott utódokat teszi.

$$\text{utódképzési ráta (u)} = \frac{\text{utódok száma}}{\text{populáció száma}}$$

két szelekció is kell

$$\text{visszahelyezési ráta (v)} = \frac{\text{lecserélendő egyedek száma}}{\text{populáció száma}}$$

- ha $u=v$, akkor feltétlen cseréről van szó
- ha $u < v$, akkor egy utód több példánya is bekerülhet
- ha $u > v$, akkor az utódok közül szelektál

további szelekció

további szelekció