

Képrekonstrukció 6. előadás



Balázs Péter

Képfeldolgozás és Számítógépes Grafika Tanszék
Szegedi Tudományegyetem

Diszkrét tomográfia (DT)

- A CT-hez több száz vetület szükséges
 - időigényes
 - költséges
 - károsíthatja a vizsgált objektumot
- Egyes alkalmazásokban a rekonstruálandó függvény csak néhány előre ismert értéket vehet fel → DT (csak 2-10 vetület szükséges)

KNOWING THE DISCRETE RANGE



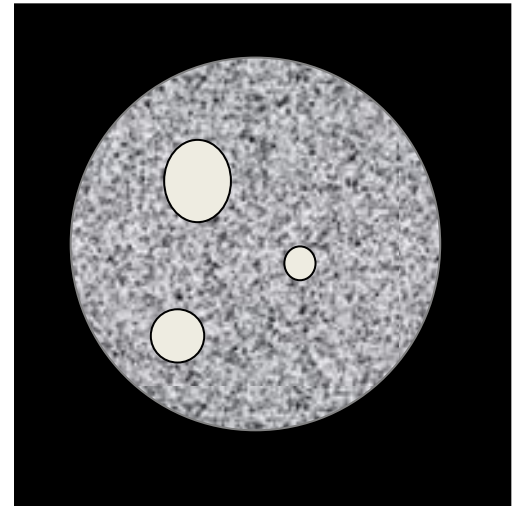
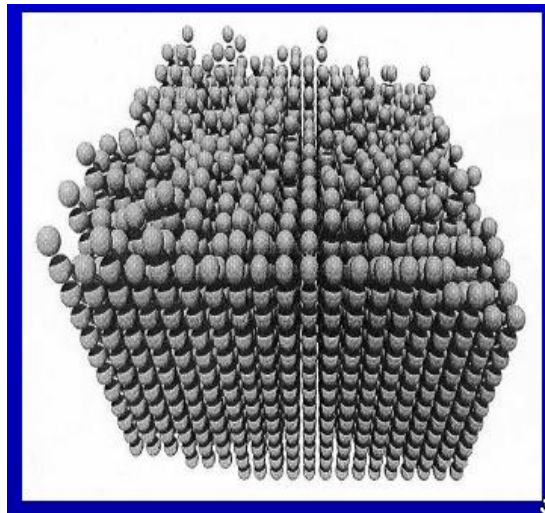
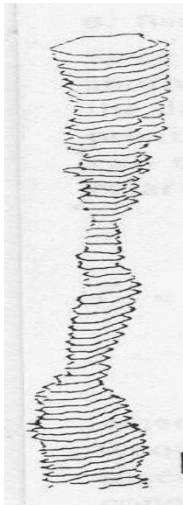
# projs.	Conv. method	Discretized image	DT method
8			
12			
16			

L. Ruskó, A.K., Z. Kiss, L. Rodek, 2003

Bináris tomográfia

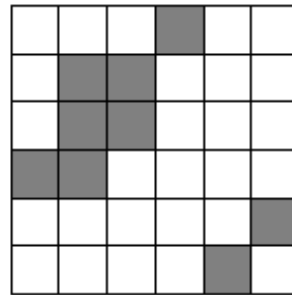
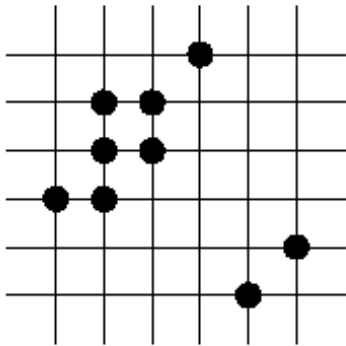
A rekonstruálandó függvény értékkészlete $\{0,1\}$ (anyag hiánya/jelenléte)

- **angiográfia:** vérerek leképezése Röntgen-sugarakkal
- **elektronmikroszkópia:** molekulák, kristályok szerkezete
- **neutroncsoló tesztelés:** homogén objektumok alakja



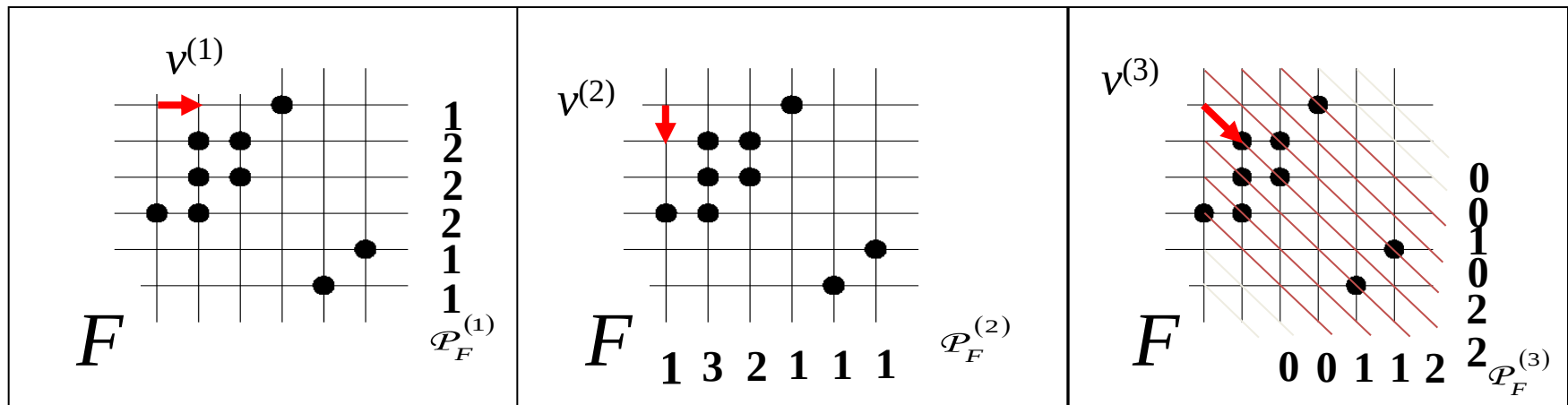
Diszkrét halmazok és vetületek

- diszkrét halmaz: $\mathbb{Z}^2$ véges részhalmaza



$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- diszkrét halmaz vetületei



Fő feladatok

Konzisztencia: Létezik-e diszkrét halmaz a megadott vetületekkel?

Egyértelműség: Egy diszkrét halmazt egyértelműen meghatároznak-e a vetületei?

Rekonstrukció: Konstruáljunk egy diszkrét halmazt a vetületeiből.

Bináris mátrixok I

- $A = (a_{ij})_{n \times m}$
- $R = (r_1, \dots, r_n)$ sorösszeg vektor – $R(A)$
- $S = (s_1, \dots, s_m)$ oszlopösszeg vektor – $S(A)$

$$r_i = \sum_{j=1}^m a_{ij}, \quad i = 1, \dots, n \quad s_j = \sum_{i=1}^n a_{ij}, \quad j = 1, \dots, m$$

- $A=B : \quad \forall i, j : a_{ij} = b_{ij}$
- $\mathcal{A}(R, S) := \{A \mid R(A) = R, S(A) = S\}$

Egyértelmű mátrixok rekonstrukciója

- Tiltott pozíciók halmaza: Q
- Egy egyértelmű mátrixban vagy nincs szabad pozíció vagy van benne *primitív* sor (oszlop), azaz amelyben
 - csak 1-ek vannak szabad pozícióban vagy
 - csak 0-k vannak szabad pozícióban.

Értékadás és behelyettesítés (S.K. Chang – 1971)

Algoritmus (R,S,Q)

while (R≠0)

{ keressünk primitív sort/oszlopot;
írjuk be a 0/1-et a sorba/oszlopba;
módosítsuk a sor és oszlopösszegeket; }

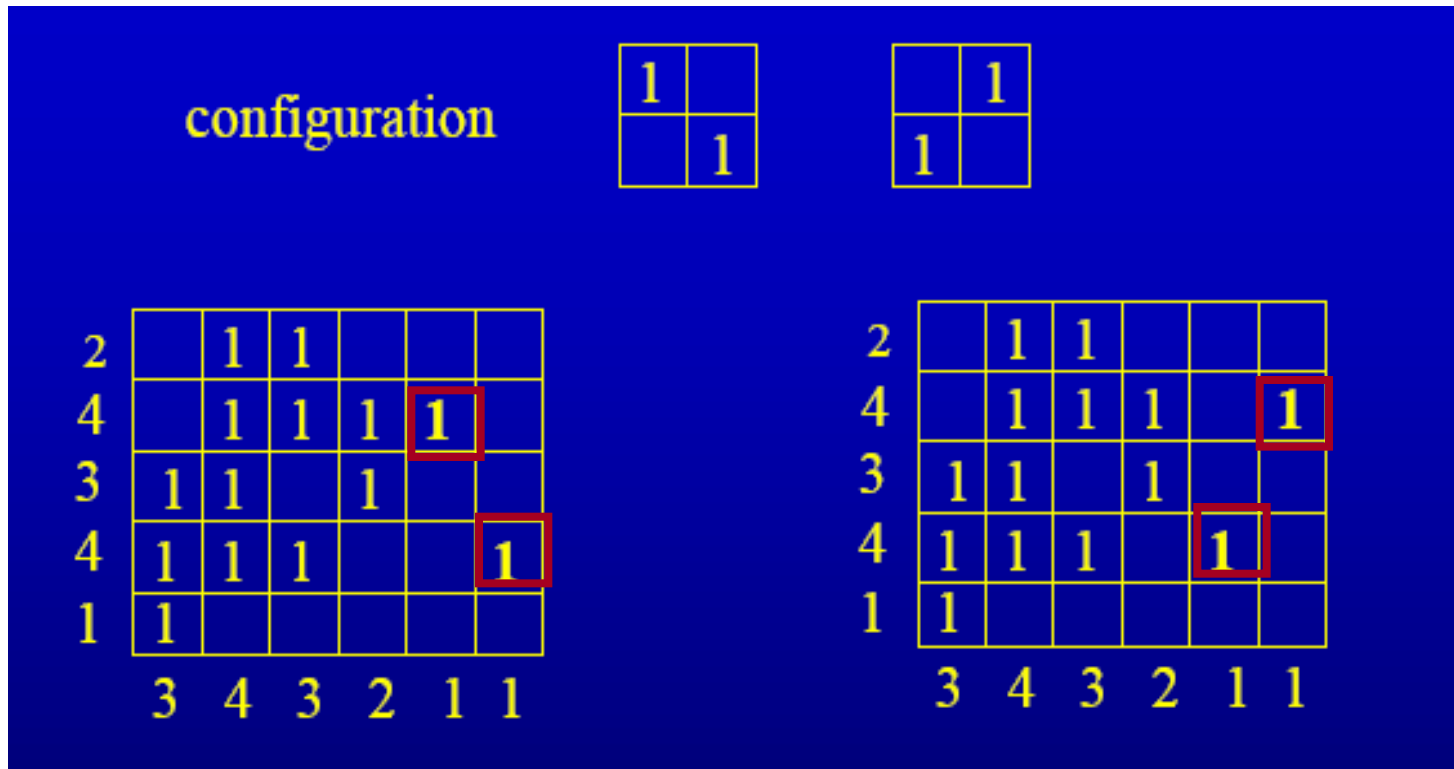
↑

Példa: $R=(2, 2, 4, 1, 2)$

$S=(1, 2, 5, 2, 1)$

$Q=\{(1,1), (2,2), (2,5), (3,4), (5,2), (5,5)\}$

Kapcsoló komponensek és unicitás



A kapcsoló komponens jelenléte szükséges és elegendő a nem-egyértelműséghez

Bizonyítás

Bizonyítás:

$\Rightarrow$: triviális

$\Leftarrow$: Legyen $A, \tilde{A} \in \mathcal{A}(R, S)$ $|A \neq \tilde{A}|$

$$\begin{array}{c} \begin{array}{c|ccc} & j_1 & j_2 & j_3 & \dots \\ \hline i_1 & 1 & \dots & 0 & \\ i_2 & & & 1 & \dots & 0 \\ \vdots & & & & \ddots & \end{array} & \Downarrow & \begin{array}{c|ccc} & j_1 & j_2 & j_3 & \dots \\ \hline i_1 & 0 & \dots & 1 & \\ i_2 & & & 0 & \dots & 1 \\ \vdots & & & & \ddots & \end{array} \end{array}$$

$\exists i_1, j_1 : a_{i_1 j_1} \neq \tilde{a}_{i_1 j_1}$
 legyen $a_{i_1 j_1} = 1, \tilde{a}_{i_1 j_1} = 0$

$\Downarrow R(A) = R(\tilde{A})$

$\exists j_2 : a_{i_1 j_2} = 0, \tilde{a}_{i_1 j_2} = 1$

$\Downarrow S(A) = S(\tilde{A})$

$\exists i_2 : a_{i_2 j_2} = 1, \tilde{a}_{i_2 j_2} = 0$

$\Downarrow$
 $\vdots$

véges sorozatot kapunk:

$(i_1, j_1), (i_1, j_2), \dots, (i_t, j_t), \dots, (i_t, j_t)$
 $\underbrace{\hspace{10em}}_{\text{átsorszámozzuk}}$

$(i'_1, j'_1), (i'_1, j'_2), \dots, (i'_k, j'_k) = (i'_1, j'_1)$

$$\begin{array}{c|cccc} i'_1 & 1 & 0 & & 0 \\ i'_2 & & 1 & 0 & 0 \\ \vdots & & & \ddots & \vdots \\ i'_k & 0 & & & 1 \end{array} \begin{array}{c} j'_1 \\ j'_2 \\ \vdots \\ j'_k \end{array}$$

feltevés: A nem tartalmaz kapcsoló komponenst

$\uparrow$ ellentmondás!

Bináris mátrixok II

- Legyen $R=(r_1,\dots,r_n)$ és $S=(s_1,\dots,s_m)$ nemnegatív egészekből álló *konzisztens* vektorpár, azaz

$$\sum_{i=1}^n r_i = \sum_{j=1}^m s_j.$$

- Jelölje $S'=(s'_1,\dots,s'_m)$ S nem-növekvő átrendezését: $s'_1 \geq s'_2 \geq \dots \geq s'_m$

- Kanonikus mátrix: $A^*=(a_{ij}^*)_{n \times m}$ $a_{ij}^* := \begin{cases} 1 & \text{ha } j = 1, 2, \dots, r_i \\ 0 & \text{különben} \end{cases}$

$$\begin{array}{c} 1 \\ 3 \\ 2 \\ 2 \end{array} \left[\begin{array}{cccc} & 1 & & \\ 1 & 1 & & 1 \\ & 1 & 1 & \\ & 1 & & 1 \end{array} \right] = A$$

$$(1 \quad 4 \quad 1 \quad 2) = S$$

$$\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \end{array} \left[\begin{array}{cccc} 1 & & & \\ 1 & 1 & & 1 \\ 1 & 1 & & \\ 1 & 1 & & \end{array} \right] = A^*$$

$$(4 \quad 3 \quad 1 \quad 0) = S^* = S(A^*)$$

Konzisztencia

- Szükséges feltétel: kompatibilitás

$$\sum_{i=1}^m r_i = \sum_{j=1}^n s_j$$

$$r_i \leq n \ (i = 1, \dots, m), \quad s_j \leq m \ (j = 1, \dots, n)$$

- Gale, Ryser, 1957: akkor és csak akkor létezik megoldás, ha

$$\sum_{j=1}^k s'_j \leq \sum_{j=1}^k s(A^*)_j \quad k = 1, \dots, n$$

3			
3			
1			
	3	3	1

$\underbrace{\hspace{10em}}_6 \quad S'$

3	1	1	1
3	1	1	1
1	1		
	3	2	2

$\underbrace{\hspace{10em}}_5 \quad S(A^*)$

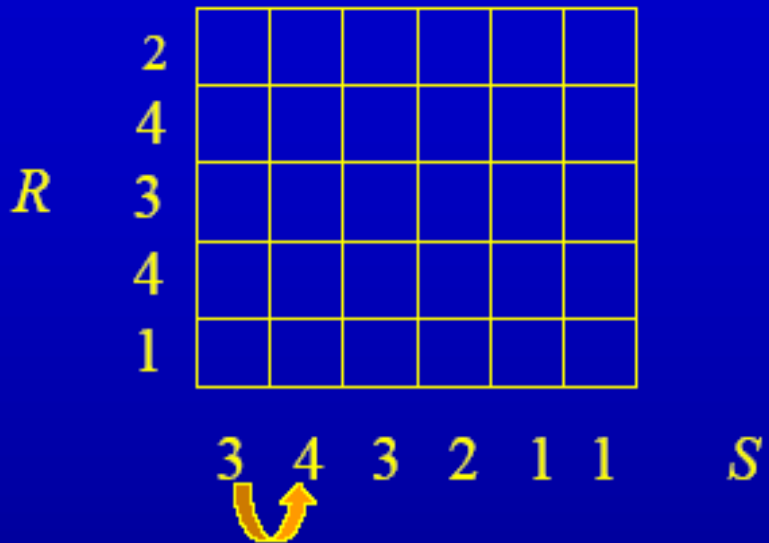
Rekonstrukció

Ryser, 1957 – az R sor- és S oszlopösszegekből

Képezzük S nemnövekvő átrendezését π permutációval $\rightarrow S'$
Töltsük fel az oszlopokat balról-jobbra $\rightarrow B$ (kanonikus mátrix)
 B legjobboldali oszlopaiból toljunk el elemeket oda, ahol $S(B) < S'$
 π inverzének segítségével rendezzük vissza az oszlopokat

Komplexitás: $O(nm + n \log n)$

R	2						
	4						
	3						
	4						
	1						
		3	4	3	2	1	1
		S					



R

2						
4						
3						
4						
1						

3 4 3 2 1 1 S



2

4

3

4

1

4 3 3 2 1 1 S'

R

2						
4						
3						
4						
1						

3 4 3 2 1 1 S



2

4

3

4

1

4 3 3 2 1 1 S'

 R

2	1	1				
4	1	1	1	1		
3	1	1	1			
4	1	1	1	1		
1	1					

 $=B$

5 4 3 2 0 0 $S(B)$

4 3 3 2 1 1 S'

R

2						
4						
3						
4						
1						

3 4 3 2 1 1 S



2
4
3
4
1

4 3 3 2 1 1 S'

 R

2	1	1				
4	1	1	1	1		
3	1	1	1			
4	1	1	1	1		
1	1					

 $=B$

5 4 3 2 0 0 $S(B)$

4 3 3 2 1 1 S'

R

2						
4						
3						
4						
1						

3 4 3 2 1 1 S



2

4

3

4

1

4 3 3 2 1 1 S'

 R

2	1	1				
4	1	1	1	1		
3	1	1	1			
4	1	1	1	1		
1	1					

 $=B$

5 4 3 2 0 0 $S(B)$

4 3 3 2 1 1 S'

2

4

3

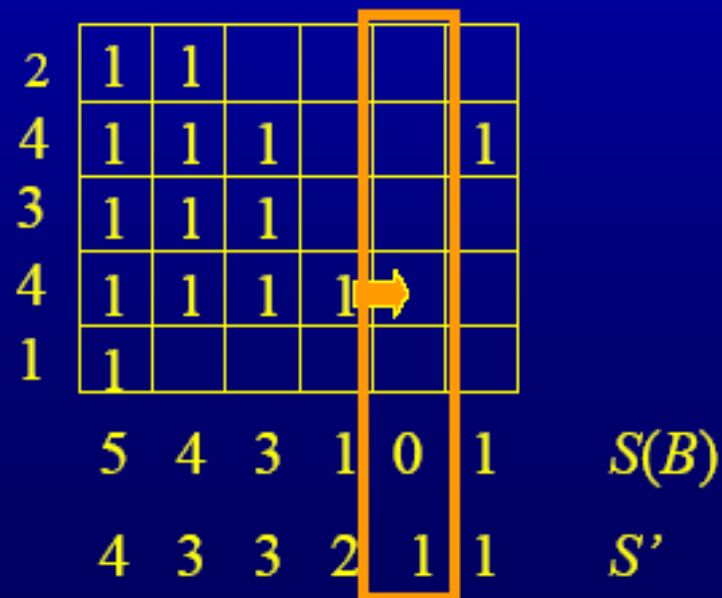
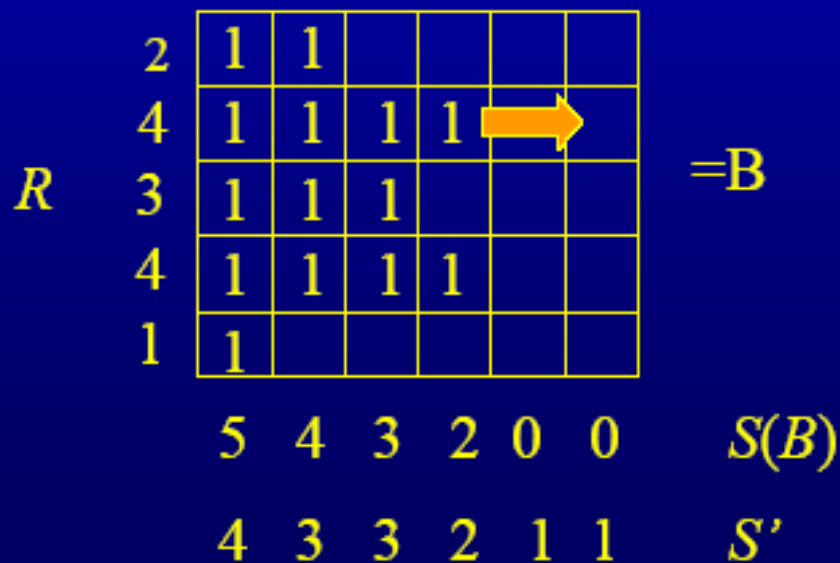
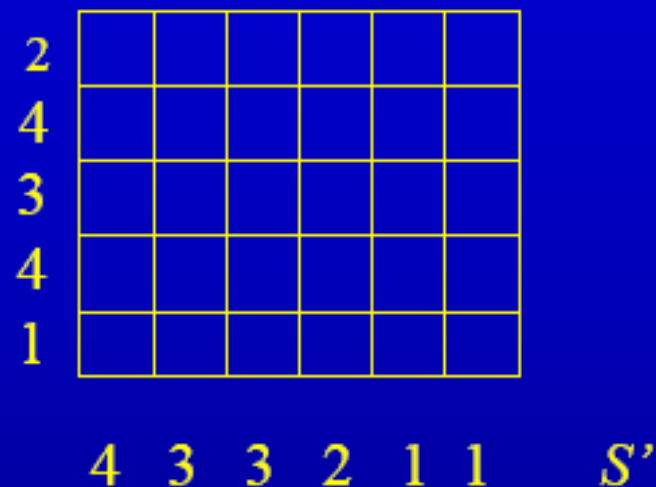
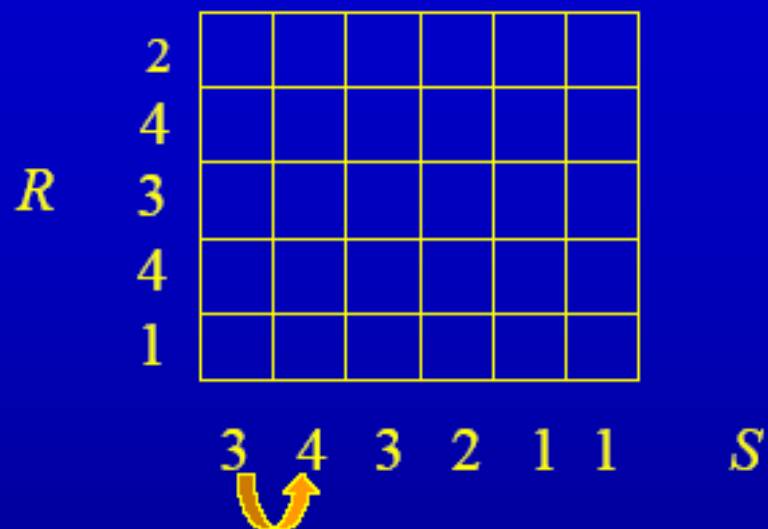
4

1

1	1				
1	1	1			1
1	1	1			
1	1	1	1		
1					

5 4 3 1 0 1 $S(B)$

4 3 3 2 1 1 S'



R	2	1	1				
	4	1	1	1			1
	3	1	1	1			
	4	1	1	1		1	
	1	1					

$5 \quad 4 \quad 3 \quad 0 \quad 1 \quad 1 \quad S(B)$
 $4 \quad 3 \quad 3 \quad 2 \quad 1 \quad 1 \quad S'$

R

2	1	1				
4	1	1	1	→		1
3	1	1	1	→		
4	1	1	1		1	
1	1					

5	4	3	0	1	1
4	3	3	2	1	1

$S(B)$

S'

R

2	1	1				
4	1	1	1	→		1
3	1	1	1	→		
4	1	1	1		1	
1	1					

5 4 3 0 1 1

4 3 3 2 1 1

$S(B)$

S'

2	1	1				
4	1	1		1		1
3	1	1		1		
4	1	1	1		1	
1	1					

5 4 1 2 1 1

4 3 3 2 1 1

$S(B)$

S'

R

2	1	1				
4	1	1	1	→		1
3	1	1	1	→		
4	1	1	1		1	
1	1					

5 4 3 0 1 1

4 3 3 2 1 1

 $S(B)$ S'

2	1	1				
4	1	1		1		1
3	1	1		1		
4	1	1	1		1	
1	1					

5 4 1 2 1 1

4 3 3 2 1 1

 $S(B)$ S' R

2	1	1	→			
4	1	1	→	1		1
3	1	1		1		
4	1	1	1		1	
1	1					

5 4 1 2 1 1

4 3 3 2 1 1

 $S(B)$ S'

R

2	1	1				
4	1	1	1			1
3	1	1	1			
4	1	1	1		1	
1	1					

5 4 3 0 1 1

 $S(B)$

4 3 3 2 1 1

 S'

2	1	1				
4	1	1		1		1
3	1	1		1		
4	1	1	1		1	
1	1					

5 4 1 2 1 1

 $S(B)$

4 3 3 2 1 1

 S' R

2	1	1				
4	1	1		1		1
3	1	1		1		
4	1	1	1		1	
1	1					

5 4 1 2 1 1

 $S(B)$

4 3 3 2 1 1

 S'

2	1		1			
4	1		1	1		1
3	1	1		1		
4	1	1	1		1	
1	1					

5 2 3 2 1 1

 $S(B)$

4 3 3 2 1 1

 S'

R

2	1	1				
4	1	1	1	→		1
3	1	1	1	→		
4	1	1	1		1	
1	1					

5 4 3 0 1 1

4 3 3 2 1 1

 $S(B)$ S'

2

4

3

4

1

2	1	1				
4	1	1		1		1
3	1	1		1		
4	1	1	1		1	
1	1					

5 4 1 2 1 1

4 3 3 2 1 1

 $S(B)$ S' R

2	1	1	→			
4	1	1	→	1		1
3	1	1		1		
4	1	1	1		1	
1	1					

5 4 1 2 1 1

4 3 3 2 1 1

 $S(B)$ S'

2

4

3

4

1

2	1	→	1			
4	1		1	1		1
3	1	1		1		
4	1	1	1		1	
1	1					

5 2 3 2 1 1

4 3 3 2 1 1

 $S(B)$ S'

S'

R

2		1	1			
4	1		1	1		1
3	1	1		1		
4	1	1	1		1	
1	1					

4 3 3 2 1 1 $S(B)$

4 3 3 2 1 1 S'

R

2		1	1			
4	1		1	1		1
3	1	1		1		
4	1	1	1		1	
1	1					



4 3 3 2 1 1

4 3 3 2 1 1

$S(B)$

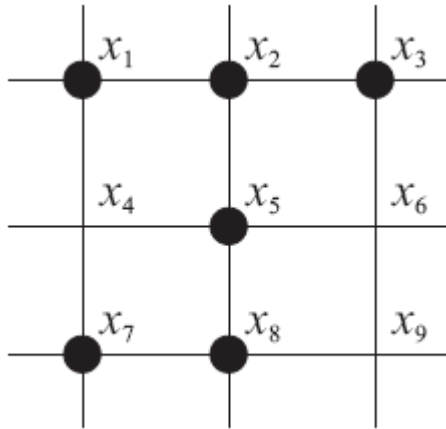
S'

2	1		1			
4		1	1	1		1
3	1	1		1		
4	1	1	1		1	
1		1				

3 4 3 2 1 1

S

Rekonstrukció egyenletrendszerrel



$$Ax = b \quad x \in \{0,1\}^{m \times n}$$

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 2 \\ 3 \\ 1 \end{pmatrix}$$

Rekonstrukció optimalizálással

Problémák az egyenletrendszer megoldása során:

- bináris változók
- nagy egyenletrendszer
- alulhatározottság (#egyenletek << #ismeretlenek)
- inkonzisztencia (pl.: zaj esetén)

$$x \in \{0,1\}^{m \times n}$$

$$C(x) = \|Px - b\|^2 + g(x) \rightarrow \min$$

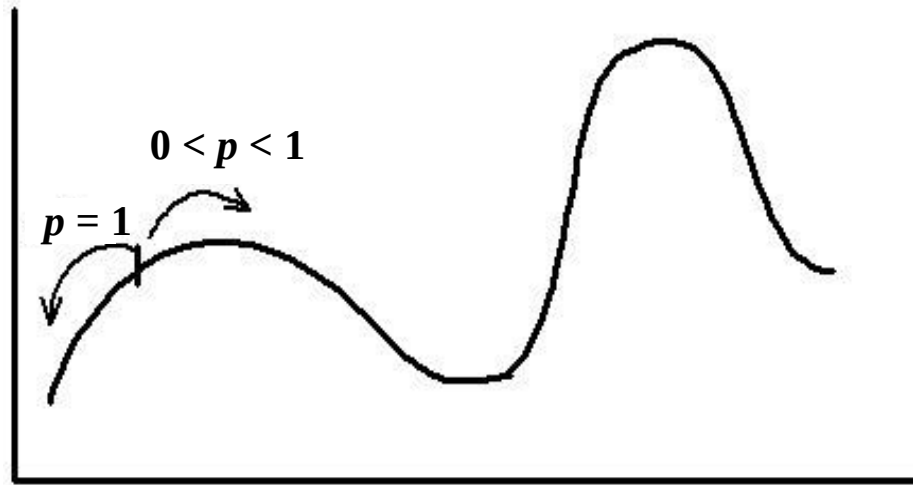


*Tag a prior
információ leírására:
konvexség, modell
képhez való
hasonlóság, stb.*

Megoldható kombinatorikus optimalizálási módszerekkel

Az optimalizálás megoldása

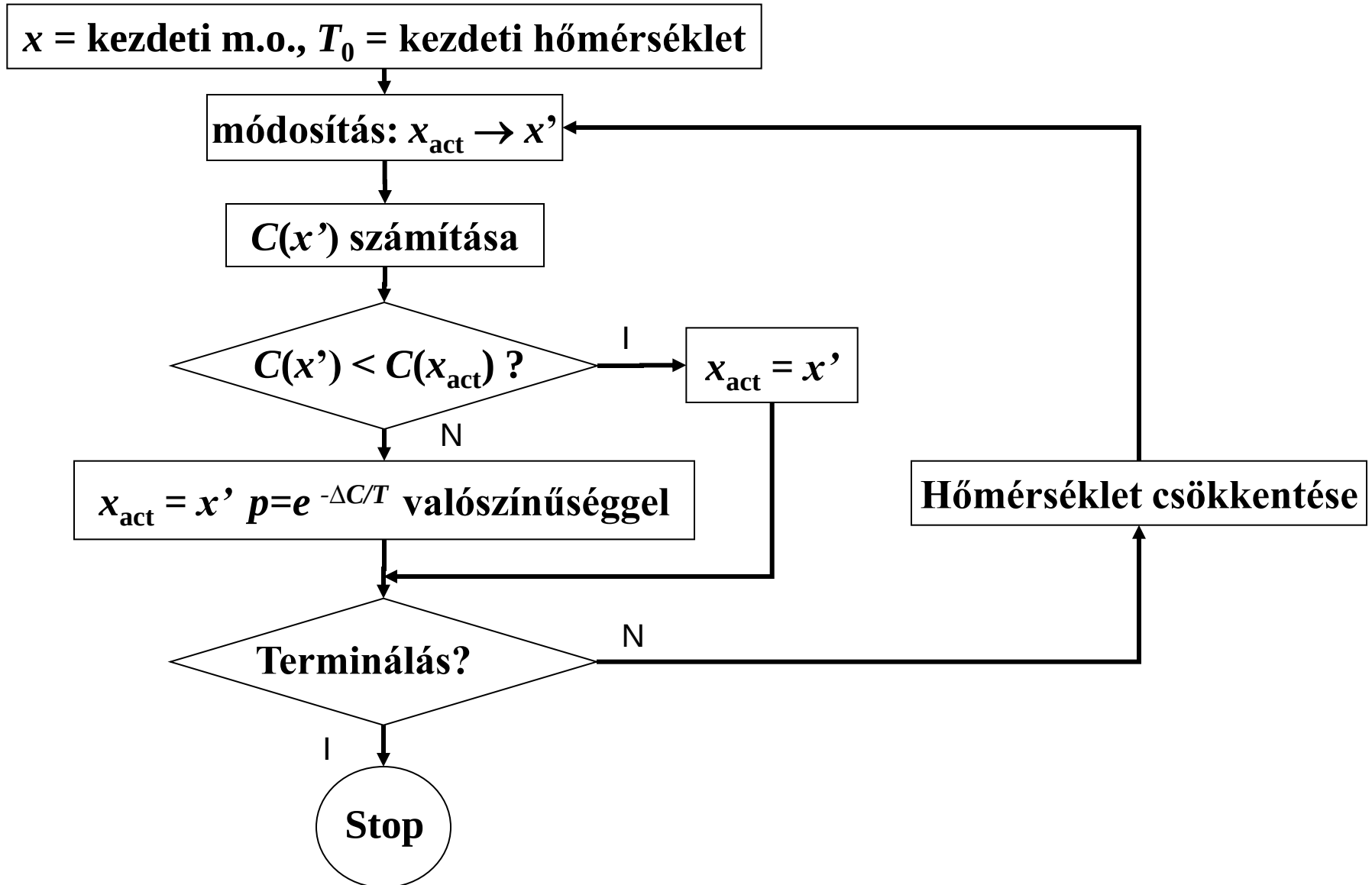
- **Probléma:** A klasszikus hegymászó algoritmusok lokális optimumba ragadhatnak
- **Ötlet:** Engedjük meg célfüggvényt növelő lépéseket is



Szimulált hűtés (SA)

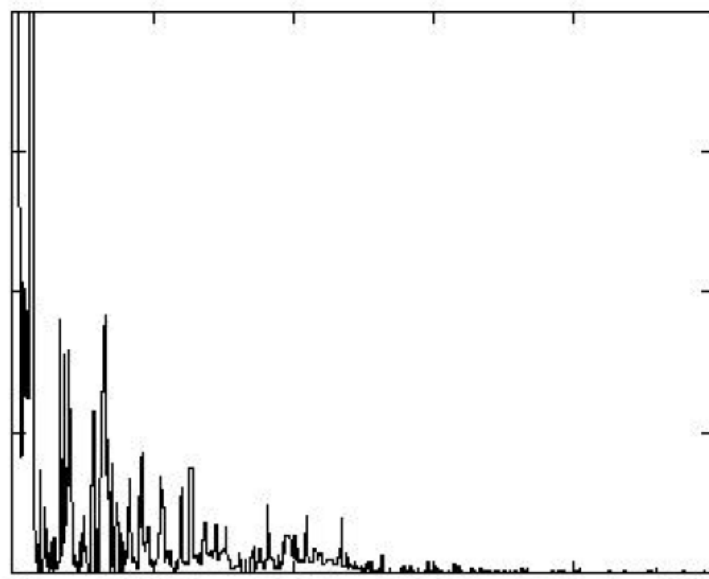
- Hűtés: termodinamikai folyamat, melynek során egy fém lehűl, majd megdermed
- A termikus zajnak köszönhetően a folyékony fém energiája a hűtés során néha növekszik
- A folyékony fém hűtési hőmérsékletét megfelelően szabályozva a fém minimális energiájú kristályrácsban dermed meg
- Szimulált hűtés: a fenti észrevételen alapuló véletlenszerű keresési technika

SA folyamata



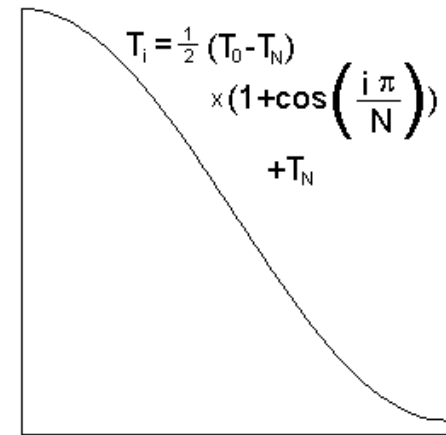
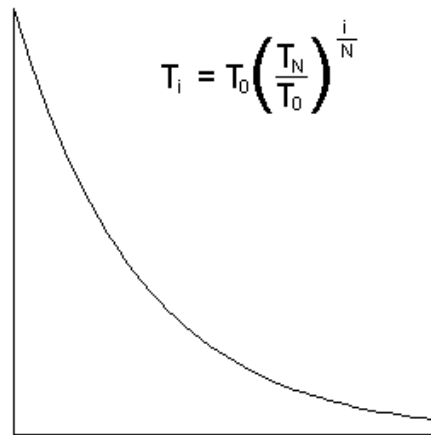
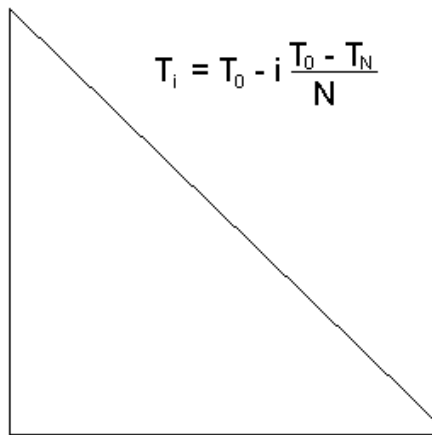
Az optimum megtalálása

- Az SA paramétereit megfelelően hangolva a globális optimum megtalálható
- A paraméterek adott optimalizálási problémára történő finom-hangolása nehézkes



Az SA paramétere

- Kezdeti hőmérséklet: T_0
- Megállási feltétel: pl.: T_N vagy ha tartósan nem találunk jobb megoldást
- Hűtési karakterisztika



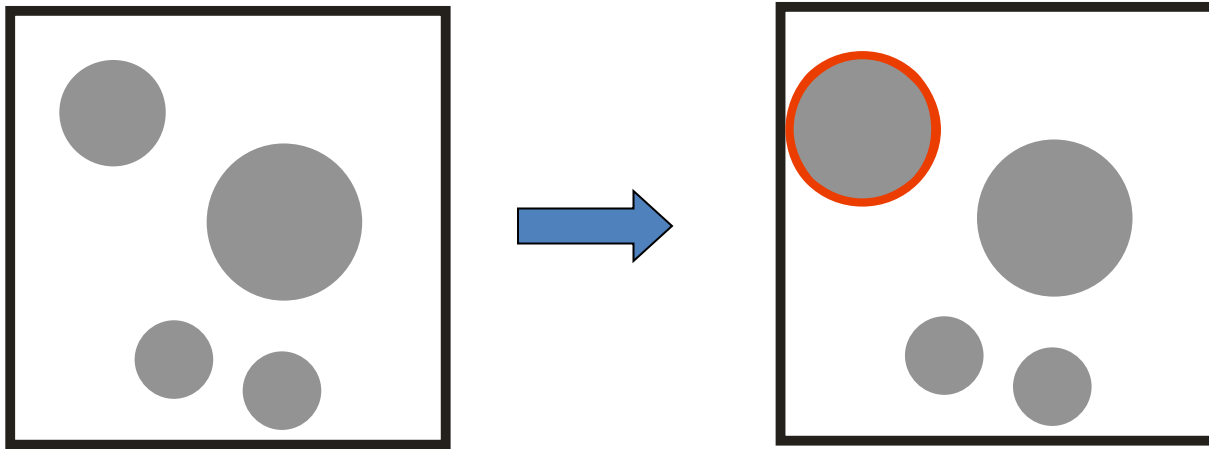
SA - Pixel alapú rekonstrukció

- A bináris képet bináris mátrix írja le
- Egy (több) véletlenszerűen választott pozícióban invertáljuk az értéket

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 1 & \boxed{1} \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

SA – Paraméter alapú rekonstrukció

- A bináris kép geometriai objektumok paramétereivel van leírva, pl. (x,y,r)
- Véletlenszerűen változtassuk az objektum(ok) paraméterét (-eit)



$[(16,53,17), (44,35,25), (26,13,12), (43,8,12)] \rightarrow [(13,50,23), (44,35,25), (26,13,12), (43,8,12)]$

Az SA hátrányai

- Nehézen paraméterezhető
- Lassú
- Egy adott rekonstrukciós feladatnak sok lokális optimuma lehet, és ezek között nagy „ellenkező irányú” lépéseket kell tenni
- A szimulált hűtés csak kis lépéseket enged meg, szükség lenne a nagy változtatást biztosító lépésekre is

Evolúciós algoritmusok (EA)

- Evolúció: Egy rendszer fokozatos fejlődése és átalakulása.
- Általános elv: élő szervezetek, tárgyak (autók, mobiltelefonok, ...), absztrakt dolgok (szoftverek, ...)
- Evolúciós algoritmusok: a fenti észrevételen alapuló véletlenszerű keresési technikák

Operátorok

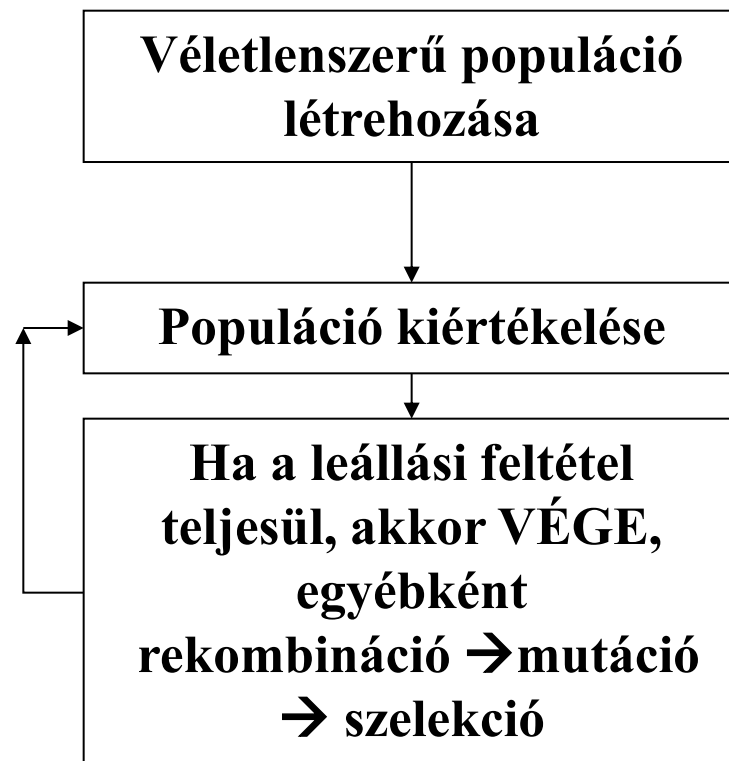
- **Mutáció:** a szülő tulajdonságai részben módosulva jelennek meg az utódban:
 $ABCDEF \rightarrow AB\textcolor{red}{E}DE\textcolor{red}{G}$
- **Rekombináció:** a szülők tulajdonságai keveredve jelennek meg az utódban:
 $ABCDEF + \textcolor{red}{DCCDFE} \rightarrow AB\textcolor{red}{C}D\textcolor{red}{F}E$
- **Szelekció:** csak az “életképes” egyedek maradnak meg

Optimalizálás EA segítségével

- Egyedek: a probléma lehetséges megoldásai
- P - a kezdetben rendelkezésre álló egyedek (kiindulási populáció) (véletlenszerűen “legyártott” egyedekből is állhat)
- f - *fitness* $fv.$: minden egyedhez hozzárendel egy “jósági” értéket, mely az optimális megoldástól való távolságot jelzi
- *Szelekció*: válasszuk ki P -ből a legjobb fitness-értékkel rendelkező egyedeket
- *Rekombináció*: képezzünk a kiválasztott egyedekből új egyedeket a keresztezés operátor segítségével
- *Mutáció*: a mutáció operátor segítségével módosítsunk néhány egyedet

EA folyamata

- **leállási feltétel:** megadott számú iteráció óta nem változik a legjobb fitness-érték, megtörtént megadott fix számú iteráció, stb.



EA előnyei és hátrányai

- Előnyök:
 - Egyszerűen kódolható
 - Széleskörűen alkalmazható (nagy keresési térben is működik és kevés információt igényel a célfüggvényről)
 - Zajos fitnessfüggvény esetén is működik
- Hátrányok
 - Lokális optimumba ragadhat
 - Néha nehéz a probléma reprezentálása
 - Paraméterezés nehéz lehet

A legegyszerűbb lenne...

- Egyedek: $\{0,1\}^{m \times n}$ elemei
- Pontszerű crossover vagy uniform crossover
- Nagyon „elromlanak” a vetületek keresztezés során
- Nem őrződik meg a kép struktúrája

Rekonstrukció modellképpel

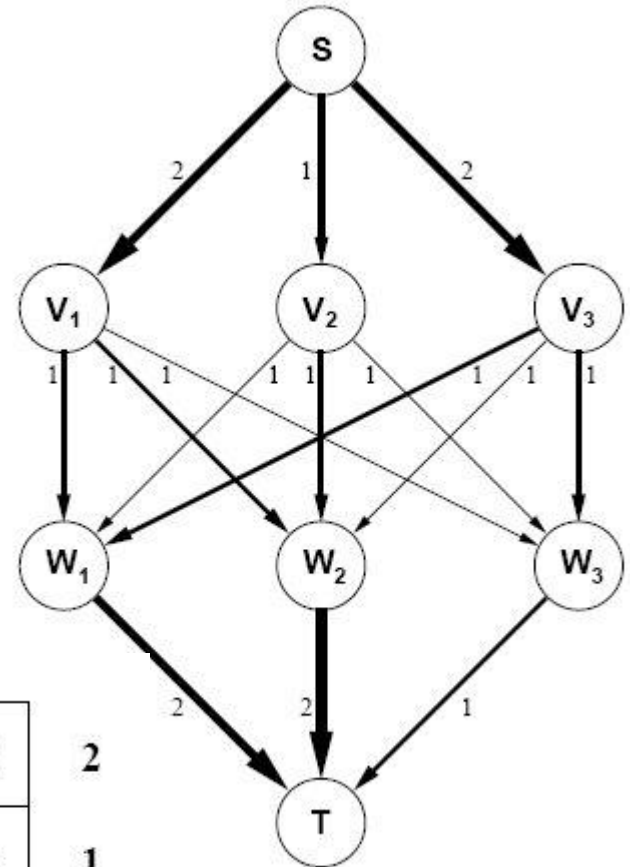
- $\mathcal{A}(R,S)$: azon mátrixok osztálya, melyek sorvetülete R , oszlopvetülete S vektorral egyezik meg
- Egy adott M mátrixhoz való hasonlóság: olyan $A \in \mathcal{A}(R,S)$ -t keresünk, melyre minimális

$$\sum_{i=1}^m \sum_{j=1}^n |m_{ij} - a_{ij}|$$

Rekonstrukció folyamproblémaként

- Maximális folyam = adott vetületepárt kielégítő rekonstrukció
- $c_{ij} = -m_{ij}$ költségekkel az éleken – minimális költségű maximális folyam: adott vetületepárt kielégítő M -mel legtöbb közös 1-est tartalmazó rekonstrukció

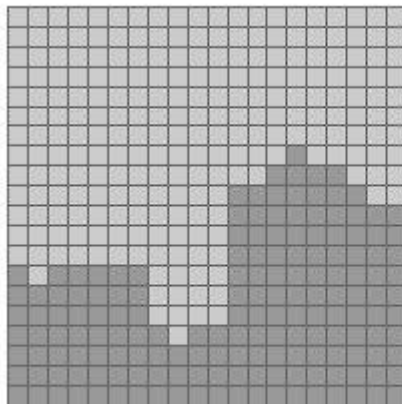
1	1	0	2
0	1	0	1
1	0	1	2
2	2	1	



Rekombináció (crossover)

2-2 kvadránsban az egyik ill. a másik szülő maszkjának kezdeménye (1-1 pont)

```
while (there are border cells) do
begin
  randomly select a border cell  $(x, y)$ ;
  mark all unassigned neighbouring cells of  $(x, y)$  as border cells;
  assign these cells to the same parent as  $(x, y)$ ;
  remove the border status of  $(x, y)$ ;
end
```



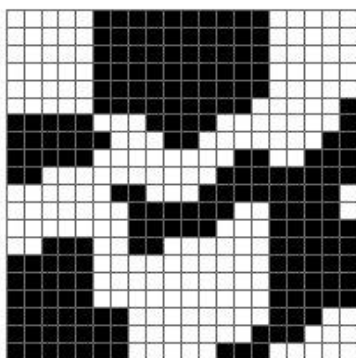
first parent
second parent

$$m_{ij} = \begin{cases} p_{ij} & \text{if } x_{ij} = 0 \\ q_{ij} & \text{if } x_{ij} = 1 \end{cases}$$

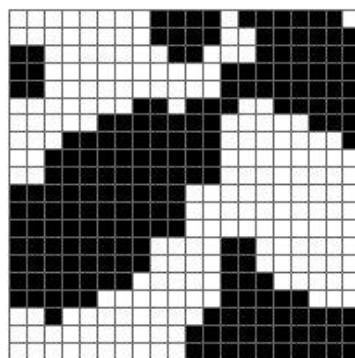
c. crossover mask

Hegymászás kapcsolásokkal

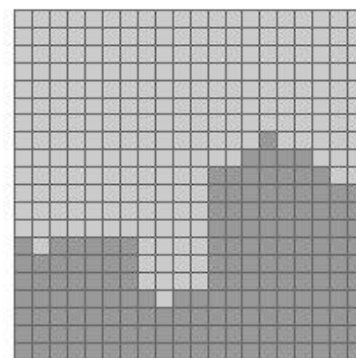
- Kiértékelőfv.: szomszédos fekete pixelek száma



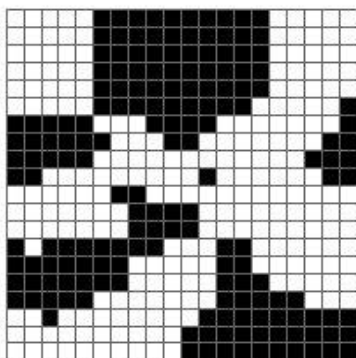
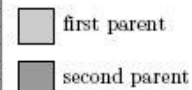
a. first parent



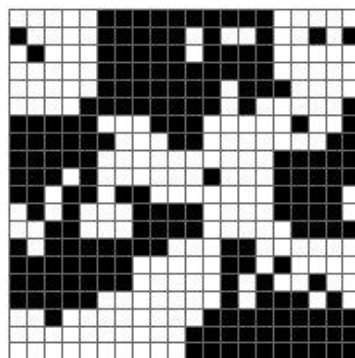
b. second parent



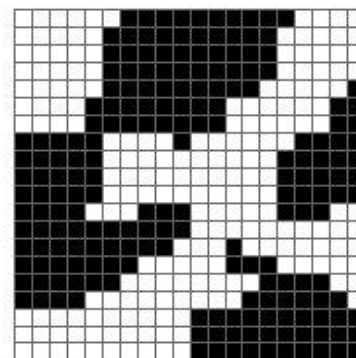
c. crossover mask



d. model image



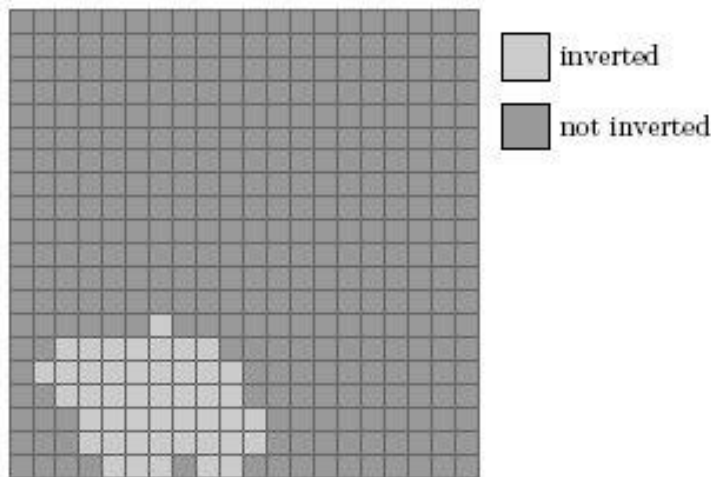
e. child before hillclimb



f. child after hillclimb

Mutáció

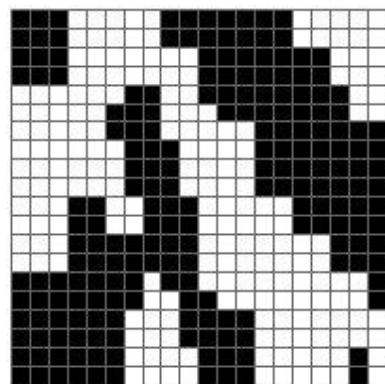
- Véletlen mező és véletlen $k \in [k_{\min}, k_{\max}]$ választása
- k -szor végrehajtjuk a határ bővítését



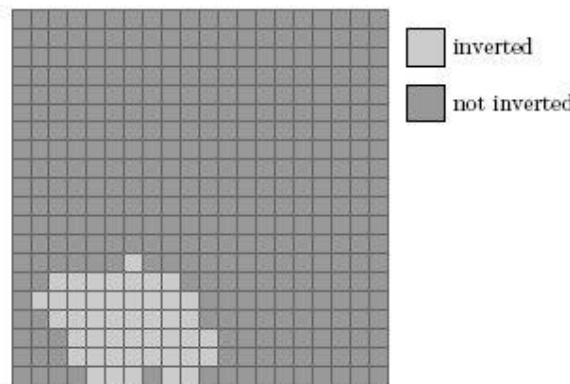
b. mutation mask

$$m_{ij} = \begin{cases} p_{ij} & \text{if } x_{ij} = 0 \\ \text{random} & \text{if } x_{ij} = 1 \end{cases}$$

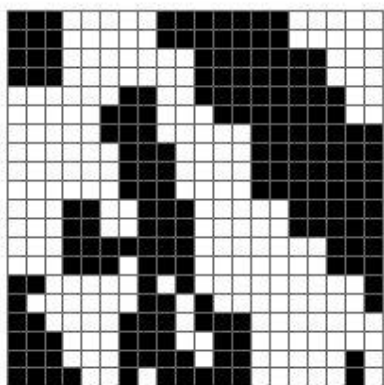
Hegymászás kapcsolásokkal



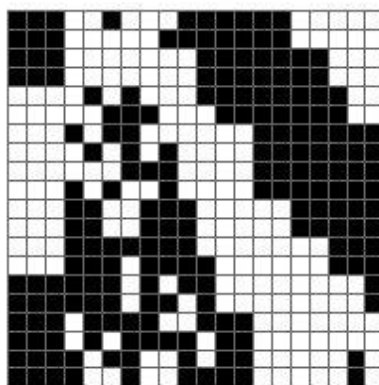
a. parent



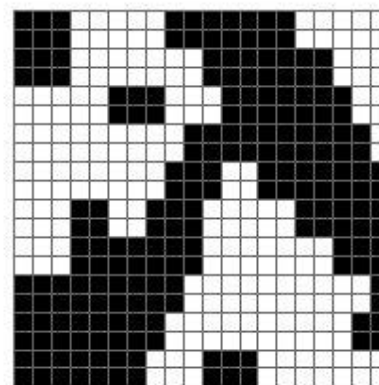
b. mutation mask



c. model image



d. child before hillclimb

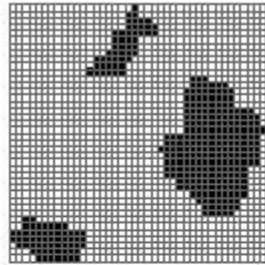


e. child after hillclimb

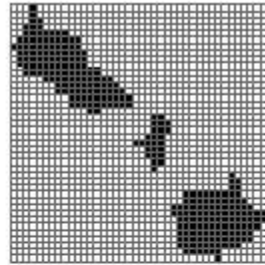
Az evolúciós algoritmus

```
generate initial population  $P_0$  of size  $\lambda$ , consisting of matrices in  $\mathcal{A}(R, S)$ ;  
perform a hillclimb operation on each matrix in  $P_0$ ;  
 $t := 0$ ;  
repeat  
   $P'_t = \emptyset$ ;  
  for  $i := 1$  to  $\mu$  do  
    begin  
      generate a child matrix  $C$ , by crossover or mutation;  
      perform a hillclimb operation on  $C$ ;  
       $P'_t := P'_t \cup \{C\}$ ;  
    end;  
  select new population  $P_{t+1}$  from  $P_t \cup P'_t$ ;  
   $t := t + 1$ ;  
until (stop criterium has been met);  
output the best individual found;
```

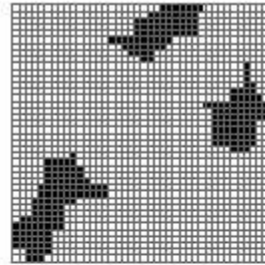
Eredmények: *hv*-konvex osztály



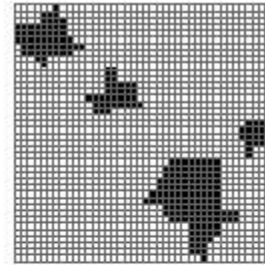
(1)



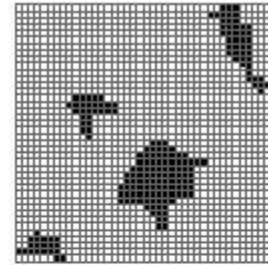
(2)



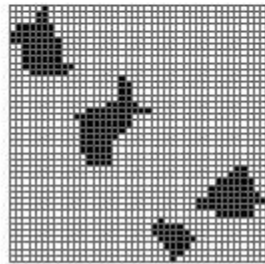
(3)



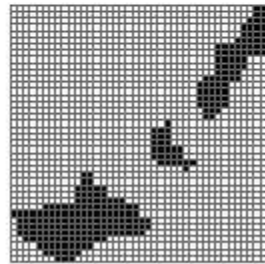
(4)



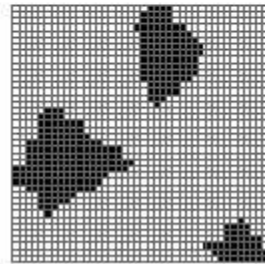
(5)



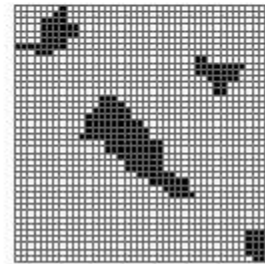
(6)



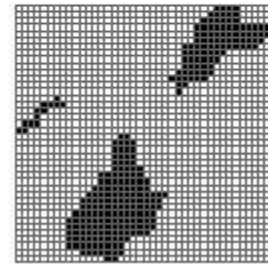
(7)



(8)



(9)

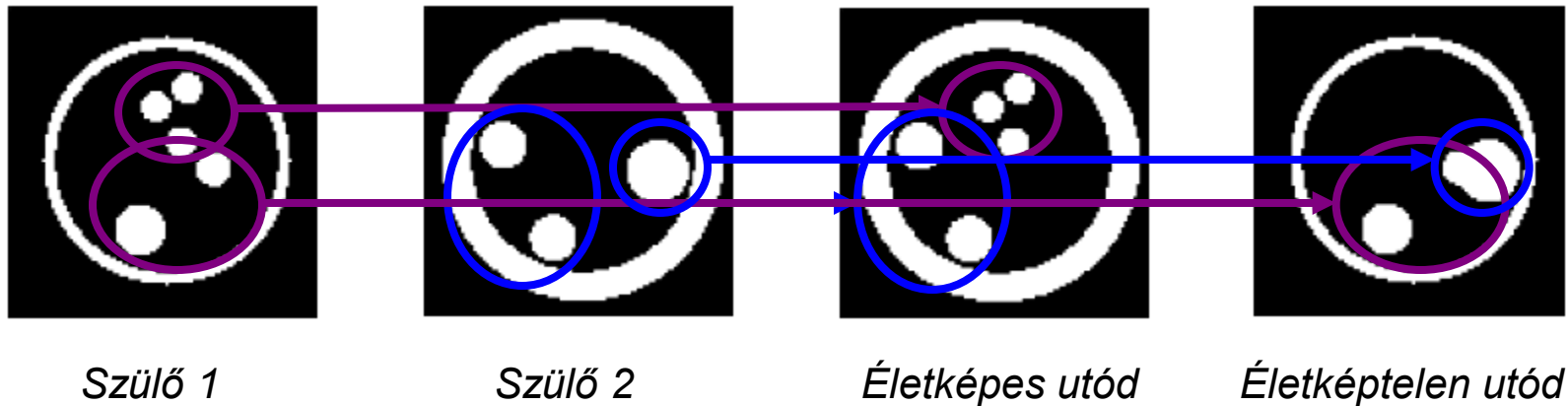


(10)

image	1	2	3	4	5	6	7	8	9	10
#perfect	9	5	8	8	9	6	6	6	6	9
avg.time (m)	25.6	16.6	15.5	8.8	8.5	10.8	17.7	17.7	8.8	15.7
avg. #gen.	24.6	17.3	18.5	14.8	15.8	18.5	30.2	15.0	21.0	23.6

Objektum-alapú genetikus rekonstrukció

- Crossover



- Mutáció

- Körlapok számának növelése/csökkentése
- Körlap középpontjának mozgatása
- Körlap átméretezése