Alkalmazott mesterséges intelligencia (AMI)

http://www.mit.bme.hu/oktatas/targyak/vimibb01

9. ea. (2019 ősz)

Kényszerkielégítési problémák
(Constraint Satisfaction Problem, CSP)

http://http://mialmanach.mit.bme.hu/aima/ch05 5. fejezet

Előadó: Pataki Béla

a fóliák
Dobrowiecki Tadeusz és
Hullám Gábor anyagainak
felhasználásával készültek

BME I.E. 414, 463-26-79
pataki@mit.bme.hu,
http://www.mit.bme.hu/general/staff/pataki

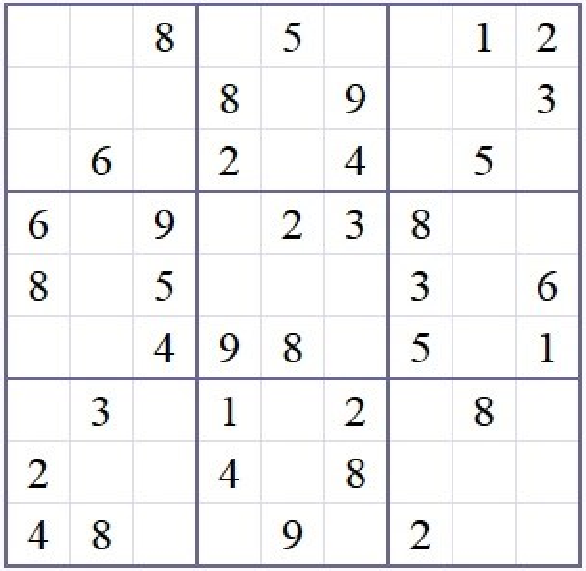
(A keresés témaköréhez tartozik)

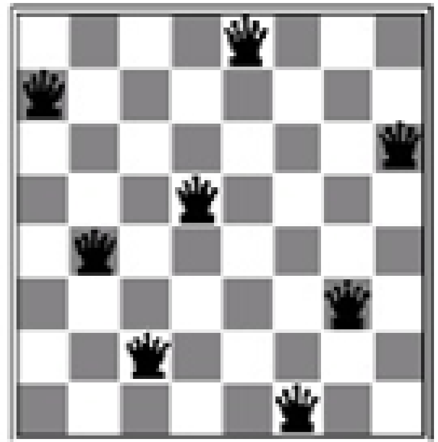
Kényszerkielégítési (korlátkielégítési) problémák Constraint Satisfaction Problems (CSP):

Állapot: Az állapot a leíró változók és a hozzájuk rendelt értékek által definiált. Az x_k változók egy-egy D_k érték-tartományból (halmazból) veszik fel értékeiket .

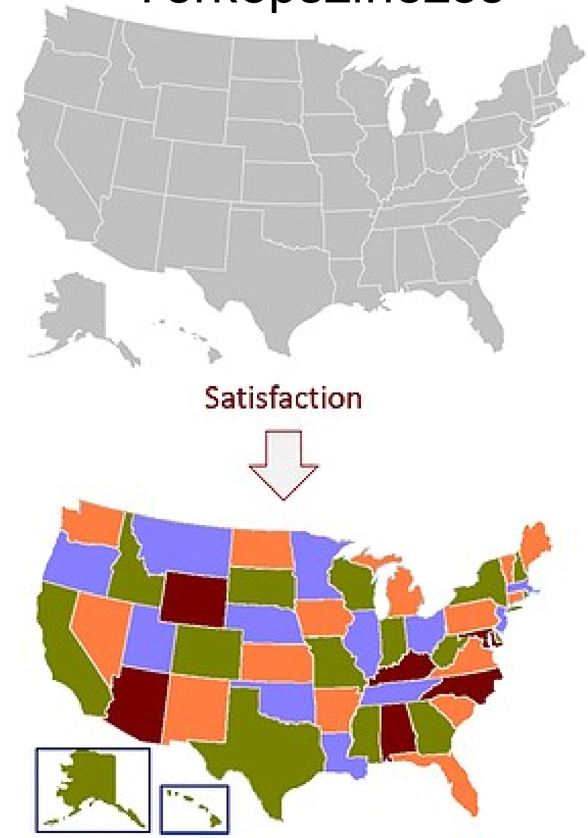
Célállapotteszt:

1. Az összes állapotot leíró változóhoz rendeltünk *számára megengedett* értéket
2. Az adott korlátok teljes halmazát kielégítettük





Térképszínezés




```
   T W O
+  T W O
--------
 F O U R
```

Órarend készítés:

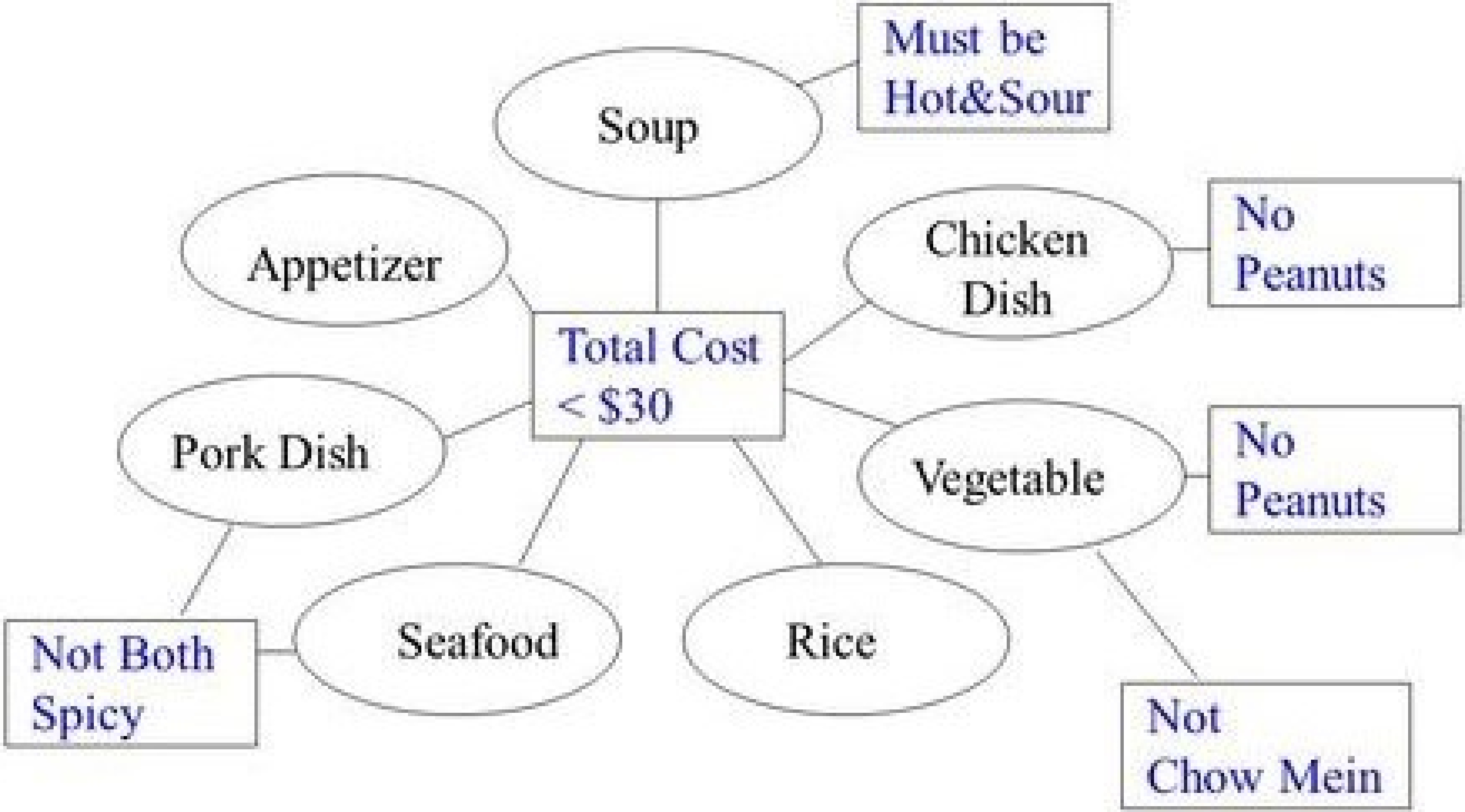
Szerda 14-16, Pataki Béla,MI-BSc5szem, Q-I

…

1. *egyszerre egy teremben csak egy évfolyam*
2. *egyszerre egy teremben csak egy oktató,*
3. *senki se lehet egyszerre két helyen*
4. *egy évfolyam egyszerre csak egy helyen lehet (?)*

- stb.

Menüösszeállítás az étlap alapján



- **Hozzárendelési problémák**, pl. ki, mit, hol tanítson
- **Menetrendi problémák**, pl. az egyetemi órarend/ teremfoglalás
- **Szállításütemezés**
- **Gyári ütemezés**

Például **Raktár-tervezési probléma**

Egy szupermarket-lánc raktárakat szeretne telepíteni a bolti hálózat kiszolgálására. Mit tudunk?

- L_1, ..., L_n lokáció, ahol raktár megépíthető.
- Csak K db. raktárra van szükség ($K < n$).
- Minden lokációra jellemző CP_i kapacitás: hány boltot képes kiszolgálni
- Minden bolthoz rendelni kell egy raktárt.
- S_j bolt ellátása L_i lokációból $P_{i,j}$ pénzbe kerül.
- Az összköltséget TC konstans alatt kell tartani.

CSP problémák típusai

Változók alapján:

Diszkrét változók

véges értéktartományok:

pl. Boole-típusú (IGEN/NEM)

végtelen értéktartományok:

egész számok, füzérek, stb.

pl. job scheduling, változók a munkaszakaszok kezdete/vége

Folytonos változók

megfigyeléseket határoló időpontok,

fizikai állapotváltozók

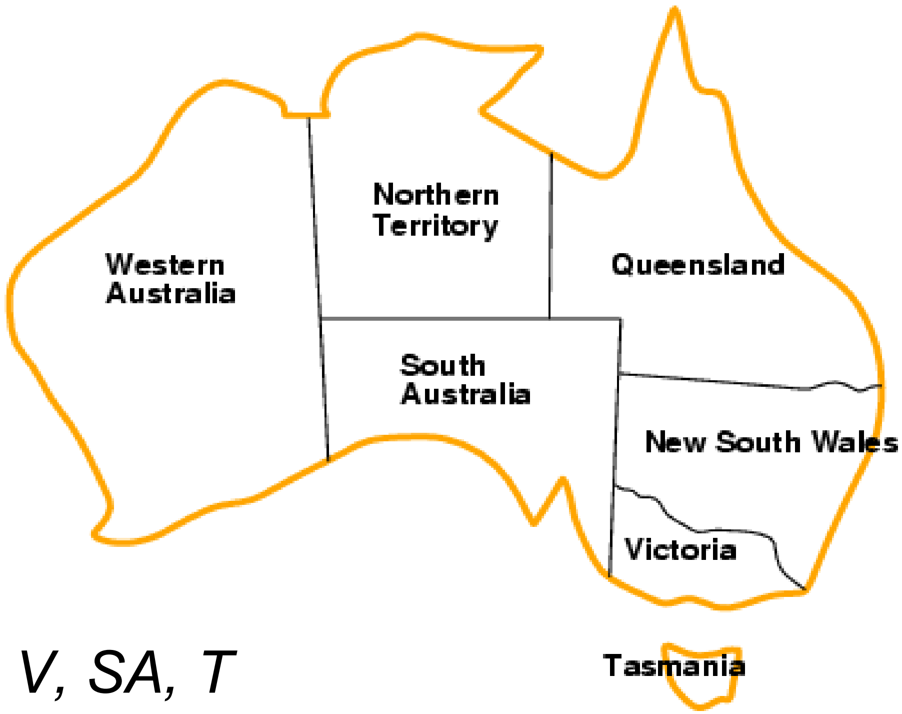
***Kényszerek, korlátok alapján*:**

Unáris korlát: egyetlen egy változóra vonatkozik, pl. SA ≠ green, x<5

Bináris korlát: két változó viszonyára vonatkozik, pl. SA ≠ WA, x<y

Magasabb-rendű korlát: 3 vagy több változó viszonyára vonatkozik, (pl. oszloponkénti változó korlátok kriptoaritmetikai feladványokban)

Példa: Térképszínezés



Változók: *WA, NT, Q, NSW, V, SA, T*

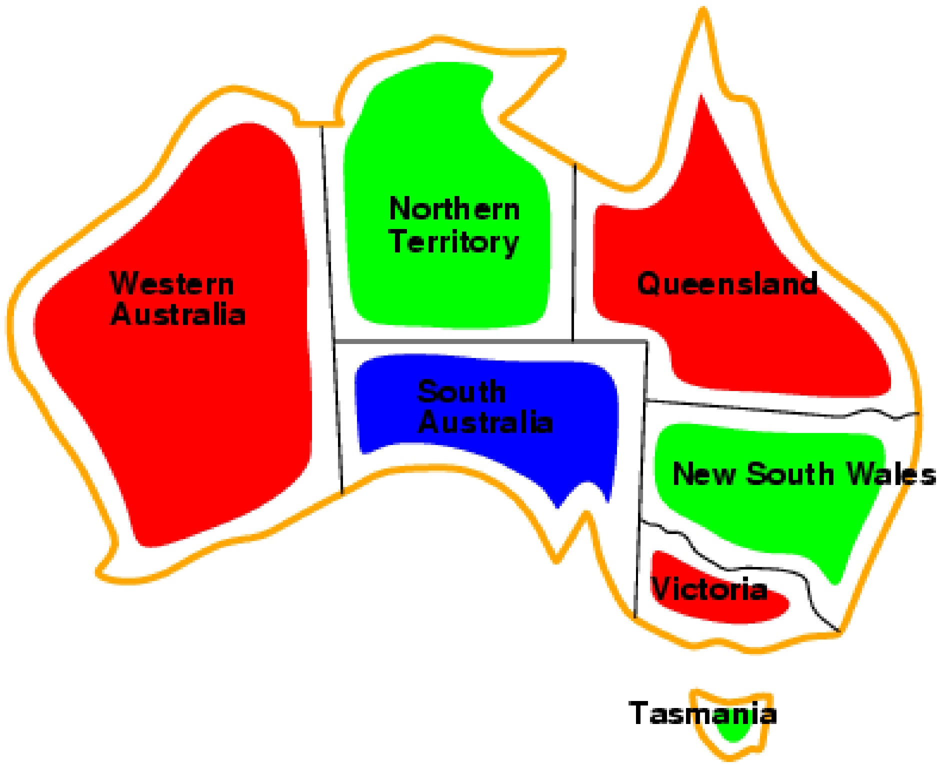
Értéktartományok D_i = {red, green, blue}

Korlátok: szomszédos területek színe legyen eltérő

pl. WA ≠ NT, ill. más megfogalmazásban: (WA,NT) értékét csakis a {(red, green), (red, blue), (green, red), (green, blue), (blue, red), (blue, green)} halmazból vehet fel.

Lehetséges-e?

pl. Térképszínezés

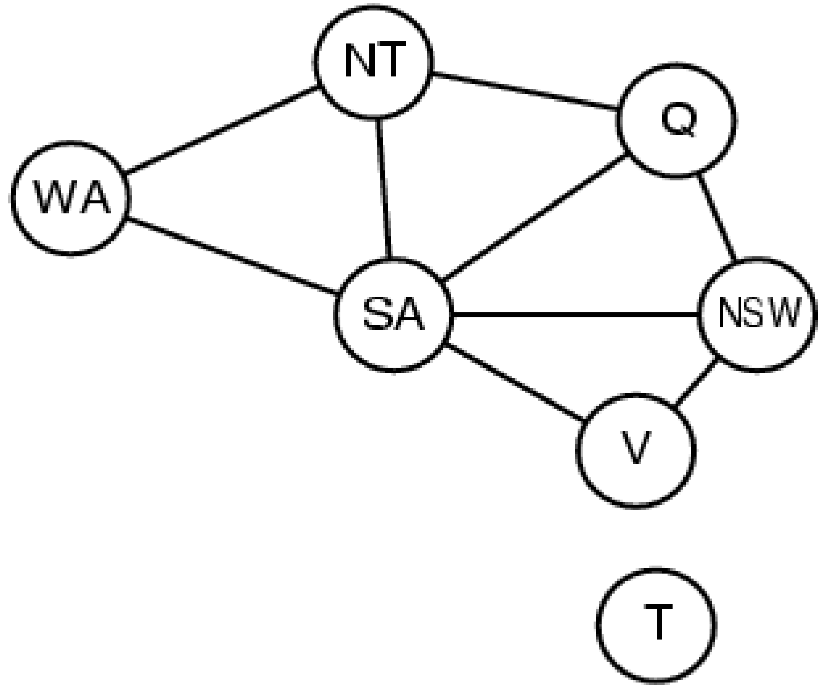


Megoldás: **teljes** és **konzisztens** változó-érték hozzárendelés

pl. WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = green

(T lehet akármi, mert nem határos senkivel - különben is több megoldás van)

Korlátok gráfja



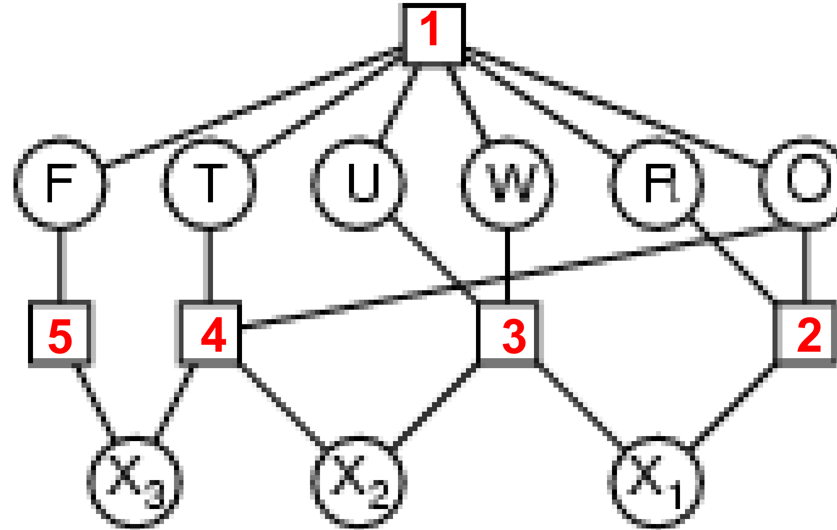
korlátgráf: csomópontjai a változók és élei a korlátok

Itt csak **bináris CSP**-k: egy-egy korlát 2 változót köt össze

…

Másik példa: kriptoaritmetika

```
    T W O
+   T W O
---------
  F O U R
```



Változók: $F \;\; T \;\; U \;\; W \;\; R \;\; O \;\; X_1 \;\; X_2 \;\; X_3$

Értéktartományok: {*0,1,2,3,4,5,6,7,8,9*}, {*0,1*}

Korlátok:

1. *Mind-eltérő(F, T, U, W, R, O)*
2. $O + O = R + 10 \cdot X_1$
3. $X_1 + W + W = U + 10 \cdot X_2$
4. $X_2 + T + T = O + 10 \cdot X_3$
5. $X_3 = F$, $T \neq 0$, $F \neq 0$

A probléma megfogalmazása

Kezdeti állapot: összes változó-hozzárendelés üres { }

Operátor: értéket hozzárendelni egy még nem lekötött változóhoz úgy, hogy az eddigi hozzárendelésekkel ne ütközzön (egyik kényszer se sérüljön)

→ kudarc, ha nincs megengedett hozzárendelés

Célállapotteszt: ha az aktuális hozzárendelés teljes és mindegyik kényszer teljesül

Keresési fa

n db változó esetén minden megoldás n mélységben fekszik
→ mi van, ha a **szélességi keresést** használjuk?

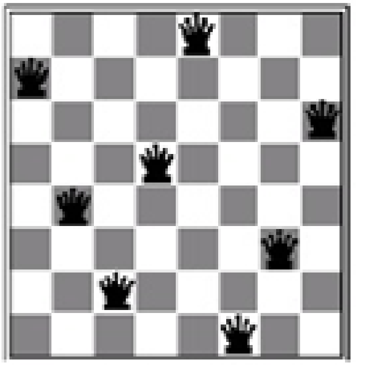
Elágazások száma L mélységben ($L=0,1,2,\dots$), ha minden változó d számú értéket vehet fel (ez a változó értékkészlete v. más néven doménje)

$b = (n - L)\, d$, (mert L változó már értéket kapott)

azaz $\boldsymbol{n!\cdot d^n}$ levélcsomópont (miközben $\boldsymbol{d^n}$ lehetséges hozzárendelés van)

(pl. 8 betűs számjegyaritmetika, n=8, d=10, levelek száma $4 \cdot 10^{12}$)

Megfogalmazás (modell) hatása a problémamegoldásra



Az n-királynő probléma tanulsága: hasonlítsunk össze három modellt

1. modell, a változók: x_{ij} (*a sakktáblamezők pozíciója*)
értékkészlet: {0, 1} (*van rajta királynő vagy nincs*)
kényszerek (sorok, oszlopok, átlók)

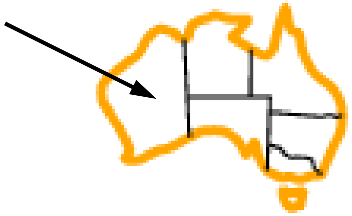
2. modell, a változók: $x_1, \ldots, x_n$ (*egy királynő által elfoglalt mező pozicíója*)
értékkészlet: $\{0, 1, 2, \ldots, n^2-1\}$ (*mezőindex*)
kényszerek (sorok, oszlopok, átlók)

3. modell, a változók: $x_1, \ldots, x_n$ (az 1., 2. stb. sorban álló *királynő oszlopindexe*)
értékkészlet: $\{1, 2, \ldots, n\}$ (*oszlop-index*)
kényszerek (sorok, oszlopok, átlók)

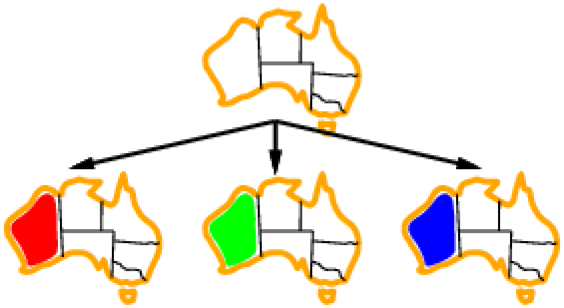
n x *n*-es sakktáblán az *n*-királynő probléma néhány *n*-re

modell	változó	értékkészl.(d)	levelek sz.(d^n)	n = 4	n = 8	n = 20
1.	n^2	2	$(2)^{(n^2)}$	$6{,}6 \cdot 10^4$	$1{,}8 \cdot 10^{19}$	$2{,}6 \cdot 10^{120}$
2.	n	n^2	$(n^2)^n$	$6{,}6 \cdot 10^4$	$2{,}8 \cdot 10^{14}$	$1{,}1 \cdot 10^{52}$
3.	n	n	n^n	256	$1{,}6 \cdot 10^7$	$1{,}0 \cdot 10^{26}$

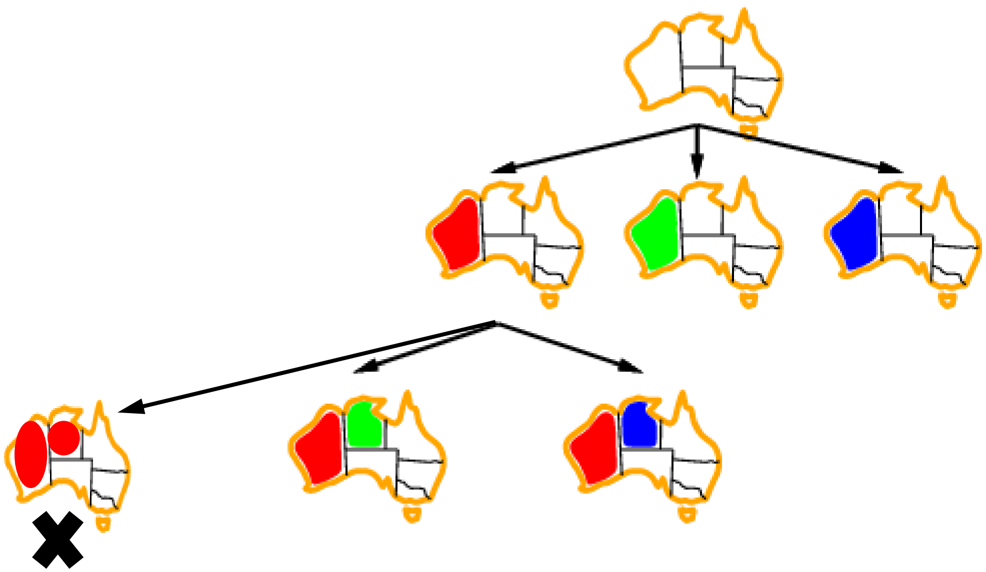
Visszalépéses keresés



Visszalépéses keresés

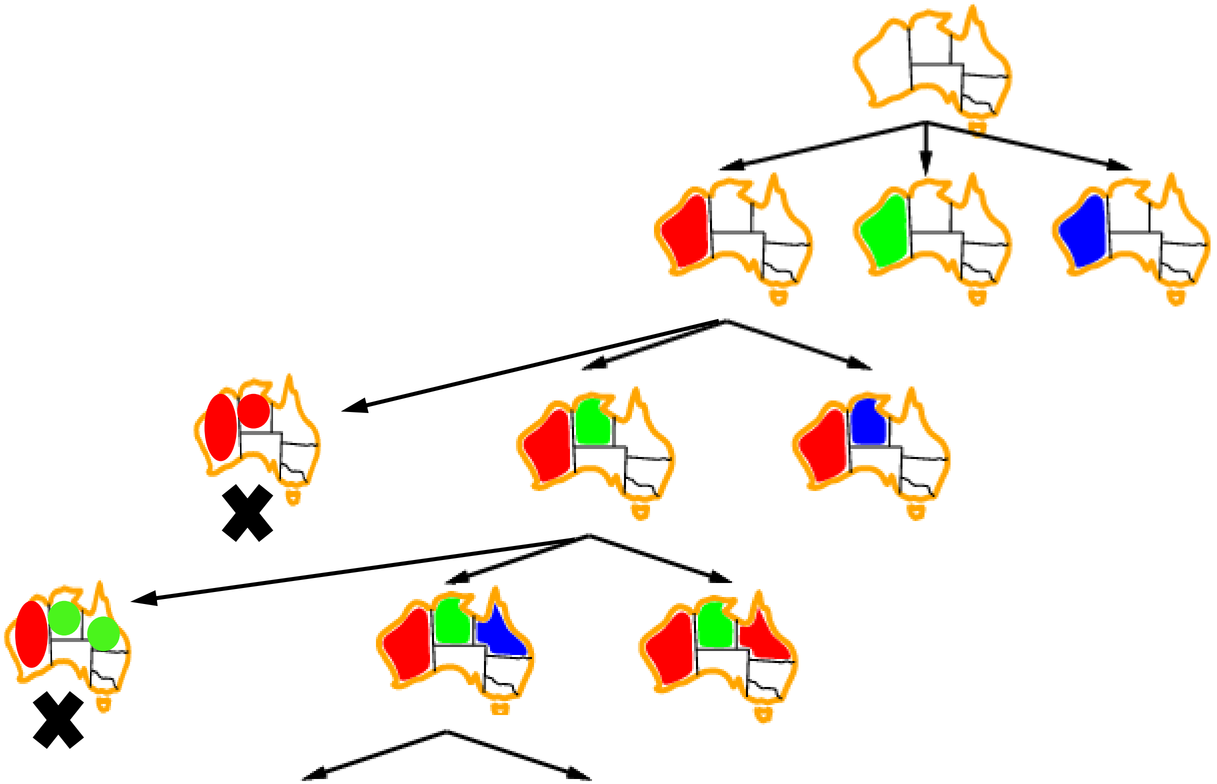


Visszalépéses keresés



Visszalépéses keresés

kvíz következik



09.1. kvíz

Milyen típusú keresés a visszalépéses CSP keresés?

A. Mélységi keresés

B. Szélességi keresés

C. Mohó keresés

D. A* keresés

Keresés

Változó-hozzárendelés **kommutatív**, *azaz* például

(WA = red) majd (NT = green)

ugyanaz, mint (NT = green) majd (WA = red)

Egy-egy csomópontban csakis egyetlen egy változó

hozzárendelése történhet meg, így:

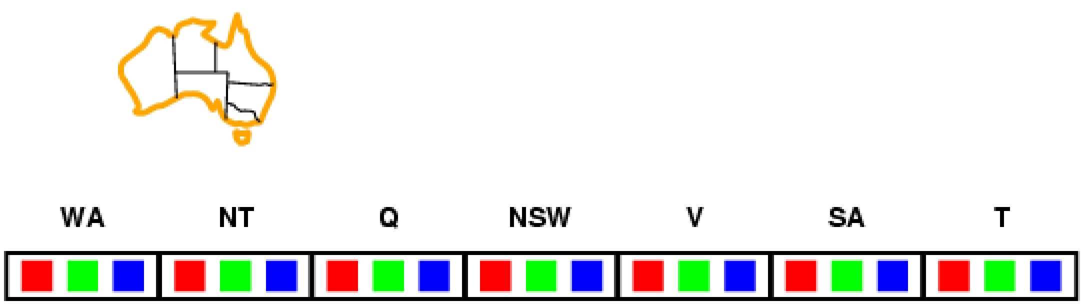
→ $b = d$ és a fának $d^{\,n}$ levele van

****

Alapvető nem informált algoritmus (keresés) CSP problémák megoldására: **visszalépéses keresés**, azaz **mélységi keresés** minden szinten egyetlen egy változó-hozzárendeléssel - ha sérül valamelyik kényszer, visszalép (egyszer sérült kényszer mélyebben nem jöhet helyre)

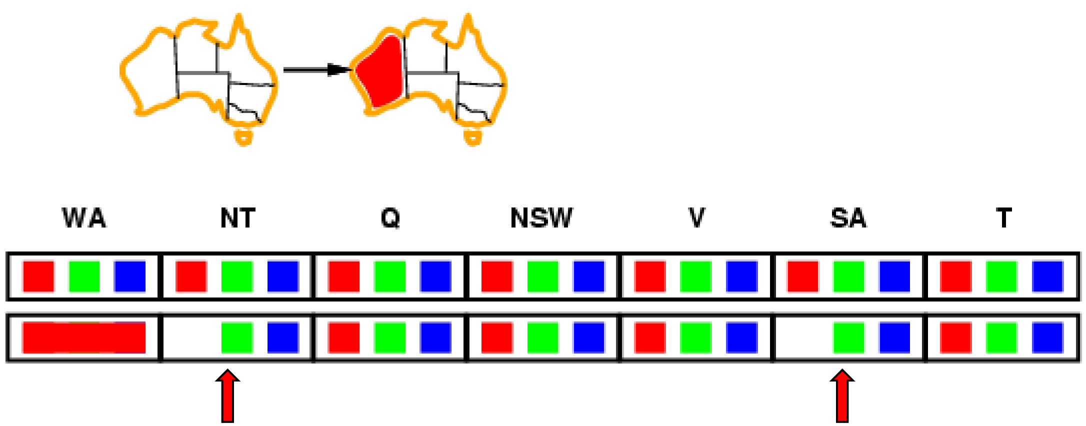
Előretekintő ellenőrzés/1 (keresés)

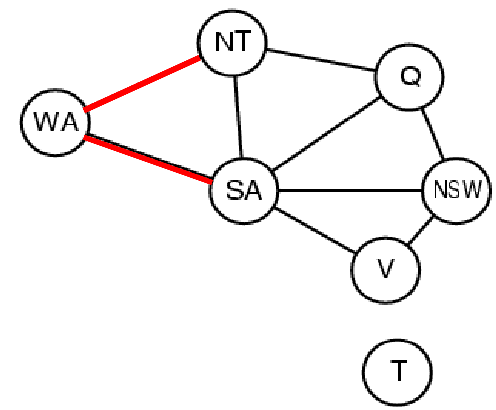
Az előretekintő ellenőrzés minden egyes alkalommal, amikor egy X változó értéket kap, minden, az X-hez kényszerrel kötött, lekötetlen Y-t megvizsgál, és Y tartományából törli az X számára választott értékkel inkonzisztens értékeket.



Előretekintő ellenőrzés/2 (keresés)

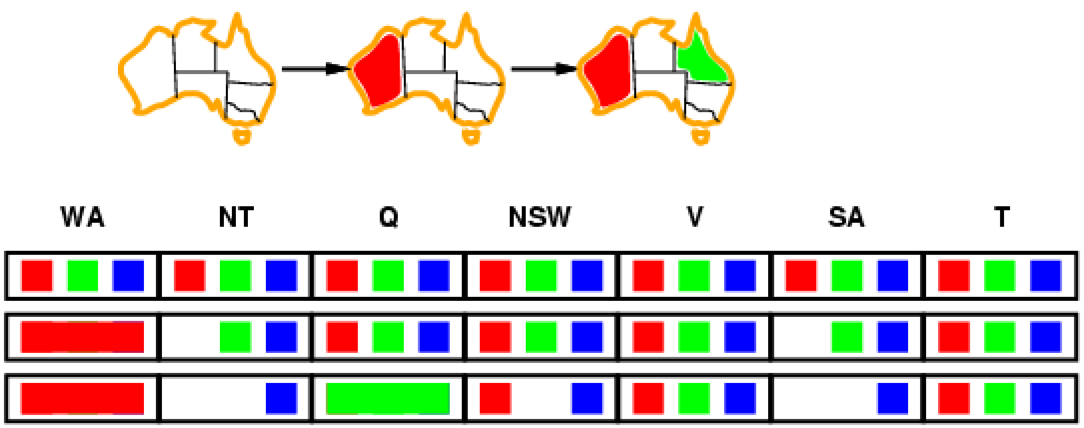
Az előretekintő keresés minden egyes alkalommal, amikor egy X változó értéket kap, minden, az X-hez kényszerrel kötött, lekötetlen Y-t megvizsgál, és Y tartományából törli az X számára választott értékkel inkonzisztens értékeket.





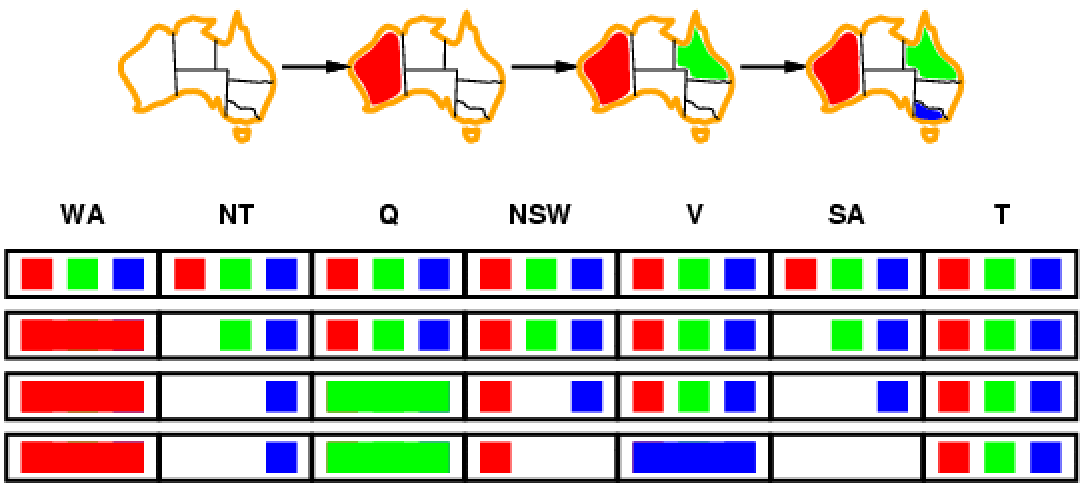
Előretekintő ellenőrzés/3 (keresés)

Az előrenéző ellenőrzés minden egyes alkalommal, amikor egy *X* változó értéket kap, minden, az *X*-hez kényszerrel kötött, még nem hozzárendelt *Y*-t megvizsgál, és *Y* lehetséges értékei közül törli az X számára választott értékkel inkonzisztens értékeket.



Előretekintő ellenőrzés/4 (keresés)

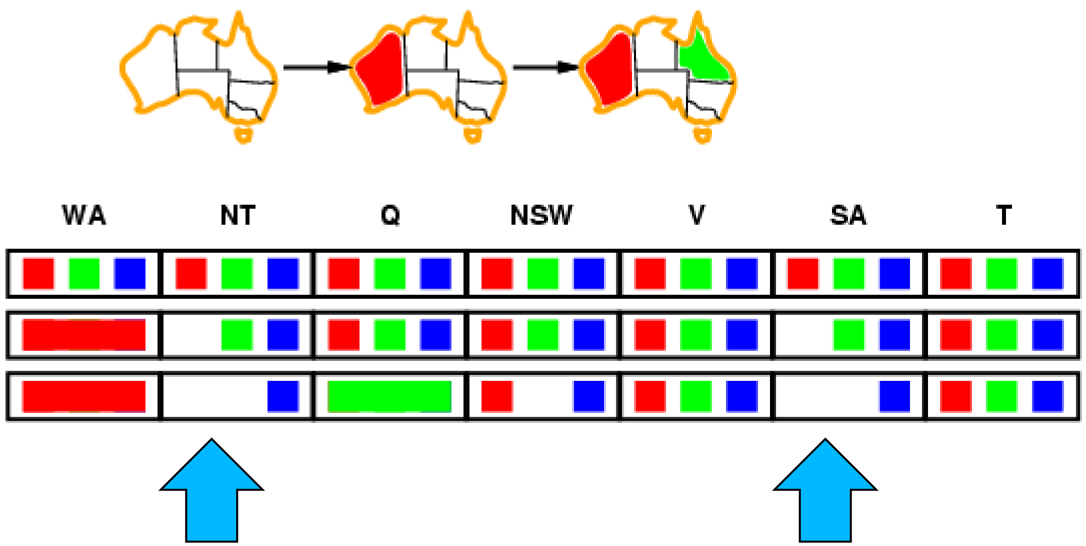
Az előrenéző ellenőrzés minden egyes alkalommal, amikor egy X változó értéket kap, minden, az X-hez kényszerrel kötött, hozzárendeletlen Y-t megvizsgál, és Y tartományából törli az X számára választott értékkel inkonzisztens értékeket.



Korlátozás előreterjesztése

Az előretekintő ellenőrzés ugyan sok inkonzisztenciát észrevesz, de nem mindent.

Ráadásul nem látja jól előre a kudarcokat.



NT és SA egyszerre nem lehet kék. Ez csak később derül majd ki, most még nem sérül kényszer.Bonyolultabb vizsgálattal ezt előre kideríthetjük.

Visszalépéses keresés hatékonyságának növelése (általános heurisztikák CSP-khez)

kvíz következik!

Általános módszerekkel is komoly gyorsítást el lehet érni:

- Melyik változóval foglalkozzunk a legközelebb?
- Milyen sorrendben vizsgáljuk az értékeit?
- Érzékelhetjük-e jó előre a kudarcokat? (korai nyesés)

ezek az ún. tárgyterület-független heurisztikák

09.2. kvíz

A még nem lekötött változóink közül melyikkel lehet érdemes kezdeni az értékadást?

A. **Amelyik a lehető legtöbb értéket kaphatja**
B. **Amelyik a lehető legkevesebb értéket kaphatja**
C. **Mindegy, hogy hány értéket kaphat, ez nem ront, nem javít**
D. **Valamelyik változóval, amelyik átlagosan sok értéket kaphat**

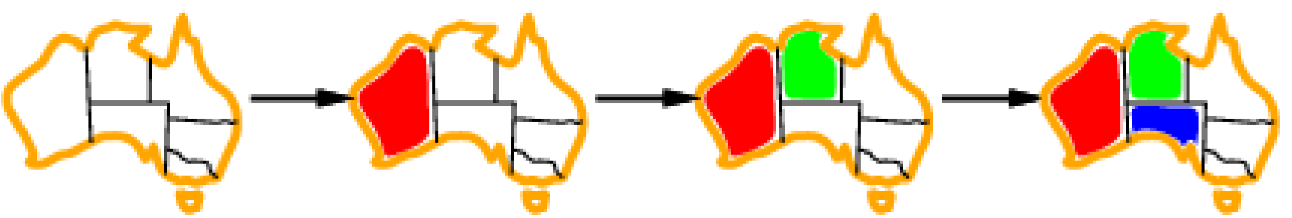
1. ***A legkevesebb fennmaradó érték*** ötlete (melyik változót választjuk?) **kvíz következik!**

A leginkább korlátozott változó:

a legkisebb számú megengedett értékkel rendelkező változóval kezdjünk, ill. folytassunk (***lokálisan kicsi az elágazási tényező!***)

⇒ **legkevesebb fennmaradó érték** heurisztika

(minimum remaining variables, **MRV**)



NT ill. SA csak 2 megengedett érték (piros már nem lehet), minden más 3. Nem érdemes Ausztrália „túlsó (keleti) oldalán folytatnunk a színezést)

09.3. kvíz

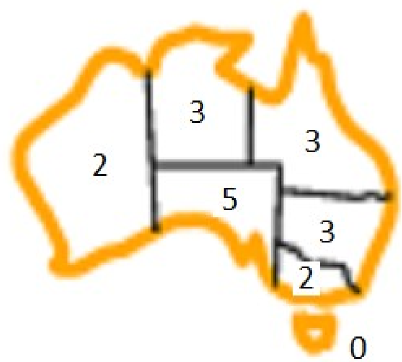
Ha a lehetséges értékek száma azonos, akkor a még nem lekötött változóink közül melyikkel lehet érdemes kezdeni az értékadást?

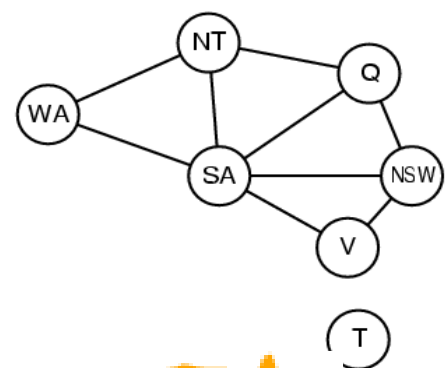
A. **Amelyik a legtöbb kényszerben szerepel**
B. **Amelyik a legkevesebb kényszerben szerepel**
C. **Mindegy, hogy hány kényszerben szerepel, ez nem ront, nem javít**
D. **Valamelyik változóval, amelyik átlagosan sok kényszerben szerepel**

2. ***Fokszám heurisztika*** ötlete (melyik <u>változót</u> válasszuk?) **kvíz következik!**

Az MRV-heurisztika semmit sem segít abban, hogy melyik régiót válasszuk ki elsőként Ausztrália kiszínezésekor, mert a kiinduláskor mindegyik régiónak három megengedett színe van.

A ***későbbi választások*** elágazási tényezőjét csökkentheti, ha azt a változót választjuk ki, amely a legtöbbször szerepel a még hozzárendeletlen változókra vonatkozó kényszerekben.





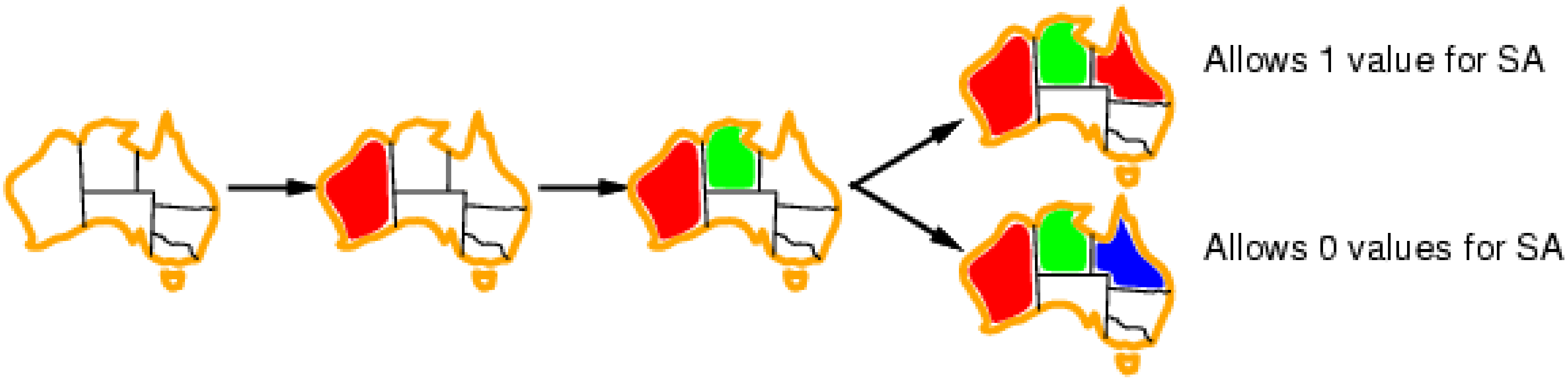
09.4. kvíz

Ha kiválasztottuk, hogy melyik változónak adunk értéket, akkor melyik lehetséges értéket adjuk?

A. **Amelyik a leginkább korlátozza a többi változót (a lehető legtöbb értéket törölhetjük)**
B. **Amelyik a legkevésbé korlátozza a többi változót (a lehető legkevesebb értéket törölhetjük)**
C. **Mindegy, hogy mennyire korlátozza a többit, ez nem ront, nem javít**
D. **Valamelyik változóval, amelyik átlagosan korlátozza a vele kényszerkapcsolatban lévő változókat**

3. ***A legkevésbé korlátozó <u>érték</u>*** ötlete **kvíz következik**

Előnyben részesítjük azt az <u>értéket</u>, amely a legkevesebb választást zárja ki a kényszergráfban a szomszédos változóknál.



09.5. kvíz: Az előző heurisztikákat alkalmazva melyik változóval foglalkozzunk először, és milyen értéket rendeljünk hozzá?

Kényszerek:

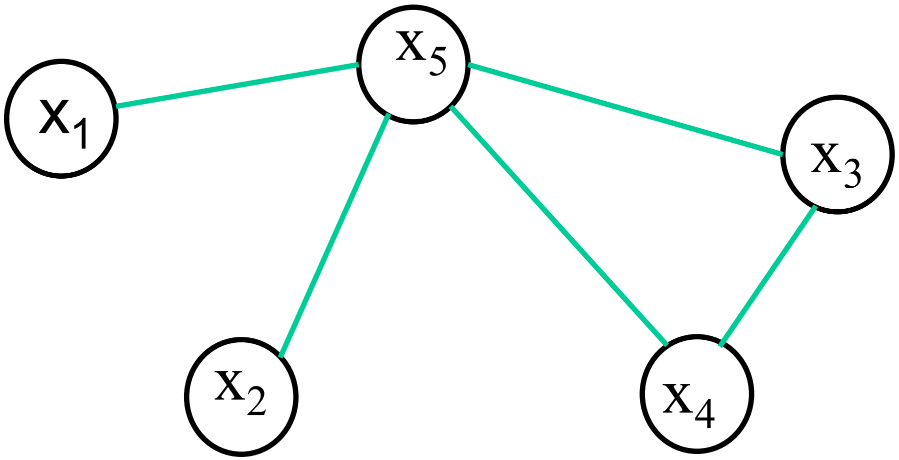
$x_1 > x_5$

$x_2 > 1 + x_5$

$x_3 > 2 \cdot x_5$

$x_4 > x_5 - 1$

$x_3 \leq 2 + x_4$



Mindegyik változó értékkészlete az egyjegyű nem negatív egészek halmaza: {0,1,2,…,9}

A. x1 $\Leftarrow$ 5

B. x2 $\Leftarrow$ 1

C. x3 $\Leftarrow$ 2

D. x5 $\Leftarrow$ 0

CSP lokális kereséssel

Kiindulás: teljes állapotleírás = minden változónak van értéke (esetleg rossz, nem teljesült kényszerekkel)

Operátorok: megváltoztatják a változók hozzárendelését, hogy csökkenjen a sérült kényszerek száma

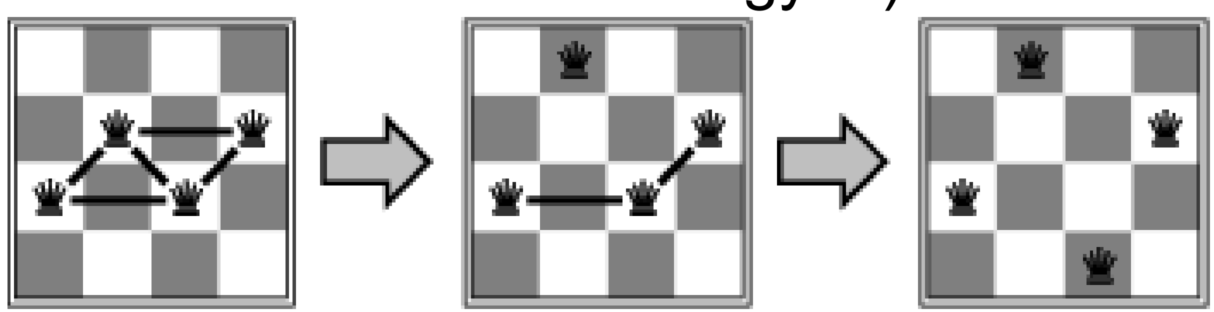
Változó szelekció: véletlen módon, bármely konfliktusban lévő (valamelyik kényszer sérül) változót választhatjuk

****

Min. konfliktus heurisztika:

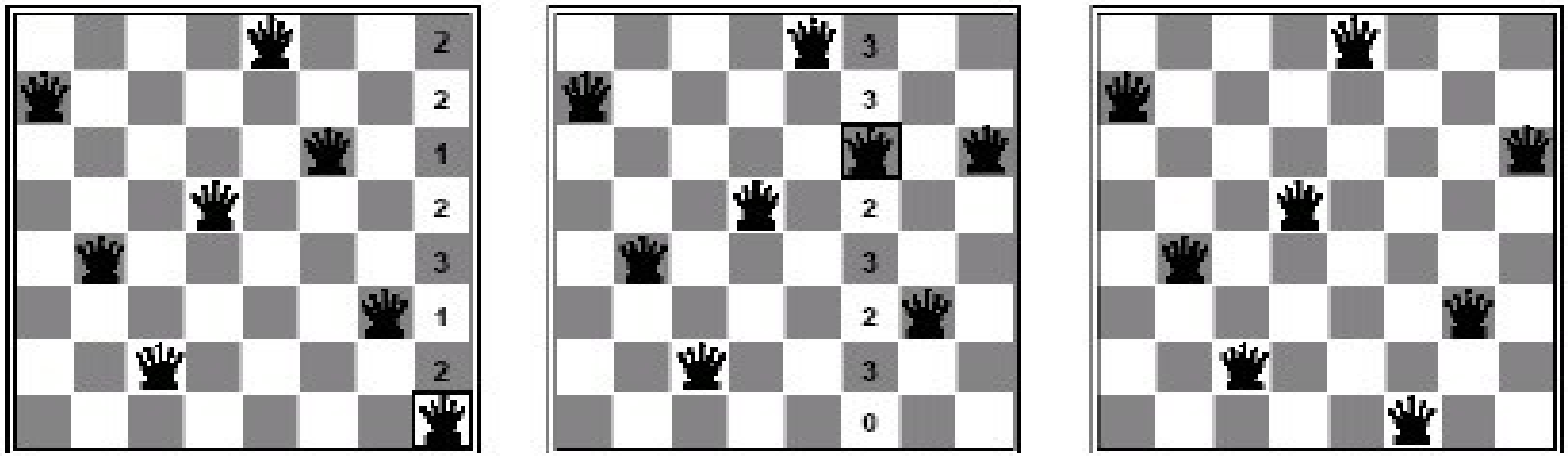
azt az értéket állítjuk be, amely a legkevesebb számú korlátot sérti, pl. hegymászó: *h(n)* = sérült korlátok száma, *h*(*n*) csökkentése a cél (itt most lefele mászunk a völgybe)

h(n) = a támadások száma



h = 5   h = 2   h = 0

CSP lokális kereséssel / Min. konfliktus heurisztika



Min. konfliktus heurisztika nagyon hatékony, nagy valószínűség-gel gyorsan old meg nagyon nagy problémaeseteket (10 millió királynő!).