

# Intelligentné rozhodovacie systémy



## Riešenie úloh s ohraničeniami

Marian.Mach@tuke.sk

<http://people.tuke.sk/marian.mach>

Október, 2018

# Reprezentácia úloh s ohraničeniami

- množina premenných  
 $V = \{V_1, V_2, \dots, V_n\}$
- množina domén  
 $D = \{D_1, D_2, \dots, D_n\}$
- množina ohraničení  
 $C = \{C_1, \dots, C_n, C_{1,2}, \dots, C_{n-1,n}, \dots, C_{1,2,\dots,n}\}$

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

# Vlastnosti ohraničení

- ohraničenie = prípustnosť hodnôt v kontexte
  - arita
  - počty
  - prázdne/plné ohraničenie
- reprezentácia
  - enumerácia (generovanie, testovanie)
  - predpis (testovanie)
- tvar ohraničenia
  - explicitný
  - implicitný (indukovaný)
- priestor prehľadávania
- špeciálne typy ohraničení – alldiff

# Sieť ohraničení

	1	2	3
1			8
2		5	3
3			

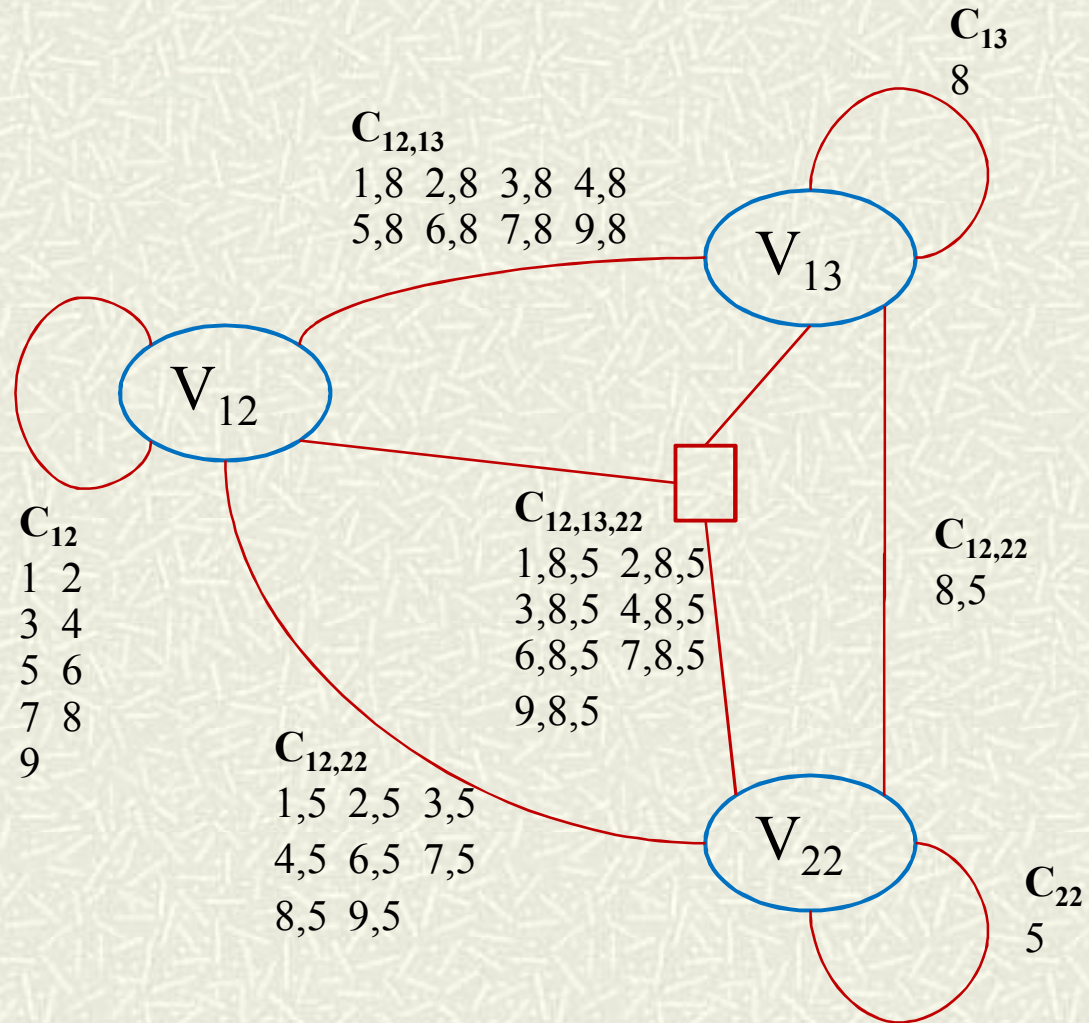
- Typ siete

- primárna

- dvojuzlová
    - hyperhrany

- duálna

- jednouzlová



# Riešenie úloh s ohraničeniami

## ▪ Prístupy

- algoritmické prístupy
  - redukčné algoritmy
    - konzistenčné algoritmy
    - substitučné algoritmy
  - prehľadavacie algoritmy
  - reštrukturalizačné algoritmy
  - ...
- metaheuristické prístupy
  - evolučné algoritmy
- znalostné prístupy
- ...



kombinované  
algoritmy

# Propagácia ohraničení

- šírenie ohraničení = sprísňovanie ohraničení  
doména  $\leftrightarrow$  ohraničenie, ohraničenie  $\leftrightarrow$  ohraničenie

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

- počiatočná situácia
  - 1 možnosť:  $C_{22} C_{51}$
  - 9 možností:  $C_{32} C_{41}$
  - 72 možností:  $C_{22,32} C_{41,51}$
  - 81 možností:  $C_{32,41}$
- propagácia
  - $C_{22} > C_{22,32} > C_{32} > C_{32,41}$
  - $C_{51} > C_{41,51} > C_{41} > C_{32,41}$

- Orezávanie priestoru prehľadávania

# Konzistencia premennej voči premennej

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

## (1,1)-konzistencia (AC)

- konzistencia  $V_i$  voči  $V_k =$  šírenie  $C_k > C_{i,k} > C_i$ 
  - ak hodnota z  $D_i$  spĺňa  $C_i$ , tak sa dá konzistentne rozšíriť o hodnotu iných premenných
    - konzistencia každej premennej voči všetkým iným premenným
- sieť ohraničení je (1,1) konzistentná, ak platí

Nech ľubovoľnej premennej  $V_i$  je priradená taká hodnota z jej domény, aby ohraničenie  $C_i$  nad touto premennou bolo splnené.

Potom pre ľubovoľnú ďalšiu premennú  $V_k$  je možné vybrať také hodnoty z jej domény, že všetky ohraničenia, definované nad vzniknutou dvojicou premenných sú splnené.



# Konzistencia premennej voči 2 premenným

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

## (1,2)-konzistencia (PIC)

- konzistencia  $V_i$  voči  $V_k$  a  $V_n =$  šírenie  
 $C_{k,n} > C_k > C_{i,k} > C_i$  a  $C_{k,n} > C_n > C_{i,n} > C_i$ 
  - ak hodnota z  $D_i$  spĺňa  $C_i$ , tak sa dá konzistentne rozšíriť o hodnotu dvoch iných premenných
    - konzistencia každej premennej voči všetkým dvojiciam iných premenných
- sieť ohraničení je (1,2) konzistentná, ak platí

Nech ľubovoľnej premennej  $V_i$  je priradená taká hodnota z jej domény, aby ohraničenie  $C_i$  nad touto premennou bolo splnené.

Potom pre ľubovoľné ďalšie dve premenné  $V_k$  a  $V_n$  je možné vybrať také hodnoty z ich domén, že všetky ohraničenia, definované nad vzniknutou trojicou premenných sú splnené.

# Konzistencia 2 premenných voči premennej

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

## (2,1)-konzistencia (PC)

- konzistencia  $V_i$  a  $V_k$  voči  $V_n$  = šírenie
  - $C_n > C_{i,n} > C_i > C_{i,k}$  a  $C_n > C_{k,n} > C_k > C_{i,k}$
  - ak hodnoty z  $D_i$  a  $D_k$  spĺňajú  $C_i$ ,  $C_k$  a  $C_{i,k}$ , tak sa dajú konzistentne rozšíriť o hodnotu inej premennej
    - konzistencia dvojíc premenných voči všetkým iným premenným
- sieť ohraničení je (2,1) konzistentná, ak platí

Nech ľubovoľným dvom premenným  $V_i$  a  $V_k$  sú priradené také hodnoty z ich domén, aby ohraničenia  $C_i$ ,  $C_k$  a  $C_{i,k}$  nad týmito premennými boli splnené.

Potom pre ľubovoľnú ďalšiu premennú  $V_n$  je možné vybrať takú hodnotu z jej domény, že všetky ohraničenia, definované nad vzniknutou trojicou premenných, sú splnené.

## (i,k)-konzistencia

- $i+k \leq$  počet premenných
- sieť ohraničení je (i,k) konzistentná, ak platí

Nech ľubovoľným  $i$  premenným sú priradené také hodnoty z ich domén, aby všetky ohraničenia nad touto  $i$ -ticou premenných boli splnené.

Potom pre ľubovoľných ďalších  $k$  premenných je možné vybrať také hodnoty z ich domén, že všetky ohraničenia, definované nad vzniknutou  $(i+k)$ -ticou premenných sú splnené.

- (i,k)-konzistencia vs. (m,n)-konzistencia
- prísna konzistencia (garancia)
- smerová konzistencia

transformácia siete  
ohraničení

# Algoritmus AC

```
(defun AC (&optional (zoznam *zoznam-premennych*))
  (unless (null zoznam)
    (let ((premenna (first zoznam)) zmena)
      (dolist (hodnota (unarne-ohranicenie premenna))
        (unless (vyhovuje-hodnota-premennym premenna hodnota)
          (redukcia-unarneho-ohranicenia premenna hodnota)
          (setf zmena t))))
      (if zmena
          (AC (union (rest zoznam) (zavisle-premenne premenna)))
          (AC (rest zoznam)))))))
```

# Algoritmus AC (pokr.)

- (defun vyhovuje-hodnota-premennym (prem hodn)
  - (every #'(lambda (prem2)
    - (vyhovuje-hodnota-premennej prem hodn prem2))
    - (zavisle-premenne prem)))
  
- (defun vyhovuje-hodnota-premennej (prem1 hodn1 prem2)
  - (some #'(lambda (hodn2)
    - (vyhovuju-2-hodnoty-ohraniceniu prem1 hodn1 prem2 hodn2))
    - (unarne-ohranicenie prem2)))
  
- (defun vyhovuju-2-hodnoty-ohraniceniu (prem1 hodn1 prem2 hodn2)
  - (member (cons hodn1 hodn2) (binarne-ohranicenie prem1 prem2)
    - :test #'equal))

# RPC / max-RPC

The image displays two 9x9 grids illustrating the RPC (Row-Column-Puzzle) and max-RPC (Maximum Row-Column Puzzle) concepts. The grids contain numbers and candidate lists (small numbers in the corners of cells).

**Left Grid (RPC):**

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

**Right Grid (max-RPC):**

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

Annotations: A yellow circle highlights the cell (row 2, column 7) in the left grid. A pink circle highlights the cell (row 4, column 8) in the left grid. A yellow dotted line traces a path through the right grid, starting from the top-right cell and ending at the bottom-left cell.



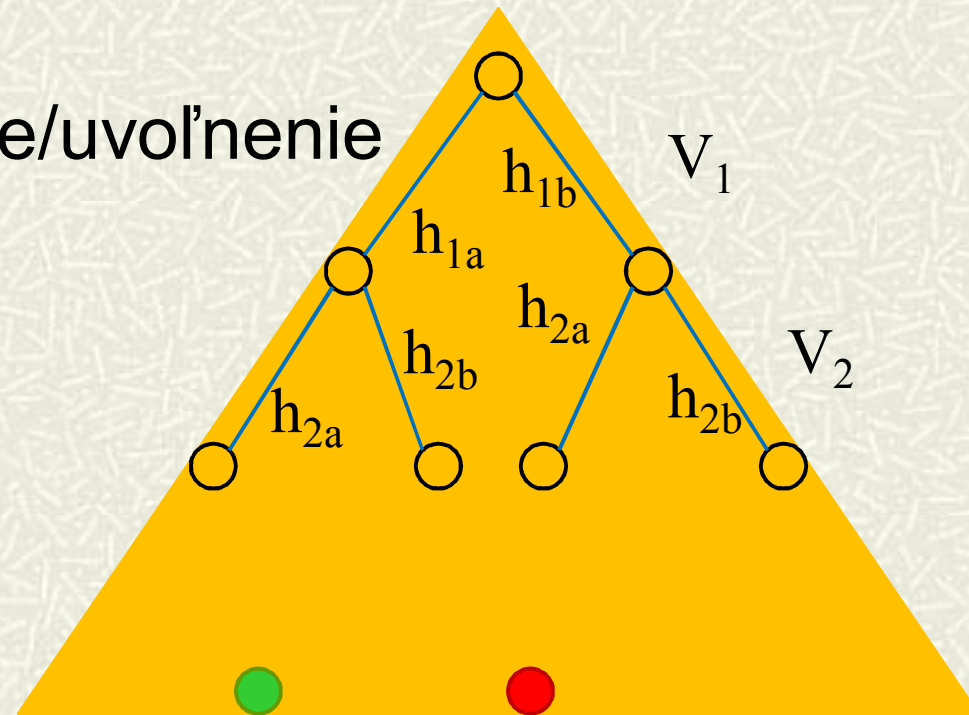
# Singletony

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

SAC

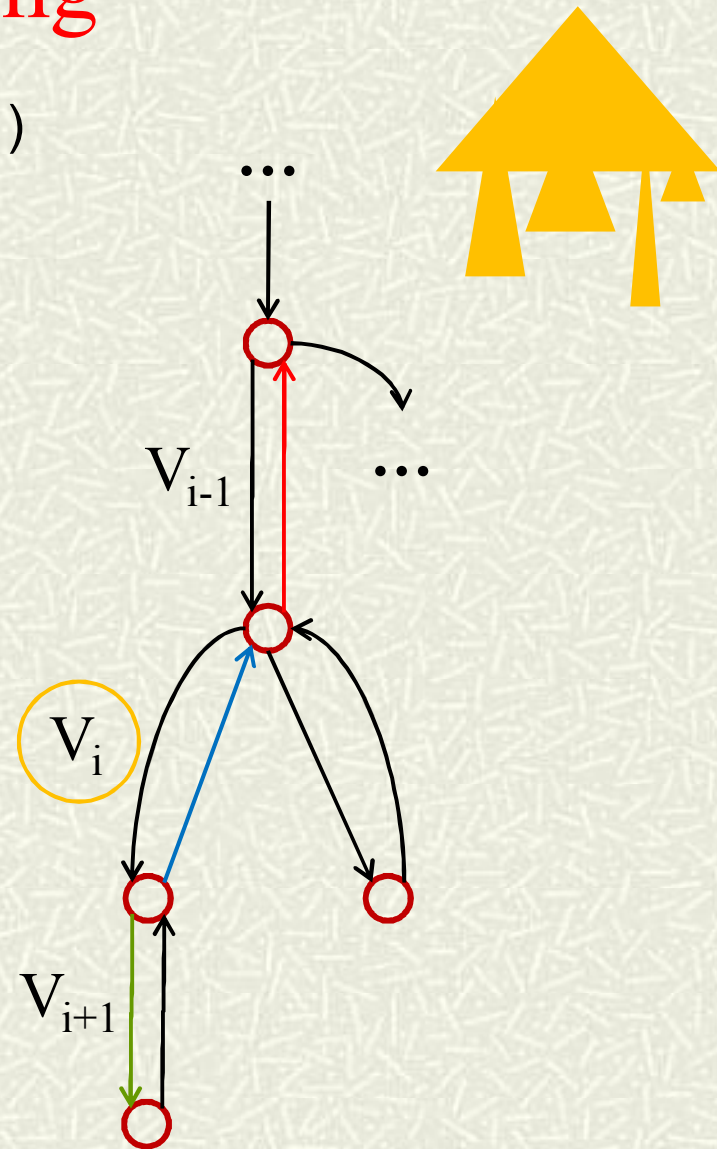
# Hierarchické prehl'adavanie

- strom prehl'adavania
  - uzly = priradenie/podpriestor
    - koreň – celý priestor
    - list – jeden bod
  - hrana = priradenie/uvol'nenie
- usporiadanie
  - premenných
  - domén



# Backtracking

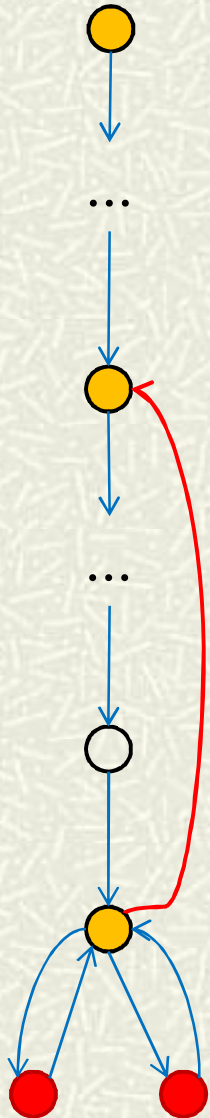
```
define method backtracking( i::<integer> )  
  if( and ( 0 < i, i <= n ) )  
    V[i] := next( D[i], V[i] );  
    if ( not ( V[i] ) )  
      backtracking ( i - 1 );  
    end if;  
    if ( satisfy_all_constraints ( ) )  
      backtracking ( i + 1 );  
    else  
      backtracking ( i );  
    end if;  
  end if;  
end method backtracking;
```



# Vlastnosti backtrackingu

- prehľadávanie do hĺbky
- tvar stromu prehľadávania závisí od usporiadania
  - premenných a ich domén
- nevýhody
  - opakované zlyhávanie
  - redundantné prehľadávanie
  - neidentifikuje príčinu neúspechu
- vylepšenia
  - skokové (znížiť počty navracaní)
  - pamäťové (znížiť počty kontrol)
  - kombinované (zvýšiť orezávanie priestoru)

# Prehľadávanie so spätným skokom



- skok k premennej, ktorá zapríčinila konflikt
  - najbližšia premenná z aktuálneho konfliktu
    - ďalšie navracanie chronologicky
  - najbližšia premenná zo susedov v sieti ohraničení
    - nerobí sa analýza aktuálneho konfliktu
    - skok na možnú príčinu
- najbližšia premenná z konfliktného zoznamu
  - udržiavanie konfliktného zoznamu
  - ďalšie navracanie skokmi
    - informácia o konfliktoch sa nestráca

Conflict-  
directed  
backjumping

# Prehľadávanie s pamäťou

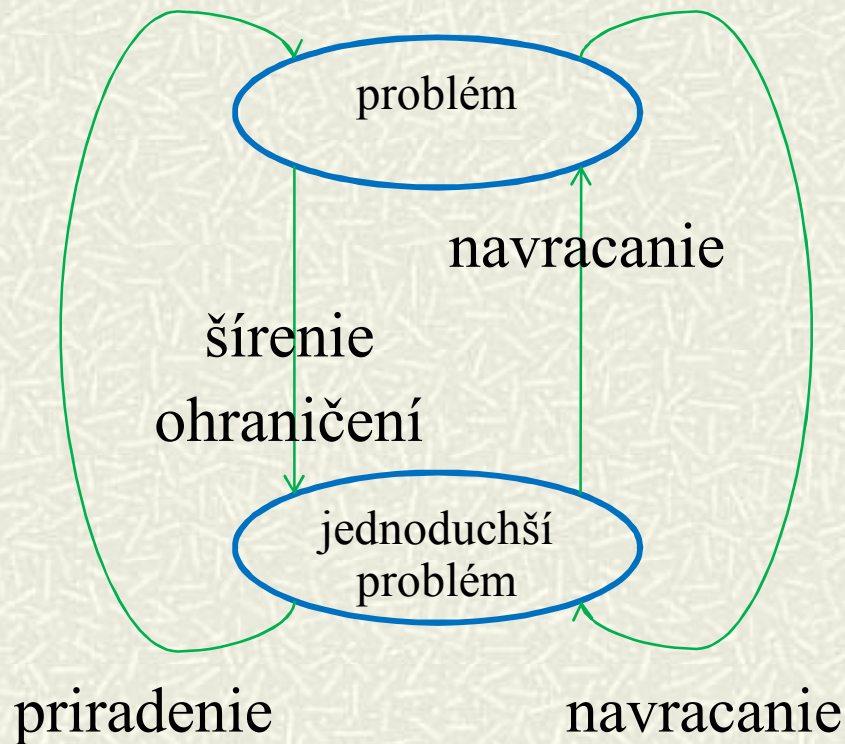
- pamäť – možnosť poučiť sa z výsledkov činnosti v skorších etapách, využiť to neskôr
- pamätanie si kombinácií hodnôt
  - učenie z neúspechu
    - ak podmnožina hodnôt porušuje ohraničenie, je pamätaná
    - v budúcnosti sa takáto kombinácia nevytvorí  
=> menej generovaných uzlov (menší strom)
  - učenie z úspechu
    - pamätá si kombinácie hodnôt, ktoré nerobili problém
    - v budúcnosti, ak vzniká taká kombinácia, už nie je testovaná  
=> menej kontrol konzistencie (nemení tvar stromu)
- indukcia a pamätanie si nových ohraničení
  - indukcia v prípade slepých koncov

Backchecking

Backmarking

Plytké a hlboké učenie

# Prehľadávanie s propagáciou ohraničení



- propagácia ohraničení v uzloch stromu prehľadávania
- navracanie
  - hodnota porušuje ohraničenie
  - vyprázdnenie ohraničenia
- redukcia medzi premennými
  - voľnými
    - zabezpečenie určitého stupňa konzistencie v koreni
  - posledne viazanou a voľnými
    - konzistencia v nekoreňových uzloch

Dopredná kontrola

Udržiavanie hranovej konzistencie

# Prehľadávanie s usporiadaním

- veľkosť stromu a jeho tvar závisí od usporiadania premenných a ich domén
- usporiadanie premenných
  - intuitívny **fail first** princíp
  - statické (poradie pred samotným prehľadávaním)
    - => minimálna šírka, maximálny stupeň
  - dynamické (počítané nanovo pre voľné premenné)
    - => minimum ostávajúcich hodnôt
- usporiadanie hodnôt
  - intuitívny **succeed-first** princíp
    - => dopredné heuristiky