Úlohy s omezujícími podmínkami

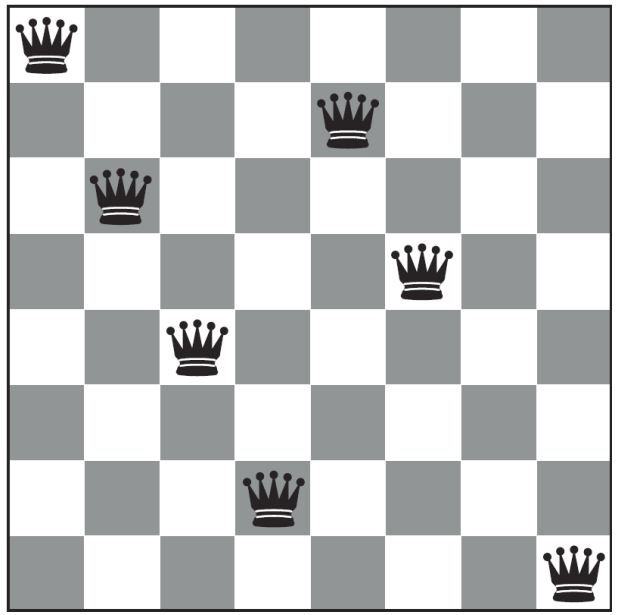
Martin Šůstek

isustek@fit.vutbr.cz

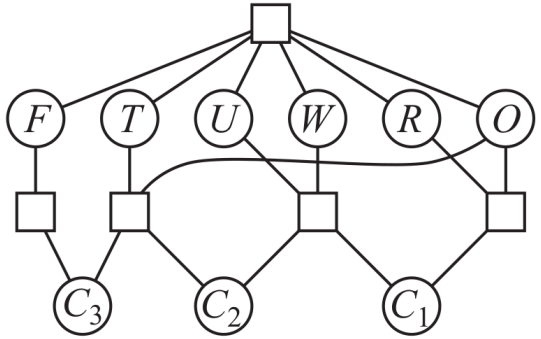
BRNO FACULTY UNIVERSITY OF INFORMATION OF TECHNOLOGY TECHNOLOGY

- Nehledáme sekvenci akcí, ale ani se nesnažíme najít nejlepší řešení
  - Hledáme řešení splňující stanovené podmínky pro kombinaci hodnot proměnných
- Faktorizovaná reprezentace stavů
  - Několik proměnných s předem definovanými rozsahy hodnot

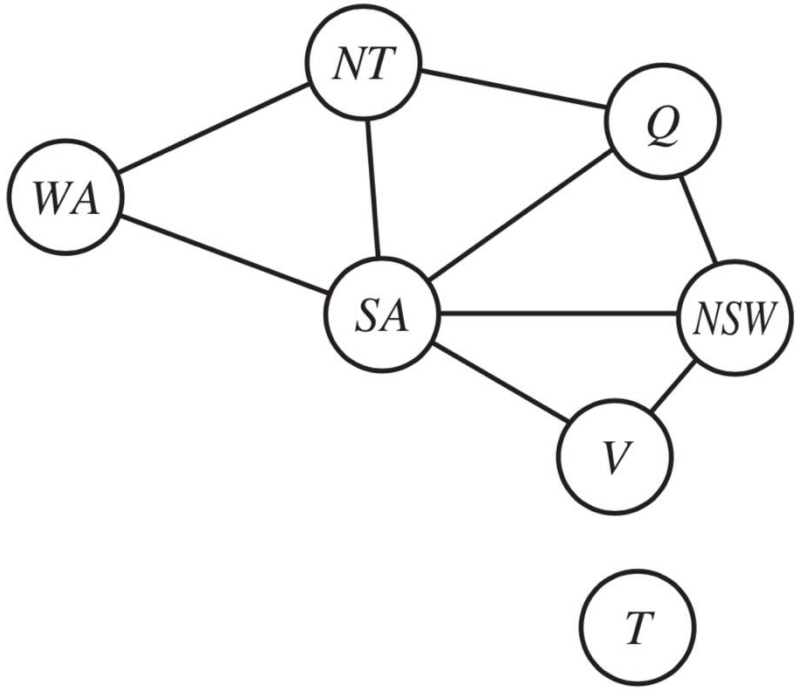
	1	2	3	4	5	6	7	8	9
A			3		2		6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7								8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1		3		



$$\begin{array}{r} T\ W\ O \\ +\ T\ W\ O \\ \hline F\ O\ U\ R \end{array}$$



- Obarvit všechny státy v mapě tak, aby sousedící státy neměly stejnou barvu
- $X = \{WA, NT, Q, NSW, V, SA, T\}, D_i = \{red\ , green\ , blue\}$
- $C = \{SA \neq WA, SA \neq NT, SA \neq Q, SA \neq NSW,\ SA \neq V, WA \neq NT, NT \neq Q, Q \neq NSW, NSW \neq V\}$
- Vizualizace pomocí constraint graph



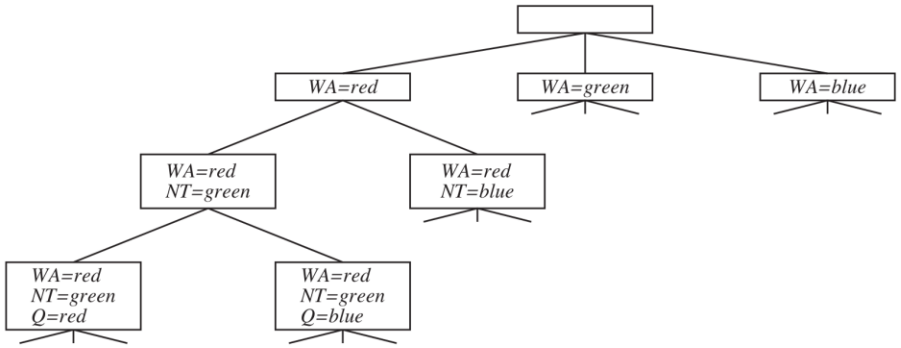
Formální definice CSP

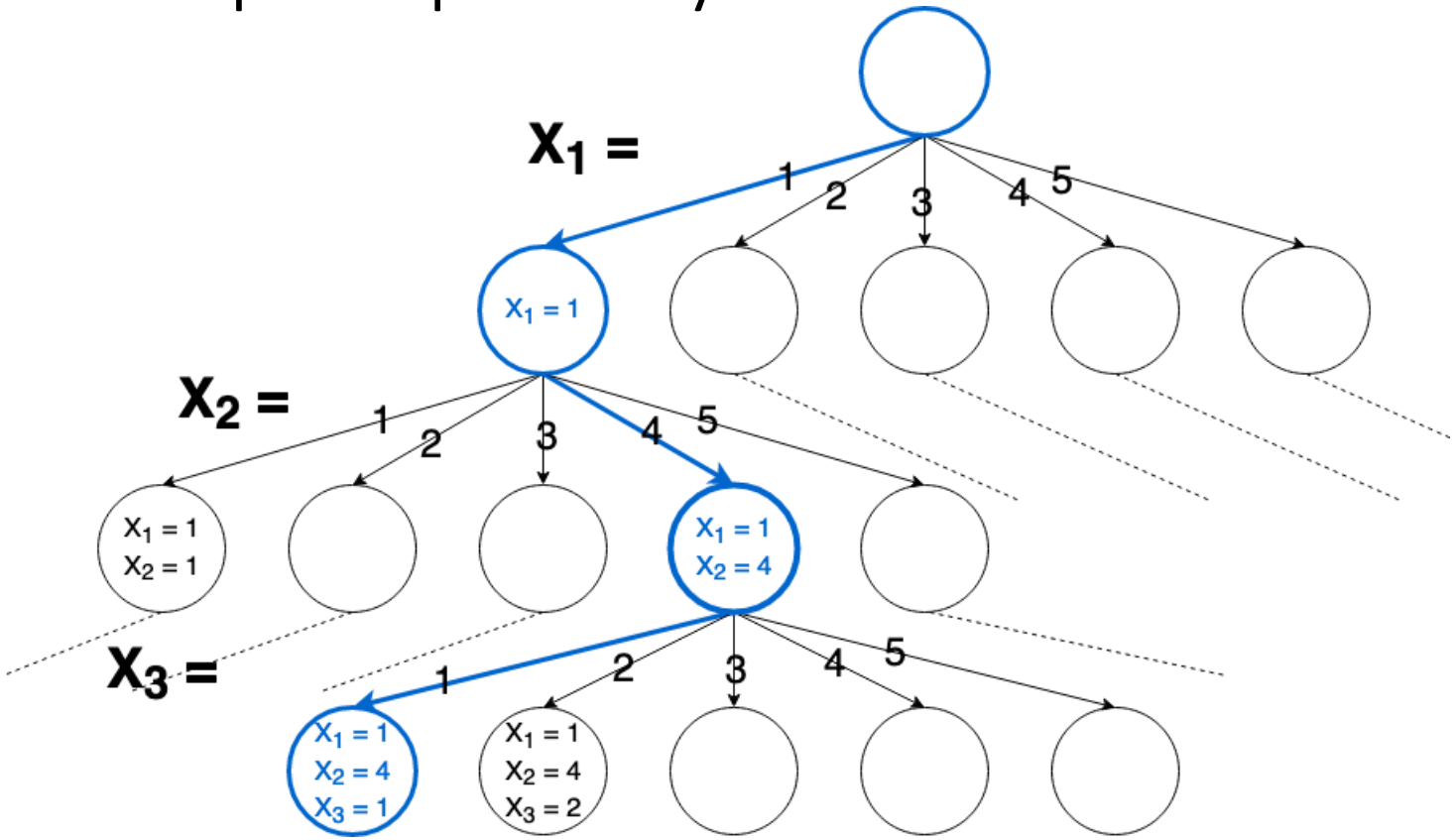
- X je množina proměnných $X = \{X_1, \ldots, X_n\}$
- D je množina domén $D = \{D_1, \ldots, D_n\}$, D_i je doména pro X_i, $1 \leq i \leq n$
- C je množina omezení, která definují povolené kombinace hodnot
  - Formálně se omezení skládá ze scope (kterých proměnných se týká) a relace
    - Relaci lze definovat různou formou, např. výčtem hodnot, rovností či nerovností
  - Příklad: $X_1 < X_2; 3X_2 + 1 = X_3; \langle (X_1, X_2), \{(0,0), (1,4)\} \rangle$
  - Využíváme zápis $Alldiff(X_1, X_2, X_3)$ = proměnné musí mít odlišné hodnoty

Přiřazení – Assignment

- Přiřazení – nastavení proměnné na přípustnou hodnotu
  - Konzistentní/legální (consistent/legal) přiřazení
    - Pokud neporušuje žádnou podmínku
  - Úplné (complete) přiřazení
    - Pokud má každá proměnná přiřazenou hodnotu
- Řešení = úplné a konzistentní přiřazení

- V každé úrovni stromu se přiřadí hodnota jedné proměnné
- Po přiřazení všech se zkoumá, jestli je řešení konzistentní
- Neuvažovat zároveň různé kombinace pořadí proměnných!
- Příklad:
  - $X = \{X_1, X_2, X_3\}$
  - $D_i = \{1, \ldots, 5\}$





Řešení CSP: Inference

- Alternativa k prohledávání
  - Snaha o snížení počtu hodnot v doménách → zrychlení prohledávání
- Inference constraint propagation CP (propagace omezení)
  - Lze provést před samotným prohledáváním nebo se může s prohledáváním prolínat
  - Konzistence uzlů (node consistency)
    - *Odpovídá 1*-consistency
  - Konzistence hran (edge consistency)
    - *Odpovídá 2*-consistency
  - Konzistence cest (path consistency)
    - *Odpovídá 3*-consistency

- Konzistentní uzly = doména proměnné neporušuje unární omezení
  - Omezení se vztahují jen k jedné proměnné – např.: $X_1 \neq 3$ nebo $X_2 = 1$
- Dosáhneme aplikováním unárních omezení na domény → úprava domén např.:
  - Příklad: $X = \{X_1, X_2, X_3\},\ D_i = \{1, \ldots, 5\}, X_1 \neq 3$ a $X_2 = 1$:
    - $D_1 = \{1,2,4,5\}, D_2 = \{1\}$
- Unární omezení provedeme pouze jednou před samotným prohledáváním

- Konzistence hran – stejné jako konzistence uzlů ale pro binární omezení
- Pro každou hodnotu z domény proměnné X_i existuje nějaká hodnota v doméně proměnné $X_j, i \neq j$ tak, že přiřazení této kombinace je konzistentní
  - = Pro každou hodnotu je na druhé straně každé hrany alespoň jeden „kamarád“
- Algoritmus AC-3 – vytvoření konzistentních hran

- Příklad: $X = \{X_1, X_2, X_3\}, D_i = \{1, \ldots, 5\}$
  - Pro $X_2 = X_1^2$:
    - $D_1 = \{1,2\},\ D_2 = \{1,4\}$
  - Pro $X_2 = 2X_1,\ X_3 = X_2 - 1,\ Alldiff(X_1, X_2, X_3)$:
    - $D_1 = \{1,2\}, D_2 = \{2,4\}, D_3 = \{1,3\}$

- Konzistence cest – ternární omezení nebo kombinace binárních dohromady
  - Např. $X_1 \neq X_2, X_1 \neq X_3, X_2 \neq X_3$ nebo $Alldiff(X_1, X_2, X_3)$
  - Pro každé konzistentní přiřazení hodnot proměnným $X_i, X_j, i \neq j$ musí existovat hodnota pro $X_k, i \neq k, j \neq k$
- Lze pokračovat dál, obecně K-konzistence
  - Konzistence cest = 3-consistency
  - Pro konzistentní přiřazení $k-1$ proměnných → konzistentní přiřazení k proměnných
  - Silná K-konzistence – nejen k-konzistentní ale taky nižší řády $(K-1, K-2, \ldots)$

- Příklad: $X = \{X_1, X_2, X_3\}, D_i = \{1, \ldots, 5\}$
  - Pro $X_2 = 2X_1,\ X_3 = X_2 - 1,\ Alldiff(X_1, X_2, X_3)$:
    - $D_1 = \{2\}, D_2 = \{4\}, D_3 = \{3\}$

Inference pro $Alldiff$ omezení: Naked tripples

- $Alldiff$ omezení → inference naked tripples
  - Příklad sudoku: domény 3 políček stejného bloku jsou podmnožinou {5,8,9} → na ostatních políčkách ve stejném bloku se nemůže nacházet 5, 8 ani 9

Obr.: https://www.sudokuwiki.org/Naked_Candidates

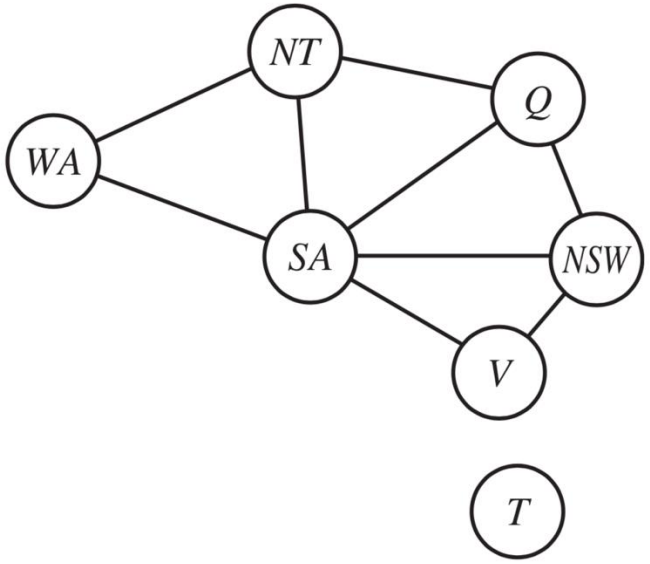
	1	2	3	4	5	6	7	8	9
A			3		2		6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7								8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1		3		

	1	2	3	4	5	6	7	8	9
A	4	8	3	9	2	1	6	5	7
B	9	6	7	3	4	5	8	2	1
C	2	5	1	8	7	6	4	9	3
D	5	4	8	1	3	2	9	7	6
E	7	2	9	5	6	4	1	3	8
F	1	3	6	7	9	8	2	4	5
G	3	7	2	6	8	9	5	1	4
H	8	1	4	2	5	3	7	6	9
I	6	9	5	4	1	7	3	8	2

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

- Při prohledávání využíváme backtracking – proč?
- Zodpovězením následujících otázek můžeme urychlit prohledávání:
  1. Kterou proměnnou přiřadit teď? X_1 nebo X_2?
  2. Kterou hodnotu přiřadit teď? $X_1 = 1$ nebo $X_1 = 2$?
  3. Máme použít nějakou inferenci v aktuálním kroku? Pokud ano, jakou?
  4. Došlo ke konfliktu, kam se vrátit? Jaká kombinace hodnot je problematická? Mohu příště neopakovat?
     - Nebudeme se zabývat
       - Metoda conflict-directed backjumping
       - Constraint learning → no-good

- Minimum-remaining-values (most constrained variable) – nejmenší větvení
  - Zvolíme proměnnou, které lze přiřadit nejméně hodnot
- Degree heuristic
  - Proměnná vyskytující se v největším počtu omezení nepřiřazených proměnných
    - V grafu nepřiřazených proměnných – největší stupeň = nejvíce hran



- Least-constraining-value – přiřadíme hodnotu, která ponechá nejvíc hodnot u sousedních nepřiřazených proměnných
  - Chceme, aby zbylo co nejvíce možností
  - Hledáme-li jedno řešení, zajímá nás pouze co nejrychlejší nalezení prvního
    - Pokud úloha nemá řešení nebo hledáme všechny → pořadí výběru hodnot je irelevantní

- Forward checking
  - Po každém přiřazení hodnoty do proměnné X_i se provádí inference, aby se zaručila konzistence hran aktuálně přiřazované proměnné X_i
    - Odebrání nekonzistentních hodnot u proměnných spojených hranou s X_i ne pro celý graf
- Maintaining Arc Consistency (MAC)
  - Po přiřazení proměnné se vynutí konzistence hran dosud nepřiřazených proměnných
    - Pomocí AC-3, aplikuje se na celý graf

Lokální prohledávání pro CSP úlohy

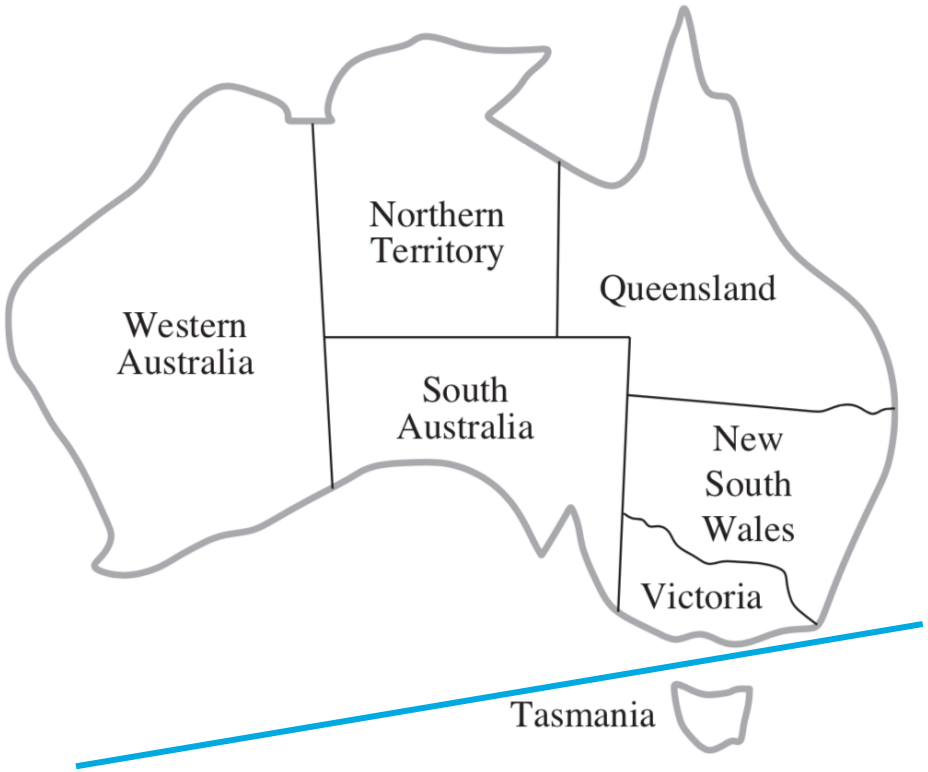
- Aktuální uzel má přiřazené všechny hodnoty proměnných – úplné přiřazení
  - Modifikujeme hodnotu zvolené proměnné → snaha snížit počet konfliktních přiřazení
    - Ideálně žádný konflikt (porušení omezujících podmínek) → konzistentní přiřazení
- Min-conflict – zvolit hodnotu minimalizující počet konfliktů
  - Připomíná hill-climbing
  - Funguje velmi dobře pro problém N-dam

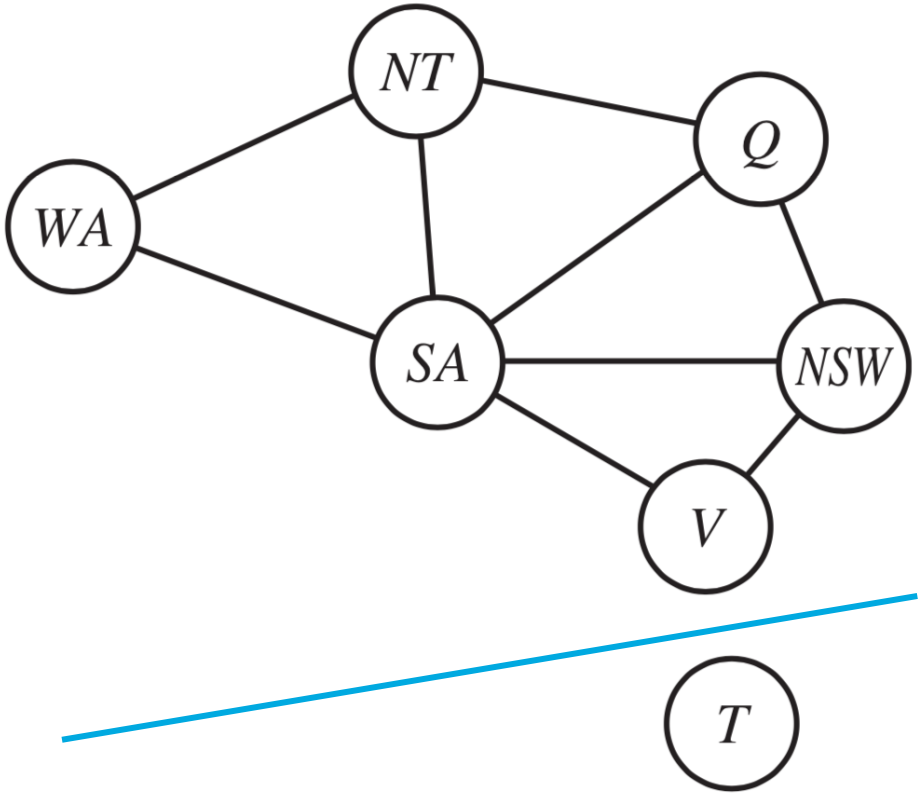
- Můžeme také využít metody uvedené v přednášce o lokálním prohledávání
- Protože se jedná o lokální prohledávání:
  - Vhodné, když se omezení mění – online settings
  - Nedokážou říct, že řešení neexistuje

Lokální prohledávání pro CSP úlohy: Možné úpravy

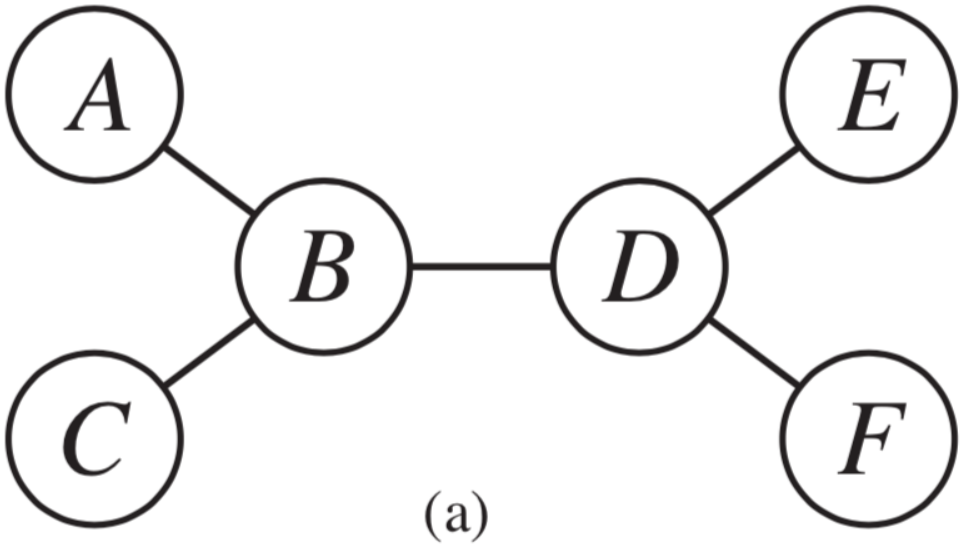
- Tabu search – uložíme několik naposledy navštívených stavů a zabráníme algoritmu, aby se do těchto stavů vrátil
- Constraint weighting – obtížnost splnění omezení je různá
  - Omezení mají váhu, bereme v potaz vážený počet konfliktů
    - Typicky se váhu učíme
    - Nedaří se nám plnit omezení → zvýšení váhy
    - Často splněná omezení → snížení váhy

- Rozdělení úlohy na nezávislé částí – zřídkakdy realizovatelné u reálných úloh
  - Oddělené řešení → zrychlení

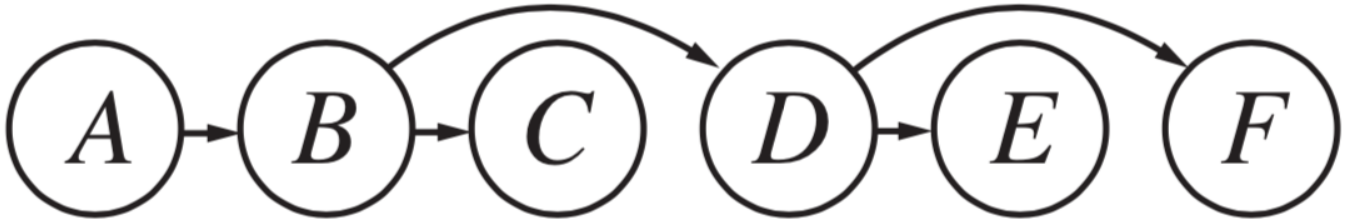




- Strom – každé dvě proměnné jsou spojeny maximálně jednou cestou
  - Rychlejší řešení díky topologickému pořadí přiřazení
    - Zvolíme kořen
    - Každý potomek se přiřazuje až po svém rodiči
  - Projdeme proměnné ve zvoleném pořadí, vyžadujeme konzistenci hran → řešení

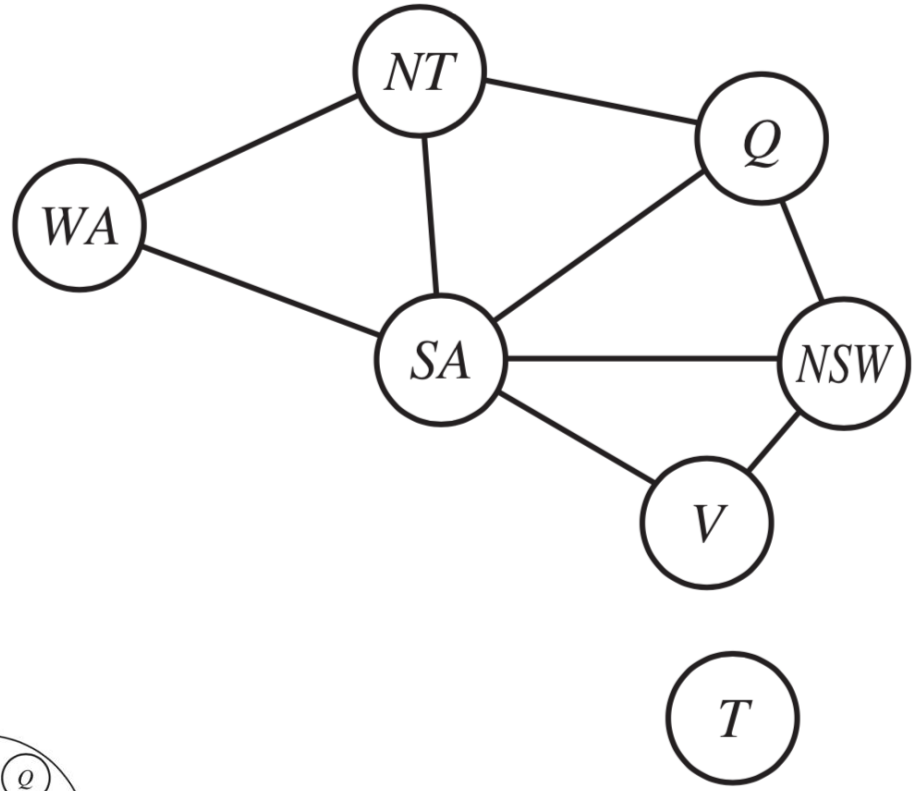


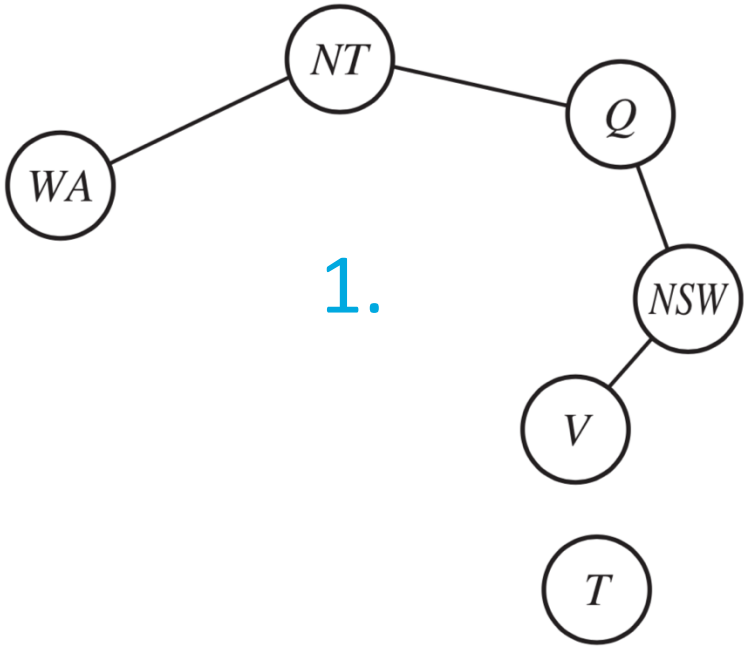
(a)

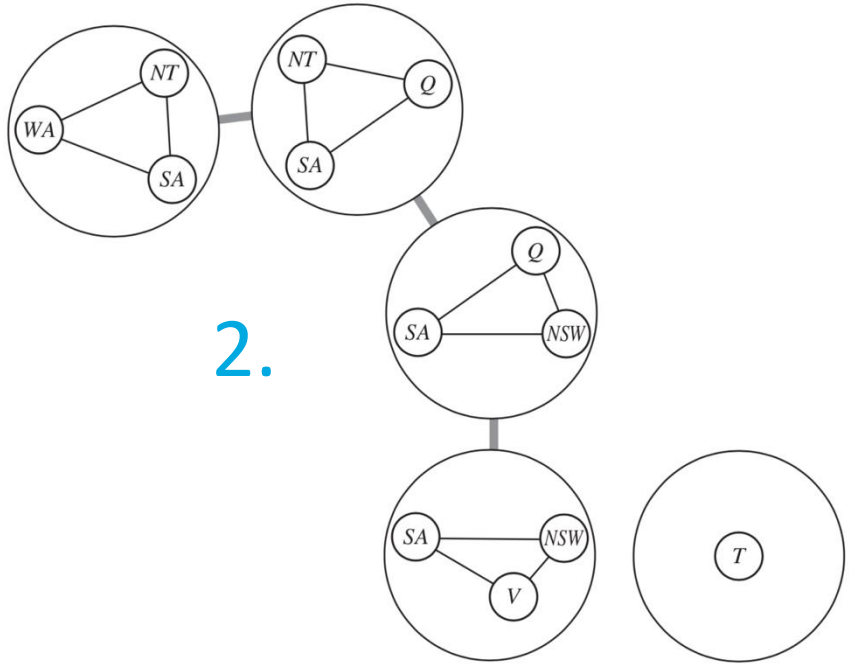


(b)

- Převod obecného grafu na strom:
  1. Odstranění uzlů, aby zbytek tvořil strom(y)
  2. Dekompozice na několik podproblémů



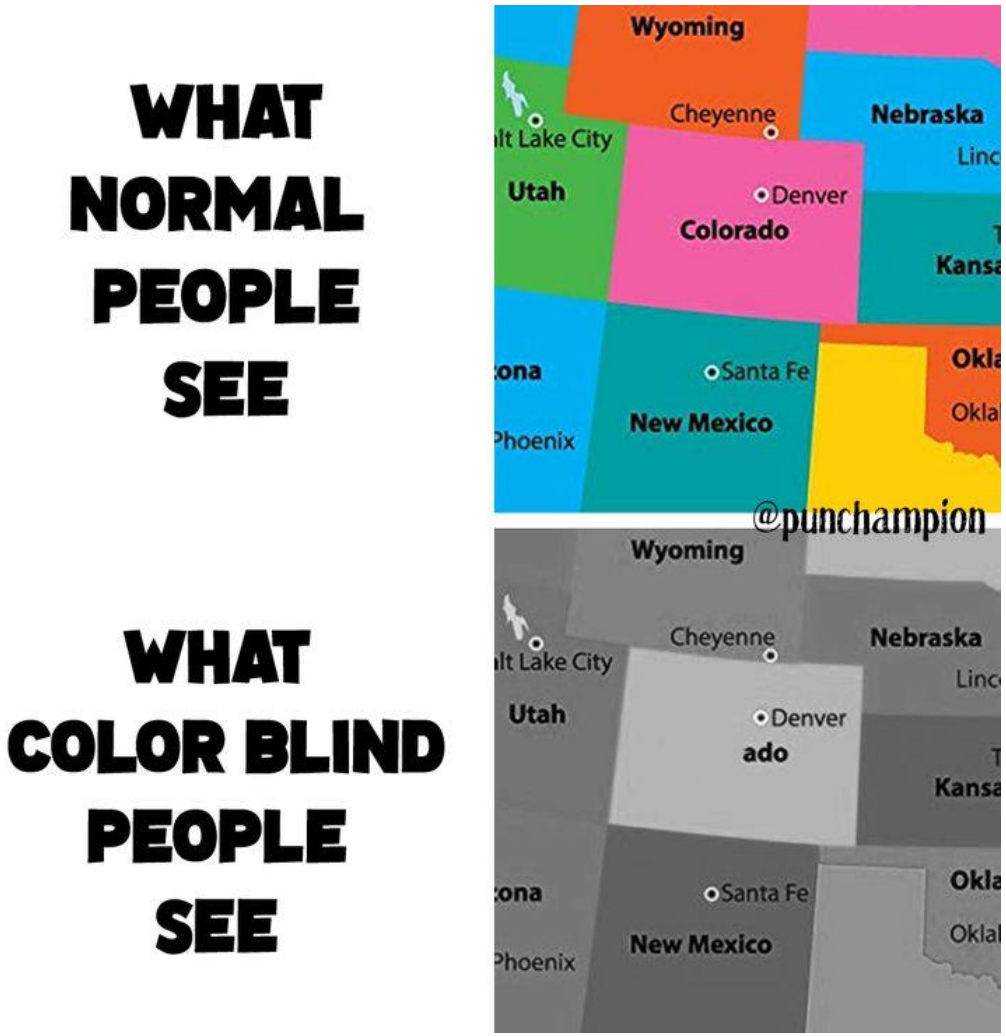




Symmetry-breaking omezení

- $Alldiff$ omezení u příkladu barvení map → několik řešení pouze výměnou barev
  - Modře obarvená města obarvíme červeně, červeně obarvená obarvíme modře, ...
    - → Přidáme symmetry-breaking constraint
- Symmetry-breaking constraint
  - Odstranění symetrie – myšlenka kanonického tvaru
  - Omezení kombinace barev, např. ordinální hodnoty barev + $NT < SA < WA$
    - Může pomoct obzvláště při dekompozici na několik podproblémů

Děkuji za pozornost!



- https://uk.pinterest.com/pin/796433515338834819/