

Hrubá síla, heuristiky a hry

Obsah

- 1 Úvod
- 2 Prohledávání stavového prostoru
 - Stavové prostory
 - Heuristiky
 - Hry
- 3 Problém splnění podmínek a backtracking
 - Problém splnění podmínek
 - Backtracking
 - Propagace podmínek
- 4 Shrnutí

Hrubá síla

- „Vyzkoušej všechny možnosti.“
- kdy použít:
 - problém relativně malý, možností málo – i když by to šlo chytřeji, nemá cenu se s tím babrat (např. Sudoku)
 - nic lepšího prostě neznáme (NP-úplné problémy)

Heuristiky

- heuristika = přístup, který většinou pomůže, ale nemusí fungovat vždy
- dobrá heuristika může řádově zlepšit hrubou sílu

NP-úplný problém

zjednodušeně:

„problém, který neumíme efektivně (polynomiálně) vyřešit“

příklady:

- problém obchodního cestujícího
- SAT – splňující přiřazení do logické formule
- zobecněné Sudoku

Prohledávání stavového prostoru

připomenutí: stavový prostor = graf

použití např.:

- Good Old-Fashioned Artificial Intelligence (GOF AI)
- verifikace protokolů

Logická úloha: Misionáři a kanibalové



- 3 misionáři, 3 kanibalové
- řeka, 1 loďka (max 2 lidé)
- víc kanibalů jak misionářů na jednom místě \Rightarrow problém

Vlk, koza, zelí

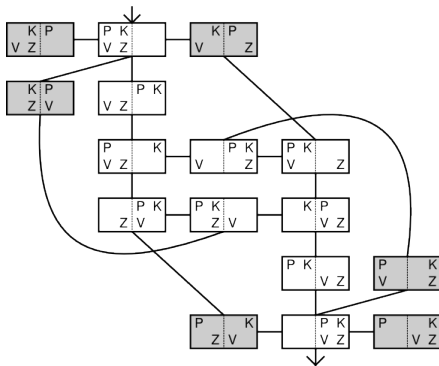
- klasická úloha typu převážení přes řeku
- zakreslit stavový prostor
- co úloha „koza a 2 zelí“?

Vlk, koza, zelí

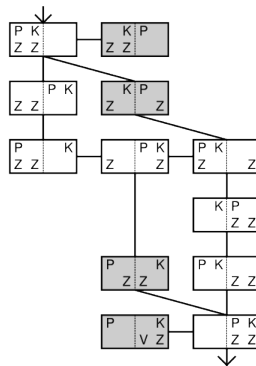
- klasická úloha typu převážení přes řeku
- zakreslit stavový prostor
- co úloha „koza a 2 zelí“?
- „izomorfní“ úlohy

Vlk, koza, zelí

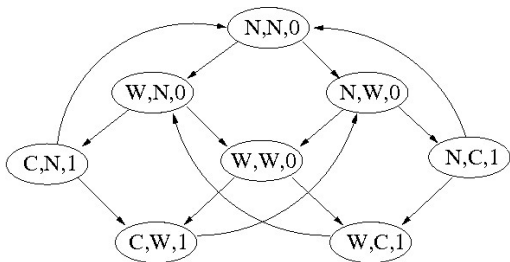
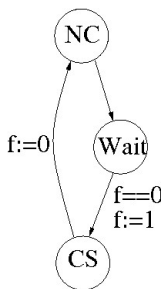
Vlk, koza a zelí



Koza a dvě zelí



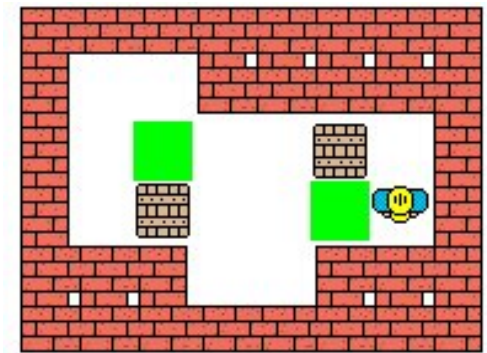
Verifikace protokolů



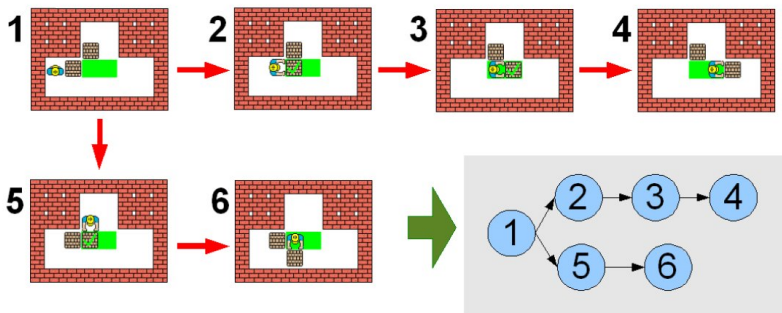
Počet stavů

- zmíněné úlohy mají jen pár stavů – lze nakreslit celý stavový prostor i ručně
- praktické problémy – mnoho stavů – nelze vygenerovat ani počítačem
- ⇒ heuristiky

Sokoban



Sokoban



Základní algoritmus

- stavový prostor = graf
- prohledávání grafu pomocí klasických prohledávání: BFS, DFS
- trochu sofistikovanější prohledávání:
 - IDA – iterative deepening
 - obousměrné hledání

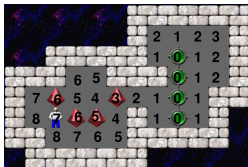
Praktické poznámky

- reprezentace – co je jeden stav (viz Sokoban)
- co potřebuji umět:
 - generování následníků stavu
 - ukládání stavů, test na přítomnost stavu (datový typ množina)

A* search

- úprava prohledávání do šířky (BFS)
- BFS postupuje rovnoměrně (povodeň)
- A* upřednostňuje “nadějnější” stavy
- heuristické ohodnocení stavů

Heuristika – příklad Sokoban

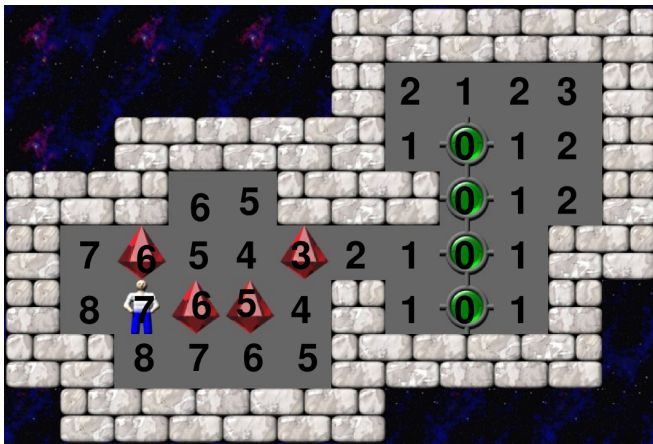


příklady:

- napočítat staticky vzdálenosti jednotlivých políček k cílovým polím, sečíst
- hledat minimální párování bedna – cílové pole
- brát v potaz polohu beden

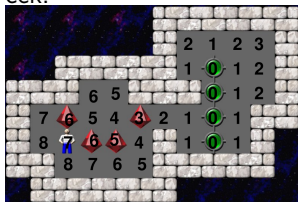
„trade-off“ přesnost heuristiky a časová náročnost jejího výpočtu

Heuristika – příklad Sokoban



Heuristika – příklad Sokoban

Vzdálenosti od cílových políček:



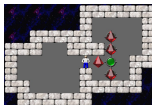
Ohodnocení: 16

$$= 1+4+5+6$$



Ohodnocení: 1

$$= 0+0+0+1$$

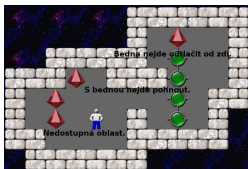


Ohodnocení: 11

$$= 0+1+4+6$$

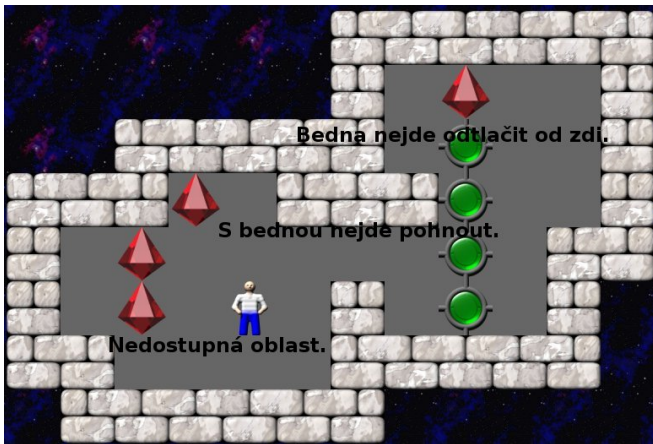


Ořezávání neperspektivních stavů



- dokážu rozhodnout, že ze stavu se nedá dostat do cíle \Rightarrow netřeba z tohoto stavu dál prohledávat
- příklad Sokoban:
 - bedna v rohu
 - bedna u zdi, od které se nemůže dostat

Ořezávání neperspektivních stavů



Hry dvou hráčů

uvažujeme hry:

- deterministické (bez náhody)
- perfektní informace
- alternující tahy
- konečné
- s nulovým součtem

příklady: šachy, go, piškvorky, hex, nim, ...

Hra Nim

jednoduchý příklad hry:

- máme n sirek (např. 20)
- povolený tah: odebrat 1, 2 nebo 3 sirky
- hráči se střídají
- kdo sebere poslední sirku vyhrává

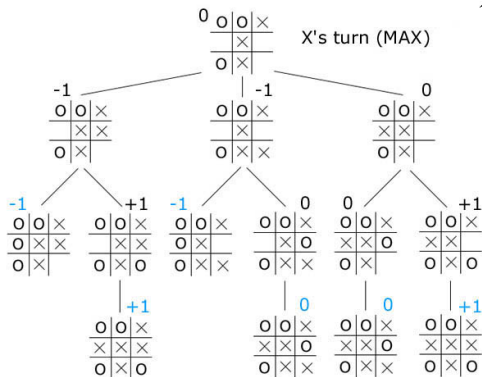
Graf hry a strategie

- stavový prostor
 - uzly = konfigurace hry, **označkování uzlů**: který hráč na tahu, výherní uzly
 - hrany = přechody podle pravidel
- strategie:
 - pro každý stav udává tah
 - výherní strategie: pro jakoukoliv strategii soupeře vyhrává

Minimax algoritmus

- ohodnocení stavů:
 - $+1$ = vyhrává první
 - -1 = vyhrává druhý
 - 0 = remíza
- počítáme „od spodu“ – od koncových (výherních) stavů směrem k počátečnímu
- podle toho, který hráč na tahu, počítáme minimum/maximum z následovníků
- vylepšení:
 - alfa-beta ořezávání
 - heuristické ohodnocení stavů

Ilustrace Piškvorky



Problém splnění podmínek

Constraint satisfaction problem (CSP)

- množina proměnných
- domény hodnot
- podmínky

Sudoku

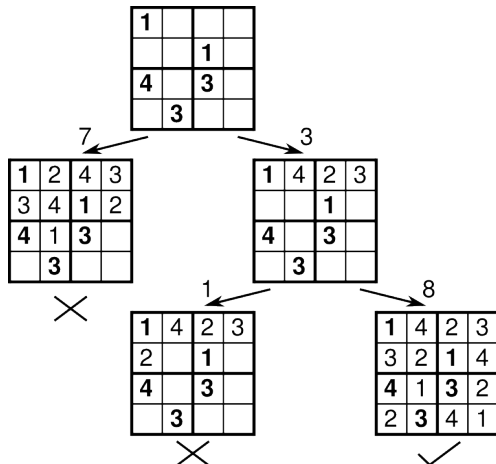
1	a	b	c
d	e	1	f
4	g	3	h
i	3	j	k

- proměnné: a, b, c, d, \dots
- domény hodnot: $\{1, 2, 3, 4\}$
- podmínky (pro j): $j \neq b, j \neq h, j \neq i, j \neq k, j \neq 3, j \neq 1$

Backtracking

- systematické hledání řešení metodou pokus-omyl
- začít s prázdným přiřazením hodnot proměnným
- postupně přiřazujeme proměnným hodnoty (systematicky)
- porušení podmínek – vrátit se zpět (backtrack) a zkusit jinou hodnotu
- podmínky kontrolujeme průběžně
- prohledávání do hloubky (DFS)

Backtracking a Sudoku



Heuristiky

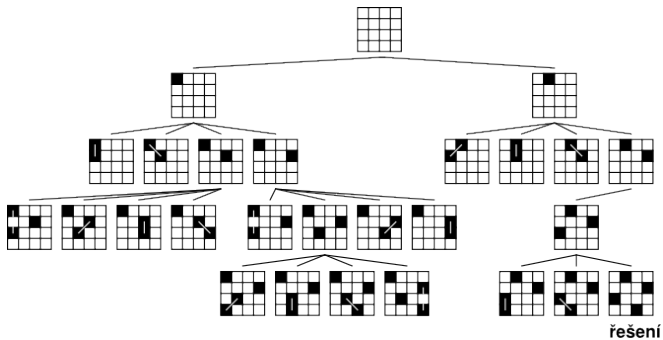
- forward checking, back jumping
- pořadí výběru proměnných

Příklad: dámy na šachovnici

- rozmístit n dam na šachovnici $n \times n$
- nesmí se vzájemně ohrožovat
- klasické zadání $n = 8$
- zkuste vyřešit pro $n = 4$
- zakreslete strom prohledání při backtrackingu

Backtracking

Čtyři dámy



Propagace podmínek

- pokus o „inteligentní“ řešení
- množina kandidátů pro každou proměnnou
- odvozování nutných hodnot
- efektivnější, ale nezaručuje řešení
- praxe: kombinace propagace podmínek a backtrackingu

Propagace podmínek

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

Další problémy CSP

- SAT – splnitelnost logických formulí
- obarvení grafu
- rozvrhování

Shrnutí

- hrubá síla
- prohledávání stavového prostoru, backtracking
- heuristiky – ohodnocení stavů, ořezávání neperspektivních, ...