Metody řešení úloh založené na prohledávání stavového prostoru - Neinformované metody

Neinformované metody nemají k dispozici žádnou informaci o cílovém stavu a nemají ani žádný prostředky, jak aktuální stavy hodnotit. Podobně metody musí někdy používat i člověk - například při prohledávání mapy, hledá-li cestu z nějakého (výchozího) místa do jiného (cílového) místa a nemá-li vůbec žádnou představu, ve kterém směru od výchozího místa cílové místo leží.

Metoda:	Obecně:	Vlastnosti algoritmu:	Užití Open/ Closed
BFS	Pro řešení složitějších úloh prakticky nepoužitelný. Přesto je algoritmus BFS považován za základní. Problém - generování již expandovaných uzlů. Eliminace užitím Closed, ale vetší „režije“ při výpočtech...	Je-li počet bezprostředních následníků každého uzlu konečný, algoritmus je **úplný a optimální**. **Časová i paměťová** náročnost je **exponenciální**.	Open nebo Open a Closed
UCS	Pracuje s podobným alg. jako metoda BFS, uvažuje však skutečné ceny přechodů (kladná čísla) a skutečné ceny cest (ceny + příslušné přechody). Pro expanzi vybírá ze seznamu OPEN uzel s nejmenším ohodnocením = uzel s min. cenou cesty.	Je-li počet bezprostředních následníků každého uzlu konečný, algoritmus je **úplný a optimální**. **Časová i paměťová** náročnost je **exponenciální**.	Open nebo Open a Closed
DFS	I když je algoritmus DFS opět velmi jednoduchý, může selhat i při řešení jednoduché úlohy. Nutné prakticky vždy použít modifikaci – úpravu bodu 5 algoritmu DFS tak, aby byly eliminovány uzly – předchůdci generovaného uzlu a uzly, které jsou již v zásobníku OPEN.	**Není ani úplný, ani optimální.** **Časová** náročnost **exponenciální**. **Paměťová** náročnost **je lineární** – O(b*m) , b – faktor větvení, m – maximální prohledávací hloubka stromu ('m' může být v některých případech nekonečné).	Jen Open, s Closed možné ale bezvýznamné
DLS	Pokud řešíme úlohu, u které dokážeme odhadnout hloubku řešení, můžeme problém s opakujícím se generováním stejných stavů vyřešit omezením hloubky Prohledávání.	**Není ani úplný, ani optimální.** **Časová** náročnost **exponenciální**. **Paměťová** náročnost **je lineární.**	Jen Open
IDS	Nelze-li stanovit hloubku řešení a nemáme-li k dispozici dostatek paměti pro použití metody BFS, můžeme se pokusit vyřešit tento problém použitím metody postupného zanořování do hloubky. Princip spočívá v opakovaném použití DLS s postupným zvyšováním hloubky prohledávání.	Má stejnou složitost, jako metoda BFS. Paměťová složitost **je lineární** - O(b*d). (d – hloubka/zanoření)	Jen Open
Backtracking	Metoda zpětného navracení je speciální metodou prohledávání do hloubky, kdy místo expanze vybraného uzlu, tj. generování všech jeho bezprostředních následníků, se generuje pouze následník jeden a teprve v případě neúspěchu se generuje následník další, atd.	**Není ani úplný, ani optimální.** **Časová** náročnost **exponenciální**. **Paměťová** náročnost **je extrémě nízká.**	Jen Open
BS	U úloh, které používají inverzní operátory je možné prohledávat stavový prostor oběma směry metodou BFS – od počátečního k cílovému stavu a současně od cílového stavu ke stavu počátečnímu.	Jako BFS. **Časová** náročnost **exponenciální**. Klesne paměťová složitost z O(bd) na O(2bd/2) ~ O(bd/2).	Open, nebo Open a Closed

Šedivě = neni uvedeno v opoře - 3.6 Shrnutí = „nemusíme umět“, ale kdo ví že....

Metody řešení úloh založené na prohledávání stavového prostoru - Informované metody

Informované metody mají k dispozici a používají nějakou informaci o cílovém stavu a mají prostředky, jak aktuální stavy prohledávání hodnotit. Právě podobné metody převážně používá i člověk – vrátíme-li se k příkladu prohledávání mapy, pak hledáme-li cestu z nějakého (výchozího) místa do jiného (cílového) místa máme obvykle alespoň přibližnou představu, ve kterém směru od výchozího místa cílové místo leží. Je zřejmé, že čím je naše představa o poloze cílového místa přesnější, tím menší oblast mapy musíme prohledávat.

Metoda:	Obecně:	Vlastnosti algoritmu:	Užití Open/ Closed
Best FS	Metody založené na výběru nejlépe ohodnoceného stavu (Best First Search) jsou velmi podobné neinformované metodě UCS.	Asi viz UCS.	Open
Greedy search	Ohodnocuje uzly pouze heuristickou funkcí, tj. odhadovanou cenou z daného uzlu do uzlu cílového - k expanzi vybírá uzel, který má toto hodnocení nejnižší. (Volně lze přeložit jako metoda lačného prohledávání.)	Viz DFS... **Není ani úplný, ani optimální.** **Časová** náročnost **exponenciální**. (Dobrá heuristika nicméně může časovou náročnost výrazně redukovat!) **Paměťová** náročnost **je lineární.**	Open
A*	Nejznámější a nejpoužívanější metodou pro řešení úloh prohledáváním stavového prostoru. K ohodnocení uzlů používá vztah f(n) = g(n) + h(n), kde heuristická funkce h(n) musí být tzv. spodním odhadem skutečné ceny cesty od ohodnocovaného uzlu k cíli – taková heuristika se pak nazývá přípustnou heuristikou (resp. heuristickou funkcí).	**Úplná** a pro přípustné heuristické funkce **i optimální.** **Časovou a prostorovou** náročnost výrazně ovlivňuje použitá heuristika.	NE

Metody řešení úloh založené na prohledávání stavového prostoru - Metody lokálního prohledávání

Existuje řada úloh, jejichž řešením je pouze nalezení cílového stavu a vlastní cesta je přitom bezvýznamná (rozmístění součástek na desce s plošnými spoji, rozmístění prvků v integrovaných obvodech, optimální rozložení zboží v regálech obchodů, optimalizace telekomunikačních sítí apod.). K řešení takových úloh se používají metody, které místo optimální cesty hledají optimální stav – tzv. metody lokálního prohledávání. Metody lokálního prohledávání neprohledávají a ani nemohou prohledávat stavový prostor systematicky, přesto mají dvě přednosti:

1. Mají zcela zanedbatelnou paměťovou náročnost.

2. Často dospějí k přijatelnému řešení v rozsáhlých (dokonce i ve spojitých, resp. nekonečných) stavových prostorech, kdy použití výše popsaných systematických algoritmů je nemožné.

Metoda:	Obecně:	Vlastnosti algoritmu:	Užití Open/ Closed
Hill-climbing	Používá k ohodnocení uzlu n, podobně jako Greedy search, pouze heuristiku h(n). Touto heuristikou je funkce, která ohodnocuje „kvalitu řešení“ spíše než vzdálenost od cílového řešení – pak čím lepší řešení ohodnocovaný uzel představuje, tím vyšší je mu přiřazena hodnota a vybírá se k expanzi uzel s nejvyšším ohodnocením.	Current & Next ... ?	NE
Simulované žíhání	Metoda simulovaného žíhání je stochastickou metodou, jejímž cílem je překonání lokálních extrémů vedoucích k častým neúspěchům metody Hill-climbing.	Current & Next ... ?	NE

Metody řešení úloh s omezujícími podmínkami

U úloh, které byly řešeny metodami založenými na prohledávání stavového prostoru, stačilo testovat, zda vybraný stav není stavem cílovým, a v opačném případě pak testovat použitelnost operátorů na tento stav. Úlohy s omezujícími podmínkami se od předcházejících úloh liší tím, že je nutné zkoumat i vnitřní strukturu jejich stavů.

Úloha s omezujícími podmínkami = **CSP** = *Constraint Satisfaction Problem*.

CSP lze řešit všemi metodami prohledávání stavového prostoru. Existují však speciální a výrazně efektivnější metody:

Metoda:	Obecně:	Vlastnosti algoritmu:	Užití Open/ Closed
Backtracking for CSP	Vkaždém kroku přiřazuje hodnotu neoznačené proměnné a pokud nový stav není legálním stavem, přiřazuje další hodnotu atd. Jinak se okamžitě vrací o krok zpět.	?	NE
Forward checking	Po každém přiřazení hodnoty nějaké proměnné vyřazuje z množin přípustných hodnot dosud volných proměnných ty hodnoty, které jsou s právě přiřazenou hodnotou v konfliktu.	?	NE
Min-conflict	Lokálním alg., který vychází z libovolného úplného (všem proměnným jsou přiřazeny nějaké hodnoty), ale nelegálního (přiřazené hodnoty nesplňují podmínky) stavu a snaží se zmenšovat počet konfliktů (tj. počet porušených podmínek). Překvapivě efektivní pro mnoho CSP.	?	NE

Metody založené na rozkladu úloh na podproblémy

Metody řešení úloh založené na rozkladu úlohy na podproblémy jsou přirozenými metodami, které při řešení obtížných problémů používá i člověk. Existují dva typy podproblémů. Problém OR -> A je řešitelný, je-li řešitelný alespoň jeden z podproblémů B, C, D, problém AND -> E je řešitelný, jsou-li řešitelné všechny podproblémy F, G a H. Plus se to může mísit a zanořovat.

Metoda:	Obecně:	Vlastnosti algoritmu:	Užití Open/ Closed
AO alg.	Pokud lze jednotlivé podproblémy ohodnocovat pomocí heuristické funkce, můžeme v každém uzlu OR „přepínat“ mezi řešenými podproblémy a řešit tak nejnadějnější podstrom. Ohodnocení bývá číselné (0=řešitelný, FUTILITY=ne). Informovaný AO algoritmus se nazývá AO* algoritmem.	?	Open a Closed

Metody hraní her

Hry se dvěma (proti)hráči, kteří se po jednotlivých tazích hry střídají. Problém spočívá v nalezení tahu hráče, který je právě na tahu (hráč A). Pro tohoto hráče bude problém považován za řešitelný, povede-li k jeho výhře alespoň jeden z jeho možných tahů (problém OR). Protože v dalším tahu táhne soupeř (hráč B), musí být pro každý tah vedoucí k výhře hráče A pro všechny tahy hráče B pro tohoto hráče neřešitelné, jinými slovy, všechny tyto tahy hráče B musí být řešitelné pro hráče A (problém AND). Hledání tahu vedoucího k výhře tak vede na klasické prohledávání AND/OR grafu.

Metoda:	Obecně:	Vlastnosti algoritmu:	Užití Open/ Closed
Easy hry	Prostě uděláme AND/OR strom, prohledáme do šířky a hloubky a třeba něco najdeme...	?	NE
MiniMax	Základem je rekurzivní procedura, která se zavolá pro aktuální stav hry a hráče A. Tato procedura vrací ohodnocení uzlu a pro hráče A i tah k uzlu s maximálním ohodnocením, tj. tah, který je v daném stavu hry pro tohoto hráče nejvýhodnější. Předpokládá se zadaná max. hloubka.	Zbytečně vyšetřované uzly	NE
Alfa a Beta	Alfa řezy zabrání zbytečnému vyšetřování tahů hráče A, beta řezy pak zbytečnému vyšetřování tahů hráče B. Tzn. rožšíření MiniMaxu.		
ExpectMiniMax je metoda pro hry s neurčitostí.			