

Použití algoritmu Monte Carlo Tree Search v plánování

*Seminář strojového učení a modelování
5. 12. 2013*

Mgr. Otakar Trunda



Motivace a cíle

- ▶ Využití Monte Carlo Tree Search (MCTS) pro plánování a rozvrhování
- ▶ Cíle:
 - prozkoumat možnosti použití MCTS v plánování
 - identifikovat případné problémy a navrhnout jejich řešení
 - výsledný systém implementovat a otestovat
- ▶ Návaznost na předchozí práci (problém Petrobras)
 - slibné výsledky na jedné plánovací doméně

Plánování

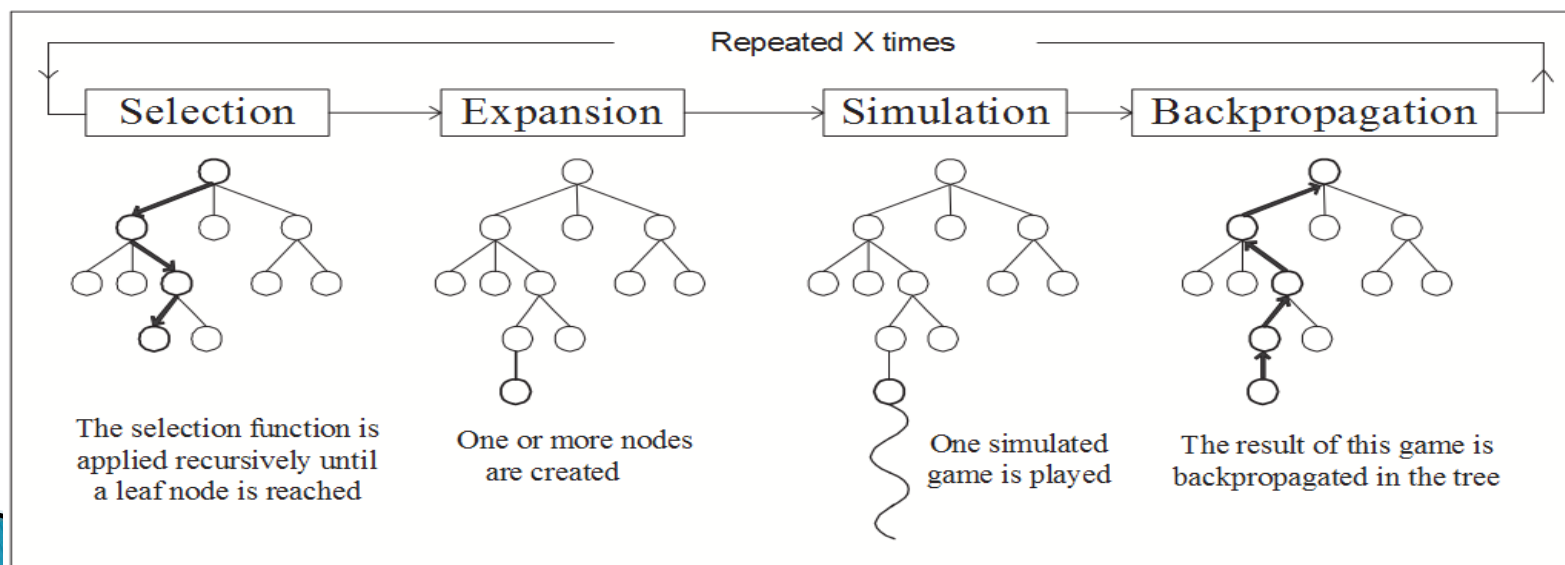
- ▶ **Zadání:**
 - popis počátečního stavu, koncových stavů a možných přechodů mezi stavy
- ▶ **Úloha (jako problém splnitelnosti):**
 - nalezení posloupnosti akcí, která vede z počátečního stavu do cílového stavu
- ▶ **Úloha (jako optimalizační problém):**
 - mezi všemi posloupnostmi vedoucími do cílového stavu nalézt tu nejlépe ohodnocenou
- ▶ Mnoho jiných typů dělení

Plánovací operátor – příklad

```
(  
: action load-cargo  
  parameters (?vessel – vessel ?c – cargo  
             ?loc – logistics_location)  
: precondition  
  (and  
    ( at ?vessel ?loc )  
    ( is-docked ?vessel ?loc)  
    ( cargo-at ?c ?loc)  
  )  
: effect  
  (and  
    (not ( cargo-at ?c ?loc) )  
    ( cargo-at ?c ?vessel)  
  )  
)
```

Algoritmus MCTS

- ▶ Anytime stochastický optimalizační algoritmus
- ▶ Stromové prohledávání stavového prostoru
- ▶ Náhodné vzorkování místo ohodnocovací funkce
- ▶ Postupně buduje asymetrický strom
- ▶ Skvělé výsledky v oblasti her (zejména Go, Hex, SameGame)



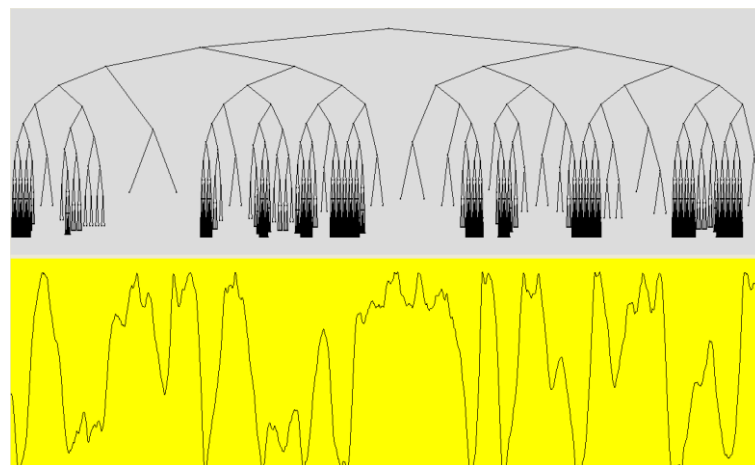
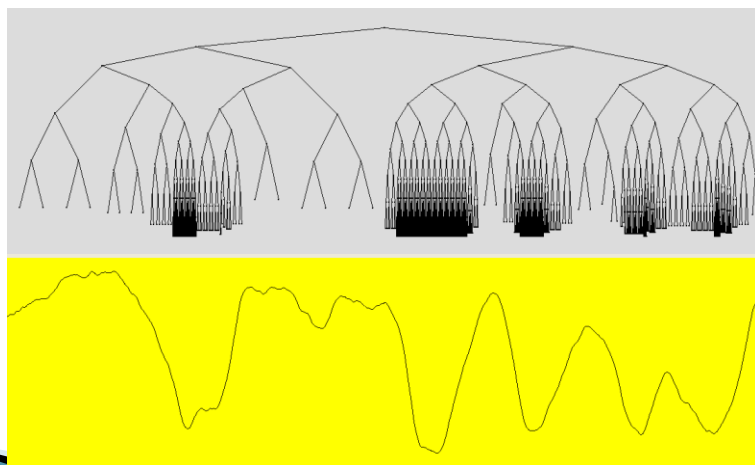
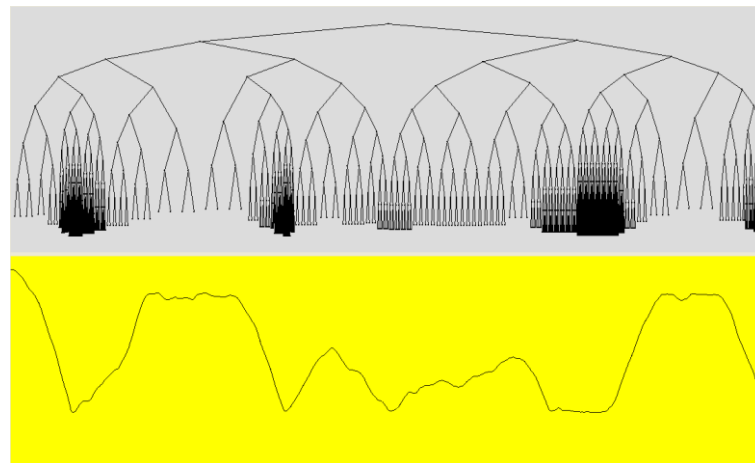
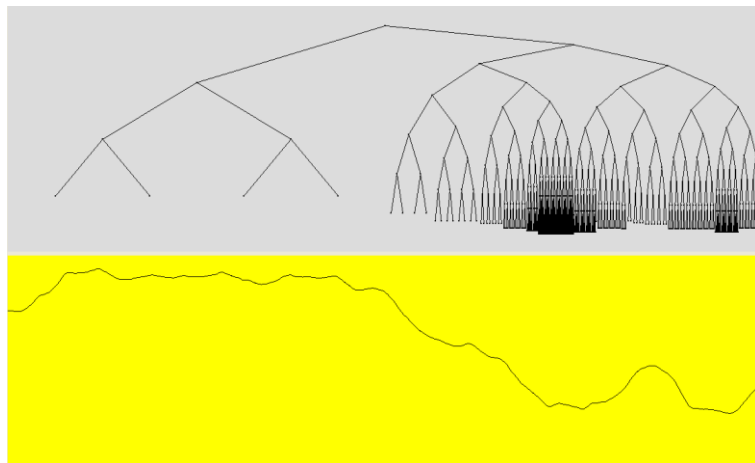
MCTS – selekce

- ▶ $v_i(N)$ – výsledek i -té simulace u vrcholu N
- ▶ $t(N)$ – počet simulací, které prošly vrcholem N
- ▶ Expectation: $\bar{v}(N) = \frac{\sum_{i=1}^{t(N)} v_i(N)}{t(N)}$

- ▶ Bias: $C \sqrt{\frac{2 \ln(t(N))}{t(N_j)}}$

- ▶ Směrodatná odchylka: $\sqrt{\frac{\sum_{i=1}^{t(N_j)} v_i(N_j)^2 - t(N_j) \bar{v}(N_j)^2}{t(N_j)}}$

MCTS – tvar stromu



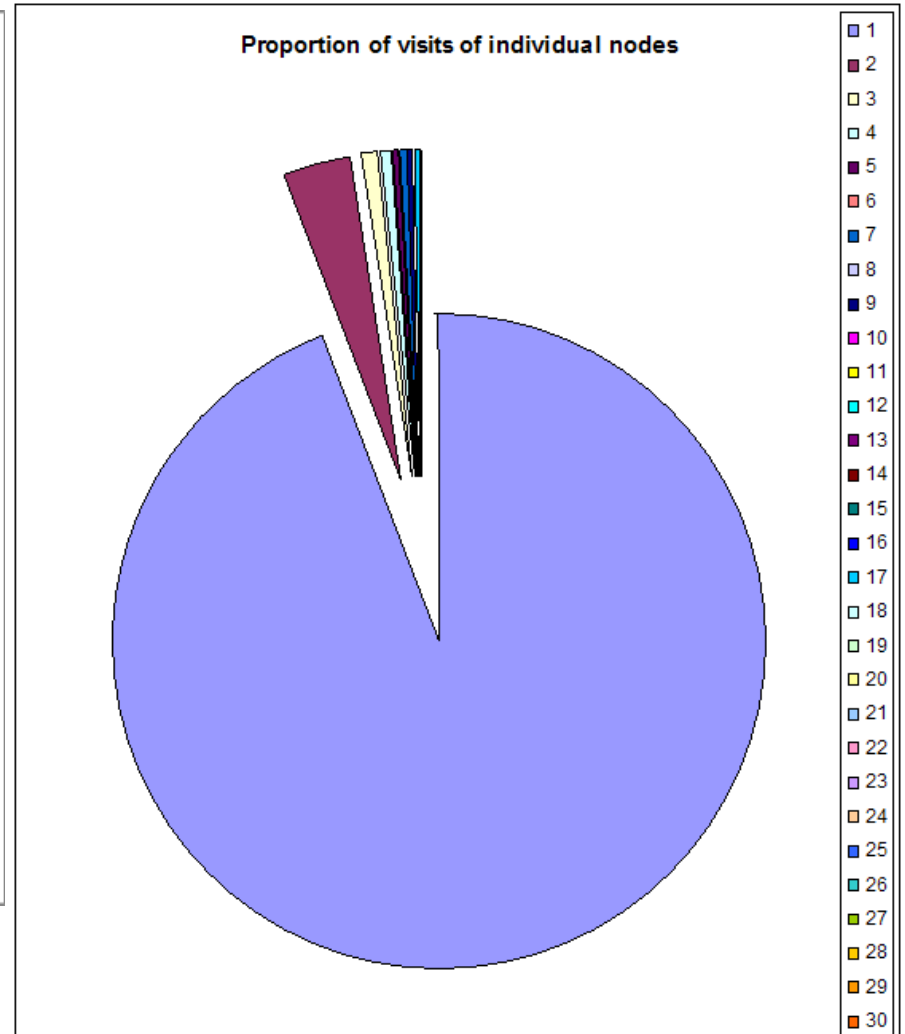
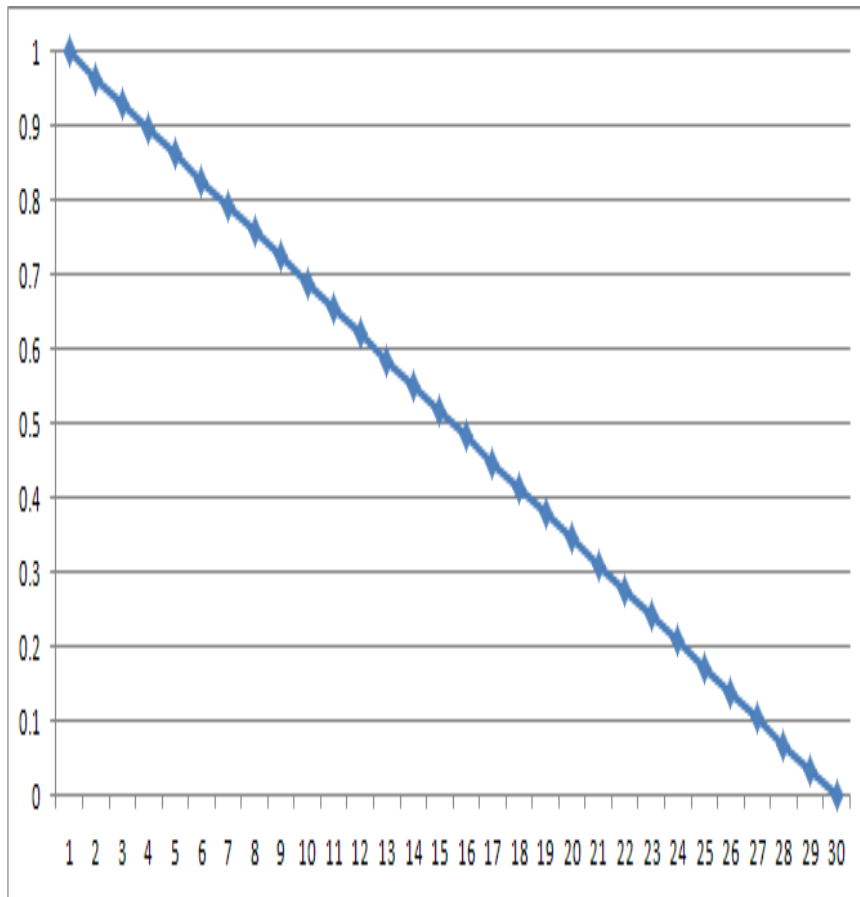
MCTS v plánování

- ▶ Možné problémy ve fázi simulace:
 - žádné omezení na délku simulace
 - hrozba uvíznutí ve slepé uličce
 - hrozba zacyklení
- ▶ Ve fázi selekce:
 - vychýlení odhadu hodnot vrcholů
- ▶ Ve fázi expanze:
 - příliš velké paměťové nároky

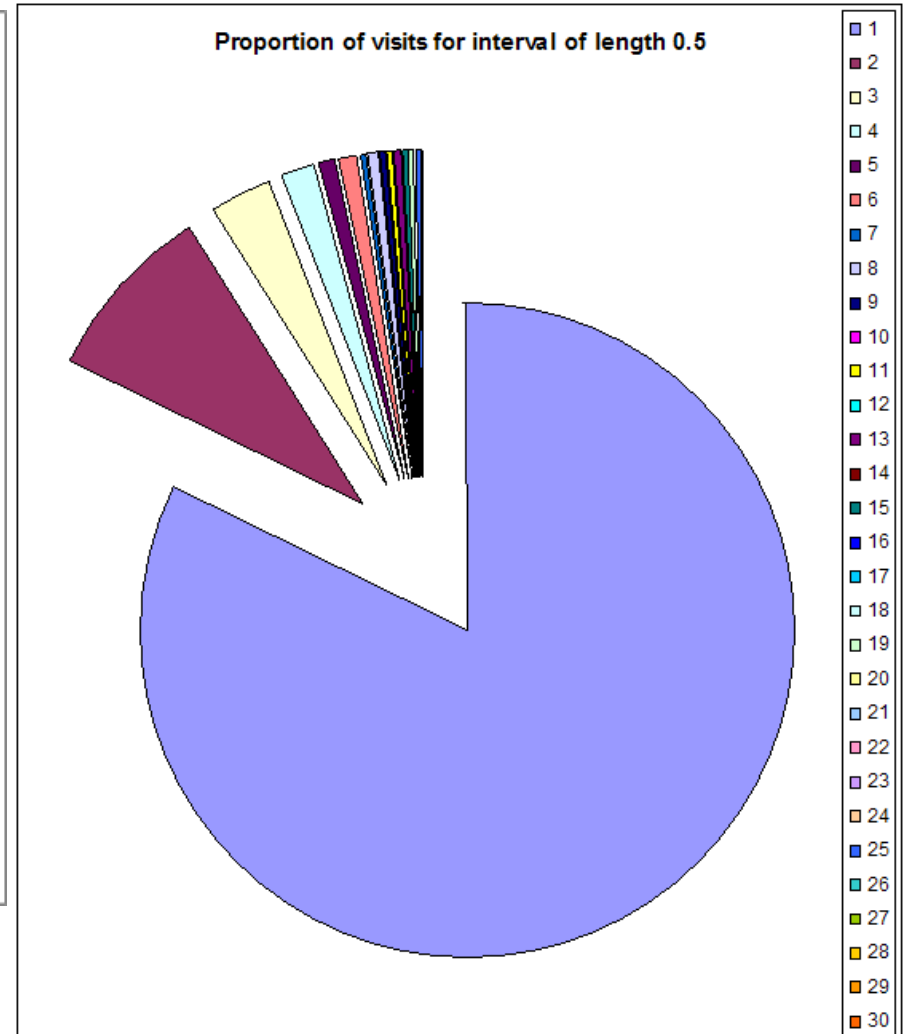
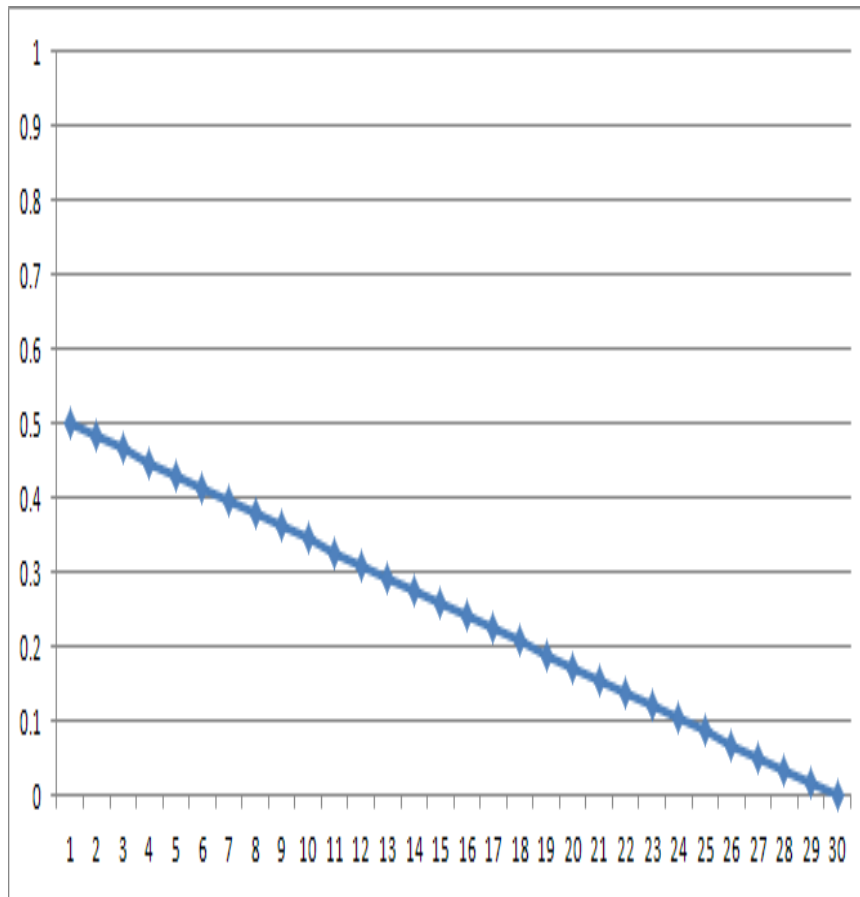
Problémy při selekci

- ▶ Mapování hodnoty *Expectation* do intervalu $[0, 1]$
- ▶ Neznámá dolní mez u *Expectation*
- ▶ Problémy při neúplném využití intervalu
 - Podobně jako u Go s handicapem
- ▶ Ohodnocení dead-end simulací
 - Při zachování efektivity

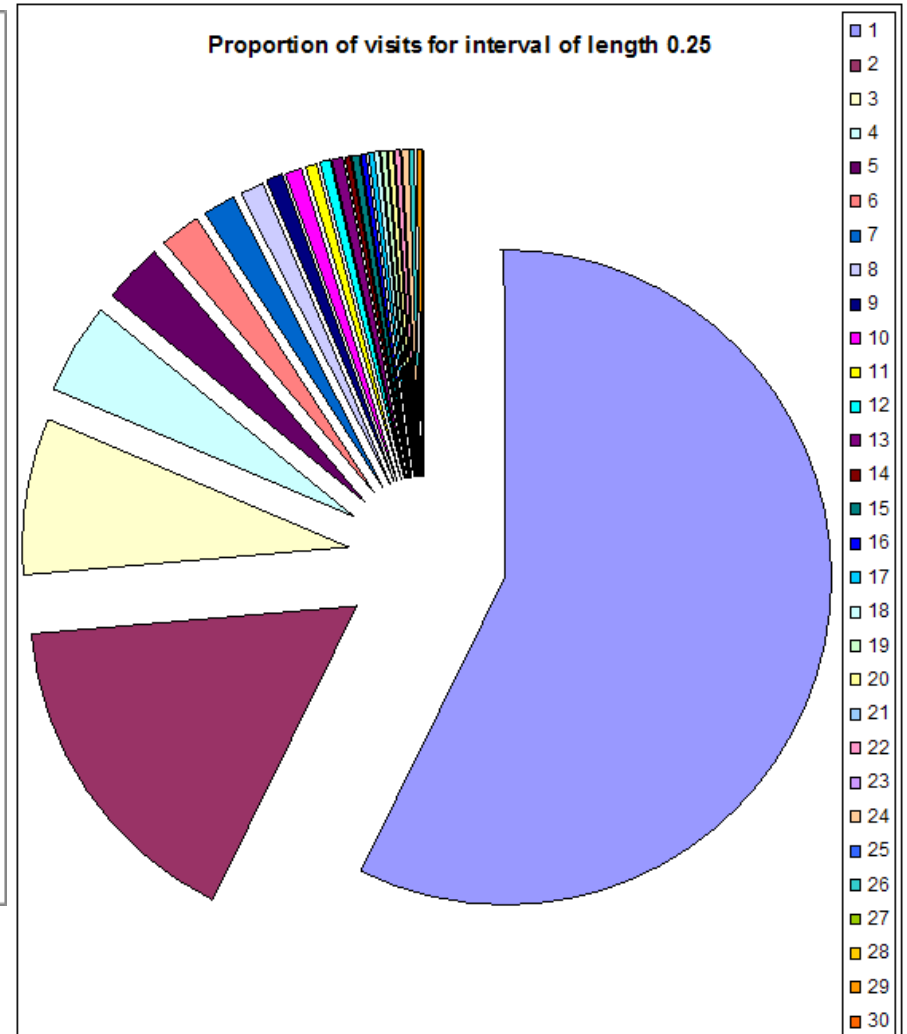
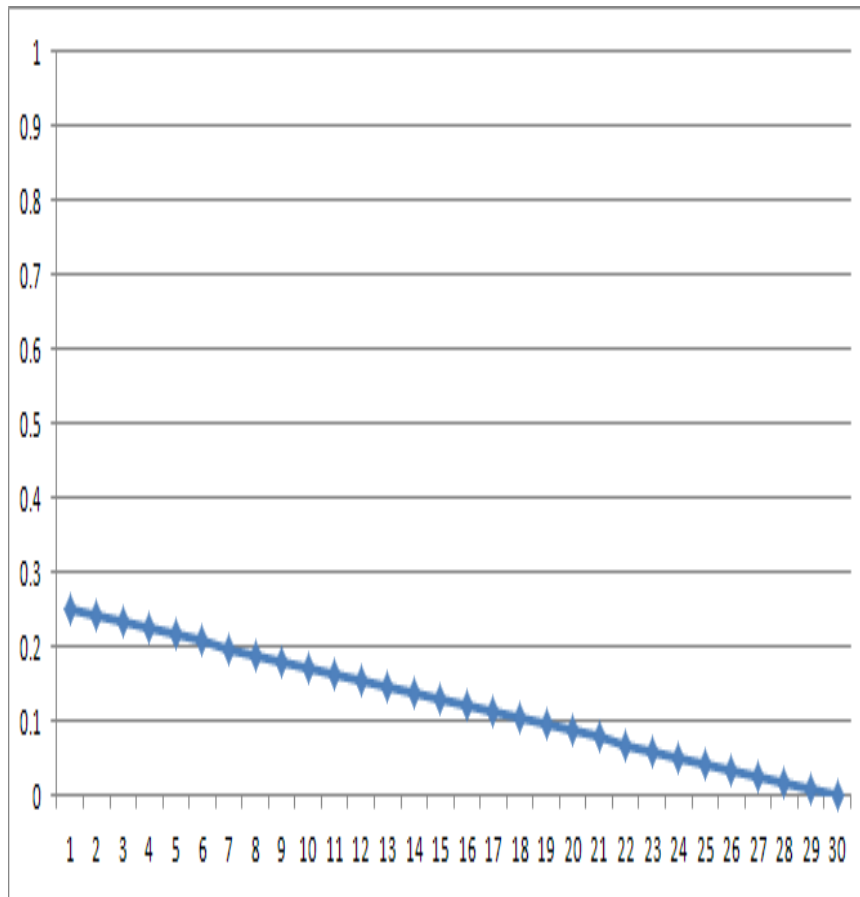
Interval délky 1



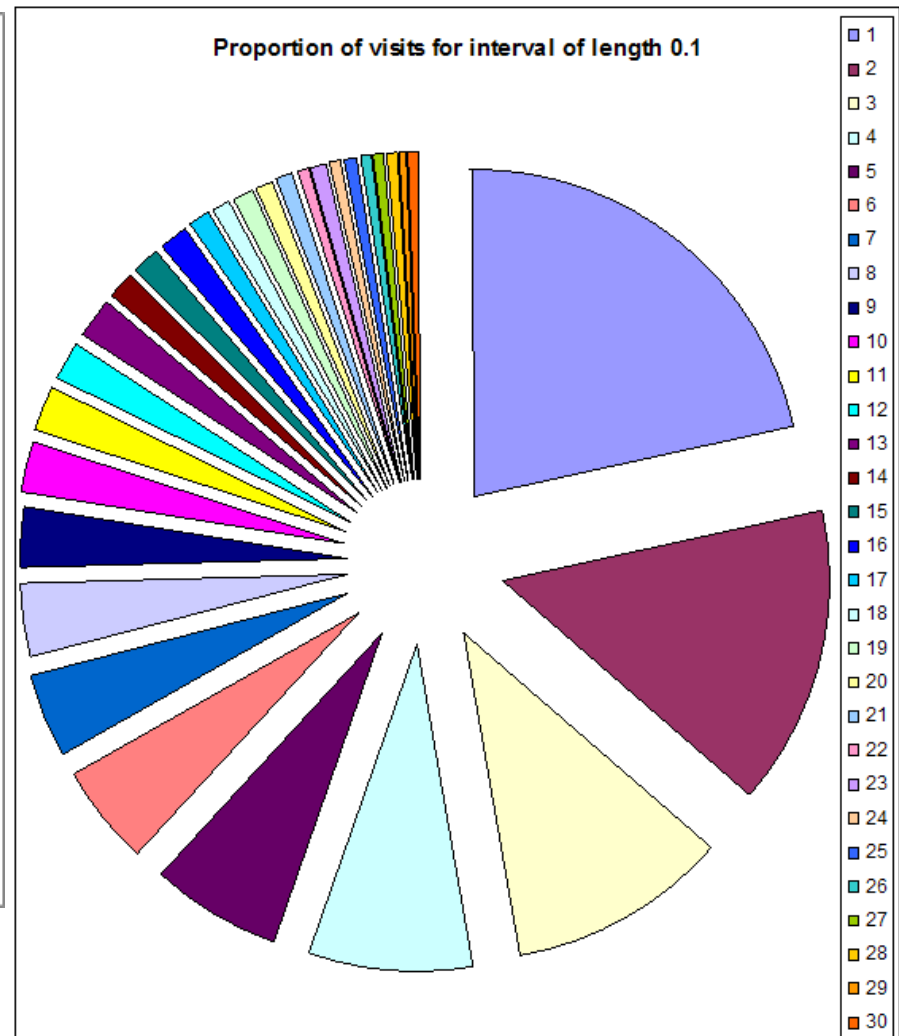
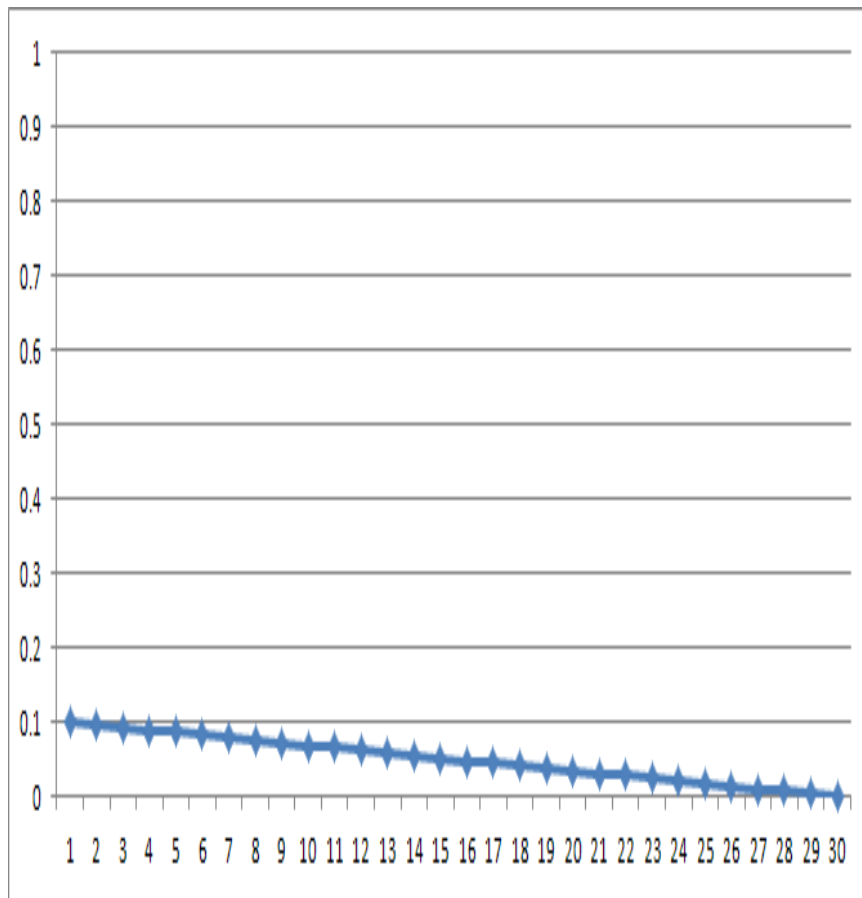
Interval délky 0.5



Interval délky 0.25



Interval délky 0.1



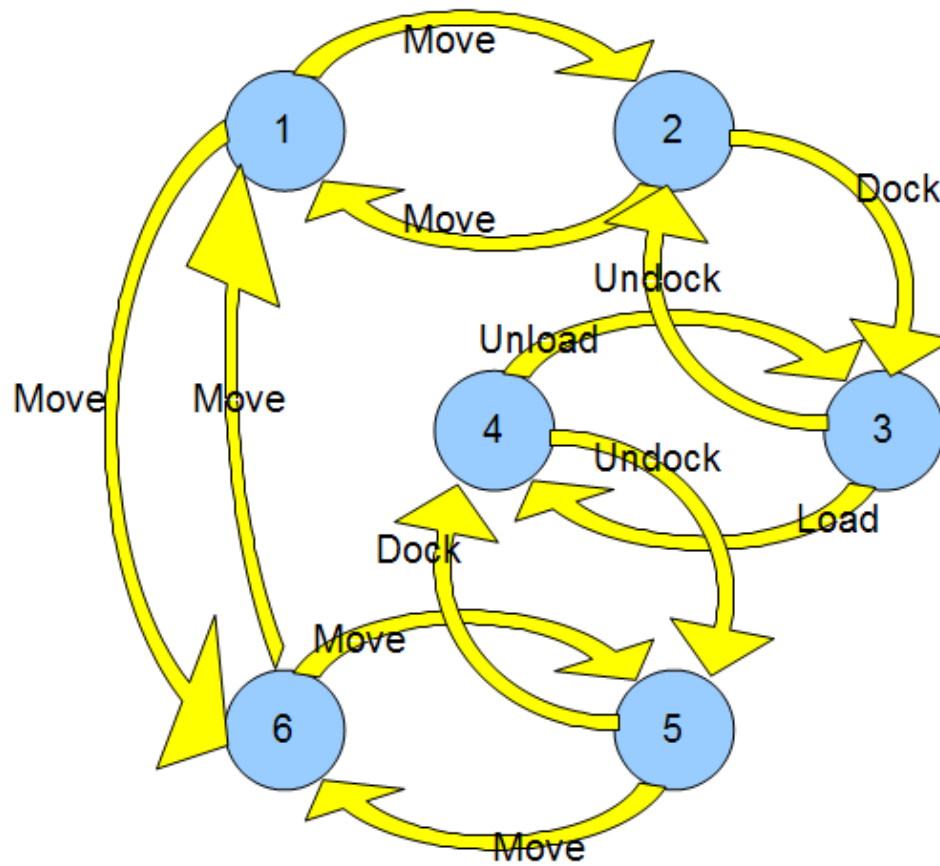
Návrh meta-akcí

- ▶ Řešení problémů pomocí úpravy stavového prostoru
- ▶ Nahrazení původních přechodů novými
- ▶ Meta-akce odpovídá posloupnosti původních akcí (tedy cestě v původním grafu)
- ▶ Cíl:
 - Zachování „dobrých“ cest, odstranění „špatných“
 - Zrychlení simulační fáze

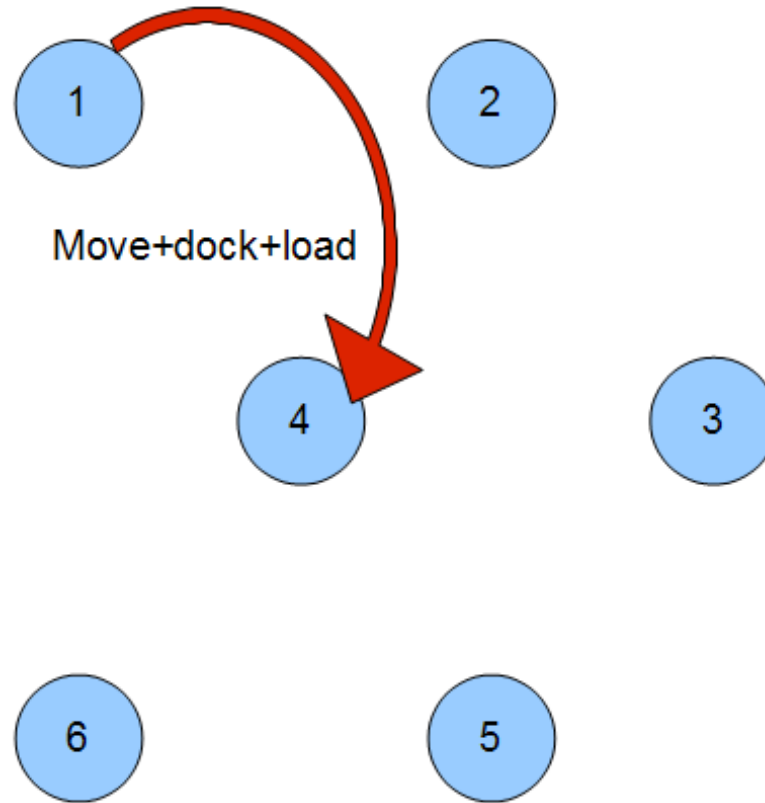
Příklad meta-akcí

- ▶ Původní akce:
 - Navigate, Dock, Undock, Load, Unload
- ▶ Některé možné sekvence:
 - Navigate, Navigate, ...
 - Dock, Undock, ...
 - Navigate, Dock, Load
- ▶ Nové akce (Meta-akce):
 - Navigate + Dock + Load
 - Undock + Navigate + Dock + Unload
 - ...

Graf přechodů – původní akce



Graf přechodů - meta-akce



Nalezení modelu meta-akcí

- ▶ Pro libovolnou doménu – těžký problém
- ▶ Omezíme se na plánování přepravy
 - Přirozeně optimalizační problém
 - Často se vyskytuje v praxi
- ▶ Využití pokročilých plánovacích technik
 - HTN-plánování, Landmarky
- ▶ Identifikace strukturálních vzorů v popisu domény
- ▶ „Lazy-learning“ pro učení meta-akcí v průběhu plánování
- ▶ Úplné prohledávání pro malé instance, extrakce znalostí

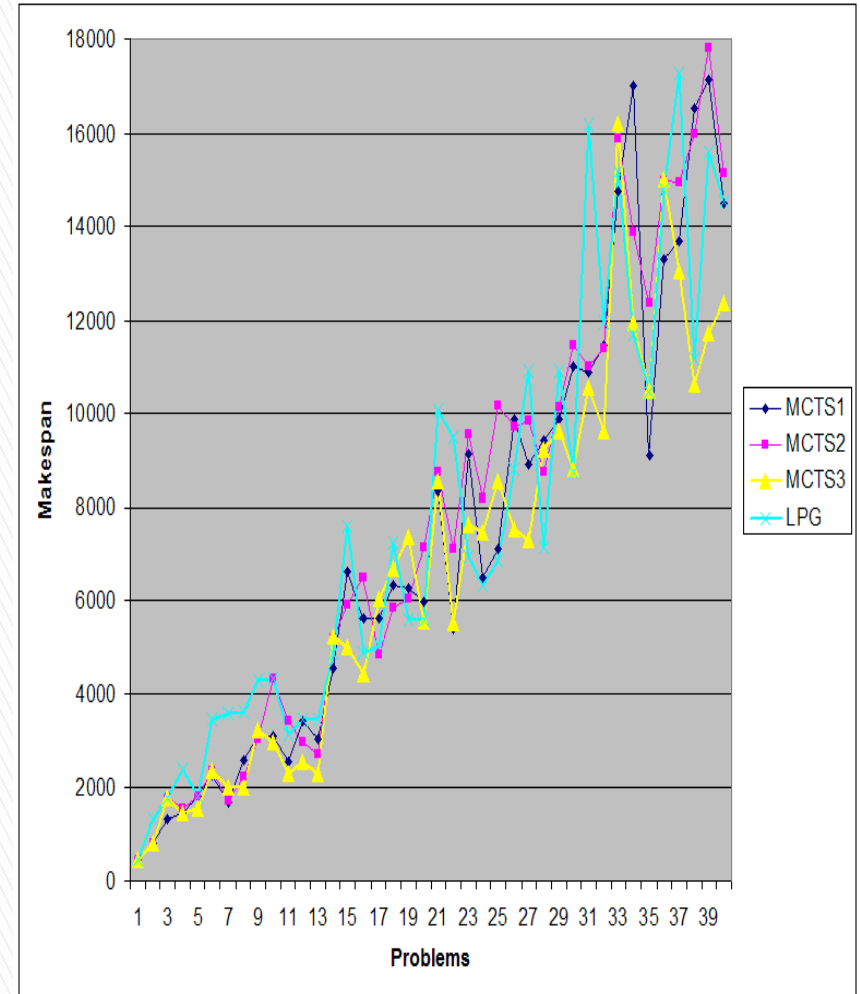
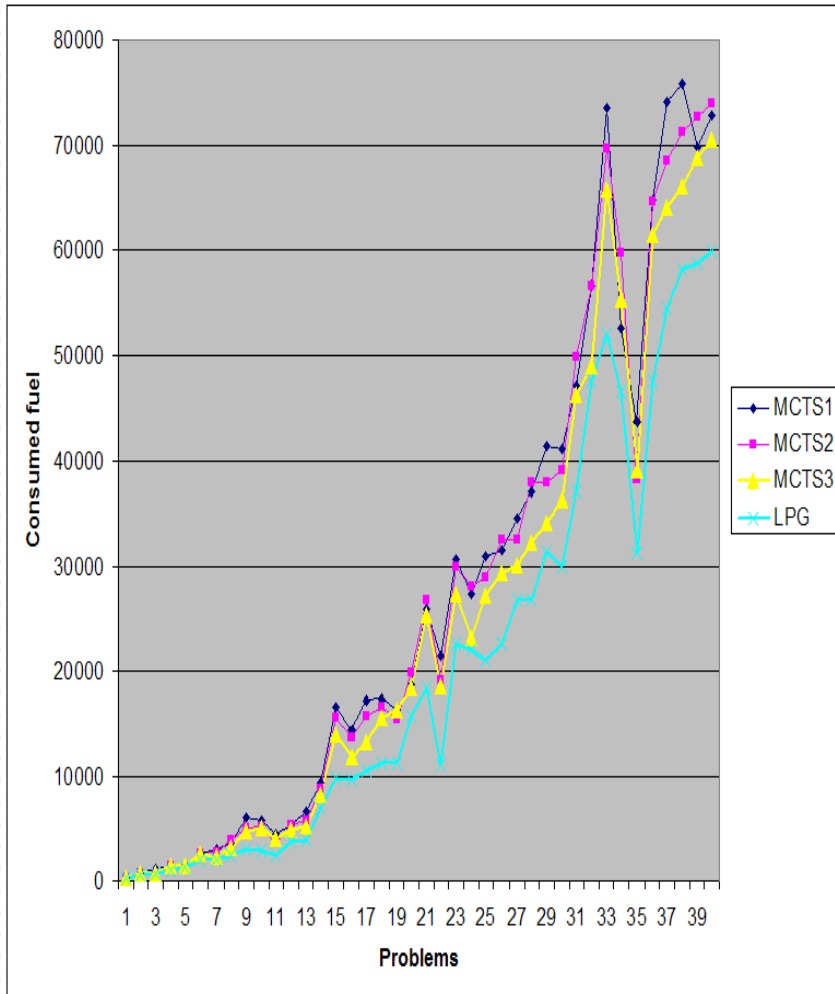
Transportační komponenty

- ▶ Syntaktický popis → sémantický popis
- ▶ Šablona pro „transportační strukturu“
 - Popsaná pomocí omezujících podmínek
- ▶ Hledání strukturálních vzorů v popisu domény
 - „Matchování“ domény na šablonu
- ▶ Využití technik CSP
 - Řešení CSP odpovídají transportačním komponentám

Shrnutí algoritmu

- ▶ Nalezení transportačních komponent
 - Jako předzpracování
- ▶ Použití algoritmu MCTS klasickým způsobem
- ▶ Během simulace používáme pouze meta-akce
- ▶ V případě, že simulace narazí na dead-end:
 - Najdeme nejbližší landmark
 - s využitím znalosti transportačních komponent
 - Najdeme cestu k nejbližšímu landmarku
 - úplné prohledávání na malé instanci
 - Uložíme tuto cestu ve formě meta-akce s budeme ji používat v dalším plánování
 - Extrakce znalostí, jejich další využití

Výsledky experimentů



Závěr

- ▶ Hlavní přínos:
 - Návrh plánovače založeného na MCTS
 - Technika pro analýzu domény – zachycení specifické struktury problému
 - Učení HTN bez učitele
 - Nový hybridní algoritmus (MCTS + plánovací techniky + učení)
- ▶ Další práce
 - Zobecnění techniky
 - Jiné způsoby pro analýzu domény
 - Použití jiných optimalizačních algoritmů