

Z · N · A · L · O · S · T · I



2013

Optimální kooperativní hledání cest řešení pomocí výrokové splnitelnosti

Pavel Surynek



Matematicko-fyzikální fakulta
Univerzita Karlova v Praze

Multi-robotické hledání cest

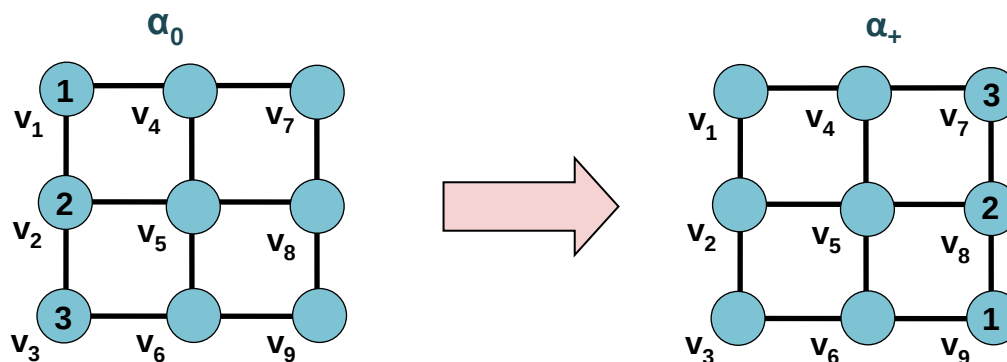
- **Mobilní roboty/agenty – všechny stejné, rozlišitelné**
 - počáteční pozice, cílové pozice
 - vyhnout se kolizím a překážkám
- **Prostředí - neorientovaný graf**
 - **Vrcholy** – pozice
 - **Hrany** – možnost pohybu na přilehlou pozici
- **CPF formálně** (*cooperative path-finding*):
 - Čtveřice $\Sigma = (G, A, \alpha_0, \alpha_+)$, kde:
 - $G=(V,E)$ je neorientovaný **graf**,
 - $A = \{a_1, a_2, \dots, a_\mu\}$, kde $\mu < |V|$ je **množina agentů/robotů**,
 - $\alpha_0: A \rightarrow V$ prostá funkce určující **počáteční rozložení** a
 - $\alpha_+: A \rightarrow V$ je prostá funkce určující **cílové rozložení**.

Pravidla a příklad

- Pravidla

- pohyb vždy do **volného** sousedního vrcholu
- žádné dva roboty do **stejného** vrcholu současně

- $\Sigma = (G, A, \alpha_0, \alpha_+)$



Řešení problému plánování cest,
kde $A=\{1,2,3\}$

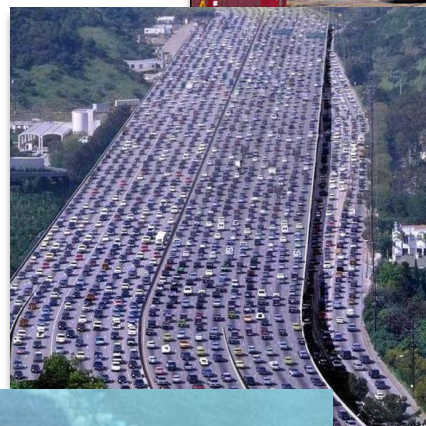
délka řešení=7

1 ...	$[v_1, v_4, v_7, v_8, v_9, v_9, v_9]$
2 ...	$[v_2, v_2, v_1, v_4, v_7, v_8, v_8]$
3 ...	$[v_3, v_3, v_3, v_2, v_1, v_4, v_7]$

Časový krok: 1 2 3 4 5 6 7

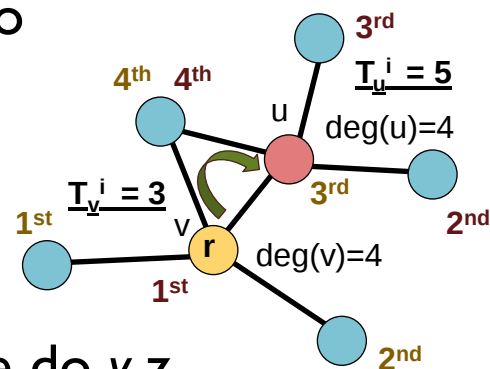
Motivace

- Přerovnávání kontejnerů
 - robot = **kontejner**
- Hustá doprava
 - robot = **automobil** (v zácpě)
- Přenos dat
 - robot = **datový paket**
- Vyhýbání plavidel
 - robot = **plavidlo**



Kódování CPF jako IP

- Rozhodujeme zda existuje plán délky k
 - kódujeme rozložení robotů v krocích $1, 2, \dots, k$
 - krok $1 \dots \alpha_0$
 - krok $k \dots \alpha_+$
- Celočíselné proměnné modelující krok i
 - $\mathbf{A}_v^i \in \{0, 1, 2, \dots, \mu\}$
 - $A_v^i = j$ když je robot a_j ve v v kroku i nebo
 - $A_v^i = 0$ je v prázdný v kroku i
 - $\mathbf{T}_v^i \in \{0, 1, 2, \dots, 2\deg(v)\}$
 - $0 < T_v^i \leq \deg(v)$ když robot opouští v do (T_v^i) -tého souseda
 - $\deg(v) \leq T_v^i \leq 2\deg(v)$ když robot vstupuje do v z $((T_v^i) - \deg(v))$ -tého souseda
 - $T_v^i = 0$ když ve v nenastává žádná akce
- Navíc podmínky pro návaznost kroků



Kódování CPF jako SAT

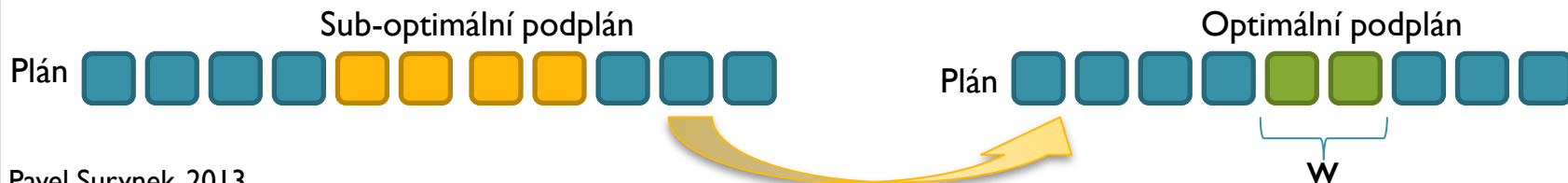
- Celočíselné proměnné (konečná doména)
 - nahradit bitovými vektory
 - například pro $\mathbf{A}_v^i \in \{0, 1, 2, \dots, \mu\}$
 - zavedeme $\lceil \log_2(\mu + 1) \rceil$ výrokových proměnných
 - přebytečné stavy zakážeme
 - $\Rightarrow$ **kompaktní reprezentace**

A 4-spojená mřížka 8×8	Kroky	SATPLAN kódování		SASE kódování		INVERSE kódování	
		Proměnné	Klauzule	Proměnné	Klauzule	Proměnné	Klauzule
4	8	5.864	55.330	11.386	53.143	5.400	38.800
8	8	10.022	165.660	19.097	105.724	5.920	48.224
12	8	14.471	356.410	26.857	168.875	5.920	46.176
16	10	30.157	1.169.198	51.662	372.140	8.122	76.192
24	10	43.451	2.473.813	73.101	588.886	8.122	71.072
32	14	99.398	8.530.312	157.083	1.385.010	12.396	137.120

- $\Rightarrow$ **těžká na vyřešení** $\Rightarrow$ SAT Competition 2013

Hledání optimální délky plánu

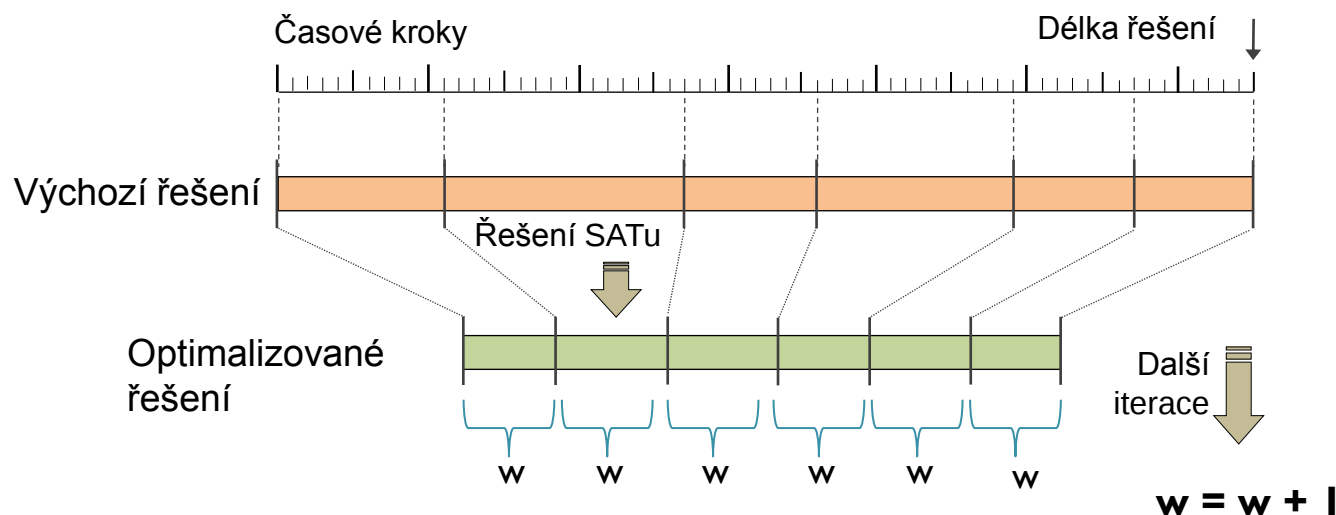
- **Zdola** (zavedeno v SATPLANu)
 - testovat počet kroků $k = 0, 1, 2, \dots$
 - zakódovat dotaz – existuje plán délky k
 - položit dotaz SAT řešiči
 - **neúplnost**
- **Shora** (nové, algoritmus COBOPT)
 - vygenerovat **sub-optimální** řešení
 - lokálně zlepšovat
 - pro velikost okna $w = 2, 3, 4, \dots$
 - nahrazování sub-optimálních podplánů **optimálními délkou w**



Strategie nahrazování

- **Hladově**

- Dána velikost optimalizačního okna w
 - **nejdelší podplán** nahraditelný optimálním délky w

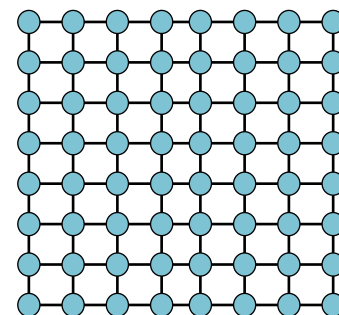


- Jestliže $w = \text{délka plánu}$
 - důkaz, že plán je optimální
- **Anytime** vlastnost
 - kdykoli můžeme vrátit nějaké (sub-optimální) řešení

Experimentální vyhodnocení

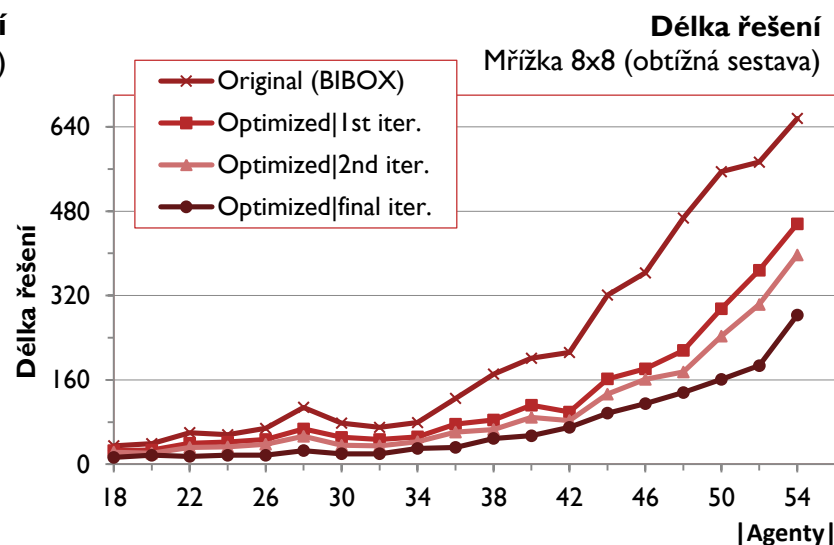
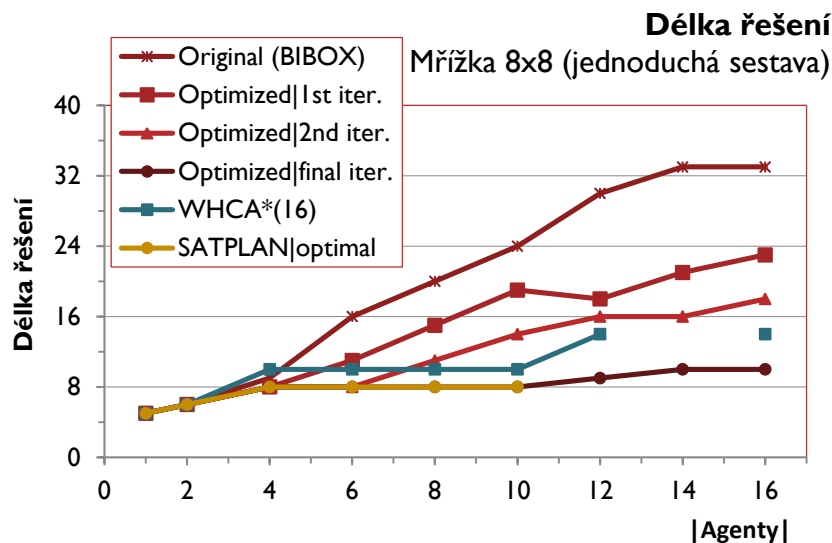
- **Testovací instance**

- 4-propojená mřížka velikosti 8×8 a 16×16
- náhodné počáteční a cílové rozložení



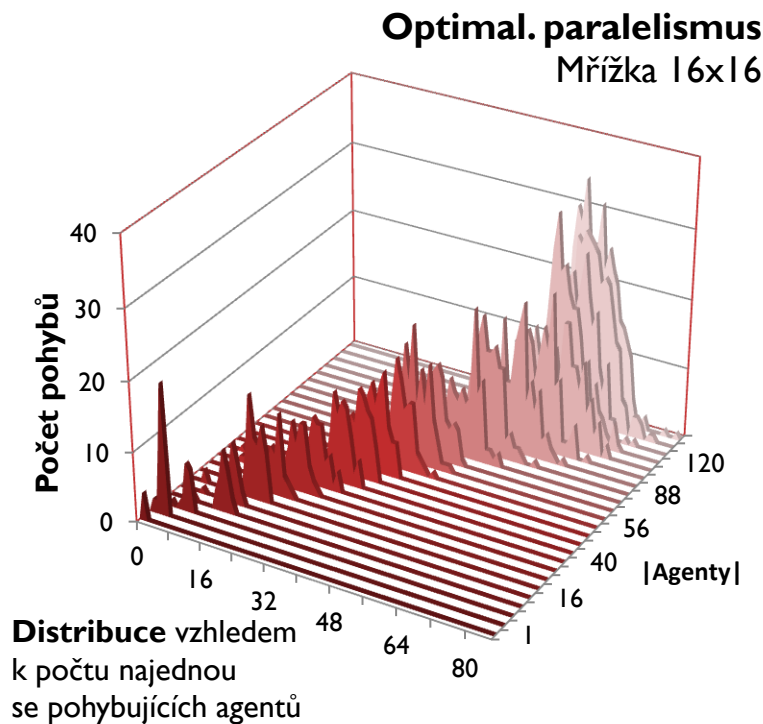
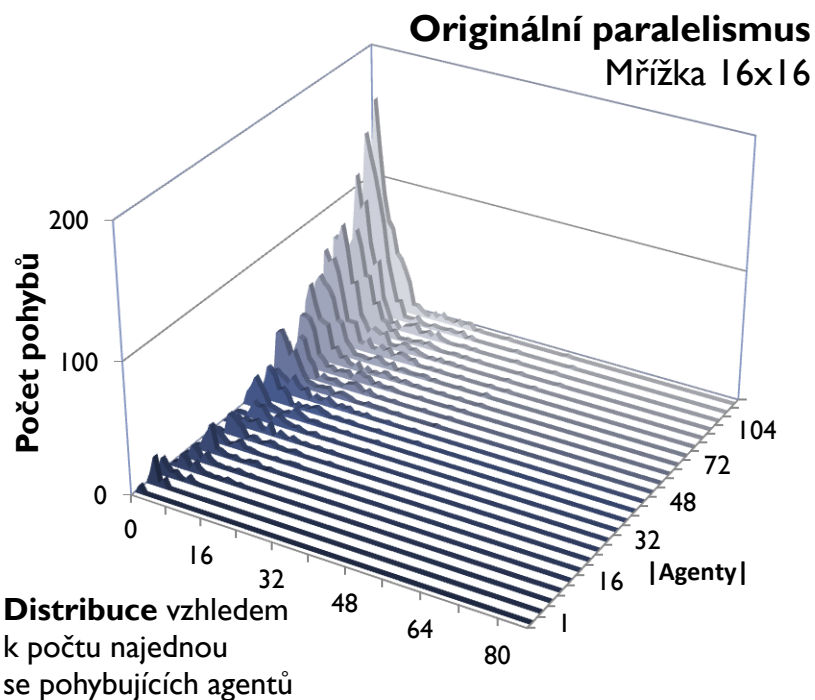
- **Porovnávané algoritmy**

- SATPLAN, WHCA*, COBOPT



Parallelismus

- Vlastnosti optimalizovaných řešení
 - **parallelismus**
 - kolik agentů najednou se pohybuje



Závěrečné poznámky

- **Výhody** návrhu
 - využití silných urychlovacích technik SAT řešičů
 - možnost snadné paralelizace
 - různé části řešení lze optimalizovat nezávisle
 - modularita
 - lepší počáteční řešení – celkové zlepšení výkonu
- **Nevýhody**
 - obtížnost zakódovaných instancí
 - energeticky náročná řešení
 - rychlý růst velikosti kódování
 - velké grafy
 - mnoho kroků