

# Constraint Programming in Planning

Omezující podmínky  
v plánování



**Pavel Surynek**

Katedra teoretické informatiky a  
matematické logiky

Matematicko-fyzikální fakulta  
Univerzita Karlova v Praze

[pavel.surynek@mff.cuni.cz](mailto:pavel.surynek@mff.cuni.cz)

# Přehled prezentace

- **Motivace k plánování**
- Intuitivní **definice** problému plánování
- **Existující** přístup k řešení
  - Plánovací graf a jeho nedostatky
- **Nové** přístupy
  - Použití hranové konzistence
  - Návrh specializované konzistence
  - Identifikace problémů řešitelných v polynomiálním čase
  - Vše provázeno experimenty
- **Aplikace** navržených technik v **booleovské splnitelnosti**
  - Předzpracování problému
- **Závěr**



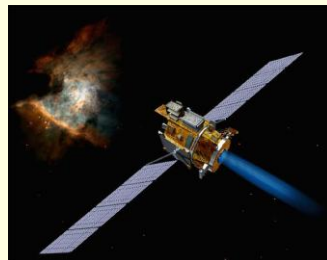
# Motivace pro plánování

## ■ Potřeba určit činnosti pro **autonomního robota**

### ■ Armáda



### Výzkum vesmíru



## ■ Určování **výrobního postupu**



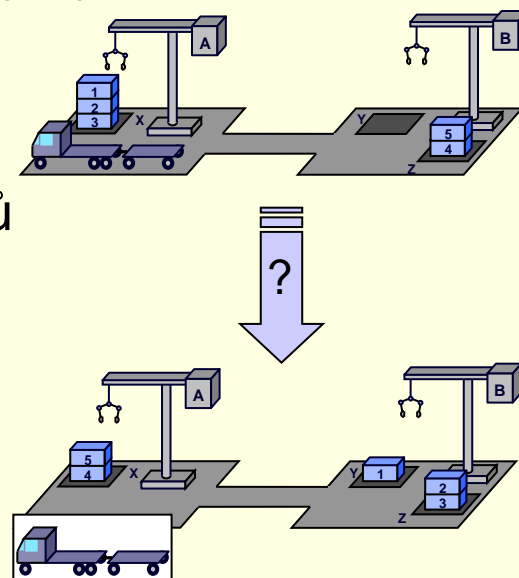
## ■ Organizace **záchranných operací**

## ■ Optimalizace **dopravy**



# Problém hledání plánu

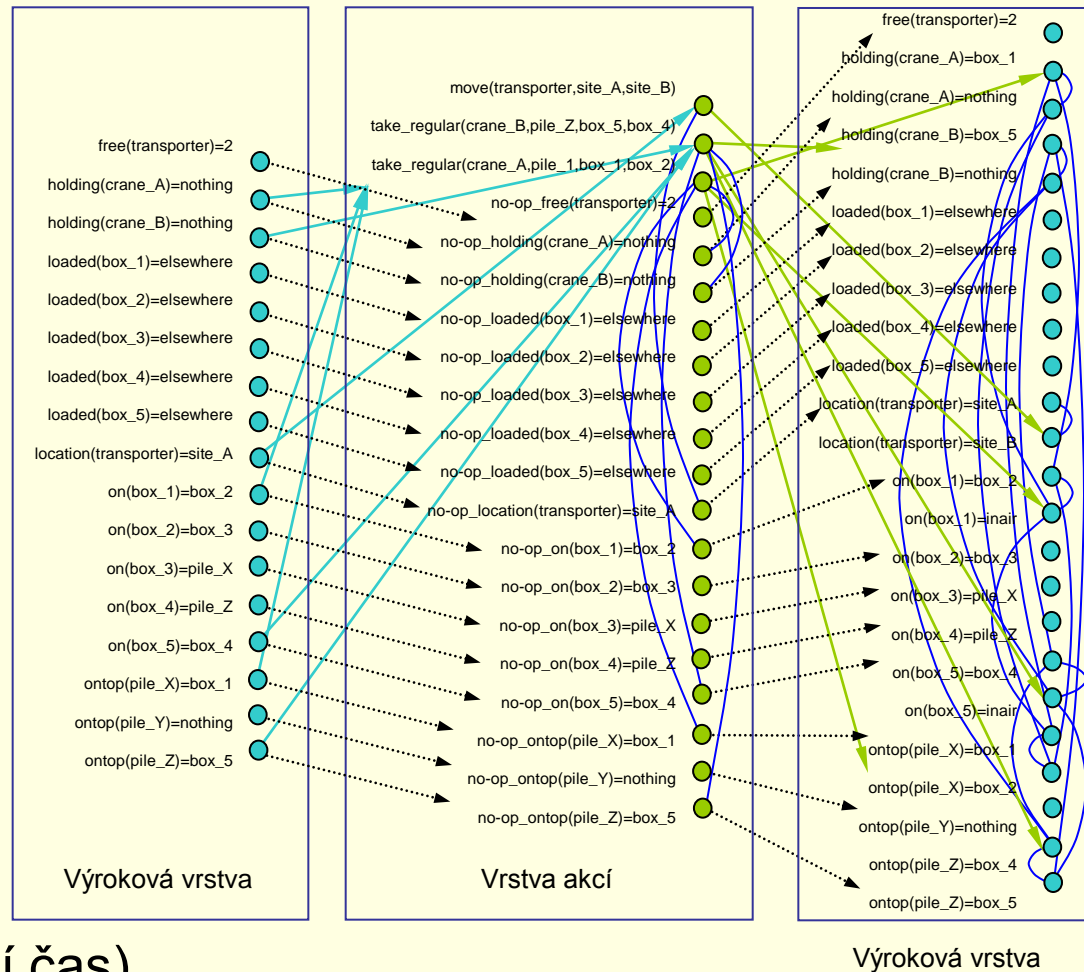
- Deterministická, plně pozorovatelná, statická abstrakce reality - **plánovací svět**
- Úloha nalézt **sled akcí (plán)**, které převedou plánovací svět z **počátečního** do **cílového** stavu
  - **Počáteční stav** = množina *základních* atomů pro **úplný popis** plánovacího světa (př.:  $on(box1, box2)$ ,  $truck\_free()$ )
  - **Cílový stav** = množina *základních* výroků (př.:  $\neg on(box1, box4)$ ,  $on(box2, box3)$ )
  - Množina povolených **akcí**, kde akce má
    - **Předpoklad** = množina výroků, které musí platit, aby šla akce použít
    - **Efekt** = množina výroků, které budou nově platit po aplikaci akce



# Existující přístup: plánovací graf a paralelismus

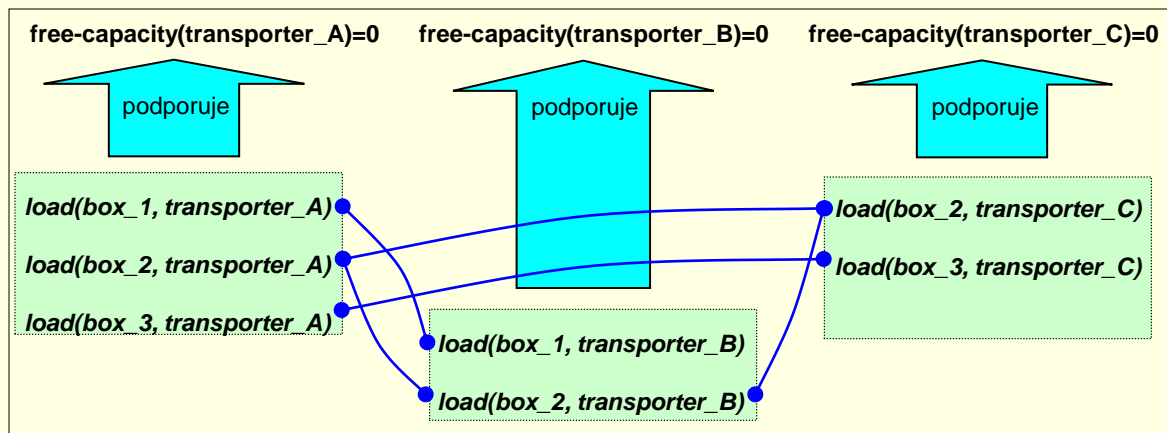
■ **Plánovací graf**=graf odvozený z problému plánování

- Rychlá (polynomiální čas) odpověď, zda **cíl dosažitelný** pomocí zadaného počtu časových kroků
- V každém časovém kroku lze **paralelně** provést **několik bezkonfliktních akcí**
- Cíl dosažitelný  $\Rightarrow$  **extrakce plánu** - pomalé (exponenciální čas)



# Nový přístup (1): použití hranové konzistence

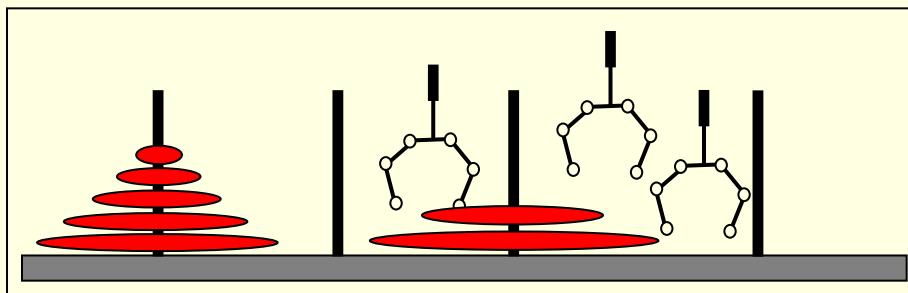
■ Při extrakci plánu opakované **hledání podporujících akcí** pro podcíle (induktivně vznikají), existující přístup (GraphPlan) používá neinformované prohledávání



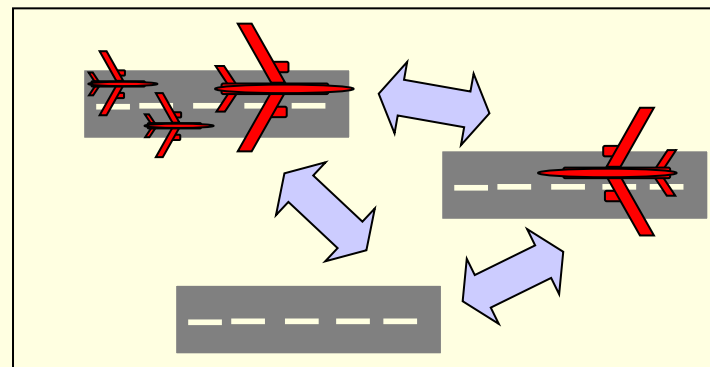
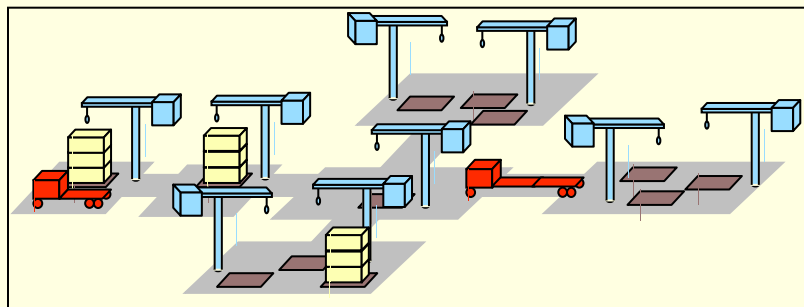
- *NP*-úplný problém (převodem **SATu** na **problém podpor**), navíc potřebujeme enumerovat více (až všechna) řešení
- **CSP** techniky efektivní pro *NP*-úplné problémy podobného typu
- Problém hledání podpor **modelujeme jako CSP** + použijme udržování **hranové konzistence** (*Arc-Consistency, AC*)

# Experimenty: plánovací problémy pro testování

- Zobecněná varianta **hanojských věží** (používáme více manipulačních rukou)



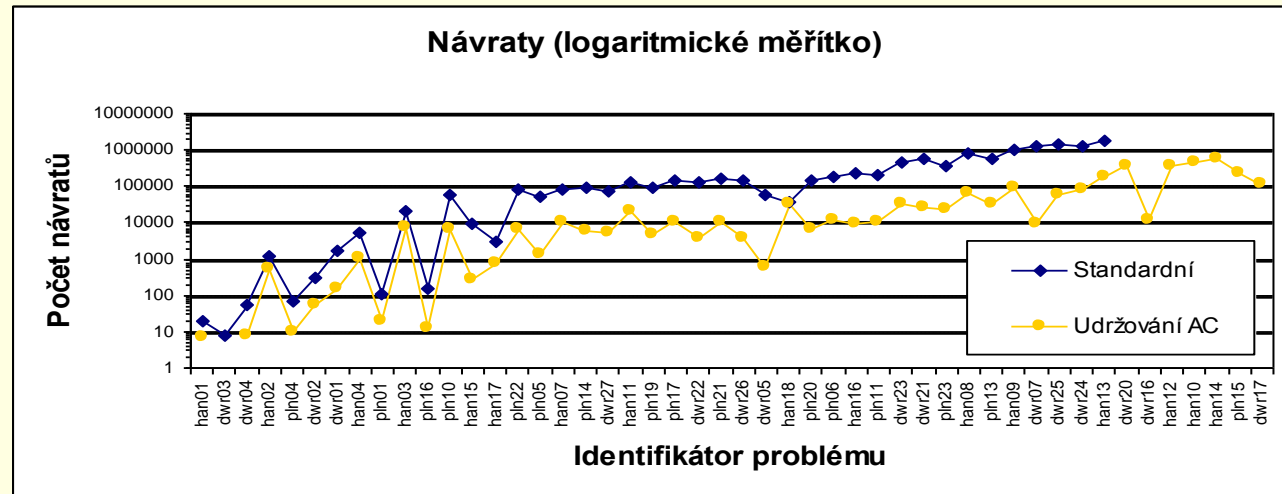
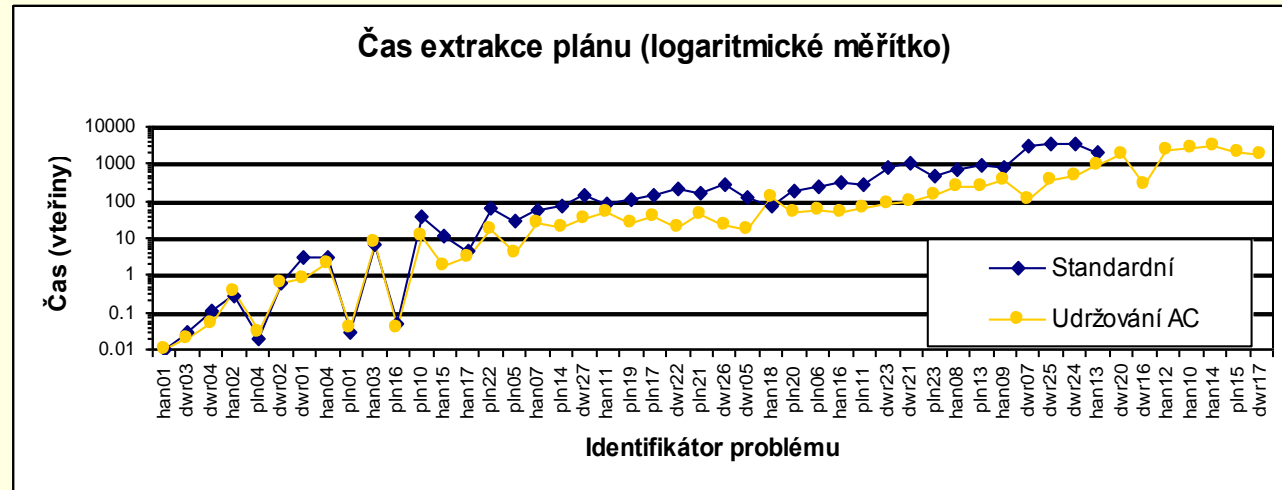
- **Roboty v docích** (Ghallab et al., 2004)



- **Letadla tankující za letu**

# Experimenty: výsledky pro AC

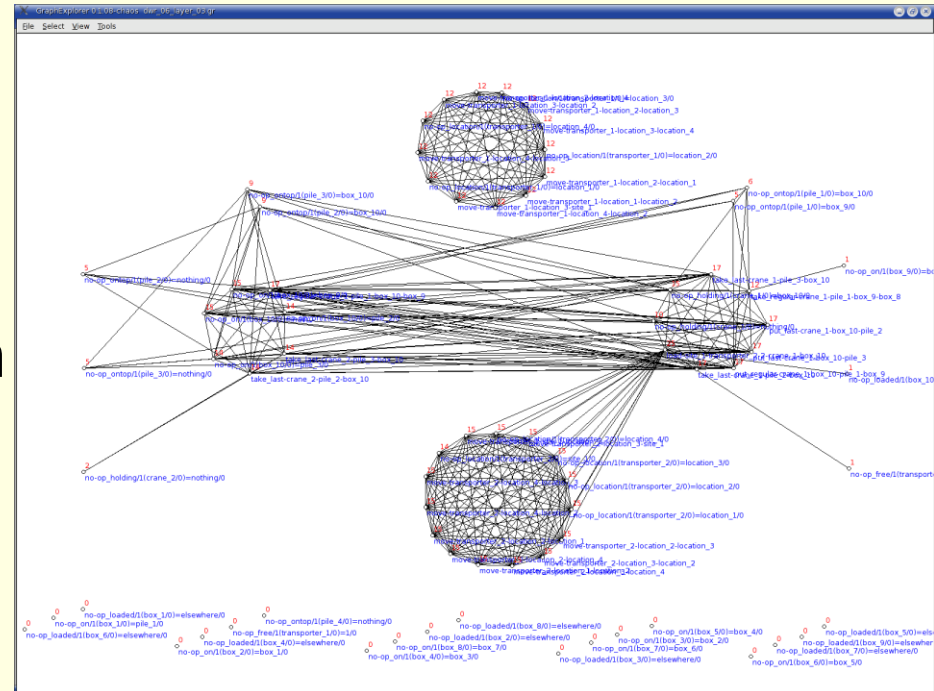
- Až násobné zvýšení výkonu při udržování AC
- Čím větší paralelismus, tím větší zlepšení
- Hanojské věže s jednou rukou - téměř žádné zlepšení
- Roboty v docích - zrychlení až 16x





# Nový přístup (2): návrh projektivní konzistence

- Interpretace problému hledání podpor jako **grafu**
- Graf je **strukturovaný** - nalezení **rozkladu** na vrcholově disjunkttní úplné podgrafy (**kliky**) ...  $C_1, C_2, \dots, C_k$  (hladově)
- Z každé kliky **nejvýše jedna akce** do řešení
- Definujeme **příspěvek** akce **a**  
 $c(a)$ =počet podporovaných atomů (atomy v efektu)
- **Příspěvek** kliky **C**  
 $c(C) = \max_{a \in C} c(a)$
- $\sum_{i=1 \dots k} c(C_i) \geq$  velikost cíle

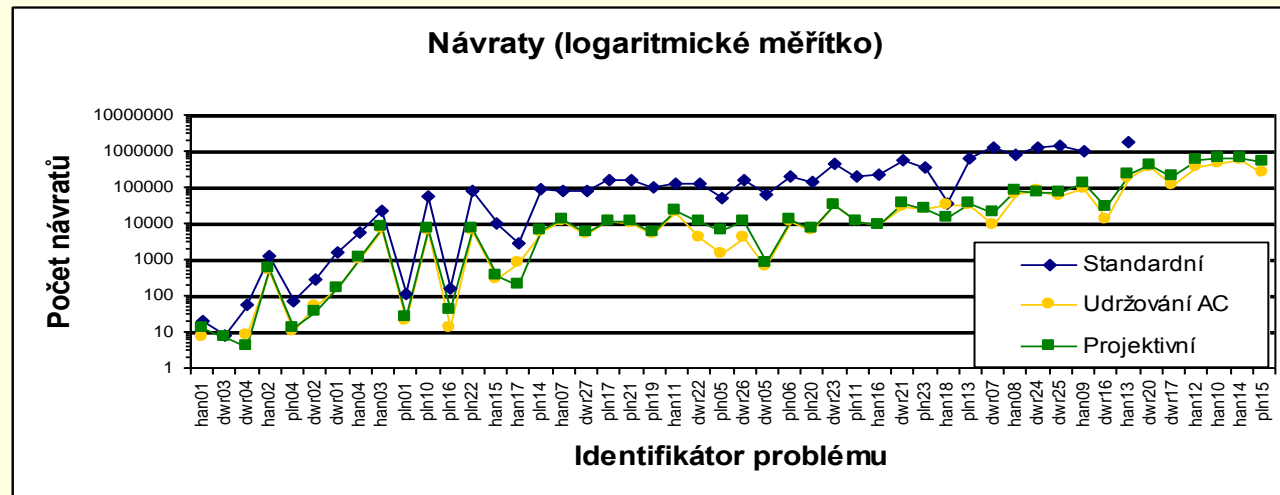
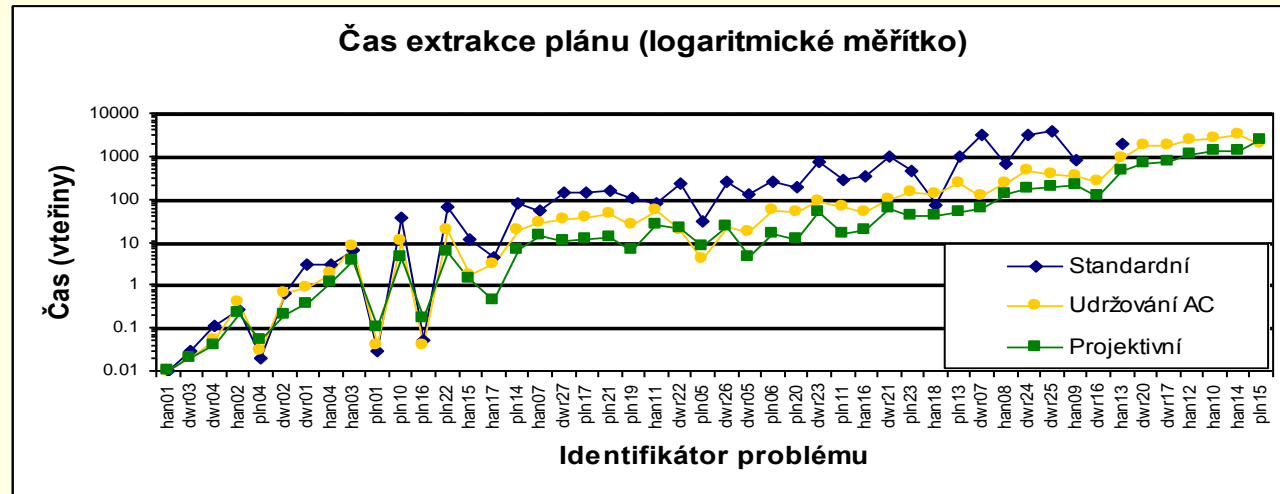


# Projektivní konzistence

- Pomocí **počítání příspěvků** lze některé akce vyloučit z uvažování
  - Zvolená akce  $a \in C_j$  společně s příspěvků ostatních klik nesplňuje dost atomů  $c(a) + \sum_{i=1 \dots k, i \neq j} c(C_i) < \text{velikost cíle}$
- Pro zesílení filtračního efektu používáme vhodnou podmnožinu daného cíle - **projektivní cíl**
  - Počítání příspěvků probíhá vzhledem k projektivnímu cíli
  - **Více projektivních cílů** - zesílení filtračního efektu
- Teoretické vlastnosti:
  - Nižší časová složitost než AC v nejhorším případě  $O(|akce|^2/|cíl|)$  oproti  $O(|akce|^3/|cíl|^2)$  (při použití AC-3)
  - Projektivní konzistence odvodí **více** než AC

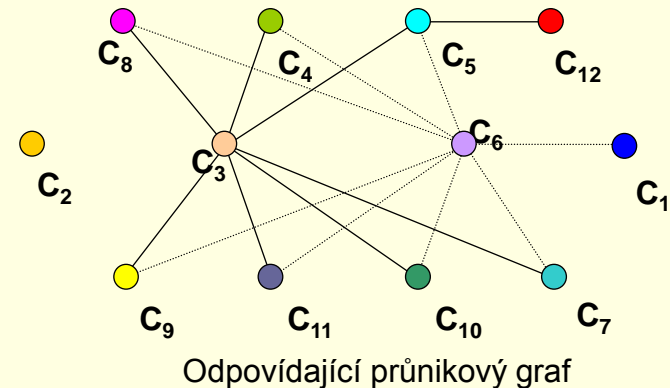
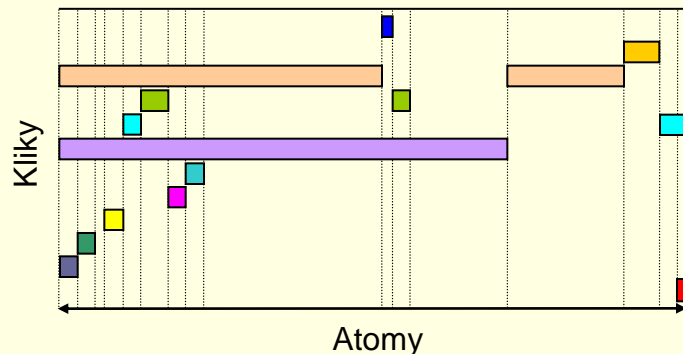
# Experimenty: výsledky pro projektivní konzistenci

- **Projektivní konzistenci** srovnáváme s **udržováním AC**
- Další zlepšení času extrakce plánu oproti **udržování AC**
- Větší zlepšení se projevuje u více **paralelních** problémů
- **Počet návratů** téměř stejný jako pro udržování AC



# Nový přístup (3): polynomiálně řešitelné problémy

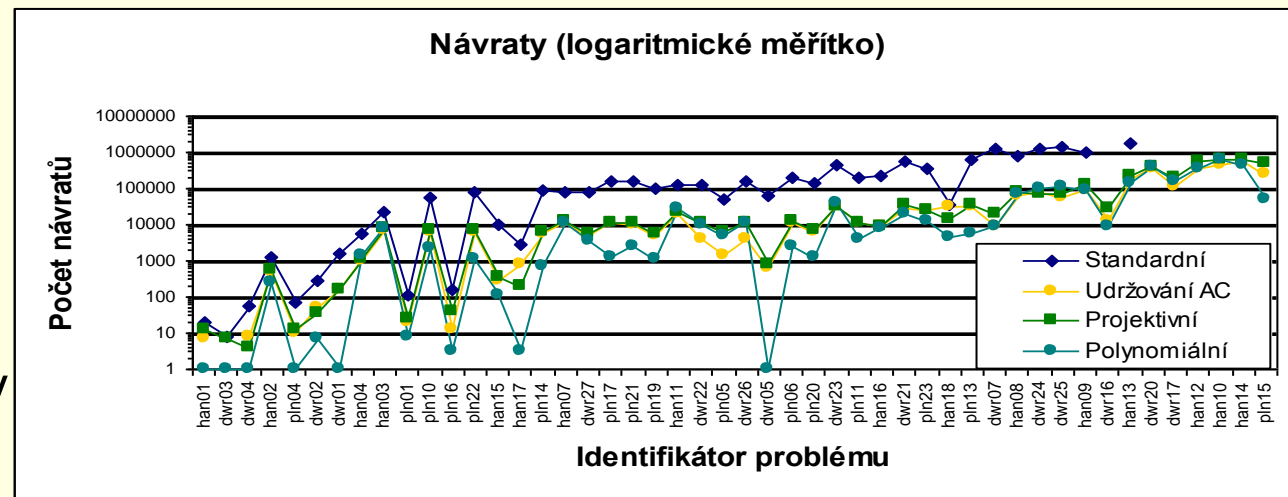
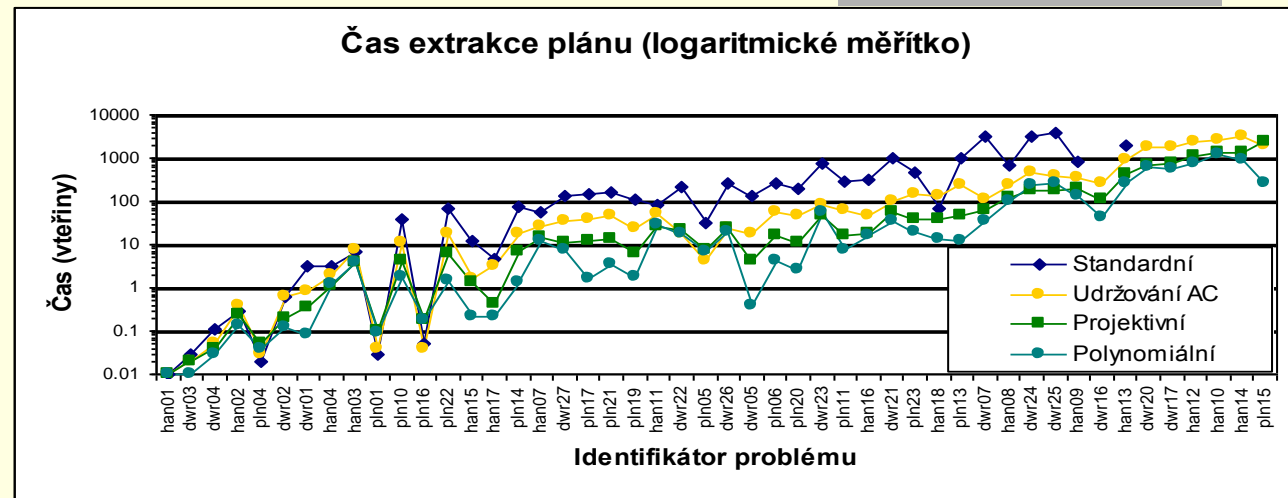
- Další rozšíření projektivní konzistence - řešené problémy se zdají být **strukturálně jednoduché** (vznikají umělým procesem) - využijme této vlastnosti
  - **Sjednocený efekt kliky** = atomy z efektů všech akcí kliky
  - **Průnikový graf** množin sjednocených efektů vypadá **skoro** jako **strom**



- **Když strom**, umíme problém podpor vyřešit v **polynomiálním čase** - stačí **vynutit projektivní konzistenci** vzhledem k vhodným projektivním cílům; **obecně** podporujeme vznik stromu

# Experimenty: výsledky pro polynomiálně řešitelný případ

- Variantu s preferencí polynomiálně řešitelného případu srovnáváme s projektivní konzistencí
- Další zlepšení času extrakce plánu oproti projektivní konzistenci
- Některé problémy vyřešeny bez návratů



# Soutěžní experimenty

- Ačkoli původně nezamýšleno, experimentální plánovač (**SPlan**) založený na projektivní konzistenci se ukázal být konkurenceschopný vzhledem ke špičkovým plánovačům podle výsledků soutěže **IPC** (*International Planning Competition*) na jistých (uměle vytvořených) instancích plánovacích problémů
  - Existující plánovače **selhávají**, pokud řešený problém **nemá řešení** nebo je nutné k nalezení řešení nejprve **zamítnout řešitelnost podproblému**

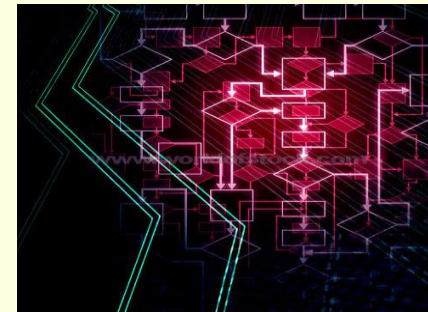
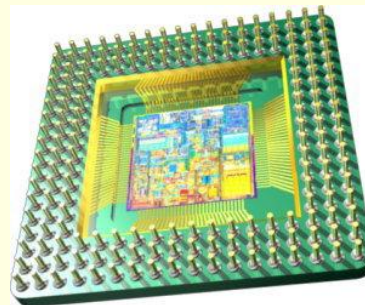
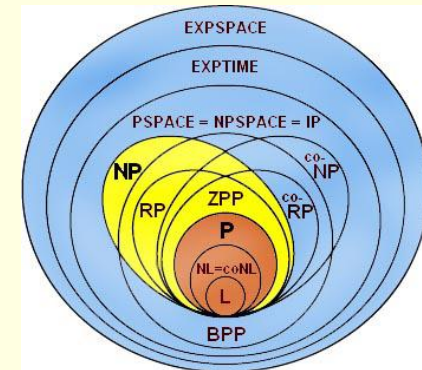
Instance	Řešitelný	SGPlan 5.1 (vteřiny)	IPP 4.1 (vteřiny)	MaxPlan/ miniSat 2 (vteřiny)	SATPlan/ Siege 4 (vteřiny)	CPT 1.0 (vteřiny)	LPG-td 1.0 (vteřiny)	SPlan (vteřiny)
ujam-02_01	ne	N/A	0.00	+ 0.00	∞	0.06	+ 0.00	0.06
ujam-03_02	ne	↓ 0.01	0.01	+ 0.02	∞	∞	+ 5.00	0.54
ujam-04_03	ne	↓ 0.01	0.64	+ 1.03	∞	∞	+ 6.00	3.39
ujam-05_04	ne	↓ 0.02	83.63	+ 8.58	∞	∞	+ 9.00	23.88
ujam-06_05	ne	↓ 0.02	> 600	> 600	∞	∞	> 600	177.9
ujam-07_06	ne	↓ 0.04	> 600	> 600	∞	∞	> 600	> 600.0
ujam-08_07	ne	↓ 0.06	> 600	> 600	∞	∞	> 600	> 600.0
ujam-09_08	ne	↓ 0.10	> 600	> 600	∞	∞	> 600	> 600.0
ujam-10_09	ne	↓ 0.16	> 600	> 600	∞	∞	> 600	> 600.0
jam-02_01	ano	↑ 0.00	0.00	0.26	0.17	0.03	↑ 0.02	0.03
jam-03_02	ano	↑ 0.00	0.00	0.25	0.16	0.04	↑ 0.01	0.09
jam-04_03	ano	↑ 0.00	0.02	0.44	0.17	0.17	↑ 0.02	0.25
jam-05_04	ano	↑ 0.00	0.65	1.10	0.23	5.03	↑ 0.02	0.60
jam-06_05	ano	↑ 0.01	25.8	2.77	0.92	228.75	↑ 0.01	1.29
jam-07_06	ano	↑ 0.01	> 600	30.92	3.01	> 600	↑ 0.02	2.48
jam-08_07	ano	↑ 0.01	> 600	228.01	14.67	> 600	↑ 0.02	4.60
jam-09_08	ano	↑ 0.01	> 600	> 600	152.01	> 600	↑ 0.03	8.77
jam-10_09	ano	↑ 0.01	> 600	> 600	> 600	> 600	↑ 0.02	17.05

# Zhodnocení výsledků v plánování

- Navržená **projektivní konzistence** přináší výrazné **urychlení řešení plánovacích problémů** nad plánovacími grafy při řešení problému hledání podpor
- Projektivní konzistence dosahuje **lepších výsledků než** srovnatelné konzistenční techniky, jako je například **hranová konzistence**
- Pomocí projektivní konzistence se podařilo identifikovat třídu problémů hledání podpor, které jsou **řešitelné v polynomiálním čase**
- Projektivní konzistence s preferencí polynomiálně řešitelného případu přináší další výrazné urychlení
- Na jistých instancích dokonce **lepší výsledky než špičkové plánovací systémy** ze soutěže IPC

# Motivace pro splnitelnost (SAT)

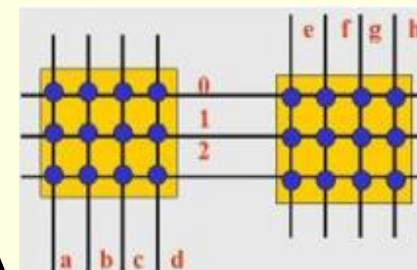
- **Problém SAT:** Dána **booleovská formule**, chceme vědět, **zda je splnitelná**, pokud ano, chceme znát ohodnocení proměnných, které formuli splňuje
- $NP$ -úplný problém, **podobá se problému hledání podpor**
- Teoreticky důležitý problém vzhledem k otázce, zda  **$P=NP$**
- Existují **efektivní řešící systémy pro SAT**, jiné  $NP$ -úplné problémy se na SAT převádějí
- Praktické aplikace při **verifikaci hardwaru a softwaru**





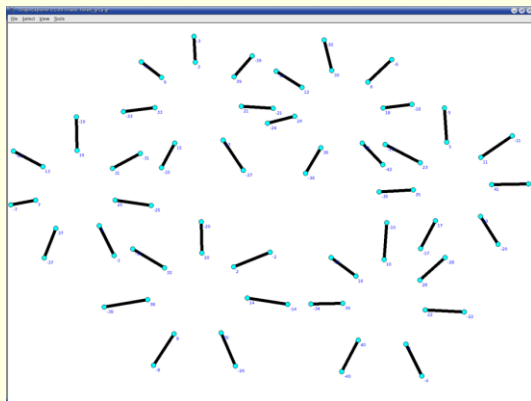
# Existující přístup k řešení a jeho nedostatky

- Používá se **prohledávání do hloubky s jednotkovou propagací** (podobné udržování hranové konzistence) a dalšími vylepšeními (detekce symetrií, učení se klauzulí, restarty a heuristiky) - metoda **DPLL**
- Existují problémy, kde současný přístup **selhává**
  - Nelze (heuristicky) **uhádnout** řešení
  - Heuristiky neuspějí, je nutné **exhaustivní prohledávání**
- Typický příklad: nesplnitelné SAT problémy kódující **Dirichletův princip** (*Pigeon-hole principle*)
  - **Neexistence řešení** - heuristika nemá co uhádnout
  - Další problémy: **směrování v FPGA**

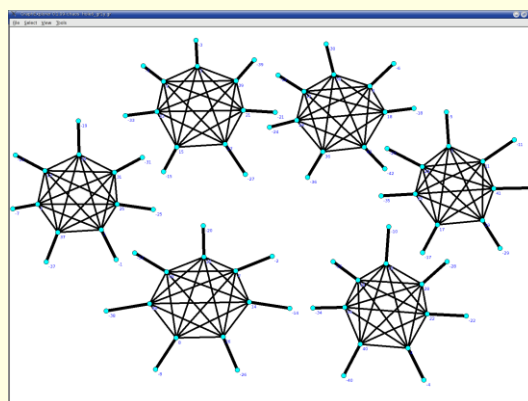


# Nový přístup: aplikace projektivní konzistence

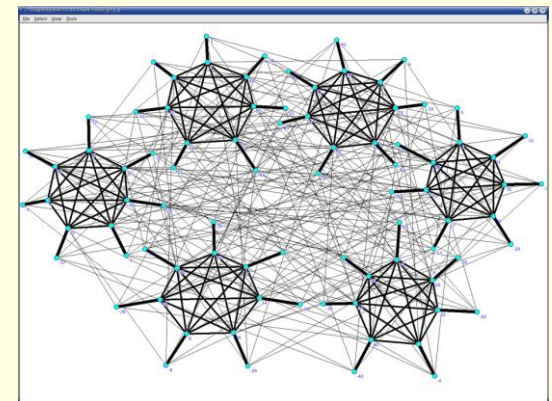
- **Podobnost SATu a problému hledání podpor**  $\Rightarrow$  adaptace projektivní konzistence pro SAT
- **Předzpracování** instance SAT  $\Rightarrow$  zjednodušený problém nebo rozhodnutí
  - Pomocí hranové konzistence **odvodíme konfliktní literály** (konfliktní literály nemohou být splněny současně)
  - Nalezneme **rozklad na kliky**, aplikujeme **adaptaci projektivní konzistence**  $\Rightarrow$  **nový problém** nebo **rozhodnutí**



7 holubů do 6 děr



Po provedení AC



Po aplikaci projektivní konzistence

# Experimenty pro SAT

- Srovnání s několika špičkovými řešícími systémy pro SAT (podle výsledků **SAT Competition 2005** a **SAT Race 2006**)
- Pro testování použity: **MiniSat**, **HaifaSat** a **zChaff**

Instance	Čas na rozhodnutí (vteřiny)	Urychlení vzhledem k MiniSAT	Urychlení vzhledem k zChaff	Urychlení vzhledem k HaifaSAT
chnl10_11	<b>0.43</b>	<b>79.76</b>	<b>17.53</b>	> 1395.34
chnl10_12	<b>0.60</b>	<b>169.68</b>	<b>8.51</b>	> 1000.00
chnl10_13	<b>0.78</b>	<b>256.79</b>	<b>14.70</b>	> 769.23
chnl11_12	<b>0.70</b>	> 857.14	<b>47.84</b>	> 857.14
urq3_5	<b>130.15</b>	<b>0.73</b>	N/A	N/A
urq4_5	> 600.00	N/A	N/A	N/A
urq5_5	> 600.00	N/A	N/A	N/A
urq6_5	> 600.00	N/A	N/A	N/A
hole9	<b>0.08</b>	<b>45.5</b>	<b>18.25</b>	<b>5977.00</b>
hole10	<b>0.13</b>	<b>301.84</b>	<b>57.92</b>	> 4615.38
hole11	<b>0.20</b>	> 3000.00	<b>161.8</b>	> 3000.00
hole12	<b>0.30</b>	> 2000.00	<b>1240.6</b>	> 2000.00
fpga10_11	<b>0.46</b>	<b>97.32</b>	<b>27.34</b>	> 1304.34
fpga10_12	<b>0.64</b>	<b>186.34</b>	<b>52.84</b>	> 937.50
fpga10_13	<b>0.84</b>	<b>431.23</b>	<b>90.65</b>	> 714.28
fpga10_15	<b>1.39</b>	> 431.65	<b>197.72</b>	> 431.65

# Zhodnocení výsledků v booleovské splnitelnosti

---

- **Techniky** původně navržené pro plánování nad plánovacími grafy se **podařilo aplikovat** při řešení problémů booleovské splnitelnosti
- Na jistých třídách obtížných problémů **výrazně vyšší výkon** než špičkové systémy pro řešení SATu (stav k 2005, 2006)
- **Aplikovatelnost** technik do jiné oblasti než je plánování ukazuje na **obecnost návrhu**

# Všeobecný závěr

- Předložená práce se zabývá **plánováním nad plánovacími grafy a booleovskou splnitelností (SAT)**
- Pro **urychlení extrakce plánu** z plánovacího grafu bylo navrženo několik technik
  - Použití **hranové konzistence (AC)**
  - Specializovaná **projektivní konzistence**
  - **Případ řešitelný v polynomiálním čase** pomocí projektivní konzistence
- Konzistenční **techniky** navržené pro plánování byly **úspěšně aplikovány** při řešení problému **SAT**
- Prezentované výsledky byly získány na základě systematické teoretické a experimentální práce (v souvislosti s prací bylo napsáno asi **5MB** experimentálních programů v C/C++)