PLÁNOVÁNÍ CEST PRO MNOHO ROBOTŮ

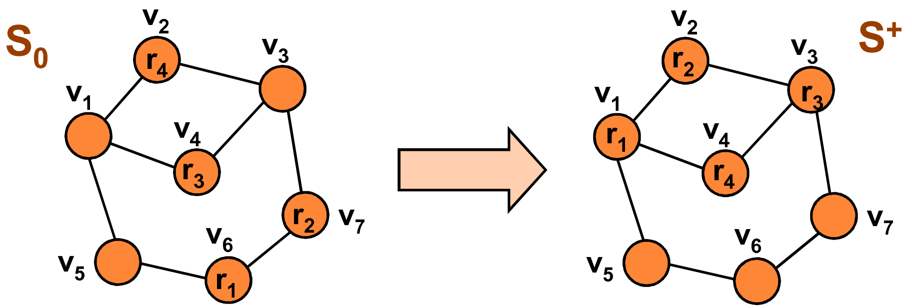
Pavel Surynek
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta
Česká republika

Plánování cest pro mnoho robotů

- **Vstup:** Graf **G=(V,E)** a množina robotů **$R=\{r_1,r_2,\ldots,r_\mu\}$**, kde **$\mu<|V|$**
  - **Každý robot** je umístěn ve **vrcholu** (nejvýše jeden robot ve vrcholu)
  - **Robot se může přesunout do volného vrcholu** skrz hranu (žádný jiný robot nesmí vstupovat do téhož vrcholu)
  - **Počáteční pozice** robotů … prostá funkce **S_0: R→V**
  - **Cílové pozice** robotů … prostá funkce **S^+: R→V**
- **Úloha: Nalézt posloupnost dovolených pohybů** pro každého robota takovou, že každý robot dosáhne své cílové pozice

PŘÍKLAD PLÁNOVÁNÍ CEST PRO ROBOTY

- **Počáteční pozice** robotů dána funkcí S_0
- **Cílové pozice** robotů dány funkcí S^+



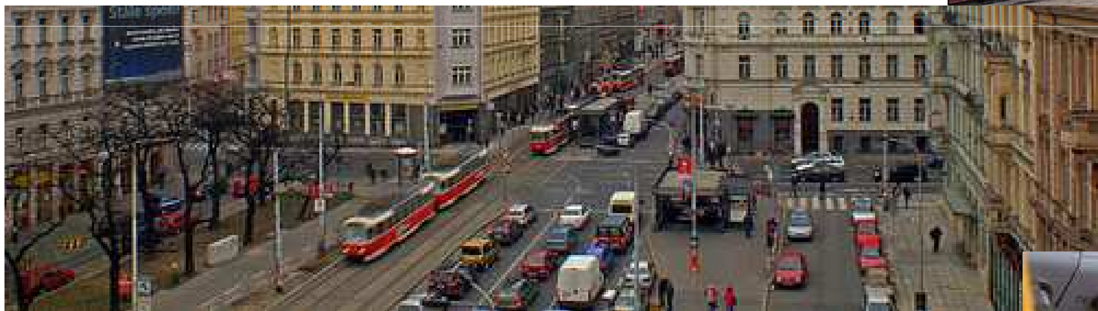
délka=12

$P_{r1}=[\mathbf{v_6},v_6,v_6,v_6,v_6,v_7,v_7,v_3,v_4,v_1,v_1,\mathbf{v_1}]$
$P_{r2}=[\mathbf{v_7},v_7,v_7,v_7,v_3,v_3,v_2,v_2,v_2,v_2,v_2,\mathbf{v_2}]$
$P_{r3}=[\mathbf{v_4},v_4,v_4,v_1,v_1,v_1,v_1,v_5,v_6,v_6,v_7,\mathbf{v_3}]$
$P_{r4}=[\mathbf{v_2},v_1,v_5,v_5,v_5,v_5,v_6,v_7,v_7,v_3,v_4,\mathbf{v_4}]$

- Zobrazeno je řešení délky **12**
  - $\mathbf{P_r}$ je posloupnost pozic robota **r** postupně ve všech diskrétních časových okamžicích
  - **Paralelismus** v rámci řešení
  - **Krátká řešení** jsou preferována (nejkratší NP-těžké)

MOTIVACE K PROBLÉMU

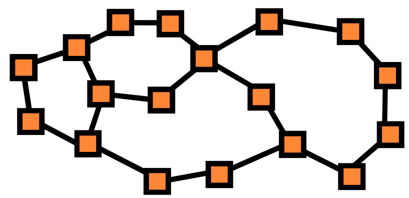
- Přerovnávání jistých agentů v těsném prostoru
- Automatické řízení velmi husté dopravy



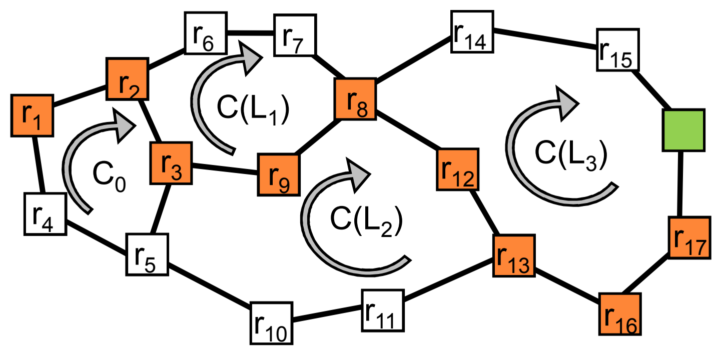
- Přenos dat v rámci komunikační sítě
- Vertikální/horizontální výtah v moderní budově

BIBOX: ALGORITMUS PRO 2-SOUVISLÉ GRAFY

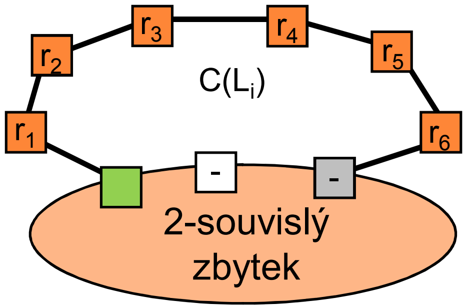
- Neorientovaný graf G=(V,E) je **2-souvislý** jestliže $|V| \geq 3$ a $\forall v \in V$ je graf $G=(V-\{v\},E')$ kde $E'=\{\{x,y\} \in E \mid x \neq v \wedge y \neq v\}$ souvislý
- **Vlastnost:** Každý 2-souvislý graf lze zkonstruovat z kružnice postupným přidáváním uch

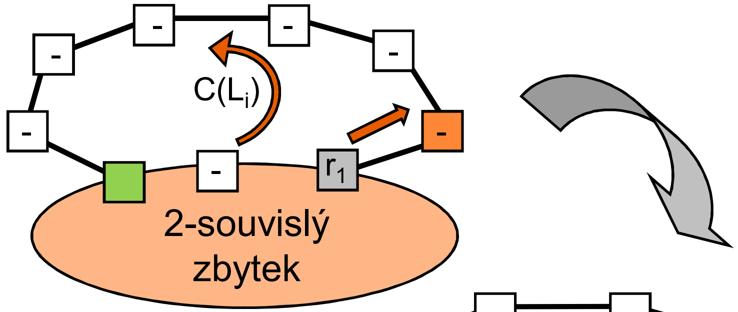


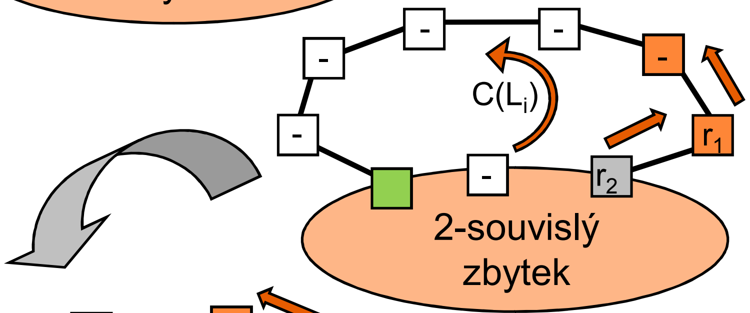
- **Rozklad** grafu **na ucha** lze rychle určit (v čase **O(|V|+|E|)**)
- Znalost rozkladu na ucha umožňuje:
  - Přesunout **volnou pozici** do libovolného vrcholu grafu
  - Přesunout **robota** do libovolného vrcholu

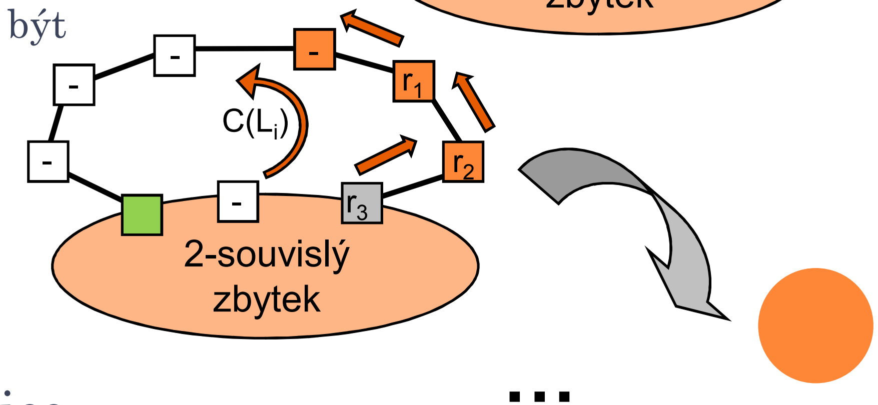


BIBOX: UMISŤOVÁNÍ ROBOTŮ V RÁMCI UCHA (≠ POČÁTEČNÍ KRUŽNICE)





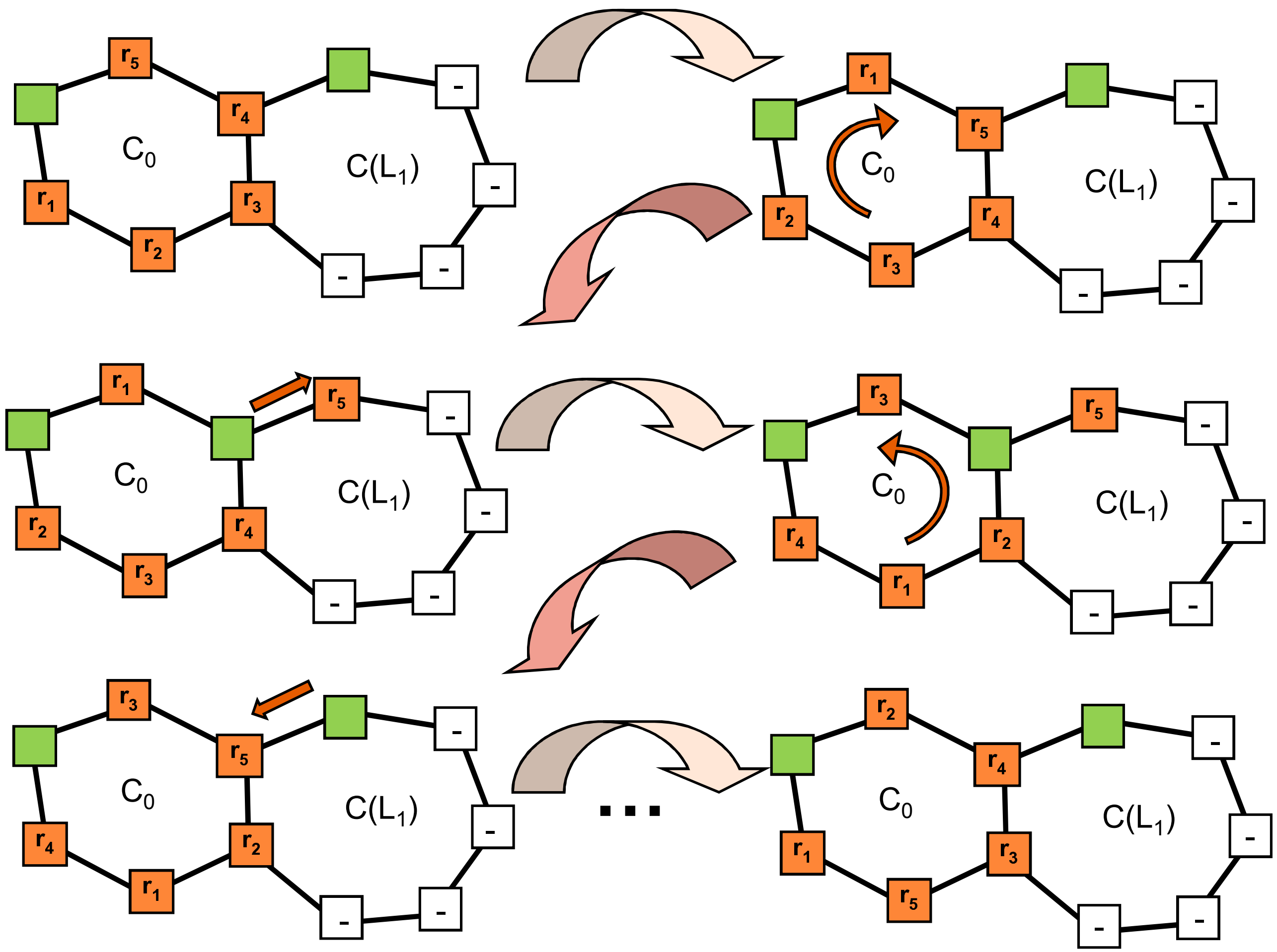




- Roboty jsou do ucha umisťovány jako na zásobník
  - Další robot je přesunut do **šedého** koncového vrcholu ucha
  - Dva případy
    - Robot je uvnitř ucha → nejprve musí být rotován mimo ucho
    - Robot již je mimo ucho
  - Robot je rotován o jeden krok dovnitř ucha (pomocí **zeleného** volného vrcholu)
- Poslední rotace umístí všechny roboty v rámci ucha na cílové pozice

BIBOX: UMISŤOVÁNÍ ROBOTŮ V RÁMCI POČÁTEČNÍ KRUŽNICE

- Výměna robotů $\mathbf{r}_5$ a $\mathbf{r}_2$ – pomocí výměn robotů lze obdržet libovolnou permutaci robotů v rámci počáteční kružnice $\mathbf{C}_0$



BIBOX: SLOŽITOST A POZNÁMKY

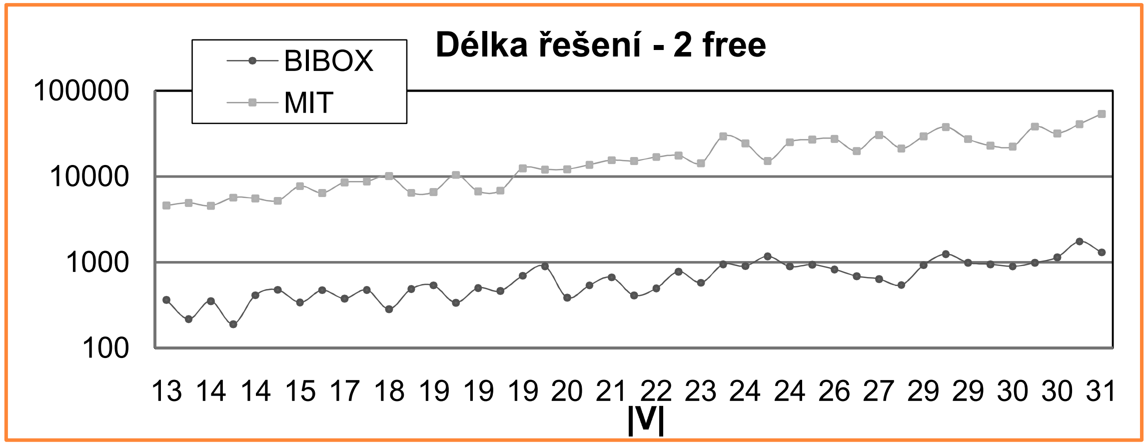
- **Časová složitost** v nejhorším případě algoritmu BIBOX je **$O(|V|^3)$**
- **Délka** generovaného **řešení** je **$O(|V|^3)$**
- **Dva volné** vrcholy jsou požadovány pouze v poslední fázi algoritmu – při umisťování robotů v počáteční kružnici
- Speciální požadavek, kdy jsou **cílové pozice volných vrcholů v rámci počáteční kružnice**, lze obejít
  - Modifikovat cílové pozice robotů dané funkcí $\mathbf{S}^+$ tak, aby byly v počáteční kružnici
  - Přesunutím volných vrcholů podél dvou disjunktních cest do počáteční kružnice
- **Více volných vrcholů** umožňuje provádět přesuny robotů **paralelně**
  - Po sobě jdoucí přesuny generovány algoritmem BIBOX jsou vyšetřovány na **nezávislost**
  - Když **nezávislé** → vykonány **paralelně**

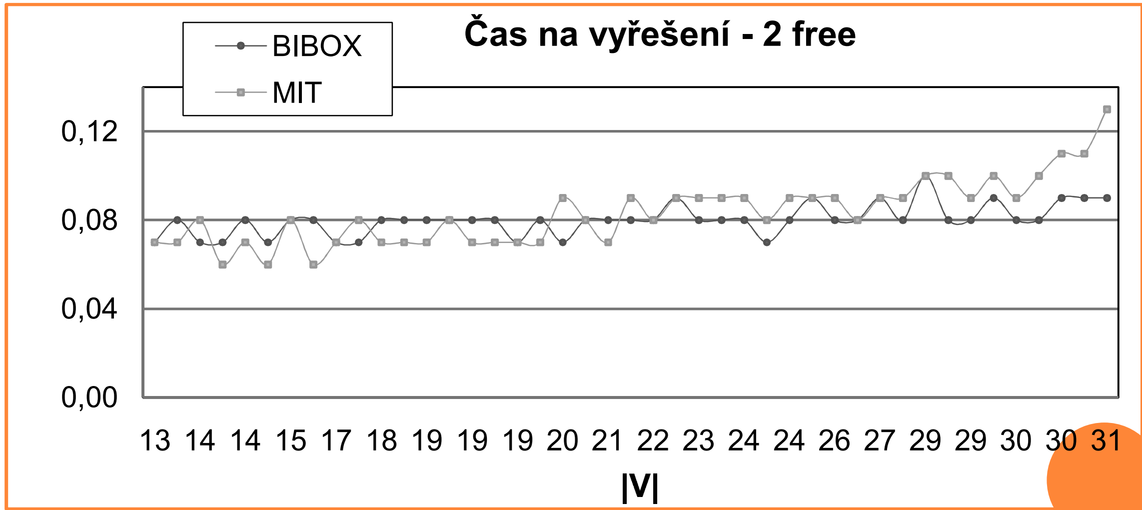
ALGORITMUS **BIBOX** A KONKURENČNÍ ALGORITMY

- Algoritmus od autorů Kornhauser, Miller, Spirakis (FOCS 1984) – zkráceně nazýváme algoritmus **MIT**
  - Pracuje s 2-souvislými grafy a alespoň **jedním volným vrcholem**
  - Založen na **3-tranzitivitě** 2-souvislých grafů (každé tři roboty lze přesunout do libovolných třech vrcholů)
  - **Časová složitost** v nejhorším případě je **$O(|V|^3)$** – stejná jako pro BIBOX, ale konstanty v odhadu jsou větší
  - **Délka** generovaného **řešení** je také **$O(|V|^3)$**
- **Doménově nezávislé plánovače** účastnící se soutěže IPC (International Planning Competition)
  - **SGPlan 5** a **LPG-td** se ukázaly být nejlepší z vítězů soutěže IPC na problému plánování cest pro mnoho robotů
- Testovací problémy
  - Náhodně generované problémy plánování cest pro mnoho robotů
  - Náhodné 2-souvislé grafy / náhodná permutace robotů
  - Ucha náhodné délky 1…8 (rovnoměrné rozdělení)

EXPERIMENTÁLNÍ SROVNÁNÍ ALGORITMŮ BIBOX A MIT

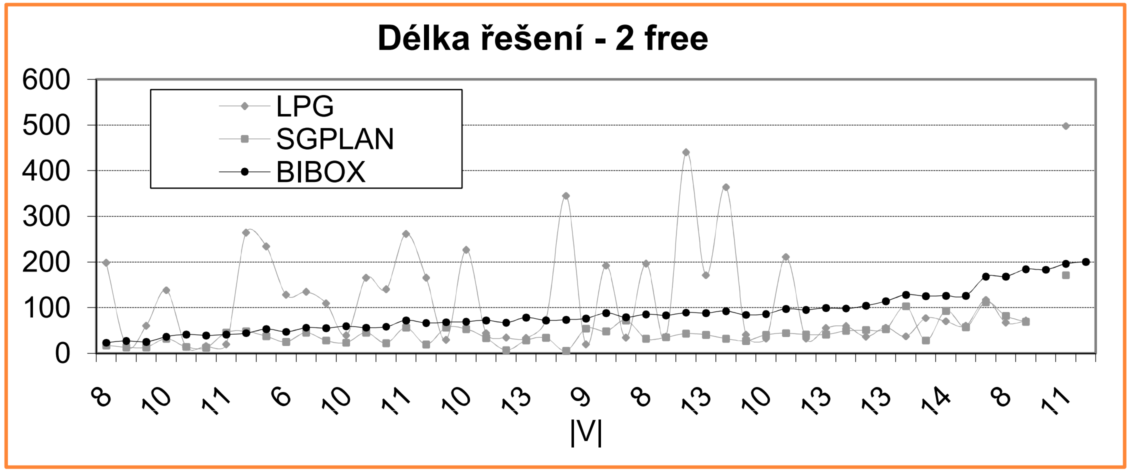
- Oba algoritmy implementovány v C++
- Srovnávány jsou **délky řešení** a **čas na vyřešení** problémů
- Náhodné grafy obsahující až **30 vrcholů**
- Algoritmus **BIBOX** produkuje až zhruba o řád **kratší řešení** než algoritmus MIT
- Algoritmus **BIBOX** je **rychlejší na větších problémech** (tento trend zřejmě pokračuje)

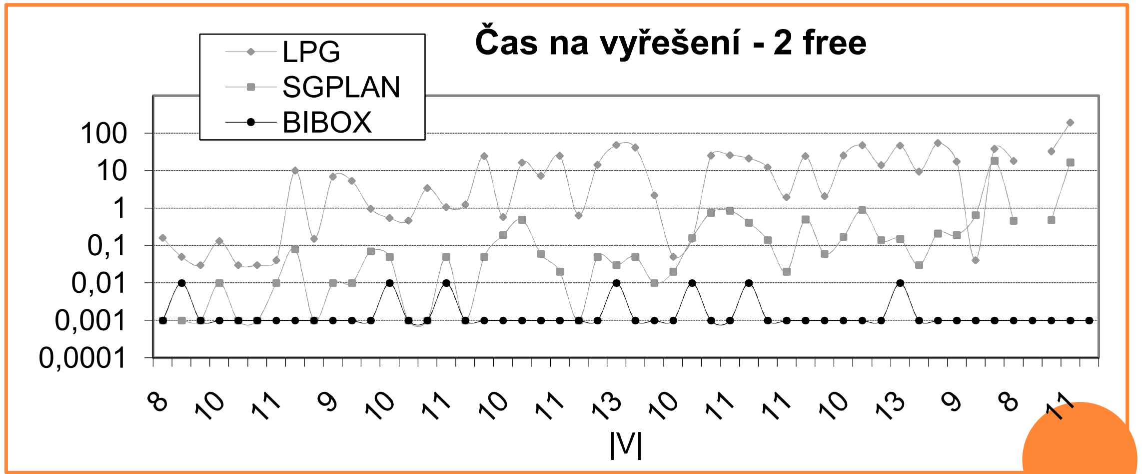




EXPERIMENTÁLNÍ SROVNÁNÍ ALGORITMU **BIBOX** A PLÁNOVAČŮ **SGPLAN** A **LPG**

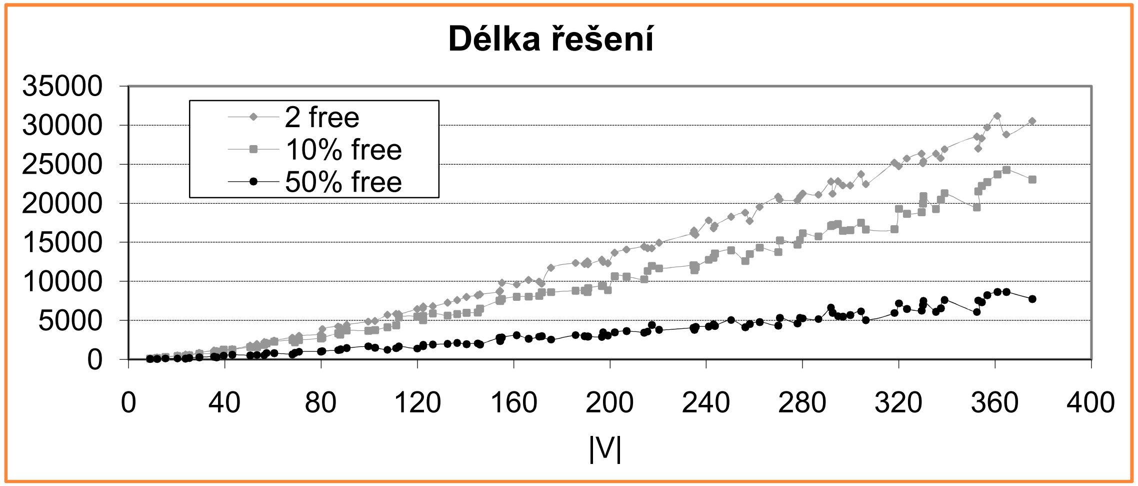
- Byl použit kód poskytnutý autory SGPlan a LPG
- SGPlan hledá nejkratší možná řešení
- Byly použity pouze malé náhodné grafy s nejvýše **15 vrcholy**
- Plánovače produkují **kratší řešení** – zvláště SGPlan
- Avšak **čas na vyřešení** je u plánovačů **velmi vysoký**

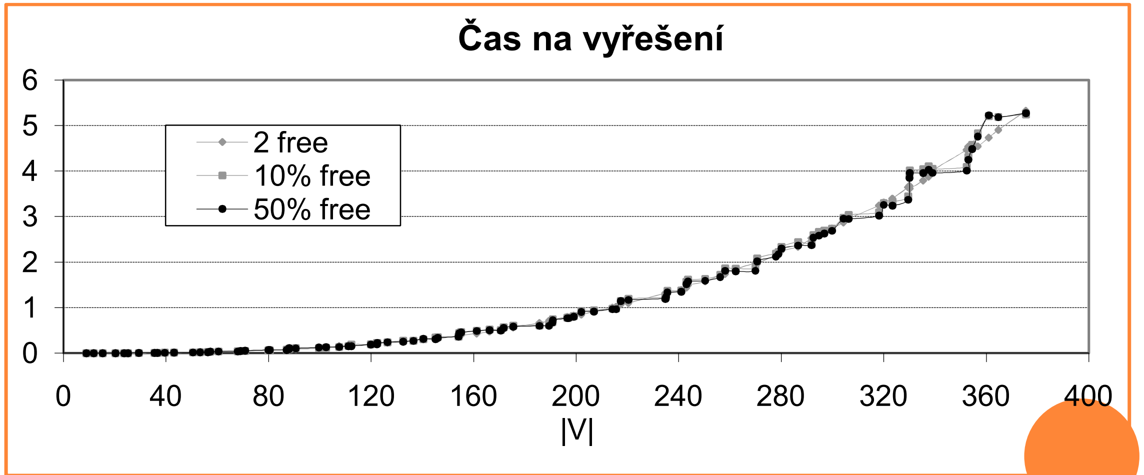




Experimenty s algoritmem BIBOX na velkých problémech

- Použity grafy s až **400 vrcholy**
- Tři sady problémů
  - 2 volné vrcholy
  - 10% volných vrcholů
  - 50% volných vrcholů
- Testován **paralelismus**
- Volné vrcholy kromě dvou zaplněny falešnými roboty – z konečného řešení odstraněny
- Všechny problémy vyřešeny za **6 vteřin**





SHRNUTÍ A ZÁVĚR

- Navržen nový algoritmu nazvaný **BIBOX** na řešení problému plánování cest pro více robotů v **2-souvislých grafech** s alespoň **dvěma volnými** vrcholy
- Experimenty ukázaly,že algoritmus **BIBOX** je lepší než konkurenční přístupy
  - **BIBOX** výrazně **překonává** dva vybrané špičkové **plánovače** (podle výsledků soutěže IPC) v čase potřebném na vyřešení
  - **BIBOX** produkuje až o řád kratší řešení něž další existující algoritmus (**MIT**), a to v o něco kratším čase
- **Budoucí práce:**
  - Upravit závěrečnou fázi algoritmu tak, aby stačil jeden volný vrchol – použít databázi vzorů
  - Zvýšit paralelismus – použít metodu kritické cesty