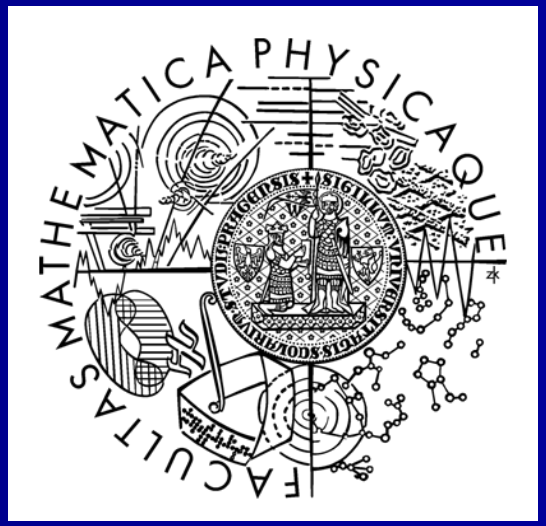
Plánování cest pro mnoho robotů
(Multi-robot path planning)

Pavel Surynek
KTIML
Matematicko-fyzikální fakulta
Univerzita Karlova v Praze



Přehled přednášky

- Motivace hravá a vážná
- Formální definice problému
- Aplikace doménově nezávislého plánování
  - selhání
- Algoritmy založené na strukturálních vlastnostech
  - Existující algoritmus - MIT
  - Nový algoritmus - BIBOX
- Využití optimálních maker
- Paralelismus
- Otevřené otázky a závěr

Formální definice problému

(Kornhauser et al.1984; Ryan, 2007)

- **Vstup:** Graf G=(V,E), V={v_1,v_2,...,v_n} a množina robotů R={r_1,r_2,...,r_μ}, kde μ<n
  - každý **robot** je umístěn **ve vrcholu** (v jednom vrcholu je nejvýše jeden robot)
  - **robot se může přesunout do sousedního volného vrcholu** skrz hranu, která vrcholy spojuje (přitom žádné dva roboty nesmí do stejného vrcholu zároveň)
  - **Počáteční pozice** robotů ... prostá funkce $\mathbf{S_0: R \rightarrow V}$
  - **Cílové pozice** robotů ... prostá funkce $\mathbf{S^+: R \rightarrow V}$
- **Úloha:** **Nalezněte posloupnost dovolených pohybů pro každého robota** tak, že všechny roboty dosáhnou cílových pozic za předpokladu, že začínaly v daných počátečních pozicích
- **Pohyb robota:** Posloupnost vrcholů, kde se postupně robot nachází

Hravá motivace

(Lucasfilm Ltd., 1999-2009)

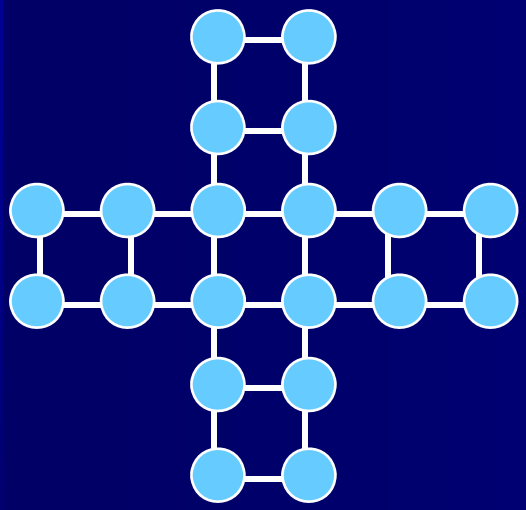
Vážná motivace

- Přemisťování předmětů v omezeném prostoru-skladiště
- Klasický příklad: pohyb skupiny robotů
- Automatické řízení husté dopravy, „zobecněné výtahy“
- Virtuální světy: přemisťování datových paketů

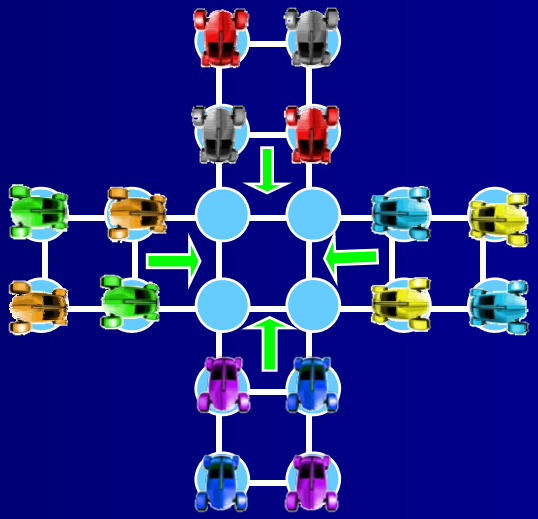
Počáteční experiment (1)

- Instance problému plánování cest pro roboty motivovaná **hustou automobilovou dopravou**
  - budeme ignorovat dopravní pravidla

Graf modelující křižovatku



Počáteční pozice robotů



Cílové pozice robotů

- Zkusíme **doménově nezávislé** plánovací systémy (SGPlan, SATPlan, LPG)

Ukázka specifikace pro plánovací systém (jazyk PDDL) (1)

Pohyb automobilů ve čtvercové síti

```
(define (domain crossing)
  (:requirements :strips)
  (:predicates
    (free ?x ?y) (at ?car ?x ?y) (adjacent ?x ?y ?p ?q))
    (:action move1
      :parameters (?x ?y ?p ?q ?car)
      :precondition (and (adjacent ?x ?y ?p ?q)
                         (at ?car ?x ?y)
                         (free ?p ?q)
                    )
      :effect (and (at ?car ?p ?q)
                   (free ?x ?y)
                   (not (free ?p ?q))
                   (not (at ?car ?x ?y))
              ) )
  ...
```

Automobil *car* se pohybuje z místa *[x,y]* na místo *[p,q]:* Místa *[x,y]* a *[p,q]* musí být spojena hranou. Automobil *car* se musí nacházet na místě *[x,y]* a místo *[p,q]* musí být volné.

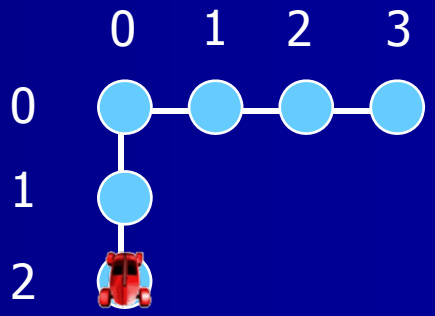
Výsledkem akce je, že automobil *car* se nachází na místě *[p,q]*, ale už se nenachází na místě *[x,y]* . Místo *[x,y]* je nyní volné, ale místo *[x,y]* již volné není.

Ukázka specifikace pro plánovací systém (jazyk PDDL) (2)

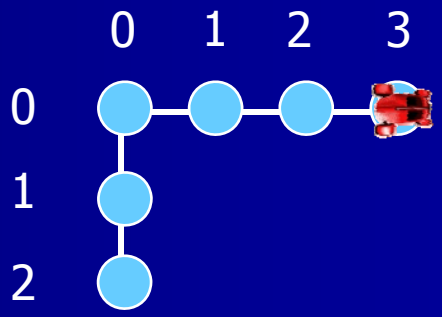
Pohyb automobilů ve čtvercové síti

```
(define (problem crossing_02)
  (:domain crossing)
  (:objects b0 b1 b2 b3 red)
  (:init
   (free b0 b0)
   (free b0 b1)
   (free b0 b2)
   (free b0 b3)
   (free b1 b0)
   (free b1 b0)
   (adjacent b0 b0 b0 b1)
   (adjacent b0 b1 b0 b2)
   (adjacent b0 b2 b0 b3)
   (adjacent b0 b0 b1 b0)
   (adjacent b1 b0 b2 b0)
   (at red b2 b0))
  (:goal (and (at red b0 b3))
))
```

Počáteční pozice robota



Cílová pozice robota



Počáteční experiment (2)

Problém	Počet vrcholů /volné	Řešitelný	Délka plánu/ optimální	SGPlan (vteřiny)	SATPlan (vteřiny)	LPG (vteřiny)
01	7/6	Ano	1/1	0.00	0.01	0.01
02	8/7	Ano	6/6	0.00	0.04	0.02
04	8/6	Ano	25/16	0.00	0.09	0.01
05	8/5	Ano	216/32	0.00	0.87	0.98
06	12/10	Ano	12/12	0.00	0.04	0.01
07	12/8	Ano	178/26	0.00	0.14	0.08
08	12/7	Ano	176/36	0.03	0.50	0.37
09	12/6	Ano	64/46	0.04	0.52	1.54
10	12/5	Ano	72/56	0.04	**>120.0**	3.46
11	12/4	Ano	112/60	0.05	**>120.0**	4.36
12	12/3	Ano	170/98	0.17	**>120.0**	5.25
13	13/4	Ano	154/112	0.61	**>120.0**	5.94
15	13/3	Ano	N/A	**>120.0**	**>120.0**	**>120.0**
16	12/10	Ano	10/10	0.00	0.04	0.02
17	12/8	Ano	30/24	0.02	0.16	0.03
18	12/4	Ano	124/N/A	**>120.0**	**>120.0**	1.45
19	12/3	Ano	114/78	0.76	**>120.0**	6.23
20	12/2	Ano	208/120	0.33	**>120.0**	7.58
22	16/2	Ne	N/A	**>120.0**	**>120.0**	**>120.0**
23	14/2	Ne	N/A	**>120.0**	**>120.0**	**>120.0**
24	28/20	Ano	72/64	0.08	**>120.0**	0.11

Analýza počátečního experimentu

- Všechny testované instance problémů byly velmi malé – grafy obsahovaly kolem 15 vrcholů
- **SGPlan** a **SATPlan** se snaží nalézt **nejkratší možné řešení** – zdroj komplikací
- **SGPlan** a **SATPlan** často problém nevyřeší **v časovém limitu 2 minut** (zvláště se to stává, pokud problém nemá řešení)
- **LPG** hledá **neoptimální řešení** (tedy ne nutně nejkratší)
- **LPG** se ukázal jako nejlepší, přesto nevyřešil všechny problémy v daném časovém **limitu 2 minut**
- **Závěr: Doménově-nezávislé plánovací systémy se zdají být neefektivním nástrojem pro plánování cest pro mnoho robotů**
- **Důvody:** Plánovací systém nevidí strukturu problému, pracuje s úlohou symbolicky pomocí odvozovacích pravidel (akcí)

Teoretičtější přístup (zkoumejme strukturu problému)

- Plánování cest pro roboty představuje obecnou variantu hry **Lloydova 15-ka**
  - dovolujeme **obecný graf**, ne jen graf izomorfní čtvercové síti
  - dovolujeme **více než jeden vrchol** grafu **volný**

- Mnoho výsledků je již známo
  - Zajímavý je především případ s **2-souvislým grafem** – téměř vždy řešitelný – na ten se dále omezíme
  - Přeuspořádání robotů ve vrcholech můžeme chápat jako **permutaci** robotů (volný vrchol v $\mathbf{S_0}$ a v $\mathbf{S^+}$ fixujeme)
  - Permutace může být buď **sudá** nebo **lichá** (vyjádřitelná jako složení sudého respektive lichého počtu transpozic prvků)

Známé výsledky

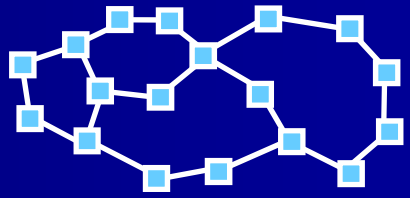
- Wilson, 1974:
  - Pokud 2-souvislý graf (neizomorfní s kružnicí) s jedním volným vrcholem **obsahuje kružnici liché délky**, potom je každý problém plánování cest pro roboty nad tímto grafem řešitelný.
  - Pokud 2-souvislý graf (neizomorfní s kružnicí) s jedním volným vrcholem **neobsahuje** lichou kružnici, potom je problém plánování cest pro roboty řešitelný, právě když $\mathbf{S}^+$ představuje **sudou permutaci** vzhledem k $\mathbf{S}_0$.
  - Řešení délky $O(|V|^5)$ je generováno v čase $O(|V|^5)$.
- Kornhauser, Miller, Spirakis, 1984 (algoritmus MIT):
  - Opět pro 2-souvislé grafy s jedním volným vrcholem.
  - Řešení délky $O(|V|^3)$ je generováno v čase $O(|V|^3)$.
  - Existují instance, kde délka nejkratšího řešení je $\Omega(|V|^3)$.
- Ratner, Warmuth, 1986:
  - Rozhodovací varianta problému, kdy hledáme nejkratší možné řešení je **NP-úplná**
  - Ukázáno pro zobecnění Lloydovy 15-ky na pole velikosti N x N

Algoritmus MIT (1)

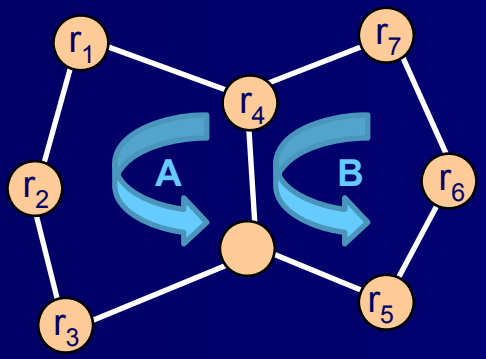
- Možná přeuspořádání robotů ve vrcholech formují permutační grupu G pro prvky $R=\{r_1,r_2,\ldots,r_\mu\}$
- **Definice:** Permutační G grupa pro prvky $R=\{r_1,r_2,\ldots,r_\mu\}$ je **k-tranzitivní** pro $k\leq\mu$, jestliže pro každé dvě k-tice prvků $a_1,a_2,\ldots,a_k$ a $b_1,b_2,\ldots,b_k$, kde $a_i\in R$, $b_i\in R$ pro i=1,2,…,k existuje permutace $\pi\in G$, že $\pi(a_i)=b_i$ pro i=1,2,…,k.
- **Tvrzení 1:** Jestliže permutační grupa G obsahuje **cyklus délky k** a je k-tranzitivní, potom obsahuje **všechny cykly délky k**.
- **Tvrzení 2:** Každou sudou permutaci pro prvky $R=\{r_1,r_2,\ldots,r_\mu\}$ lze získat jako produkt nejvýše μ-2 **cyklů délky 3** (3-cyklus)
- **Naznačíme**, že 2-souvislý graf obsahuje 3-cyklus a že je 3-tranzitivní
  - Stačí k vytvoření libovolné sudé permutace.

Algoritmus MIT (2)

- **Definice:** Graf G=(V,E) je 2-souvislý, jestliže |V|≥3 a pro ∀v∈V je G=(V-{v},E'), kde E'={{x,y}∈E | x,y ≠ v}, je **souvislý**.
- **Tvrzení 3:** Každý 2-souvislý graf lze zkonstruovat z kružnice postupným přidáváním **uch**.



Cyklus + 1 ucho



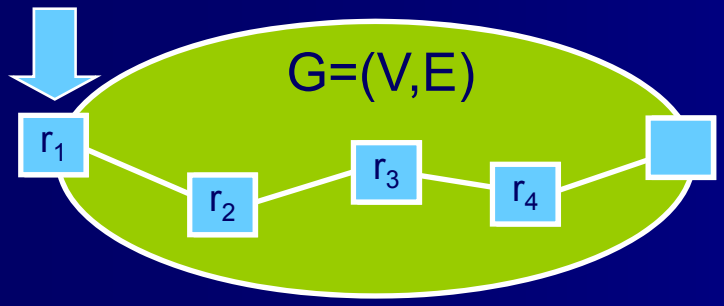
- Ilustrace **3-cyklu:** $ABA^{-1}B^{-1}=(r_4,r_7,r_3)$
- Ilustrace **3-tranzitivity:**
  - Libovolné 3 roboty (x,y,z) umíme narovnat do ucha délky aspoň 3 … permutace P
  - Totéž pro roboty (u,v,w)…permutace Q
  - PQ^{-1} dává 3-tranzitivitu (x,y,z) na (u,v,w)

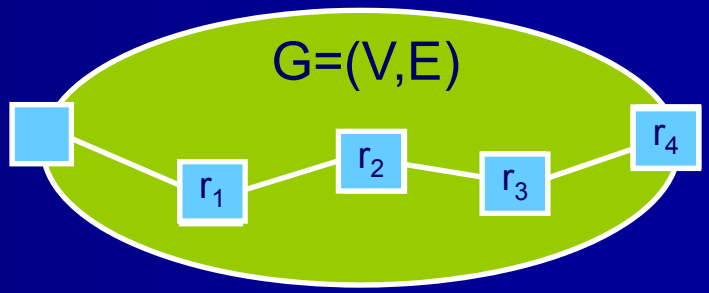
Algoritmus MIT (3)

- Ilustrován byl **nejjednodušší případ**
  - Rozbor případů pro různé situace (délky uch, speciální podgrafy), podstata stejná jako na ilustrovaném příkladu
- Časová složitost:
  - Generování 3-cyklu
    - konstantní počet rotací uch
    - rotace ucha spotřebuje O(|V|) pohybů, lze určit v čase O(|V|)
    - celkem tedy O(|V|) pohybů i času
  - Generování 3-tranzitivity
    - 3 přesuny robota k uchu, 3 rotace ucha
    - přesun robota podél hrany spotřebuje O(|V|) pohybů
    - robot se přesouvá až přes |V| hran
    - celkem tedy $O(|V|^2)$ pohybů i času
  - Celkem potřebujeme složit μ-2 3-cyklů, kde na každý potřebujeme $O(|V|^2)$ pohybů i času, celkem je potřeba $O(|V|^3)$ pohybů i času
- Nevýhoda: relativně velké konstanty v odhadech

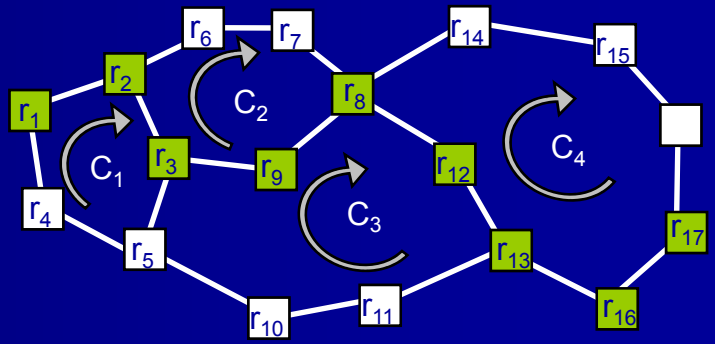
Modernější přístup: algoritmus BIBOX (1)

- Předpokládejme, že známe rozklad 2-souvislého grafu G=(V,E) na počáteční kružnice C_0 a posloupnost uch $L_1,L_2,\ldots,L_l$ které se mají postupně přidávat abychom dostali G.



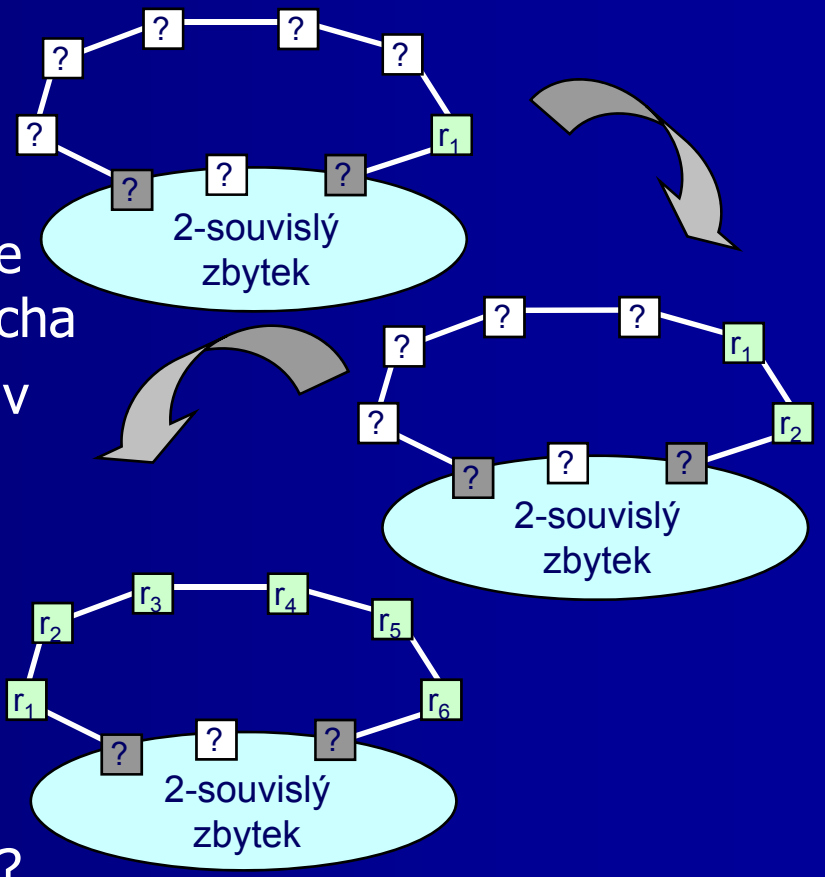


- **Umíme:** „přesouvat" volný vrchol na libovolné místo v grafu.
- **Umíme:** libovolného robota přemístit do libovolného vrcholu (využijeme 2-souvislosti)



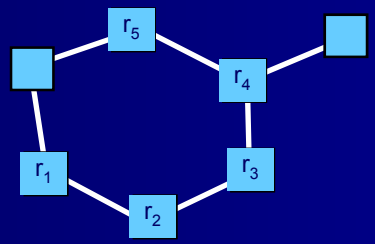
Algoritmus BIBOX (2)

- Postupujeme indukcí podle počtu uch
  - Postupně umisťujeme roboty jejichž cílové pozice jsou v rámci posledního ucha
  - Po umístění všech robotů v uchu dostáváme menší problém stejného typu (na 2-souvislém zbytku)
  - Technický detail: co když volný vrchol má být v uchu?
  - Co s počáteční kružnicí C_0?



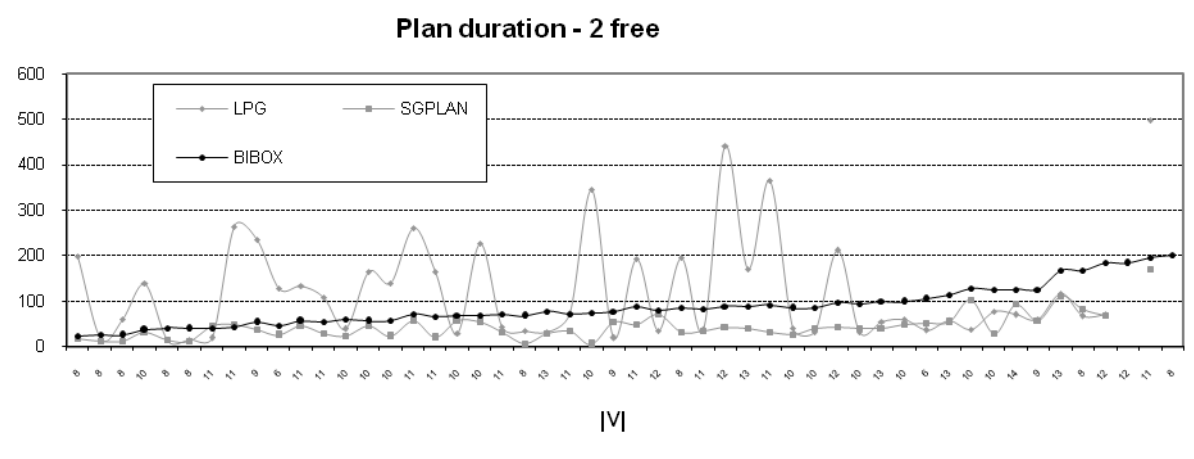
Algoritmus BIBOX (3)

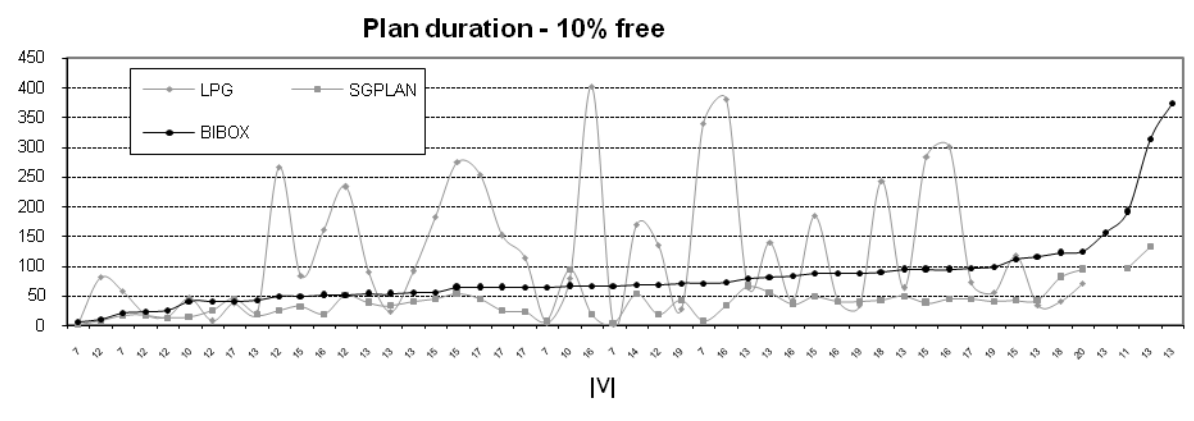
- Pro počáteční kružnice C_0 potřebujeme dva volné vrcholy (chceme-li zachovat jednoduchost postupu).



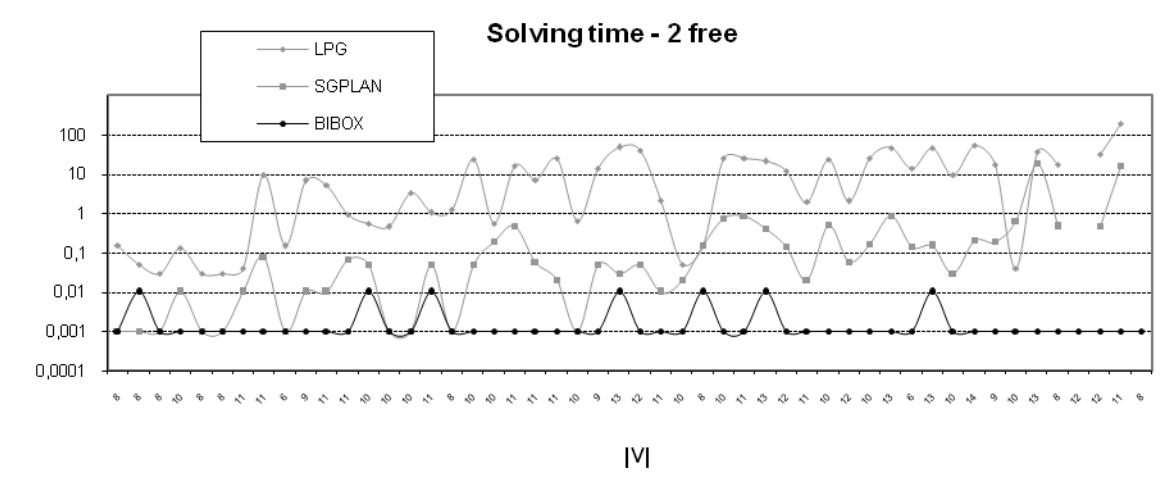
- Časová složitost:
  - umístění robota do ucha spotřebuje $O(|V|^2)$ pohybů
  - pro umístění všech robotů v uchách $O(|V|^3)$ pohybů
  - počáteční kružnice spotřebuje také $O(|V|^3)$ pohybů
  - celkem tedy $O(|V|^3)$ pohybů, čas je také $O(|V|^3)$
- V odhadech menší konstanty než u algoritmu MIT

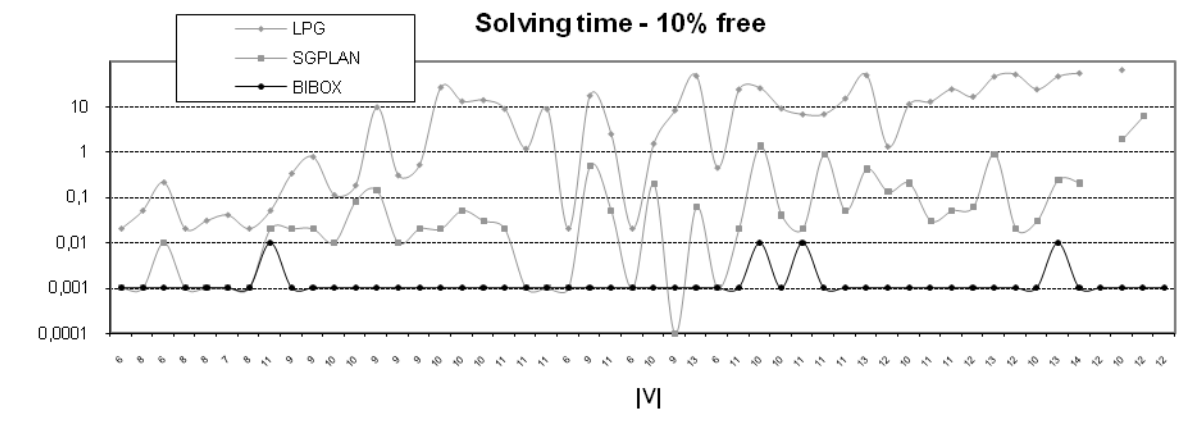
Experimenty:
BIBOX vs. plánovače (1)



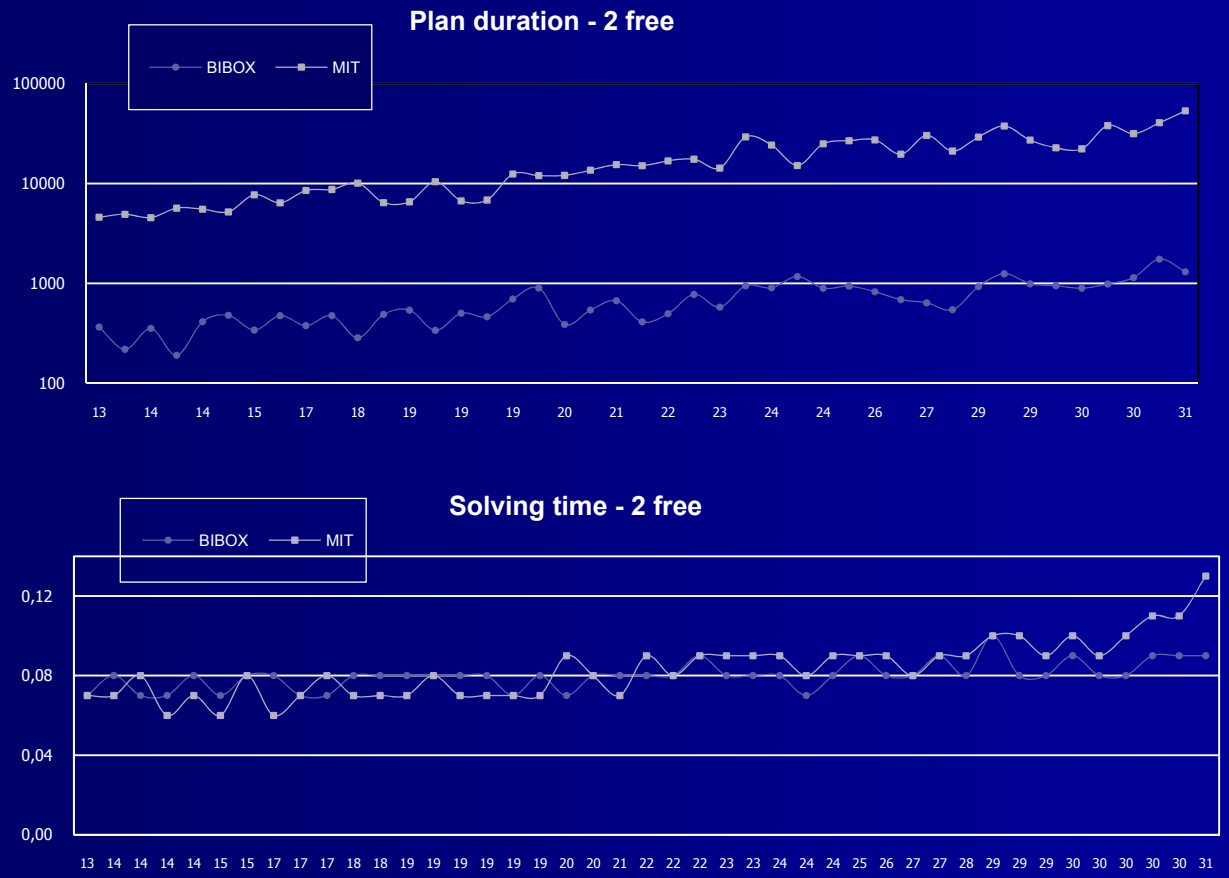


Experimenty:
BIBOX vs. plánovače (2)

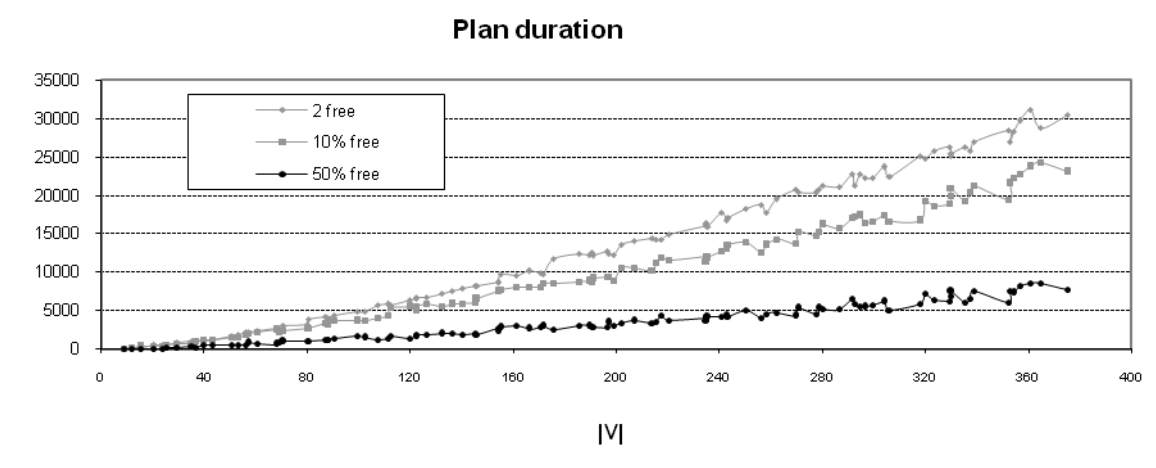


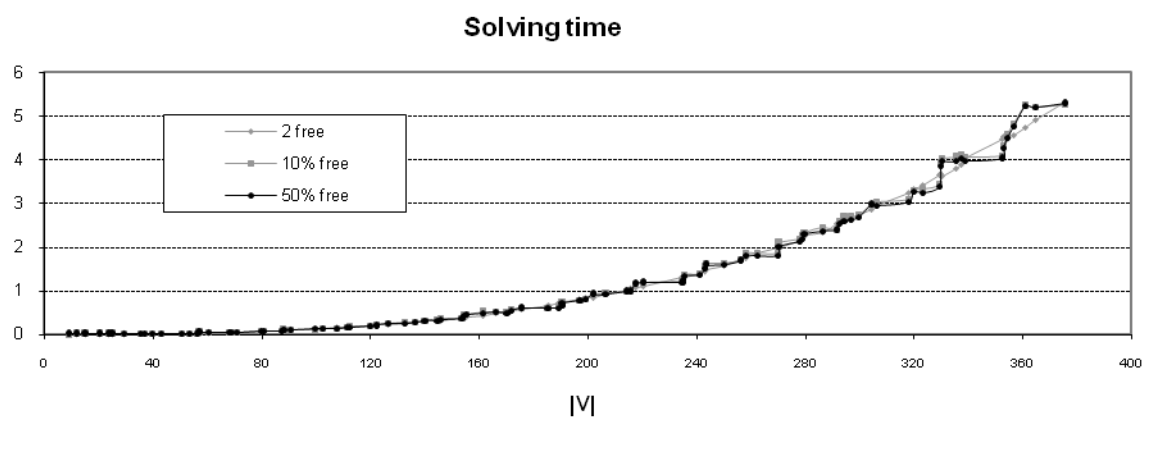


Experimenty:
BIBOX vs. MIT



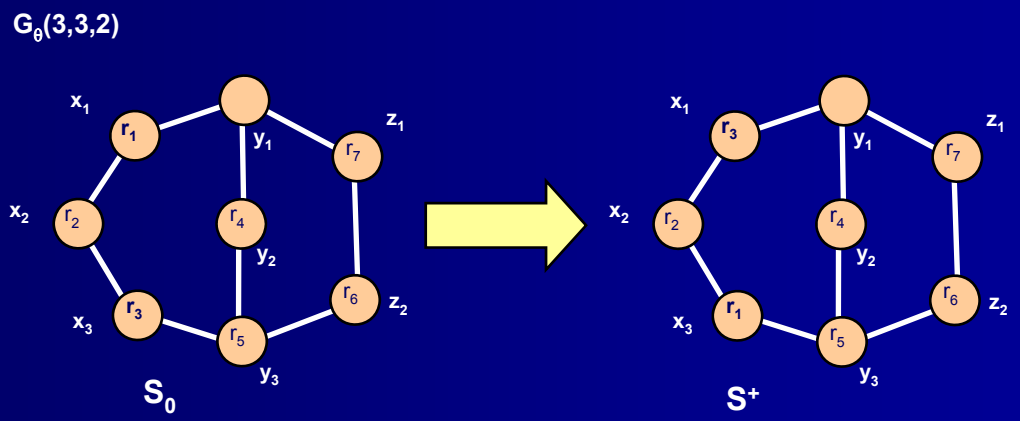
Experimenty:
Škálovatelnost algoritmu BIBOX





Využití optimálních maker (1)

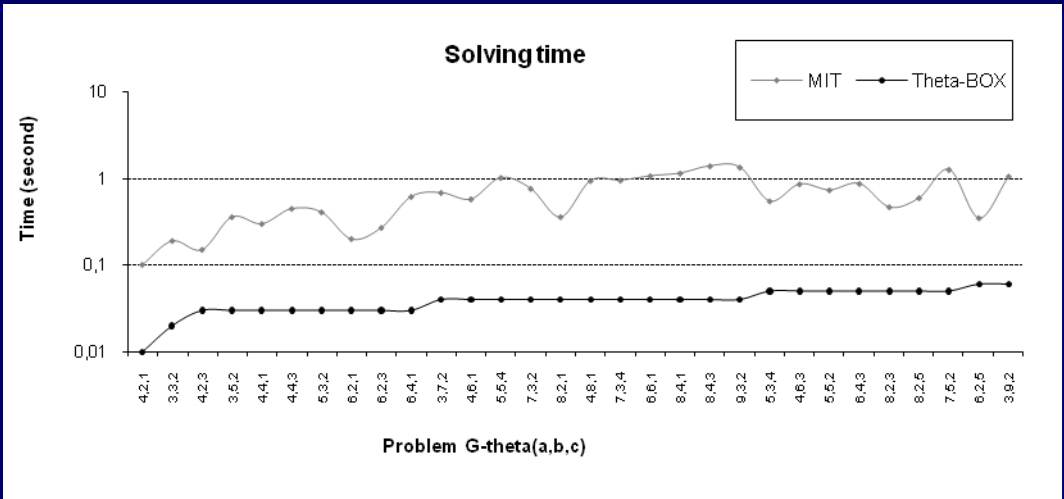
- Požadavek na **dva volné** vrcholy je příliš omezující.
  - Lze používat 3-tranzitivitu a 3-cykly z algoritmu MIT
  - Příliš zvětší počet pohybů v řešení
  - Sestavme řešení z předvypočtených optimálních řešení (optimální makra) pro 2-cykly (transpozice) a 3-cykly v počáteční kružnici a uchu (označujeme jako $\mathbf{G}_\theta$) – algoritmus Theta-BOX

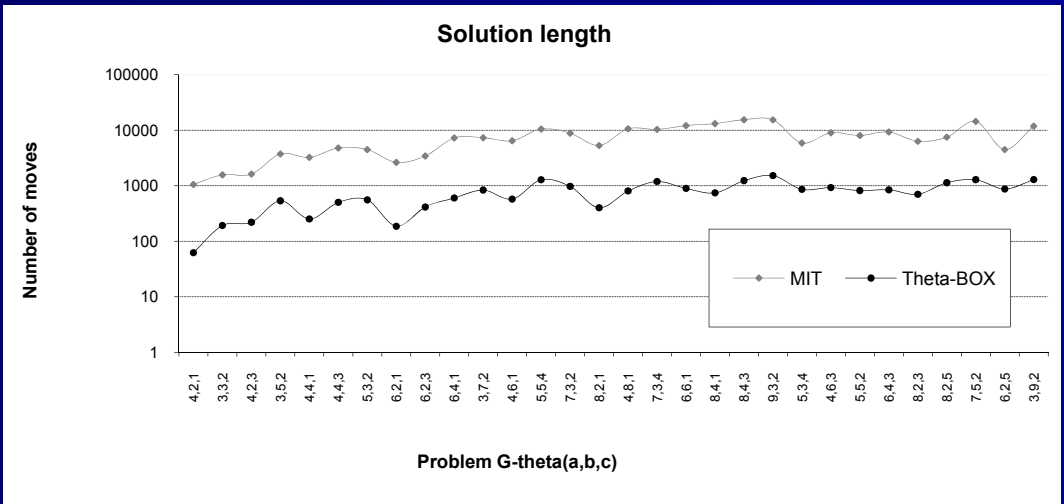


Využití optimálních maker (2)

- Použijeme **tvrzení 2** pro 3-cykly resp. obdobné tvrzení pro transpozice (2-cykly)
- Velikost databáze řešení pro graf G=(V,E)
  - Počet záznamů * velikost záznamu = $O(|V|^2)*\Omega(|V|^3)$ pro 2-cykly, možné jen když máme lichou kružnici
  - Počet záznamů * velikost záznamu = $O(|V|^3)*\Omega(|V|^3)$ pro 3-cykly
- Nalezení optimálních maker není snadné
  - Speciální varianta algoritmu IDA* s přidaným učením
  - Nelze používat on-line pro větší instance
- 3-cykly výhodnější vzhledem k délce výsledného řešení
- 2-cykly dávají menší databázi
- Není-li v databázi maker potřebné makro přítomné, použijeme MIT algoritmus

Experimenty:
Theta-BOX vs. MIT

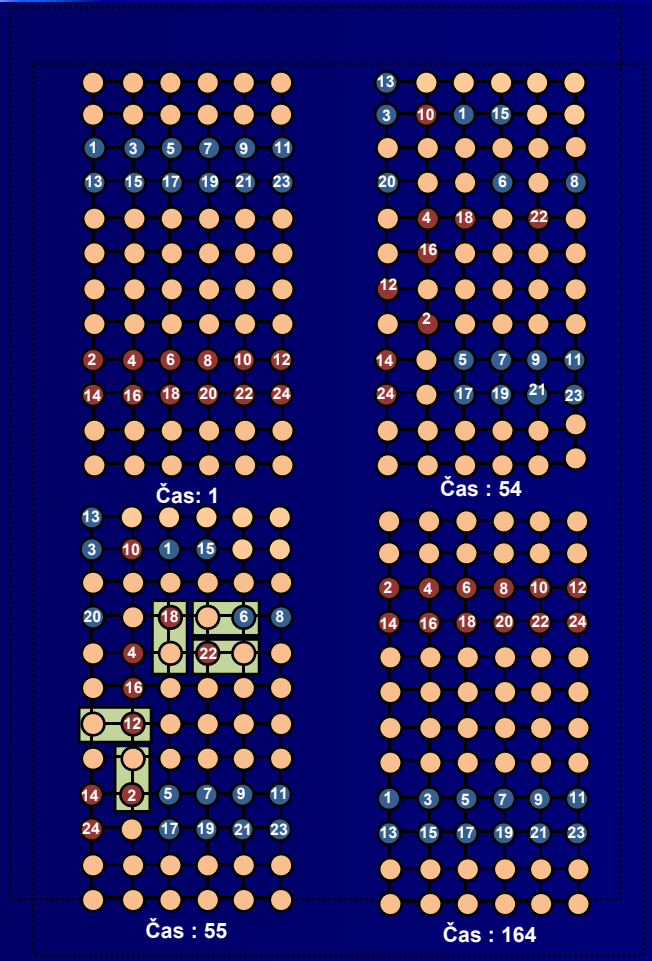




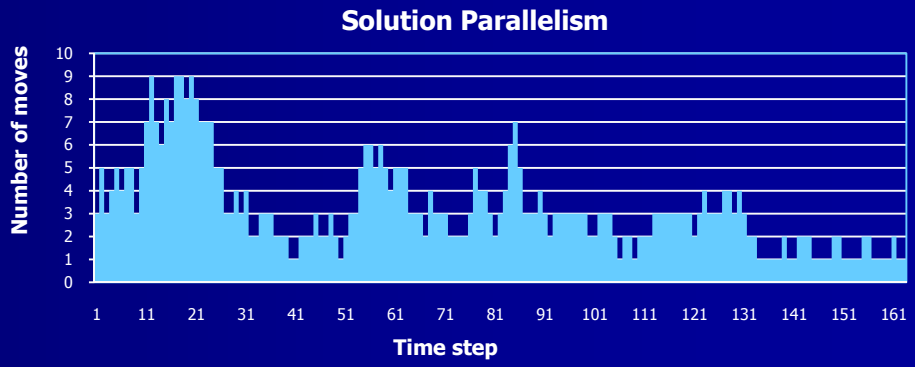
Paralelismus (1)

- Pokud graf obsahuje **více volných vrcholů**, lze provádět více pohybů robotů současně
  - Zkrácení délky řešení
- Místo volných vrcholů uvažujeme fiktivní roboty, jejichž pohyb bude ve výsledku ignorován
- Vyprodukujeme sekvenční řešení algoritmem BIBOX
- Řešení M (posloupnost pohybů – trojice … robot, odkud, kam … [r,u,v]) paralelizujeme **algoritmem kritické cesty**
  - Orientovaný acyklický graf T=(V,E), kde V budou pohyby z M a $[r_1,u_1,v_1]$-$[r_2,u_2,v_2]$ bude orientovaná hrana v E, jestliže $[r_1,u_1,v_1]$ se nachází před $[r_2,u_2,v_2]$ v M a $\{u_1,v_1\} \cap \{u_2,v_2\} \neq \emptyset$.
  - Vrcholům přiřadíme čas
    - Vrchol bez předchůdců bude mít čas 0
    - Jinak čas(v)=$\max_{\text{u předchůdce v}}$ čas(u) + 1
    - Lze v čase O(|V|+|E|)

Paralelismus (2)



Dvě skupiny robotů pohybující se proti sobě (červená a modrá skupina)



Otázky bez odpovědí (zatím)

- Jaká je složitost nalezení optimálního řešení problému cest pro roboty v 2-souvislém grafu, je-li více vrcholů volných?
  - nekonstantní počet volných vrcholů
  - jak tuto otázku správně položit
- Jaká je složitost nalezení optimálního řešení problému cest pro roboty v grafu $\mathbf{G}_\theta$?
- Jaká je složitost nalezení optimálního řešení transpozice resp. 3-cyklu robotů v grafu $\mathbf{G}_\theta$?
  - instance je zde zadaná 4 resp. 5 čísly

Závěr a poznámky

- Popis existujícího algoritmu MIT pro plánování cest pro roboty – využití 3-tranzitivity a 3-cyklu
  - 2-souvislé grafy
  - 1 volný vrchol
  - Nekvalitní řešení
- Nový algoritmus BIBOX – přímé umisťování robotů podle rozkladu grafu na ucha
  - 2-souvislé grafy
  - vyžaduje aspoň 2 volné vrcholy, s jedním vystačíme při použití optimálních maker
  - řešení lze dobře paralelizovat
- Lze upustit od požadavku, že cílový vrchol pohybu má být volný -> další zkrácení řešení

Literatura

- Wilson, R. M., 1974. *Graph Puzzles, Homotopy, and the Alternating Group.* Journal of Combinatorial Theory, Ser. B 16, pp. 86-96, Elsevier.
- Kornhauser, D., Miller, G. L., Spirakis, P. G., 1984. *Coordinating Pebble Motion on Graphs, the Diameter of Permutation Groups, and Applications.* Proceedings of the 25th Annual Symposium on Foundations of Computer Science (FOCS 1984), pp. 241-250, IEEE Press.
- Ratner, D., Warmuth, M. K., 1986. *Finding a Shortest Solution for the N × N Extension of the 15-PUZZLE Is Intractable.* Proceedings of the 5th National Conference on Artificial Intelligence (AAAI 1986), pp. 168-172, Morgan Kaufmann Publishers.
- Ryan, M. R. K., 2007. *Graph Decomposition for Efficient Multi-Robot Path Planning.* Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, pp. 2003-2008, 2007.
- Surynek, P. 2009*. A Novel Approach to Path Planning for Multiple Robots in Bi-connected Graphs.* Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan, IEEE Press, 2009, to appear.
- Surynek, P. 2009*. Towards Shorter Solutions for Problems of Path Planning for Multiple Robots in θ-like Environments.* Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference (FLAIRS 2009), Sanibel Island, FL, USA, AAAI Press, 2009, to appear.