



Vizualizace problémů pohybu po grafu

Pavel Surynek

Matematicko-fyzikální fakulta
Univerzita Karlova v Praze

Problémy pohybu po grafu

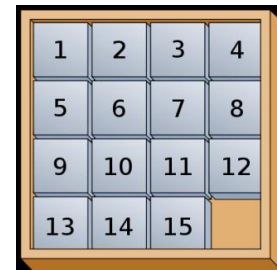
- ▶ **Abstrakce** k úlohám o pohybu více (autonomních nebo pasivních) entit v jistém prostředí (fyzickém či virtuálním).
 - ▶ Entity mají dané **počáteční** a **cílové** rozmístění v prostředí.
 - ▶ Chceme **naplánovat pohyby entit v čase**, aby dosáhly cílového rozmístění a přitom **respektovaly fyzikální omezení**.
- ▶ **Fyzikální omezení** jsou:
 - ▶ Entity spolu **nesmějí kolidovat**.
 - ▶ Entity **nesmějí narážet do překážek** vyskytujících se prostředí.
- ▶ Existují **dvě abstrakce** k úloze o pohybu:
 - ▶ **Pohyb kamenů po grafu** (*pebble motion on a graph*)
 - ▶ **Plánování cest pro mnoho robotů** (*multi-robot path planning*)



Úloha o pohybu kamenů po grafu (1)

Wilson, 1974; Kornhauser et al., 1984

- ▶ Populární pohybová úloha, jež lze abstrahovat jako problém pohybu kamenů po grafu je známá jako **Lloydova patnáctka (devítka)**.

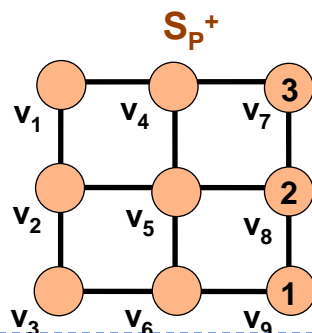
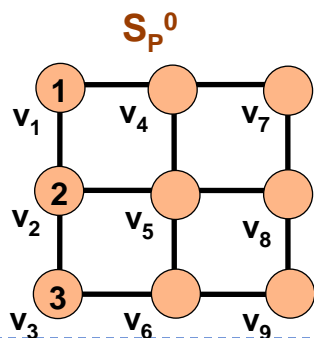


- ▶ Entita je kámen s číslem (*pebble*).
- ▶ Prostředí modelujeme jako neorientovaný graf, kde **vrcholy reprezentují pozice** v prostředí a **hrany možnost průchodu entity mezi pozicemi**.
- ▶ **Formální definice** problému pohybu kamenů po grafu:
 - ▶ Je to čtveřice $\Pi = (G, P, S_p^0, S_p^+)$, kde:
 - ▶ $G=(V,E)$ je neorientovaný **graf**,
 - ▶ $P = \{p_1, p_2, \dots, p_\mu\}$, kde $\mu < |V|$ je **množina kamenů**,
 - ▶ $S_p^0: P \rightarrow V$ je prostá funkce určující **počáteční rozložení** kamenů a
 - ▶ $S_p^+: P \rightarrow V$ je prostá funkce určující **cílové rozložení** kamenů.

Úloha o pohybu kamenů po grafu (2)

Wilson, 1974; Kornhauser et al., 1984

- ▶ Čas modelujeme diskrétně. **Časové kroky** a jejich uspořádání jsou izomorfní přirozeným číslům.
- ▶ **Dynamicita** úlohy je následující:
 - ▶ Kámen nacházející se v časovém kroku i v jistém vrcholu se může přesunout do sousedního vrcholu v časovém kroku $i+1$, jestliže cílový vrchol je v časovém kroku i **neobsazený** a **žádný jiný kámen** se současně nepřesouvá do stejného cílového vrcholu.
- ▶ Pro dané $\Pi = (G, P, S_p^0, S_p^+)$, chceme najít:
 - ▶ Posloupnost pohybů pro každý kámen tak, že je splněna podmínka dynamicity úlohy a kámen dosáhne cílového vrcholu.



Řešení problému pohybu po grafu, kde $P=\{1,2,3\}$

délka řešení=7

$M_1=[v_1, v_4, v_7, v_8, v_9, v_9, v_9]$
$M_2=[v_2, v_2, v_1, v_4, v_7, v_8, v_8]$
$M_3=[v_3, v_3, v_3, v_2, v_1, v_4, v_7]$

Časový krok: 1 2 3 4 5 6 7

Úloha plánování cest pro mnoho robotů (1)

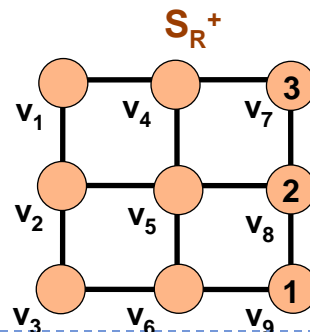
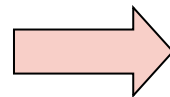
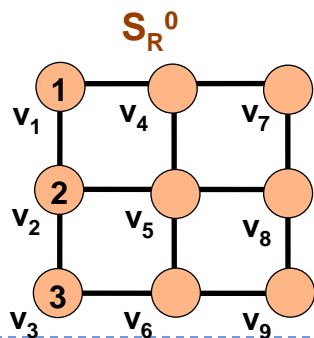
Ryan, 2007

- ▶ **Formální definice** problému plánování cest pro mnoho robotů:
 - ▶ Je to čtveřice $\Sigma = (G, R, S_R^0, S_R^+)$, kde:
 - ▶ $G=(V,E)$ je neorientovaný **graf**,
 - ▶ $R = \{r_1, r_2, \dots, r_v\}$, kde $v < |V|$ je **množina robotů**,
 - ▶ $S_R^0: R \rightarrow V$ je prostá funkce určující **počáteční rozmístění** robotů a
 - ▶ $S_R^+: R \rightarrow V$ je prostá funkce určující **cílové rozmístění** robotů.
- ▶ **Zeslabíme** podmínku na **dynamicitu**:
 - ▶ Robot nacházející se v časovém kroku i v jistém vrcholu se může přesunout do sousedního vrcholu v časovém kroku $i+1$, jestliže je cílový vrchol je **neobsazený** nebo **právě opouštěný** a **žádný jiný robot** v témže časovém nevstupuje do stejného cílového vrcholu.

Úloha plánování cest pro mnoho robotů (2)

Ryan, 2007

- ▶ Pro dané $\Sigma = (G, R, S_R^0, S_R^+)$, chceme najít:
 - ▶ Posloupnost pohybů pro každého robota tak, že je splněna podmínka dynamicity úlohy a robot dosáhne cílového vrcholu.
- ▶ Podmínka dynamicity pro pohyb kamenů po grafu **implikuje** podmínku dynamicity pro plánování cest pro mnoho robotů.
 - ▶ Řešení úlohy pohybu kamenů po grafu je také řešením odpovídající úlohy plánování cest pro mnoho robotů.
- ▶ **V čem je tedy rozdíl?**
 - ▶ Plánování cest pro mnoho robotů umožňuje vyšší **paralelismus**.



Řešení problému plánování cest pro mnoho robotů, kde $R=\{1,2,3\}$

délka řešení=5

$O_1=[v_1, v_4, v_7, v_8, v_9]$
$O_2=[v_2, v_1, v_4, v_7, v_8]$
$O_3=[v_3, v_2, v_1, v_4, v_7]$

Časový krok: 1 2 3 4 5

Má smysl řešit úlohy pohybu po grafu?

- ▶ Přesouvání kontejnerů
(entita = **kontejner**)
- ▶ Intenzivní doprava
(entita = **automobil** (v zácpě))
- ▶ Přesuny dat
(entita = **datový paket**)
- ▶ Zobecněné výtahy
(entita = **výtah**)



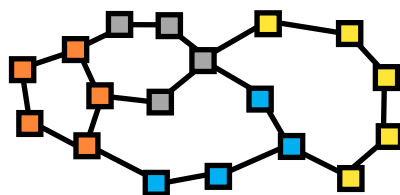
Jsou úlohy pohybu po grafu **lehké** či **těžké**?

- ▶ **Základní** varianta obou úloh je **snadno řešitelná**:
 - ▶ Existuje algoritmus pracující v polynomiálním čase ($O(|V|^3)$), který vygeneruje řešení délky $O(|V|^3)$ problému pohybu kamenů po grafu (Kornhauser et al., 1984).
 - ▶ Podle uvedených pozorování je použitelný i na plánování cest pro mnoho robotů.
- ▶ Chceme-li najít **řešení nejkratší možné délky** obtížnost roste:
 - ▶ Optimální varianta úloh pohybu po grafu je **NP-těžká** a navíc **neexistuje** pro ni **polynomiální aproximační schéma** (pokud $P \neq NP$) (Ratner a Warmuth, 1986).
- ▶ Zaměřili jsme se na vygenerování **kvalitnějších** (tj. kratších) **neoptimálních** řešení:
 - ▶ Omezení na **2-souvislé grafy** - téměř vždy je úloha řešitelná.

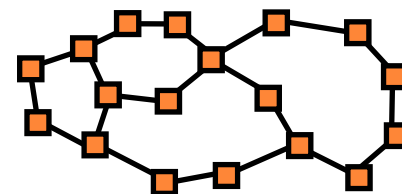


Případ 2-souvislého grafu

- ▶ Úlohy pohybu po grafu na **2-souvislých** grafech jsou z praktického hlediska nejdůležitější.
 - ▶ Téměř všechny cílová rozložení entit v grafu jsou dosažitelná z libovolného počátečního rozložení.
- ▶ **Další omezení** je, že dovolujeme pouze **jeden volný vrchol** (toto představuje neobtížnější situaci).
- ▶ Neorientovaný graf $G=(V,E)$ je **2-souvislý**, jestliže $|V| \geq 3$ a $\forall v \in V$ je graf $G=(V-\{v\},E')$, kde $E'=\{\{x,y\} \in E \mid x,y \neq v\}$, souvislý.
- ▶ **Důležitá vlastnost:** Každý 2-souvislý graf sestojit postupným přidáváním uch k počáteční kružnici
→ rozklad na ucha

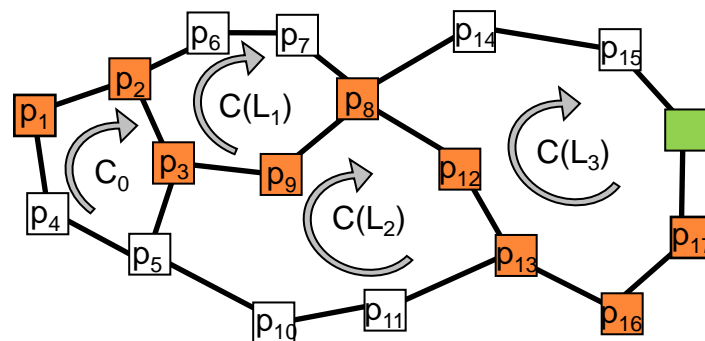


- počáteční kružnice
- 1. ucho
- 2. ucho
- 3. ucho



Algoritmus **BIBOX** (1)

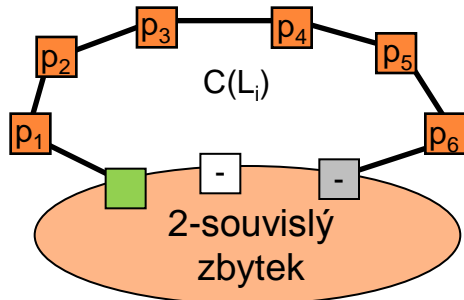
- ▶ Algoritmus **BIBOX** řeší obě úlohy pohybu po grafu.
 - ▶ Předpokládá, že prostředí je modelováno **2-souvislým** grafem.
 - ▶ Opírá se o **znalost rozkladu na ucha**.
 - ▶ Předpokládáme právě jeden neobsazený vrchol.
 - ▶ Není-li tomu tak, obsadí se vrcholy fiktivními kameny/roboty. Jejich pohyb se pak z finálního řešení odfiltruje.
 - ▶ Algoritmus opět pracuje v polynomiálním čase ($O(|V|^3)$), ovšem konstanta v odhadu je nižší než u algoritmu z (Kornhauser et al., 1984).
 - ▶ Základní dovednost je přesun kamenu/roboty do vybraného vrcholu grafu:
 - ▶ **Přesouvání volného vrcholu** a
 - ▶ **rotace podél uch.**



Algoritmus BIBOX (2)

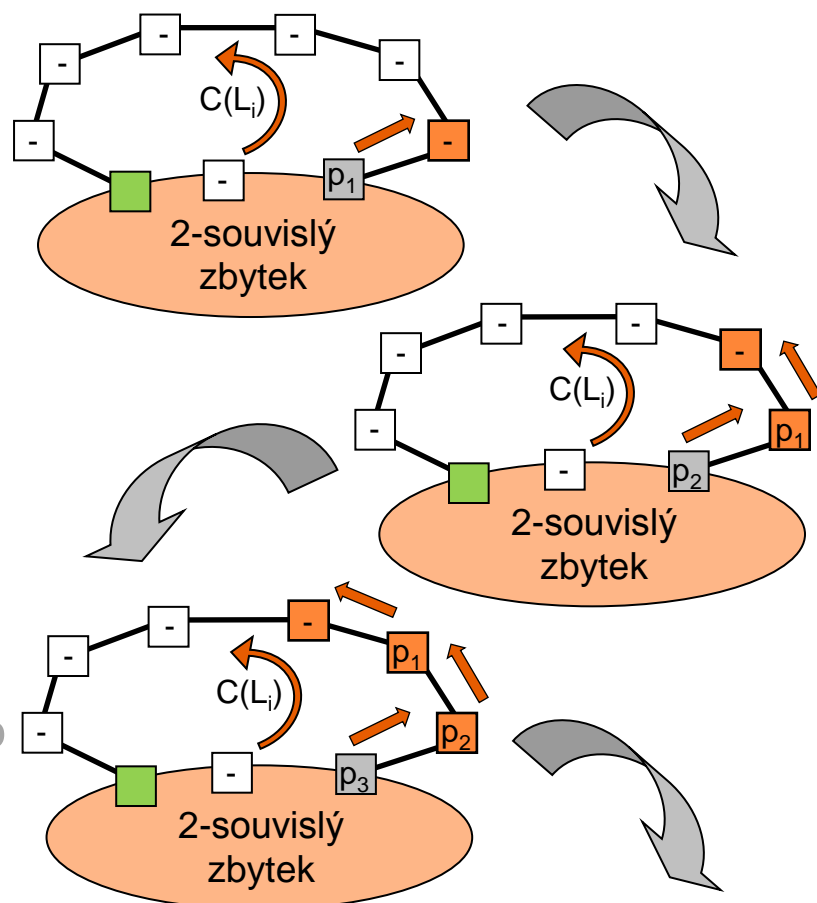
- ▶ Zvládneme-li přesun individuálního kamenu/roboty lze provádět složitější přesuny:

- ▶ Skládání kamenů/robotů do ucha ve správném pořadí.



- ▶ **Zásobníkové skládání:**

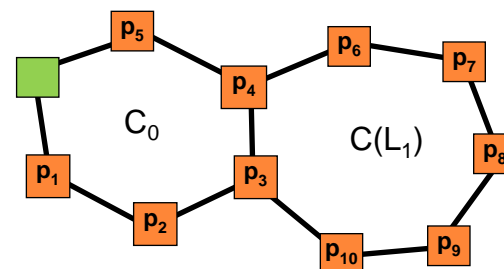
- ▶ Vezmeme **poslední ucho** rozkladu.
- ▶ Přesuneme kámen/roboty do **šedého** vrcholu.
- ▶ Provedeme rotaci ucha (pomocí **zeleného** volného vrcholu).



...

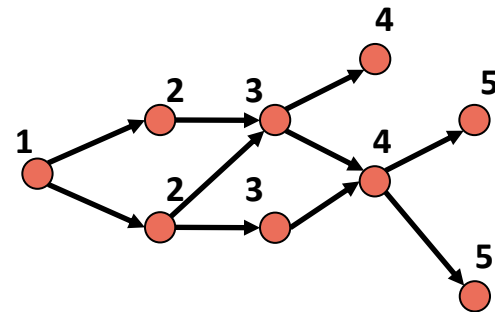
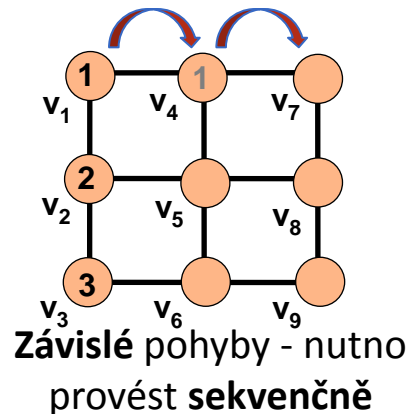
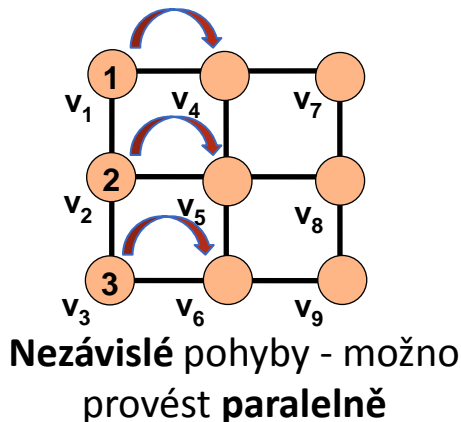
Algoritmus BIBOX (3)

- ▶ Počáteční kružnice a první ucho rozkladu představují zvláštní situaci.
- ▶ Zásobníkové skládání, zde **nefunguje**.
- ▶ Výslednou (sudou) **permutaci** kamenů/robotů nutno poskládat z jednotlivých **rotací podél 3-cyklů** (bez detailů).
 - ▶ **Úzké hrdlo** algoritmu, zlepšování pomocí předvypočtené databáze optimálních řešení.
 - ▶ Podstatné vylepšení: používat optimální řešení tzv. **slabých rotací podél 3-cyklů**.



Zvyšování paralelismu v řešeních

- ▶ Řešení generovaná algoritmem BIBOX nezohledňují možnosti **paralelismu** (více než jeden pohyb je proveden mezi dvojicí po sobě jdoucích časových kroků).
 - ▶ V úloze pohybu kamenů po grafu je paralelismus možný, když graf obsahuje alespoň **dva neobsazené vrcholy**.
 - ▶ U plánování cest pro mnoho robotů je **paralelismus možný vždy**.
- ▶ Jak zkonstruovat paralelní řešení?
 - ▶ Definuje se závislost mezi pohyby – závislé pohyby musejí následovat jeden po druhém.
 - ▶ Metodou **kritické cesty** spočítáme **nejdřívější čas** provedení pohybů.



Nedostatky popsaného přístupu

- ▶ Není-li graf zcela zaplněn kameny/roboty může navržený postup vytvářet jisté redundance v rámci řešení.
 - ▶ Přidáme fiktivní kameny/roboty, úlohu vyřešíme.
 - ▶ Z vygenerovaného řešení odstraníme pohyby fiktivních kamenů/robotů.
 - ▶ Řešení zparalelizujeme metodou kritické cesty.
- ▶ Pomocí vizualizačního programu GraphRec se podařilo identifikovat několik typů **redundancí**:
 - ▶ **(i) Inverzní pohyby** – pohyb negující bezprostředně předcházející pohyb.
 - ▶ **(ii) Redundantní pohyby** – sekvence pohybů, která kámen/robot vrátil do výchozí pozice.
 - ▶ **(iii) Dlouhá posloupnost pohybů** – sekvence pohybů, která kámen/robot přesouvá do jiného vrcholu, ale přitom existuje kratší sekvence pohybů provádějící totéž.

Odstraňování redundancí

- ▶ Uvedené tři druhy redundancí jsou postupnými zobecněními, tj. redundance typu **(iii) zahrnuje všechny předchozí**.
- ▶ Odstranění redundancí lze provádět přímočarými algoritmy (podle definice dané redundance).
 - ▶ **Časové nároky** algoritmů na odstranění jednotlivých typů redundancí **rostou** (redundance typu **(iii) má největší časové nároky** při odstraňování).
 - ▶ Proto se algoritmy **aplikují postupně**.
 - ▶ 1. odstranit inverzní pohyby (i)
 - ▶ 2. odstranit redundantní pohyby (ii)
 - ▶ 3. odstranit dlouhé posloupnosti (iii)
 - ▶ Náročnější algoritmus běží na již potenciálně zkráceném řešení.
- ▶ Provedli jsme **srovnání**, jaký přínos odstranění redundancí má.



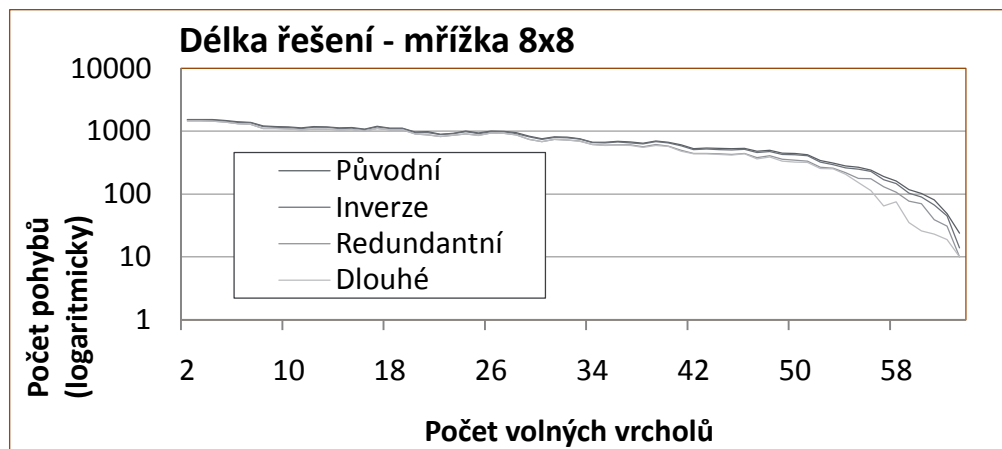
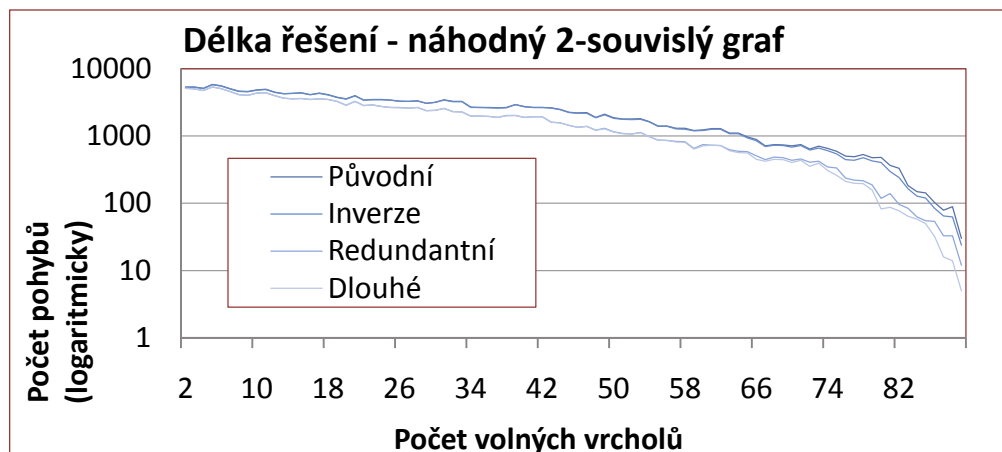
Experimenty (pro pohyb kamenů po grafu)

▶ Náhodný 2-souvislý graf:

- ▶ Postupné přidávání uch náhodné délky k již zkonstruovanému grafu.
- ▶ Počáteční a cílové rozložení kamenů voleno jako náhodná permutace.

▶ Mřížka 8x8:

- ▶ Opět náhodné počáteční a cílové rozložení.
- ▶ Postupně zmenšujeme počet kamenů.
- ▶ Obsahuje-li graf více volných vrcholů, dosahujeme až několikanásobného zkrácení řešení.



Literatura (1)

- ▶ **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 $N \times N$ 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.
-

Literatura (2)

- ▶ **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, pp. 3613-3619, IEEE Press, 2009.
 - ▶ **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, pp. 207-212, AAAI Press, 2009.
 - ▶ **Surynek, P.** 2009. Making Solutions of Multi-robot Path Planning Problems Shorter Using Weak Transpositions and Critical Path Parallelism. Proceedings of the 2009 International Symposium on Combinatorial Search (SoCS 2009), Lake Arrowhead, CA, USA, 6 pages, University of Southern California, 2009.
 - ▶ **Surynek, P.** 2009. An Application of Pebble Motion on Graphs to Abstract Multi-robot Path Planning. Proceedings of the 21st International Conference on Tools with Artificial Intelligence (ICTAI 2009), Newark, NJ, USA, pp. 151-158, IEEE Press, 2009, ISBN 978-0-7695-3920-1, ISSN 1082-3409.
-