Towards Shorter Solutions for Problems of Path Planning for Multiple Robots in θ-like Environments

Pavel Surynek

Charles University in Prague

Faculty of Mathematics and Physics

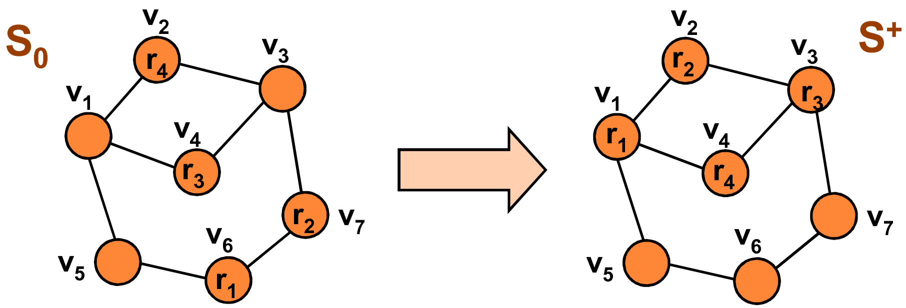
The Czech Republic

Path Planning for Multiple Robots

- **Input:** Graph **G=(V,E)** and a set of robots $\mathbf{R=\{r_1,r_2,\ldots,r_\mu\}}$, where $\mathbf{\mu<|V|}$
  - **each robot** is placed **in a vertex** (at most one robot in a vertex)
  - a **robot can move into an unoccupied** vertex through an edge (no other robot is allowed to enter the vertex)
  - **initial positions** of robots … simple function $\mathbf{S_0: R \to V}$
  - **goal positions** of robots … simple function $\mathbf{S^+: R \to V}$
- **Task: Find a sequence of allowed moves** for robots such that all the robots reach their goal positions starting from the given initial positions

EXAMPLE OF MULTI-ROBOT PATH PLANNING

- **Initial positions** of robots given by $\mathbf{S_0}$
- **Goal positions** of robots given by $\mathbf{S^+}$



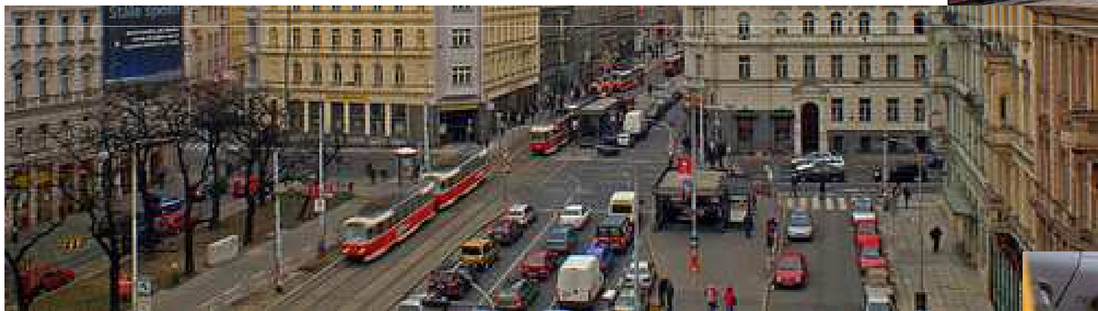
length=12

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

- A solution of length **12** is shown
  - $\mathbf{P_r}$ is a sequence of positions of the robot **r** in all the discrete time steps
  - Notice the **parallelism** within the solution
  - **Short solutions** are preferred (shortest NP-complete)

Motivation for the Problem

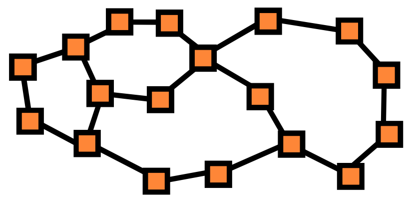
- Rearrangement of agents in tight space
- Automated control of heavy traffic



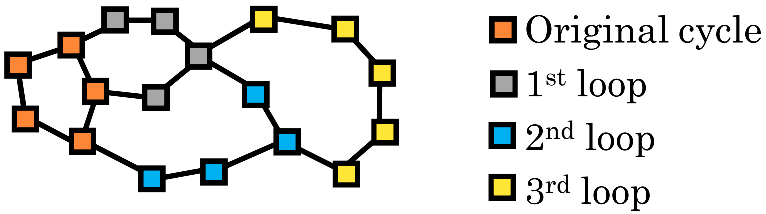
- Data transfer with limited size of the cache memory
- Generalized lifts in future city-size buildings

A Case with a Bi-connected Graph

- Multi-robot path planning problems with **bi-connected graphs** are most important for practice
- Almost **all the goal arrangements** of robots are reachable using allowed moves
- An undirected graph G=(V,E) is **bi-connected** if and only if |V|≥3 and ∀v∈V the graph G=(V-{v},E') where E'={{x,y}∈E | x,y ≠ v} is connected

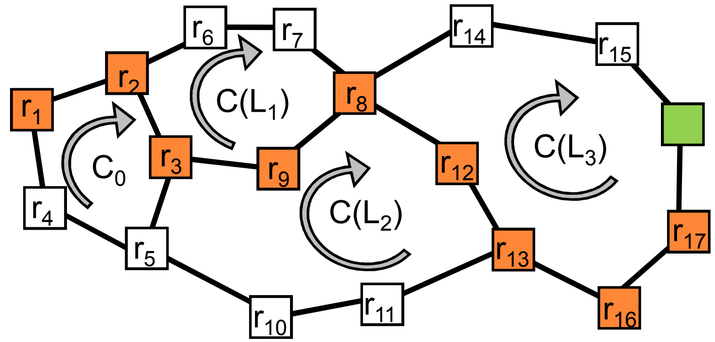


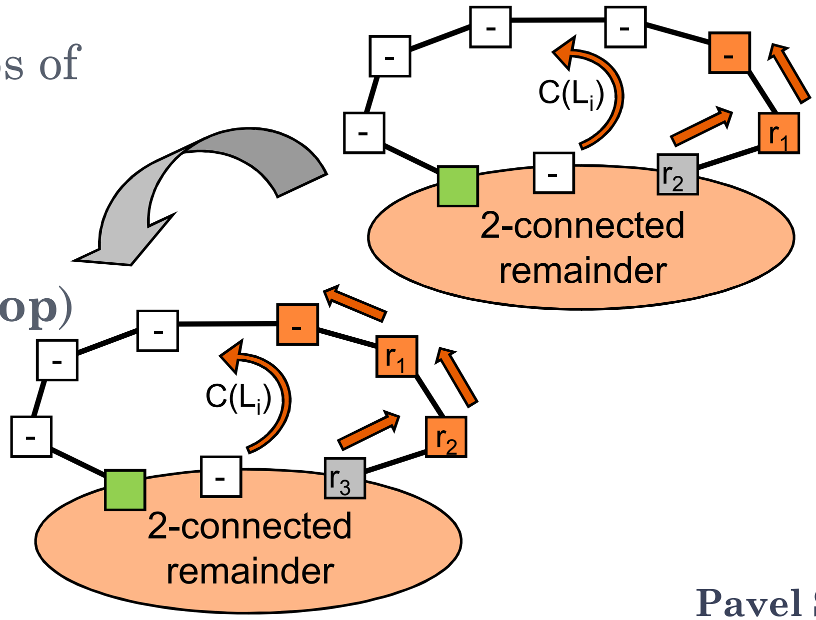
- **Property:** Every bi-connected graph can be constructed from a cycle by consecutive adding of loops → **loop decomposition**

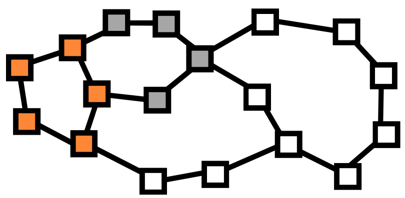


A Trouble with a Bi-connected Graph and Single Unoccupied Vertex

- Suppose a multi-robot path planning problem with a bi-connected graph
- The knowledge of loop decomposition allows us:
  - to **move an unoccupied** vertex to an arbitrary position
  - to **move a robot** to an arbitrary position
  - to **stack robots** in the loops of the decomposition
  - **original cycle** of the loop decomposition remains problematic (plus the **1st loop**)

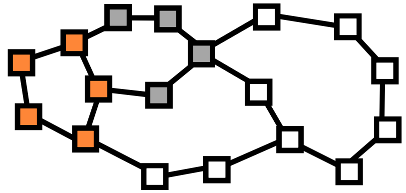


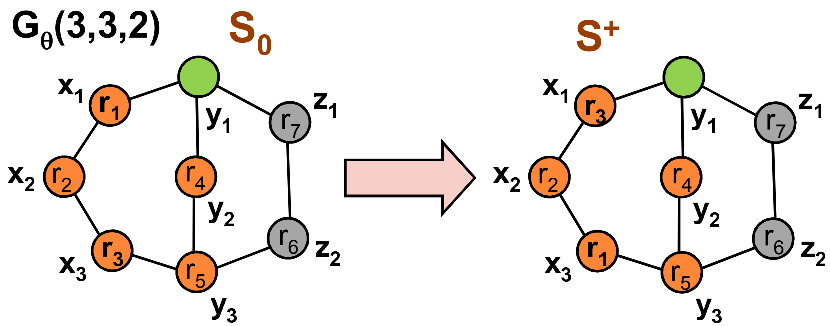




A NOTION OF θ-LIKE GRAPH

- Let us concentrate on an **original cycle and the 1st loop** of the decomposition (the remaining graph can be solved using stacking robots in the loops)
- This graph is called a **θ-like graph** – $G_\theta(a,b,c)$ specified by three parameters **a, b, c** where:
  - **a** is the size of the left loop … vertices $\{x_1,x_2,\ldots, x_a\}$
  - **b** is the size of the middle bone … vertices $\{y_1,y_2,\ldots, y_b\}$
  - **c** is the size of the right loop … vertices $\{z_1,z_2,\ldots, z_c\}$
  - the vertex y_1 is preserved **unoccupied**





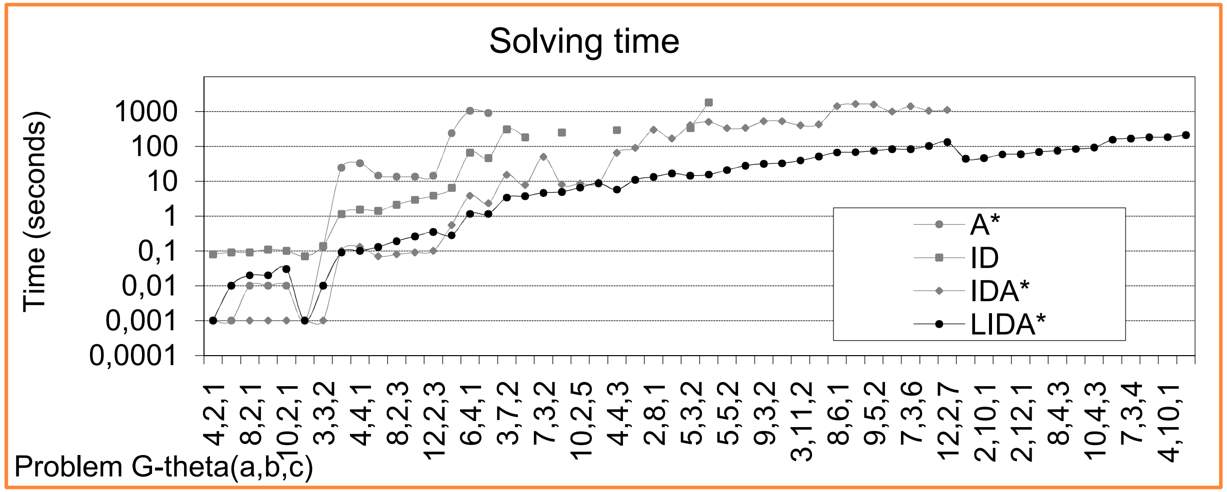
- Can be solved using **3-transitivity** of bi-connected graphs (any three robots can be moved to any three vertices)
- 3-transitivity induces **3-cycles**, 3-cycles induce **even** permutations
- However, the use of 3-transitivity generates **lengthy** solutions

θ-like Graphs and Pattern Database with Optimal Solutions

- Interpret arrangement of robots a **θ-like graph** as a **permutation**
- **Proposition 1**
  - **Any permutation** over **μ** elements can be obtained as a composition of at most **μ-1 transpositions**.
- **Proposition 2**
  - Any **even permutation** over **μ** elements can be obtained as a composition of at most **μ-1 rotations along a triple** (**3-cycle**).
- **Proposition 3**
  - Rotation along a **triple** is always solvable in a **θ-like graph**; **transposition** is solvable, if the **θ-like graph** contains an **odd cycle**.
- **Transposition** and **3-cycle rotation** case are good candidates to be stored in a pattern database (**optimal solutions** to these cases are stored)
  - **polynomial number** of records in the database (**$O(|V|^5) + O(|V|^6)$**)
  - they completely solve every multi-robot problem on a θ-like graph

Search for Optimal Solutions of Transposition and 3-cycle Cases

- It is not easy to generate the pattern database with optimal solution to transposition and 3-cycle cases
- It seems that this problem is computationally difficult; some indications are:
  - finding and **optimal solution** of multi-robot path planning problem on an **arbitrary bi-connected graph** is **NP-hard**
  - the size of the input of the problem is the **number of bits** necessary to encode **5** (**transposition** case) or **6** (**3-cycle** rotation case) **integers**
- Currently, nothing better than the **search**
  - a special variant of the **IDA* algorithm** enhanced with **learning** – called **Learning IDA* (LIDA*)** – better than A*, iterative deepening, IDA*

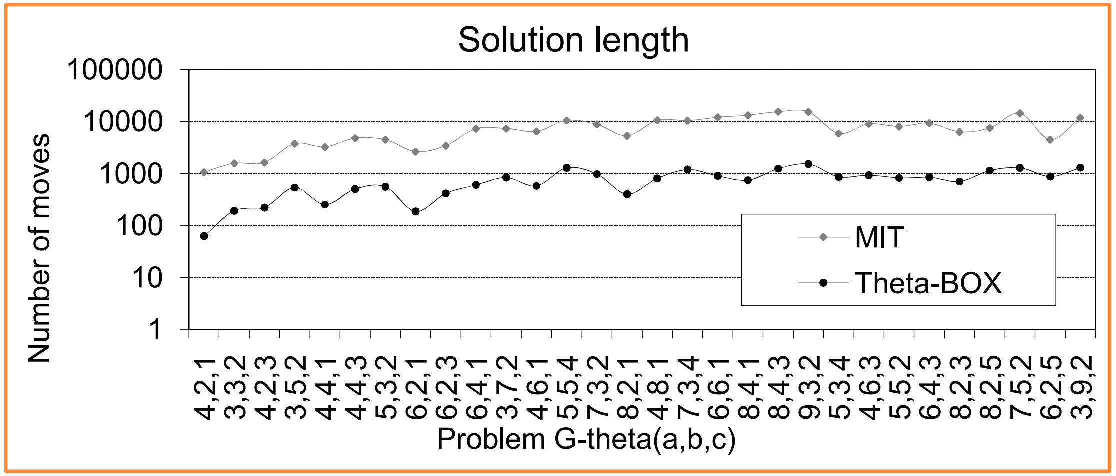


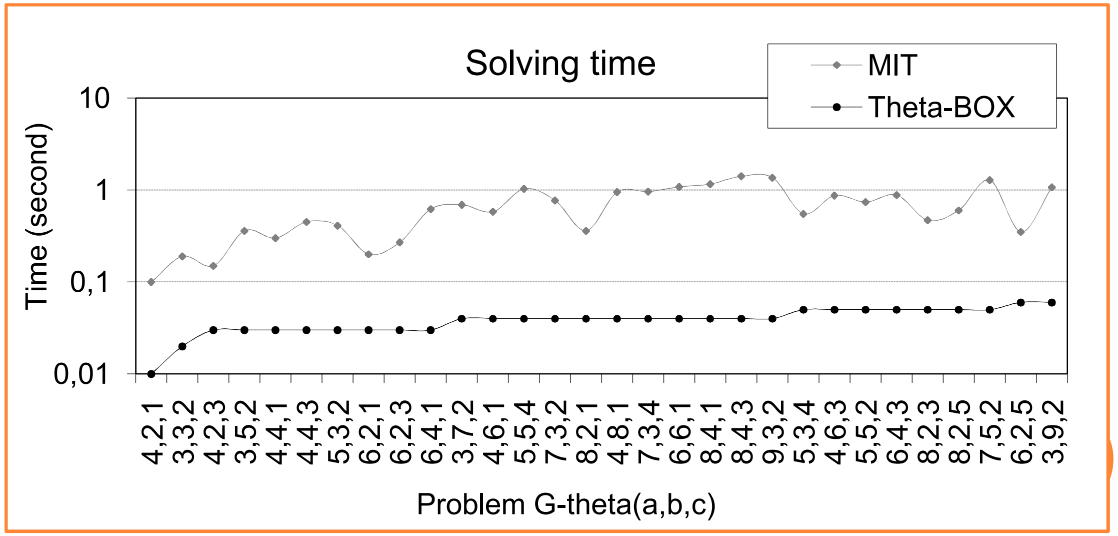
Pattern Database Compared with a Method Based on 3-Transitivity

- Suppose that all the records necessary to solve a given problem are stored in the pattern database
- Optimal solutions from the pattern database are used to compose a sub-optimal solution of a given problem – the algorithm called **θ-BOX**
  - Worst case **time complexity of $O(|V|^4)$**
  - The **length** of the generated solution is also $O(|V|^4)$
- A competitor: an algorithm of Kornhauser, Miller, Spirakis (FOCS 1984) – we call it the **MIT** algorithm
  - Works on bi-connected graphs with at least **one unoccupied**
  - Based on a property of **3-transitivity** of bi-connected graphs (any three robots can be moved to any three vertices)
  - Worst case **time complexity** of $O(|V|^3)$
  - The **length** of the generated **solution** is also $O(|V|^3)$
- The MIT algorithm is **theoretically better,** however it has **high constants in the asymptotic estimations**

EXPERIMENTAL COMPARISON OF θ-BOX WITH MIT

- Both algorithms implemented in C++
- **Length of solutions** and **solving time** are compared
- Random **θ-like** graphs with up to 30 vertices
- The **θ-BOX** algorithm produces order of magnitude **shorter solutions** than **MIT**
- In the terms of **runtime** the **MIT** algorithm is also outperformed by **θ-BOX**





Conclusions and Remarks

- A **novel approach** (θ-**BOX algorithm**) how to solve multi-robot path planning problem in a **θ-like environment** (θ-like graph) has been proposed
  - based on **composing a sub-optimal solution** of the optimal solutions of special cases pre-calculated into the pattern database (**transposition** and **3-cycle** case)
  - existing state-of-the-art algorithm (MIT) has been outperformed in terms of **speed** and **quality** (length) of produced **solutions**
- Topics for **future work:**
  - we don't know the **theoretical complexity** of producing an optimal solution to the transposition and 3-cycle case
  - notice that all the robots except that 2 or 3 **are forced to hold their positions** – too strong condition; relaxing leads to **shorter solutions** (however the size of database increases)
  - **integration** of **θ-BOX** algorithm into an algorithm for bi-connected graphs - more aggressive use of pattern database