A Novel Approach to Path Planning for Multiple Robots in Bi-connected Graphs

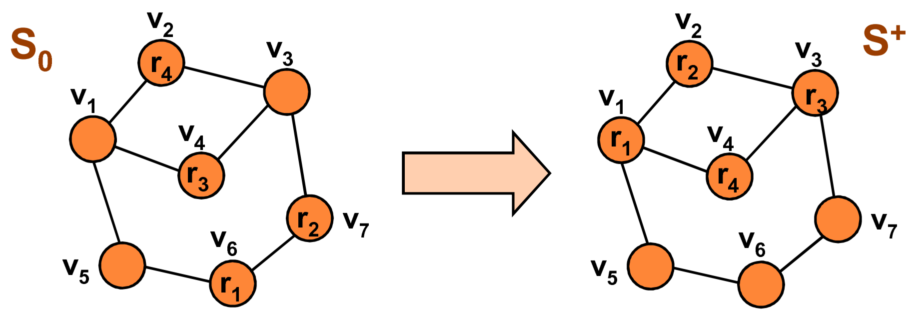
Pavel Surynek
Charles University in Prague
Faculty of Mathematics and Physics
The Czech Republic

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 $\boldsymbol{\mu<|V|}$
  - **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 $\mathbf{S_0: R\rightarrow V}$
  - **goal positions** of robots … simple function $\mathbf{S^+: R\rightarrow V}$
- **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 $\mathbf{S^+}$



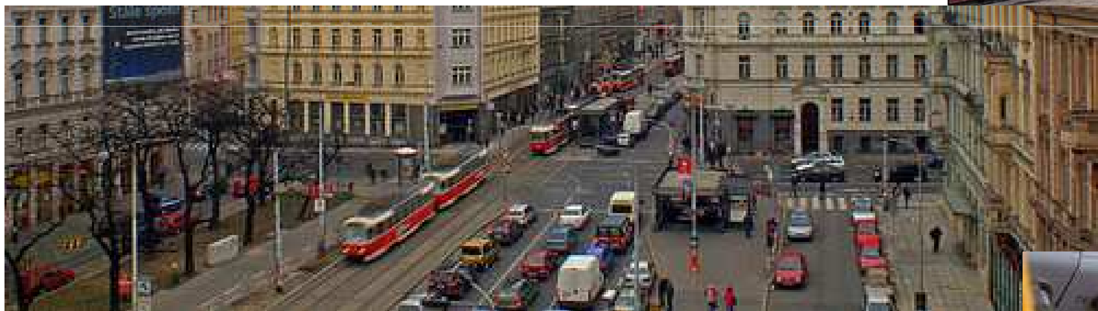
length=12

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

- 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

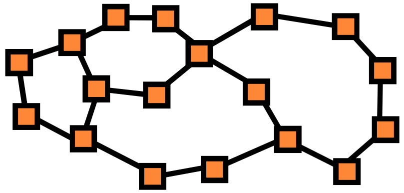
- Rearrangement of agents in tight space
- Automated control of heavy traffic



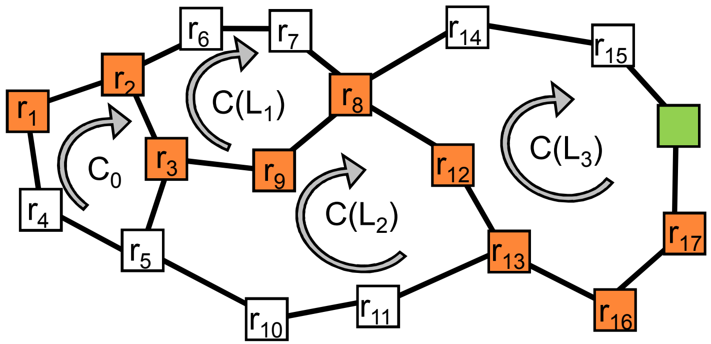
- Data transfer with limited size of the cache memory
- Generalized lifts in future city-size buildings

BIBOX: An Algorithm for Bi-connected Graphs

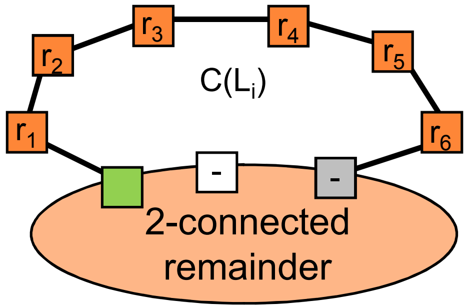
- An undirected graph G=(V,E) is **bi-connected** if and only if |V|≥3 and ∀v∈V the graph G=(V-{v},E') where E'={{x,y}∈E | x,y ≠ v} is connected
- **Property:** Every bi-connected graph can be constructed from a cycle by consecutive adding of loops



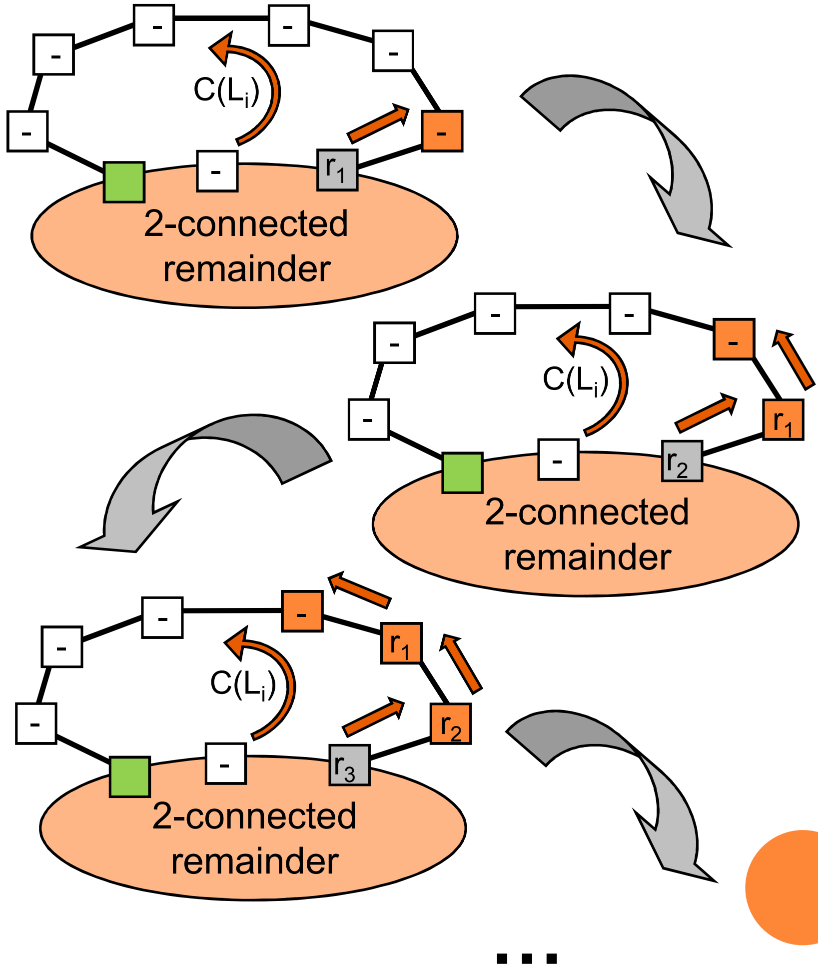
- The **loop decomposition** can be obtained in worst case time of **O(|V|+|E|)**
- The knowledge of loop decomposition allows us:
  - to **move an unoccupied** vertex to an arbitrary position
  - to **move a robot** to an arbitrary position



BIBOX: Placing Robots in a Regular Loop (≠ Original Cycle)

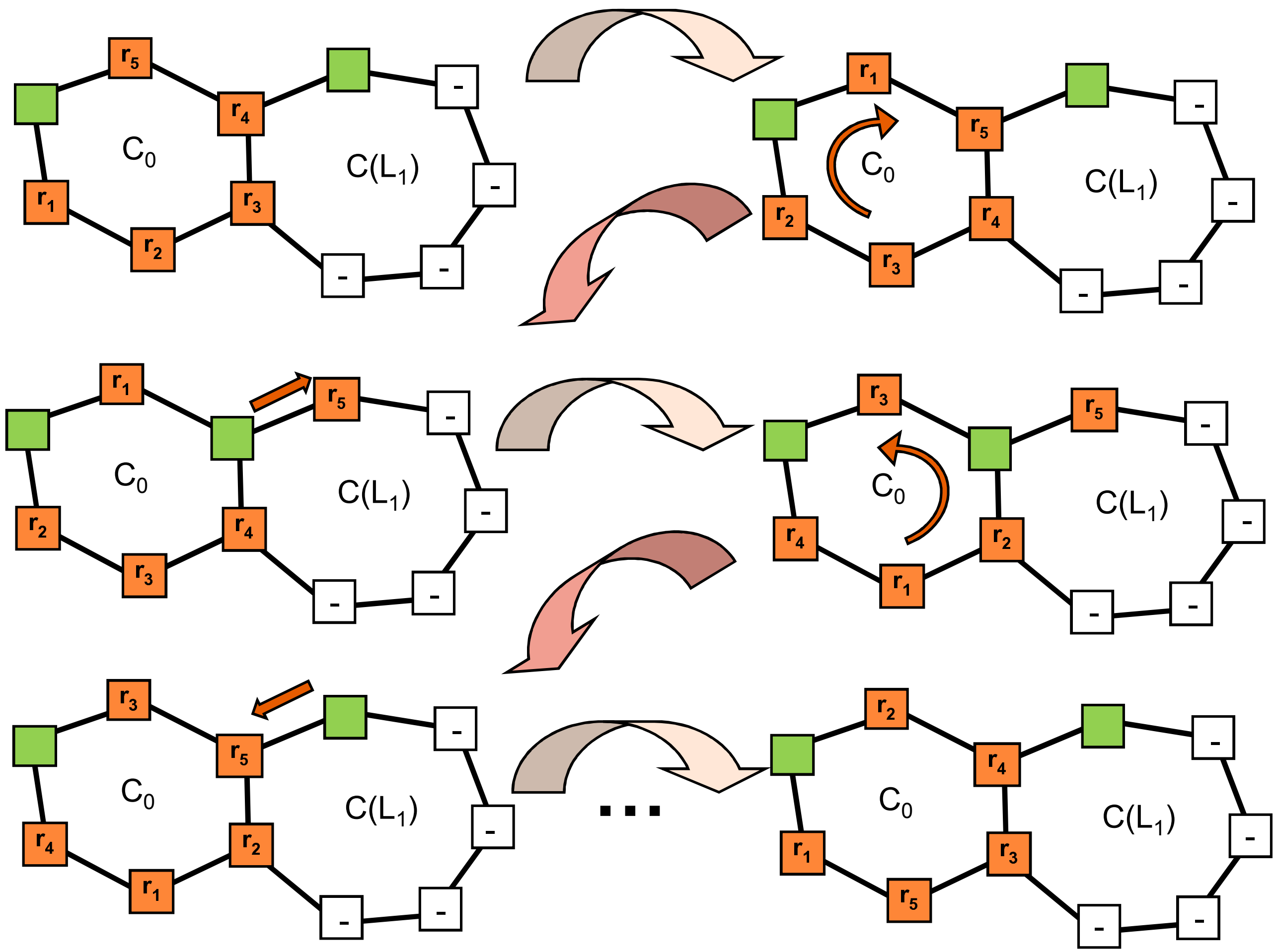


- Robots are placed in stack like manner into the loop
  - The next robot is moved into the gray connection vertex
  - Two cases
    - the robot is somewhere in the loop → must be rotated out of the loop first
    - the robot is outside the loop
  - Then the robot is rotated into the loop (using **green** unoccupied vertex)
- The final rotation places all the robots to their goal positions



BIBOX: Placing Robots within the Original Cycle

- Exchanging robots $\mathbf{r_5}$ and $\mathbf{r_2}$ - using exchanges of robots every permutation of robots within $\mathbf{C_0}$ can be reached



BIBOX: Complexity and Remarks

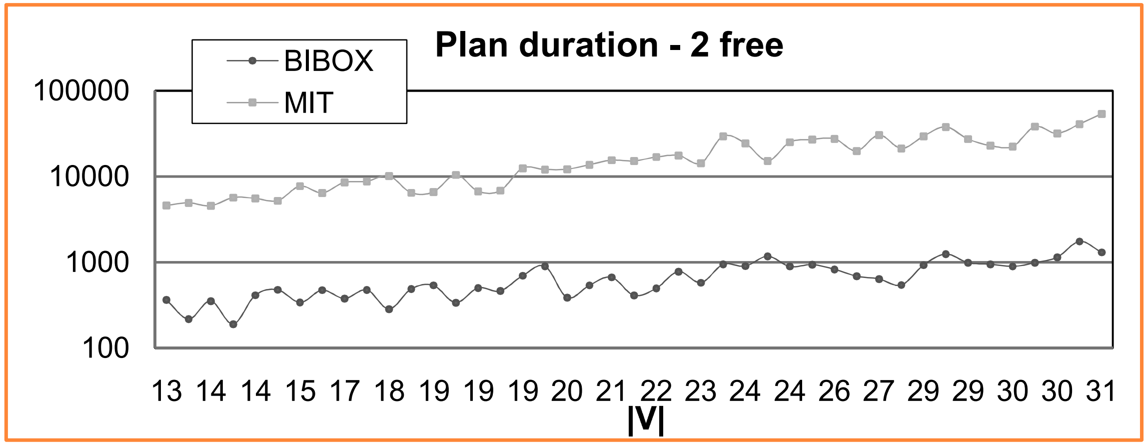
- Worst case **time complexity** of the BIBOX is $\mathbf{O(|V|^3)}$
- The **length** of the generated **solution** is $\mathbf{O(|V|^3)}$
- **Two unoccupied** vertices are required only in the last phase – in the original cycle
- The special requirement that unoccupied vertices are finally in the original cycle can be treated
  - modify the goal arrangement of robots given by $\mathbf{S^+}$ to contain free vertices in the original cycle
  - move free vertices along two disjoint paths into the original cycle
- Having more than one unoccupied vertex parallel moves can be done
  - consecutive moves produced by the BIBOX algorithm are checked for **independence**
  - if **independent** → performed in **parallel**

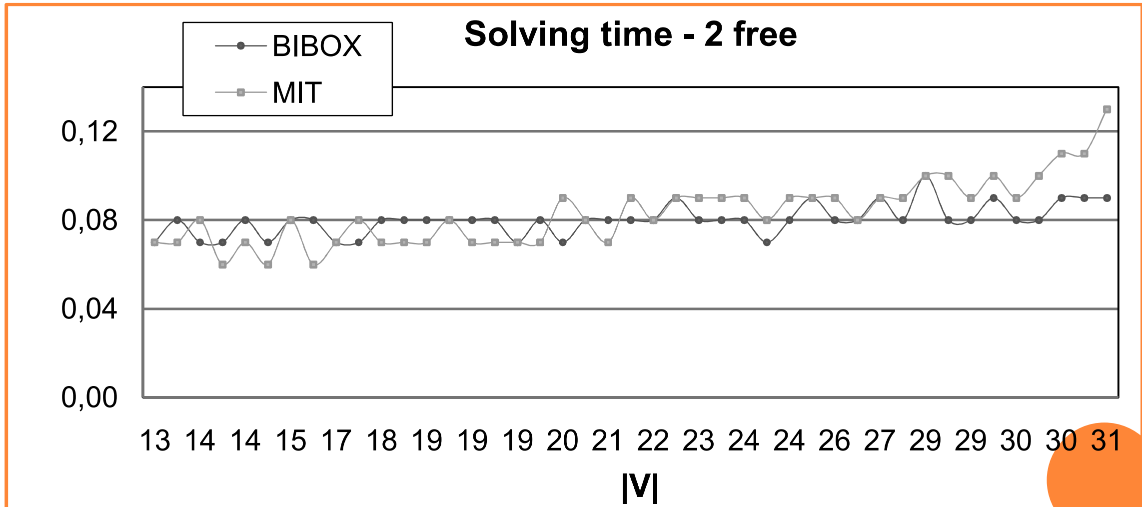
THE **BIBOX** ALGORITHM AND ITS COMPETITORS

- 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 **$O(|V|^3)$** – same as BIBOX however the constant is higher
  - The **length** of the generated **solution** is also **$O(|V|^3)$**
- **Domain independent planners** participating in the IPC (International Planning Competition)
  - **SGPlan 5** and **LPG-td** proved to be best from the winners of the IPC on the multi-robot path planning problem
- The testing problems
  - several randomly generated multi-robot path planning problems
  - random bi-connected graph / random permutation of robots
  - loops of random length of 1…8 distributed uniformly

EXPERIMENTAL COMPARISON OF BIBOX WITH MIT

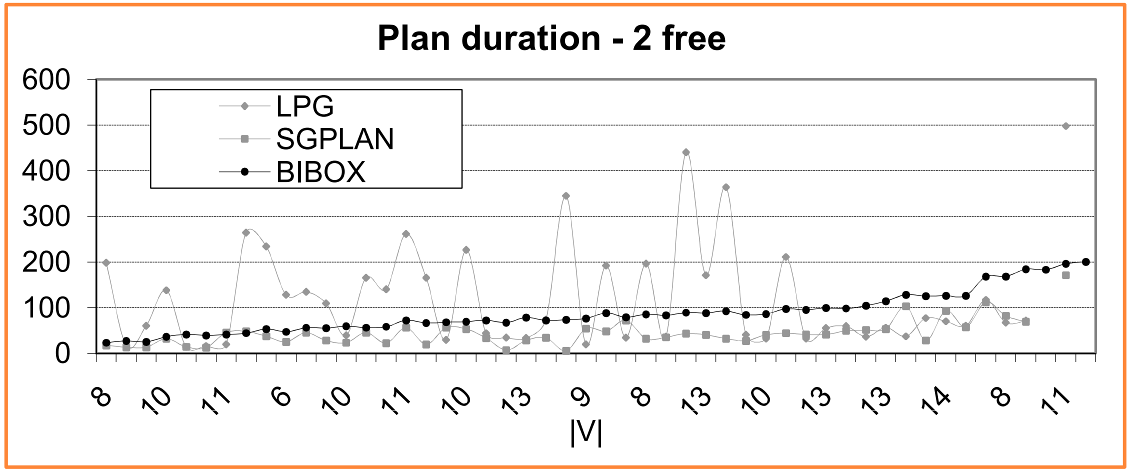
- Both algorithms implemented in C++
- **Length of solutions** and **solving time** are compared
- Random graphs of the size up to **30 vertices**
- The **BIBOX** algorithm produces approximately order of magnitude **shorter solutions** than MIT
- The BIBOX algorithm is **faster on larger problems** (this trend seems to continue)

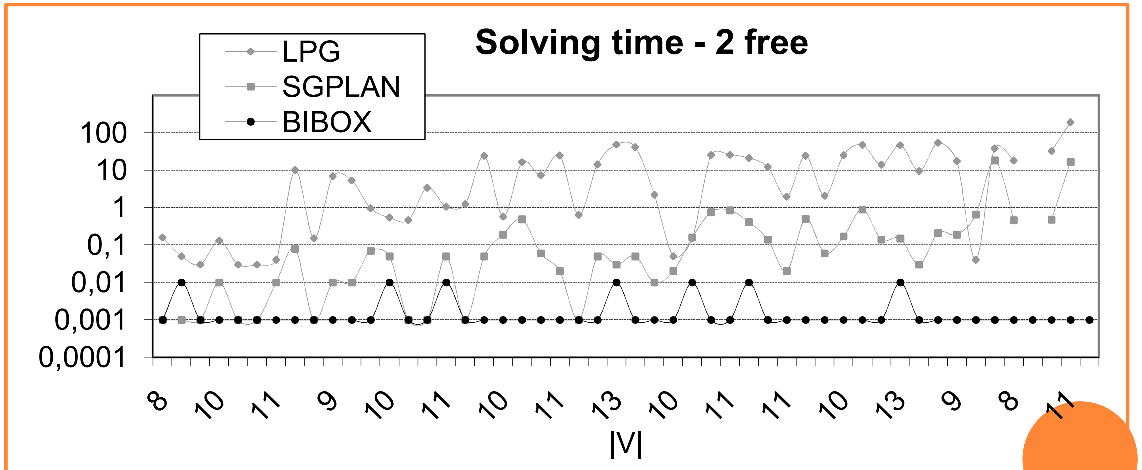




EXPERIMENTAL COMPARISON OF **BIBOX** WITH PLANNERS **SGPLAN** AND **LPG**

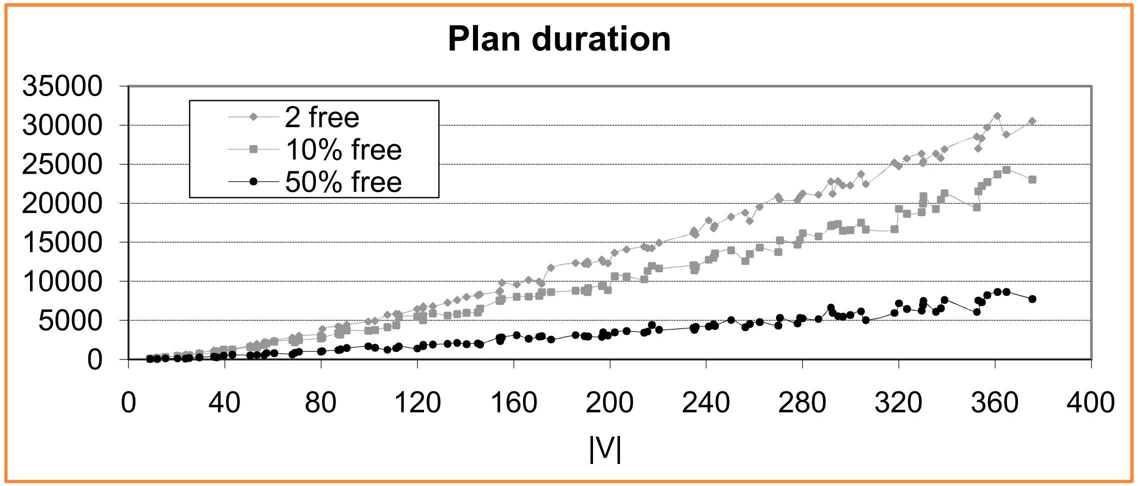
- The code provided by authors of SGPlan and LPG was used
- SGPlan generates shortest possible solution
- Only small random graphs of the size up to **15 vertices** were used
- The planners produce **shorter solutions** – especially SGPlan
- However, it terms of **runtime** the planners are **completely uncompetitive**

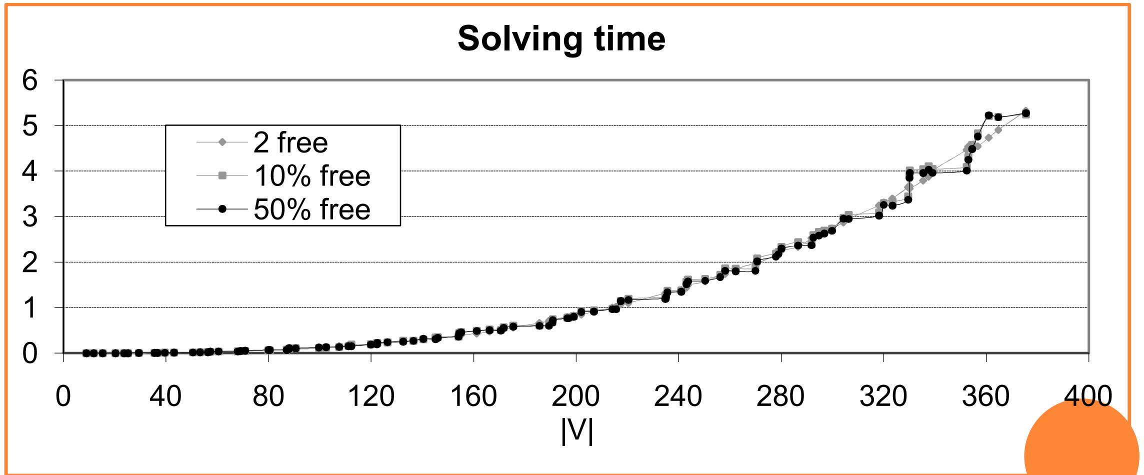




Experiments with BIBOX on Large Problems

- Graphs with up to **400 vertices** were used
- Three setups
  - 2 vertices unoccupied
  - 10% vertices unoccupied
  - 50% vertices unoccupied
- Simple parallelism was tested
- Unoccupied vertices treated as dummy robots – removed in the final solution
- All the problems solved within **6 seconds**





Summary and Conclusions

- A novel algorithm called **BIBOX** for multi-robot path planning in **bi-connected graphs** with at least **two unoccupied** vertices has been proposed
- Experiments proved that **BIBOX** is better several alternative state-of-the-art approaches
  - The **BIBOX** significantly **outperformed** two selected state-of-the-art **planners** (according to IPC) in terms of runtime
  - It produces order of magnitude **better solutions** than another domain dependent state-of-the-art algorithm (**MIT**) in slightly better runtime
- **Future work:**
  - Adapt the last phase to suffice with only one vertex – use a pattern database
  - Increase parallelism – use a method of critical path