Unifying Search-based and Compilation-based Approaches to Multi-agent Path Finding through Satisfiability Modulo Theories

Pavel Surynek
Faculty of Information Technology
Czech Technical University in Prague

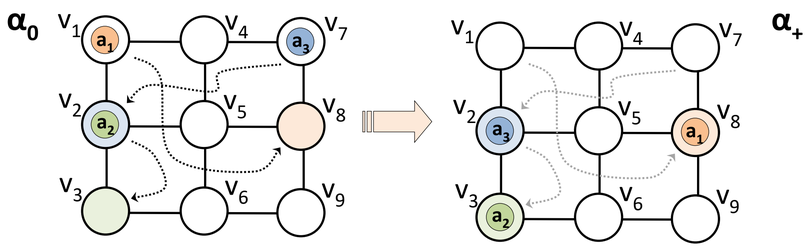
FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Multi-Agent Path Finding

MAPF

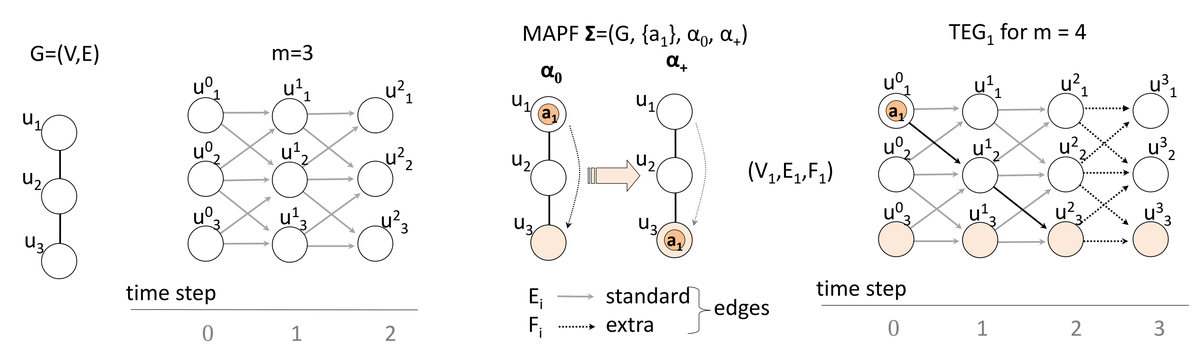
- a task to **relocate agents** to their goals in a non-colliding way
- agents move over undirected graph

CPF $\mathbf{\Sigma}=(G, \{a_1,a_2,a_3\}, \alpha_0, \alpha_+)$



	α_0	α_1	α_2	α_3	$\alpha_4 = \alpha_+$
a_1	v_1	v_1	v_2	v_5	v_8
a_2	v_2	v_3	v_3	v_3	v_3
a_3	v_7	v_4	v_4	v_1	v_2

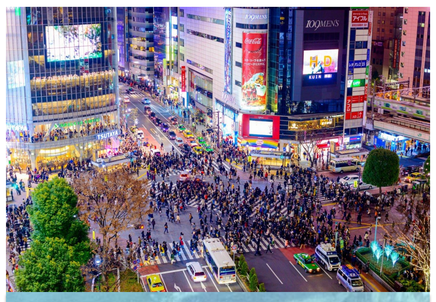
Time Expanded Graph

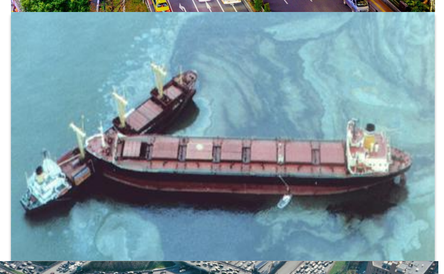


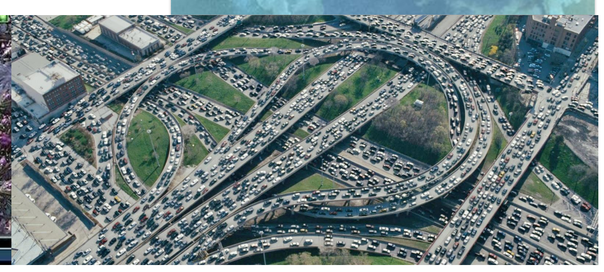
- **positions** of all agents at all time-steps are represented in the *time expanded graph* **– TEG**
- introduce a **propositional** variable for each node in TEGs
  - TRUE iff **agent** is in the vertex at the given time-step
  - introduce constraints for **valid movements** etc.

Motivation

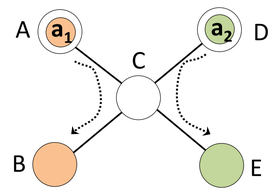
- **rearranging** containers (agent = container)
- heavy traffic **control** (agent = car)
- ship **avoidance** at sea (agent = ship)
- **data transfer** planning (agent = data packet)



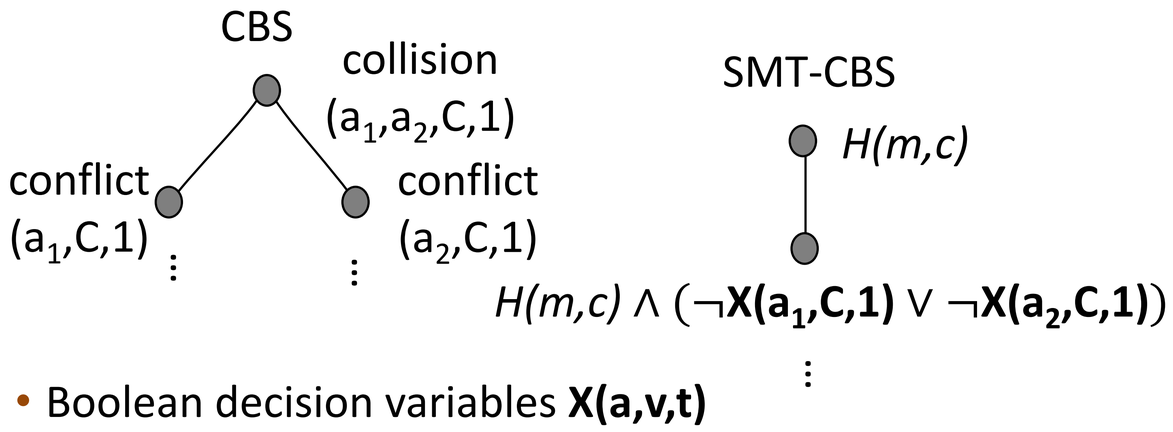




SMT-CBS vs. CBS



- building encoding *F(m,c)* **lazily**
  - ignore collisions in *F(m,c)* → *H(m,c)*
- **no branching** at the high level

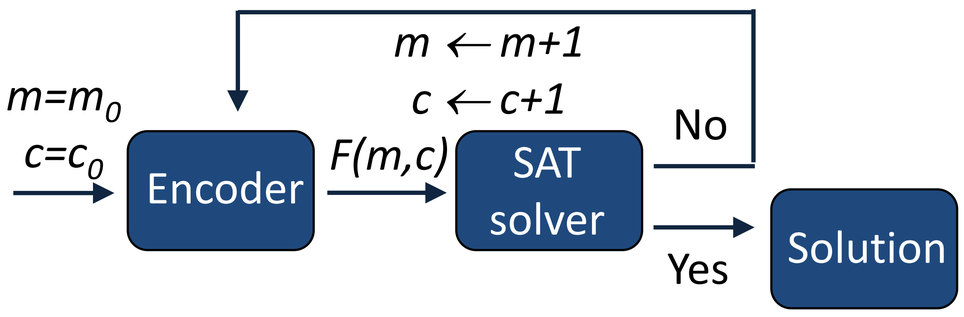


- Boolean decision variables **X(a,v,t)**
  - agent **a** in vertex **v** at discrete time step **t**

Reducing MAPF to SAT

- **expand** the graph modeling the environment in time
  - number of expansions *m* is specified
  - total cost bound *c* is specified
- **encode** relocation of agents through expanded graphs as a propositional formula *F(m,c)*
  - ask **SAT solver** whether *F(m,c)* is solvable
    - m_0 = length of the longest of shortest paths
    - c_0 = sum of lengths of shortest paths

} connecting starts and goals



Experiments