Towards Optimal Cooperative Path Planning in Hard Setups through Satisfiability Solving

Pavel Surynek

Charles University in Prague, Czech Republic
and Kobe University, Japan

Problem of Cooperative Path-planning (CPP)

- **Abstraction** for tasks of motion of multiple (autonomous or passive) entities in a certain environment (real or virtual).
  - Entities are given an **initial** and a **goal** arrangement in the environment.
  - We need to **plan movements of entities in time**, so that entities reach the goal arrangement while **physical limitations are observed**.

- **Physical limitations** are:
  - Entities must **not collide with each other**.
  - Entities must **not collide with obstacles** in the environment.

CPP – Formal definition (1)

Wilson, 1974; Kornhauser et al., 1984; Ryan, 2008

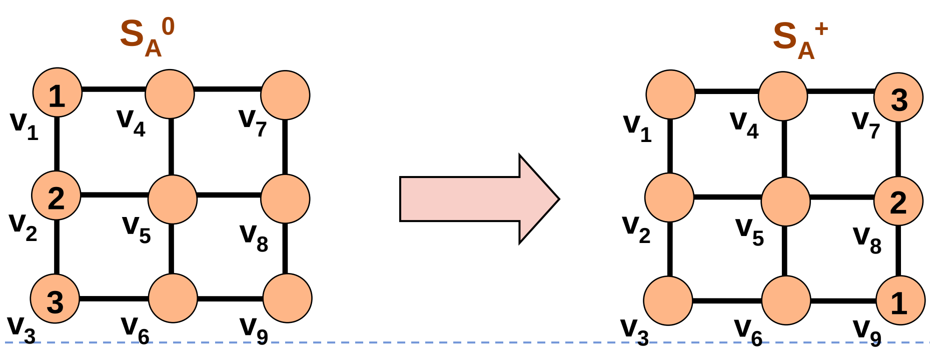
- The environment is modeled as an **undirected graph** where **vertices represent locations** in the environment occupied by agents and **edges** enable agents to go to the **neighboring location**.

- **Formal definition** of the task of CPP
  - It is a quadruple $\Pi = (G, A, S_A^0, S_A^+)$, where:
    - $G=(V,E)$ is an **undirected graph**,
    - $A = \{a_1,a_2,\ldots,a_\mu\}$, where $\mu<|V|$ is a **set of agents**,
    - $S_A^0: A \rightarrow V$ is a uniquely invertible function determining the **initial arrangement of agents** in vertices of G, and
    - $S_A^+: A \rightarrow V$ is a uniquely invertible function determining the **goal arrangement of agents** in vertices of G.

CPP – Formal Definition (2)

Wilson, 1974; Kornhauser et al., 1984; Ryan, 2008

- The **dynamicity** of the task is as follows:
  - An agent occupying a vertex at time step *i* can move into a neighboring vertex (the move is finished at time step *i+1*) if the target vertex is **unoccupied** at time step *i* and **no other agent** is moving simultaneously into the same target vertex
- For the given $\Pi = (G, A, S_A^0, S_A^+)$, we need to find:
  - A sequence of moves for every agent such that dynamicity constraint is satisfied and every agent reaches its goal vertex.



Solution of an instance of cooperative path-planning on a graph with A={1,2,3}

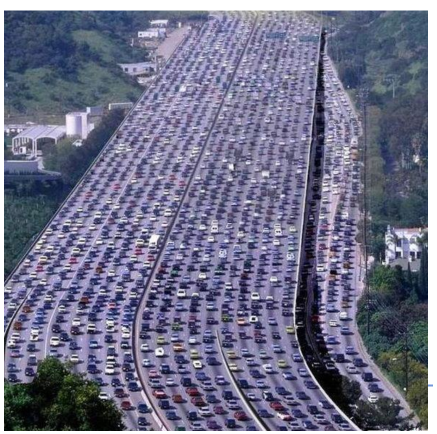
makespan=7

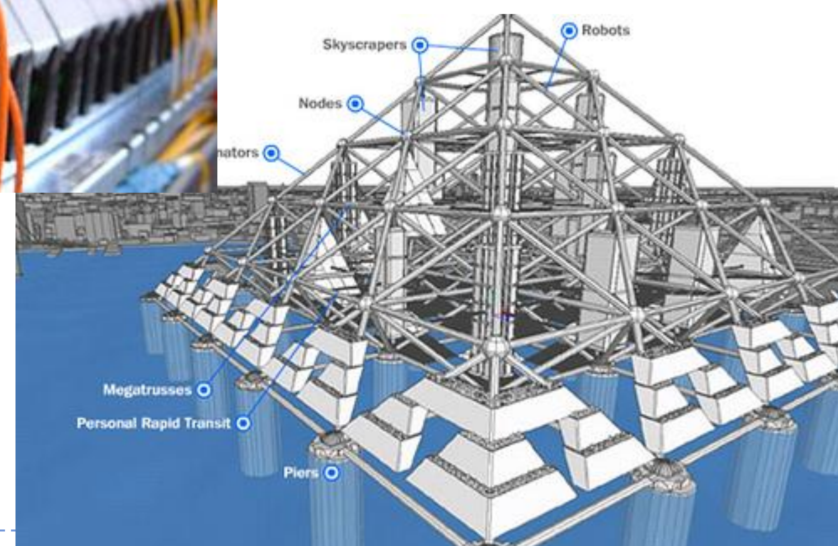
$\mathbf{M_1}$=[$\mathbf{v_1}$, v_4, v_7, v_8, v_9, v_9, $\mathbf{v_9}$]
$\mathbf{M_2}$=[$\mathbf{v_2}$, v_2, v_1, v_4, v_7, v_8, $\mathbf{v_8}$]
$\mathbf{M_3}$=[$\mathbf{v_3}$, v_3, v_3, v_2, v_1, v_4, $\mathbf{v_7}$]

Time step: **1 2 3 4 5 6 7**

Motivation

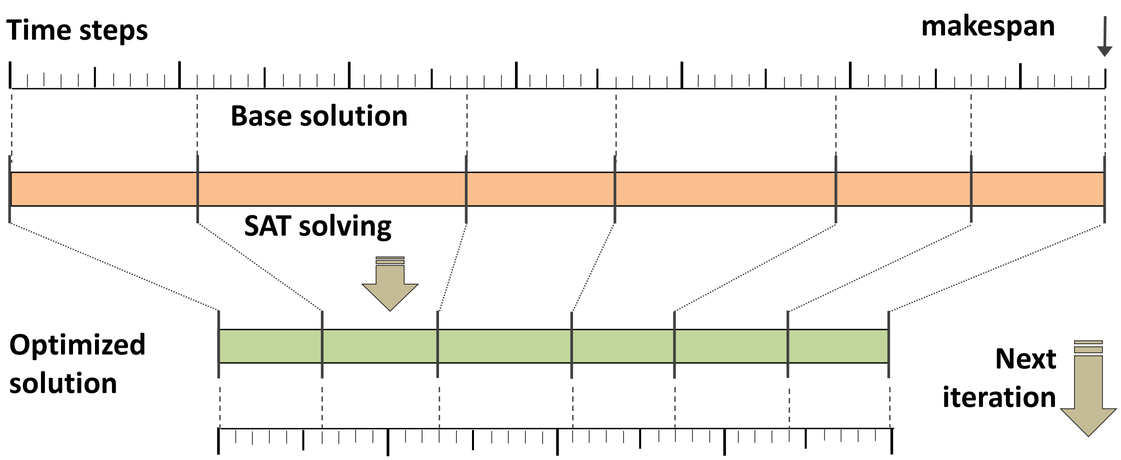
- Container rearrangement
  (entity = **container**)
- Heavy traffic
  (entity = **automobile** (in jam))
- Data transfer
  (entity = **data packet**)
- Generalized lifts
  (entity = **lift**)





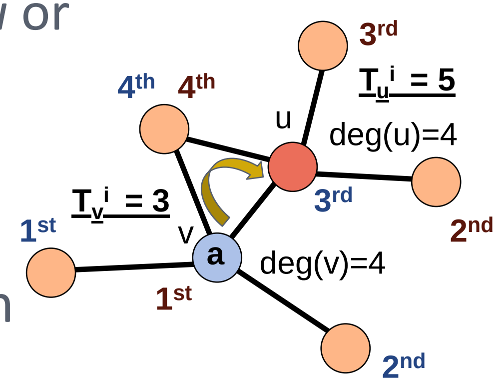
COBOPT – CPP as Propositional Satisfiability

- Construct a **propositional formula** such that
  - **satisfiable** iff there exists a solution to CPP of a given **makespan**
- Suppose we are provided with **makespan suboptimal solution** (base solution – can be generated in polynomial time [ICRA 2009, ICTAI 2009])
  - find a **makespan optimal** replacement of the given sub-sequence of the base solution



Inverse Encoding of CPP

- Makespan **m**, **initial state** and **goal state** are given
  - encode states of the planning world at time steps 1,2,...,m
  - **step 1** equals to the **initial state**
  - **step m** equals to the **goal state**
- State variables: **"what agent** is located in the given vertex"
- **Step *i*** consists of the following **integer variables** for each $v \in V$:
  - $A_v^i \in \{0,1,2,..., \mu\}$
    - $A_v^i = j$ iff the agent a_j is located in v at the time step *i* or
    - $A_v^i = 0$ iff there is no agent in v
  - $T_v^i \in \{0,1,2,..., 2deg(v)\}$
    - $0 < T_v^i \leq deg(v)$ iff the agent **goes out** of v into (T_v^i)-th neighbor
    - $deg(v) \leq T_v^i \leq 2deg(v)$ iff the agent **goes into** v from $((T_v^i)-deg(v))$-th neighbor
    - $T_v^i = 0$ iff no-operation is selected for v
- + **constraints** to enforce valid transitions between states



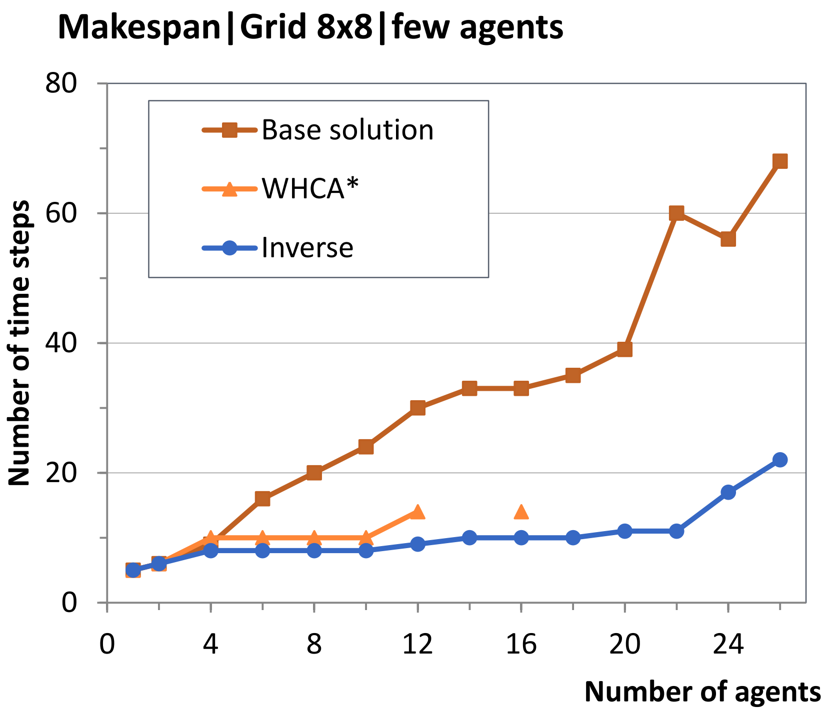
Encoding Size Comparison

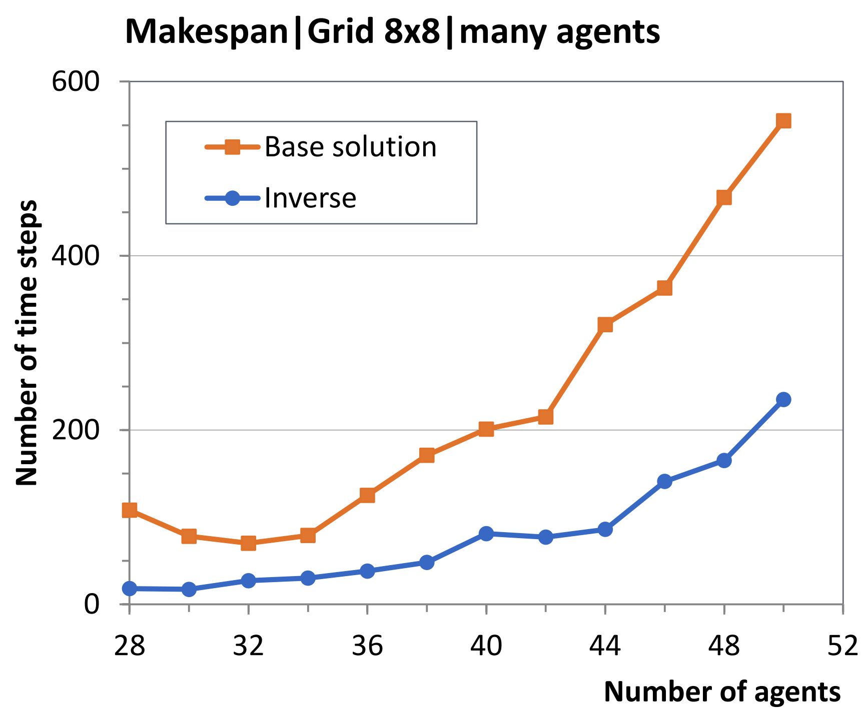
- Experimental setup:
  - 4-connected **grid** of size 8x8
  - **random** initial and goal arrangement of agents

|A| in the 4-connected grid 8×8	Number of layers	SATPLAN encoding		SASE encoding		INVERSE encoding	
		|Variables|	|Clauses|	|Variables|	|Clauses|	|Variables|	|Clauses|
4	8	5.864	55.330	11.386	53.143	**5.400**	**38.800**
8	8	10.022	165.660	19.097	105.724	**5.920**	**48.224**
12	8	14.471	356.410	26.857	168.875	**5.920**	**46.176**
16	10	30.157	1.169.198	51.662	372.140	**8.122**	**76.192**
24	10	43.451	2.473.813	73.101	588.886	**8.122**	**71.072**
32	14	99.398	8.530.312	157.083	1.385.010	**12.396**	**137.120**

Makespan Comparison – grid 8x8

- Compared against WHCA*
  - WHCA* is decoupled
    - often produces near makespan optimal solution





Concluding Remarks

- Improving sub-optimal solutions of **CPP** by modeling the problem as propositional satisfiability.
- **COBOPT**: short subsequences of a sub-optimal solution are replaced by the makespan **optimal** ones.
- Novel **SAT encoding** – inverse encoding
  - **fewer** variables and clauses than domain independent encodings SASE and SATPLAN
- Our related works:
  - [SoCS 2012] – a generalization of COBOPT for **domain-independent** planning, **iCOBOPT** – improved variant of COBOPT
  - [ICTAI 2012] – an alternative encoding based on all-different constraints
  - [ECAI 2012] – a new encoding style of the **all-different** constraint