APPLICATION OF PROPOSITIONAL SATISFIABILITY TO SPECIAL CASES OF COOPERATIVE PATH-PLANNING

Pavel Surynek

Charles University in Prague, Czech Republic
and Kobe University, Japan
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.
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 $\Pi = (G, A, S^0_A, S^+_{A})$, where:

- $G = (V, E)$ is an undirected graph,
- $A = \{a_1, a_2, ..., a_{\mu}\}$, where $\mu < |V|$ is a set of agents,
- $S^0_A : A \rightarrow V$ 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 $G$. 
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 $A\{1,2,3\}$

<table>
<thead>
<tr>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[v_1, v_4, v_7, v_8, v_9, v_9, v_9]$</td>
<td>$[v_2, v_1, v_4, v_7, v_8, v_8]$</td>
<td>$[v_3, v_3, v_2, v_1, v_4, v_7]$</td>
</tr>
</tbody>
</table>

makespan = 7

Time step: 1 2 3 4 5 6 7

CPP – Formal Definition (2)
Wilson, 1974; Kornhauser et al., 1984; Ryan, 2008
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

- Suppose that we are able to construct a propositional formula such that
  - satisfiable iff there exists a solution to CPP of a given makespan

- Next suppose that we are provided with makespan suboptimal solution (base solution – can be generated in polynomial time)
  - we can find makespan optimal replacement of the given sub-sequence of the base solution
All-Different Encoding of CPP

- Encodes “where is the given agent”
- The state at the given time step \( i \) is described by the following integer variables for each \( a \in A \):
  - \( L_a^i \in \{1, 2, \ldots, |V|\} \) with the interpretation that \( L_a^i = j \) iff the agent \( a \) is located in the \( j \)-th vertex of the graph \( G \)
- The requirement that there is at most one agent per vertex is modeled as All-Different(\( L_{a1}^i, L_{a2}^i, \ldots, L_{a\mu}^i \))
- Other constraints are more complicated
  - it is necessary to express that agents can move along edges only
  - and that target vertex of the movement must be empty
- Augmenting by heuristics
  - some vertices are unreachable by the agent in the given time step
## Encoding Size Comparison

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

| \(|A|\) in the 4-connected grid \(8\times8\) | Number of layers | SATPLAN encoding | SASE encoding | INVERSE encoding | ALL-DIFFERENT encoding |
|---|---|---|---|---|---|
| | | \(|\text{Variables}\)| \(|\text{Clauses}\)| \(|\text{Variables}\)| \(|\text{Clauses}\)| \(|\text{Variables}\)| \(|\text{Clauses}\)| \(|\text{Variables}\)| \(|\text{Clauses}\)|
| 4 | 8 | 5864 | 55330 | 11386 | 53143 | 5400 | 38800 | 11128 | 54356 |
| 8 | 8 | 10022 | 165660 | 19097 | 105724 | 5920 | 48224 | 25136 | 114952 |
| 12 | 8 | 14471 | 356410 | 26857 | 168875 | 5920 | 46176 | 42024 | 181788 |
| 16 | 10 | 30157 | 1169198 | 51662 | 372140 | 8122 | 76192 | 79008 | 326736 |
| 24 | 10 | 43451 | 2473813 | 73101 | 588886 | 8122 | 71072 | 140400 | 537528 |
| 32 | 14 | 99398 | 8530312 | 157083 | 1385010 | 12396 | 137120 | 309824 | 1120672 |
Makespan Comparison – grid 8x8

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

![Graphs showing makespan comparison](image-url)
Concluding Remarks

- Improving sub-optimal solutions of cooperative path-planning by modeling the problem as propositional satisfiability.

- COBOPT: short subsequences of a sub-optimal solution are replaced by the makespan optimal ones.

- SAT encoding (and its variants)
  - All-Different encoding

- COBOPT solution optimization represents state-of-the-art in generating short solutions to CPP