Multi-Agent Path Finding (MAPF)

- a task to relocate agents to their goals in a non-colliding way
- agents move over undirected 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 if 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)

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
    - \( m = m_0 \):
      - length of the longest of shortest paths
    - \( c = c_0 \):
      - sum of lengths of shortest paths
- introduces constraints for valid movements

SMT-CBS vs. CBS

- building encoding F(m,c) lazily
  - ignore collisions in F(m,c) \( \rightarrow H(m,c) \)
- no branching at the high level
- Boolean decision variables X(a,v,t)
  - agent a in vertex v at discrete time step t

Experiments

- runtime vs. instance size
- Brc202d | MAPF
- Runtime vs. Instance
- Runtime Dens20d
- Runtime Opt003d
- MDD-SAT + SMT-CBS