Implicitly Coordinated Multi-Agent Path Finding under Destination Uncertainty

The Dancing Robots

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Bernhard Nebel

University of Freiburg, Germany

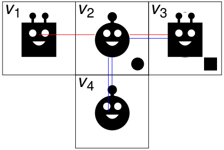
2018

Motivation

- A central problem in many applications is the coordinated movement of agents/robots/vehicles in a given environment.

An abstraction of this problem is the Multi-Agent Path Finding (MAPF) problem:

- Given a set of agents A, a (perhaps directed) graph $G = (V, E)$, an initial state modelled by an injective function $\alpha_0 : A \to V$ and a goal state modelled by another injective funtion α_*, can α_0 be transformed into α_* by movements of single agents without collisions?



- Can we find a (central) plan to move the square robot S to v_3 and the circle robot C to v_2?

Going beyond: MAPF/DU

MAPF with destination uncertainty (MAPF/DU):

- All agents know their own destinations, but these are not common knowledge any longer.
- For each agent, there exists a set of possible destinations, which do not overlap with possible destinations of other agents. These possible destinations are common knowledge.
- All agents plan and re-plan without communicating with their peers.
- A success announcement action becomes necessary, which the agents can use to announce that they have reached their destination (and after that they are not allowed to move anymore).

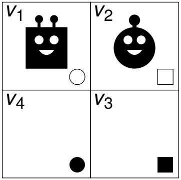
→ Models multi-robot interactions without communication

MAPF/DU: Conceptual problems

- We need a solution concept for the agents: implicitly coordinated branching plans.
- We need to define what the joint execution of such plans means.
- We need to find conditions that guarantee success of such joint executions.
- We have to determine the computational complexity for finding plans and deciding solvability.

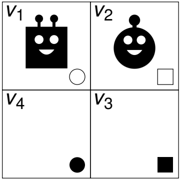
→ Since MAPF/DU is a special case of epistemic planning (initial state uncertainty which is monotonically decreasing), we can use concepts and results from this area.

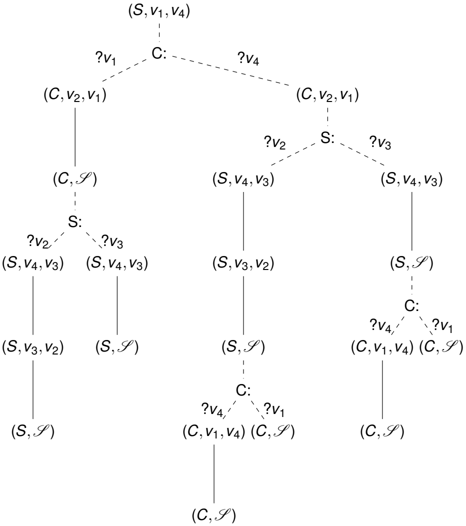
MAPF/DU: Implicitly coordinated branching plans



- Square agent S wants to go to v_3 and knows that circle agent C wants to go to v_1 or v_4.
- C wants to go to v_4 and knows that S wants to go to v_2 or v_3.
- Let us assume S forms a plan in which it moves in order to empower C to reach their common goal.
- S needs shifting its perspective in order to plan for all possible destinations of C (branching on destinations).
- Planning for C, S must forget about its own destination.

Branching plan: Example





Joint execution

- Let us assume, all agents have planed and a subset of them came up with a family of plans $(\pi_i)_{i \in A}$.
- Among the agents that have a plan with their own action as the next action to execute, one is chosen.
- The action of the chosen agent is executed.
- Agents, which have anticipated the action, track that in their plans.
- All other agents have to replan from the new state.
- Since all plans branch over all possible destinations and result in goal states, no acting agent will ever execute an action that leads to a dead end.
- However, agents might wait for each other infinitely long or undo each others action leading to execution cycles.

Results

- Provided all agents plan in a particular way, we can guarantee success with polynomial executions.
- They have to plan
  - optimally, i.e., generate (worst-case) shortest plans;
  - conservatively, i.e., replan from the initial state using the executed actions as a prefix;
  - eagerly, i.e., always plan to act when they can act (respecting optimality and conservativity).
- Deciding whether an implicitly coordinated plan exists is PSPACE-complete (same for eager plans with depth *k*).

Preprint: `http://tr.informatik.uni-freiburg.de/2018/`
Talk on Epistemic Planning: **Friday, 2pm**