Implicitly Coordinated Multi-Agent Path Finding Under Destination Uncertainty

Bernhard Nebel

In a spatial multi-agent environment, e.g., a warehouse, a street intersection, an airport, or a video game, agents have to move to different destinations in a collision-free manner. Such scenarios can be formalized as the *multi-agent path finding* (MAPF) (Ma and Koenig 2017) problem.

This problem and a number of variants have been studied quite extensively, the computational complexity of these problems has been determined (Kornhauser, Miller, and Spirakis 1984; Ratner and Warmuth 1986; Surynek 2010; Yu and LaValle 2013), and a number of optimal and sub-optimal algorithms have been proposed (Felner et al. 2017). The assumption has usually been that movements are computed centrally before execution starts. Furthermore, because of this assumption, destinations are considered to be common knowledge. In this talk we drop both assumptions and analyze whether the agents are still able to coordinate their movements in order to reach their respective destinations.[1] We call the resulting path-finding problem *MAPF under destination uncertainty* or simply *MAPF/DU*. Such a scenario is, for instance, plausible in human-robot interactions or when agents do not share a common communication channel.

In order to illustrate this point, let's have a look at the situation depicted in Figure 1. The circle agent C wants to

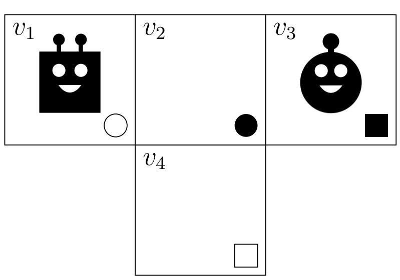

Figure 1: Multi-agent coordination example

go to v_2 and the square agent S wants to go to v_3 (the solid circle and square). However, both are uncertain about the destination of the other agent. S knows only that C has v_1 or v_2 as its destination and C knows that S wants to go either to v_3 or to v_4.

As a first step, we have to come up with a solution concept. We introduce *i-strong branching plans* that correspond to *implicitly coordinated policies* as they have been proposed in the area of epistemic planning (Engesser et al. 2017). The idea of these plans is that an agent moves to a location in order to empower the other agents to reach the common goal of moving all agents to their destination. For this purpose, the agent has to plan actions of the other agents, i.e., making a *perspective shift*. In doing so, the agent has to plan for all possible destinations of the agent, i.e., branch on possible destinations.

One interesting property of these plans is that one can reduce them to skeletons that are composed out of *stepping stones*, configurations in which one agent can reach its destination with certainty and success for the rest is guaranteed. Using this result, one can show that the worst-case execution cost of a branching plan for a MAPF/DU instance is polynomially bounded.

As a next step, we have to define what a joint execution of these branching plans should be. Basically, one agent who wants to act is chosen, and its action is executed. Afterwards, the other agents track the execution of this action in their plans, if it was an action they anticipated. If the action was not anticipated, the agents replan from the new situation. Since every action that is executed is part of a successful plan, it can never happen that this action leads to a dead-end.

It can, however, happen that agents block each other forever, if they have come up with the plan such that the other agents starts acting. We call such agents *lazy*. Conversely, if they are never lazy, we call them *eager* (Bolander et al. 2018). While eager agents never block each other, they may end up in infinite executions. There are two ways to avoid that. One is to require that the agents plan optimally, i.e., that generate only plans with a minimal depth. If the knowledge of the agents is *uniform*, then one can guarantee success. Unfortunately, this is not the case for MAPF/DU. Moreover, we found an example where we could indeed demonstrate that two agents could end up in an inifinite execution.

The other possible way to avoid cycles is to require that plans do not contain cycles and that in replanning the agents always start at the initial state, using the already executed

[1]This is joint work with Thomas Bolander, Thorsten Engesser and Robert Mattmüller. See `http://tr.informatik.uni-freiburg.de/2018/`

sequence of actions as a prefix of the plans they are generating. We call agents that plan in this fashion *conservative*. While this avoids cyclic executions, it might lead to execution sequences that are exponentially longer than the shortest possible plan.

Combining both properties, however, leads to agents for which we can give a tight success guarantee. Optimal, eager agents which replan conservatively are guranteed to succeed using at most polynomially many steps, provided there exists a plan at all.

Finally, we have a look at the computational complexity of deciding the existence of (bounded) i-strong plans. We show that these problems are PSPACE-complete. This demonstrates that communication about destinations pays off significantly. For the case that we deal only with few agents, we show that deciding existence and bounded existence for a fixed number of agents is polynomial.

References

[Bolander et al. 2018] Bolander, T.; Engesser, T.; Mattmüller, R.; and Nebel, B. 2018. Better eager than lazy? How agent types impact the successfulness of implicit coordination. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, (KR-2018)*. To appear.

[Engesser et al. 2017] Engesser, T.; Bolander, T.; Mattmüller, R.; and Nebel, B. 2017. Cooperative epistemic multi-agent planning for implicit coordination. In *Proceedings of the Ninth Workshop on Methods for Modalities (M4MICLA-17)*, 75–90.

[Felner et al. 2017] Felner, A.; Stern, R.; Shimony, S. E.; Boyarski, E.; Goldenberg, M.; Sharon, G.; Sturtevant, N. R.; Wagner, G.; and Surynek, P. 2017. Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges. In *Proceedings of the Tenth International Symposium on Combinatorial Search (SOCS-17)*, 29–37.

[Kornhauser, Miller, and Spirakis 1984] Kornhauser, D.; Miller, G. L.; and Spirakis, P. G. 1984. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. In *25th Annual Symposium on Foundations of Computer Science (FOCS-84)*, 241–250.

[Ma and Koenig 2017] Ma, H., and Koenig, S. 2017. AI buzzwords explained: multi-agent path finding (MAPF). *AI Matters* 3(3):15–19.

[Ratner and Warmuth 1986] Ratner, D., and Warmuth, M. K. 1986. Finding a shortest solution for the N × N extension of the 15-puzzle is intractable. In *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI-86).*, 168–172.

[Surynek 2010] Surynek, P. 2010. An optimization variant of multi-robot path planning is intractable. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI-10)*.

[Yu and LaValle 2013] Yu, J., and LaValle, S. M. 2013. Structure and intractability of optimal multi-robot path planning on graphs. In *Proceedings of the Twenty-Seventh Conference on Artificial Intelligence (AAAI-13)*.