Learning Conflicts From Experience

Yann-Michaël De Hauwere and **Ann Nowé**
Computational Modeling Lab - Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brusssels, BELGIUM
{ydehauwe,anowe}@vub.ac.be

Abstract

Multi-agent path finding has been proven to be a PSPACE-hard problem. Generating such a centralised multi-agent plan can be avoided, by allowing agents to plan their paths separately. However, this results in an increased number of collisions and agents must replan frequently. In this paper we present a framework for multi-agent path planning, which allows agents to plan independently and solve conflicts locally when they occur. The framework is a generalisation of the CQ-learning algorithm which learns sparse interactions between agents in a multi-agent reinforcement learning setting. [1]

Introduction

The goal of multi-agent path planning is to navigate agents from a starting position to their respective goals, while avoiding each other and any static obstacles that may be present in the environment. Compared to single agent path planning, one major additional problem arises: the number of states and actions grows exponential in the number of agents. The multi-agent path finding problem is PSPACE hard (Hopcroft, Schwartz, and Sharir 1984). This makes generating one plan for all agents at once intractable for all but the smallest number of agents. This problem occurs in various application domains such as robotics (Bennewitz, Burgard, and Thrun 2002), air traffic control (Pallottino et al. 2007), disaster rescue (Kitano et al. 1999) and computer games (Silver 2005).

In multi-agent reinforcement learning (MARL) a novel paradigm has been developed, called sparse interactions (Melo and Veloso 2010). The idea of this framework is to learn the situations in which agents influence each other and how to handle these conflict situations as they occur. In all other situations, agents can act independently, ignoring the other agents. The principle is shown graphically in Figure 1.

This paper describes how the principles of a concrete MARL algorithm that learns these sparse interactions, named CQ-learning (De Hauwere, Vrancx, and Nowé 2010), can be used in multi-agent path finding.

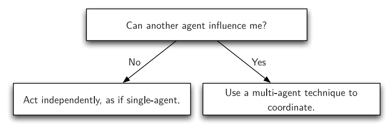

Figure 1: Decomposition of path planning into independent planning for non-conflict situations and multi-agent planning for situations where agent can collide.

Learning conflict situations

The main idea of the framework presented here is that as long as agents are not being influenced, i.e. they do not collide with each other, they can act independently. If a collision occurs, the agents are to solve the problem locally. After the conflict is resolved, the agents can continue acting independently and follow their individual plan again.

The problem of single agent planning has already been extensively explored and many efficient approaches already exist that can solve this problem (LaValle 2006). In this paper we assume that agents can plan their paths from their initial locations to their respective goals in the scenario where no other agents are present in the environment, i.e. they act individually in the environment. This initial plan can be generated offline, or online through learning algoritms. The only requirement is that agents are aware of the post-conditions of every action they take in this single agent setting. When multiple agents are acting together and planning a path to the goal, they follow their single agent plan, and verify that the post-conditions are still valid. This means that there was no influence from other agents. A statistical test[2] on these conditions is used to identify potential influences and informs the agent when to switch to a multi-agent plan. These post-conditions could be the distance traveled during the last timestep, the time needed to travel, a signal informing the agent whether its last navigation action was succesfull. Or in reinforcement learning or dynamic programmic context this is the value function of a state. If these post-conditions are statistical significantly different compared to the single-

[1] Originally published as "Learning multi-agent state space representations". In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 715722.

[2] In the experiments we used a Student's t-test.

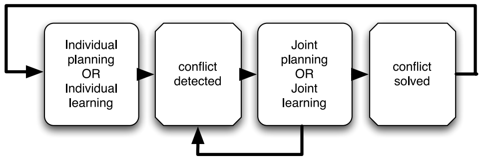

Figure 2: Evolution of conflict detection and action selection in CQ-learning/planning

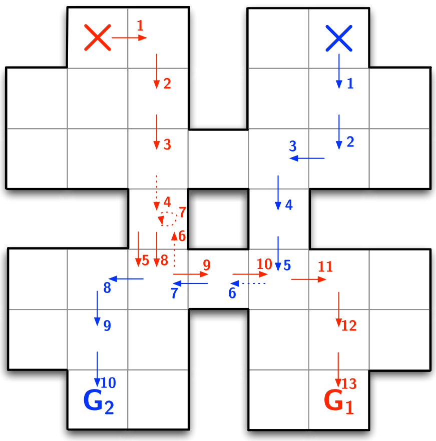

Figure 3: Trace of a solution found by CQ-learning in a two-agent environment

agent plan, the agent changes it state space representation to include the information of the conflicting agent. This means that in these states the agent will learn or plan using a multi-agent approach.

This local problem solving approach favors the agent to find a quick solution to the conflict by augmenting the state information for planning. In the next state, if the conflict is resolved it can follow its individual plan again as shown in Figure 2.

Experiments

We tested CQ-learning in several maze environments in (De Hauwere, Vrancx, and Nowé 2010) and we illustrate a sample solution found in one of these environments containing two agents, see Figure 3. Their initial positions are given by the **X**, their respective goals, marked by the letter **G** in the corresponding color. Actions taken, following the individual plan are marked by arrows with full tails, whereas actions taken using augmented state information are indicated with dotted tails. Agent 1 (in red) uses a multi-agent approach during time steps 4 to 7, after which it follows its individual plan again. Agent 2 uses a multi-agent approach during time step 6. It acts independent during the rest of the episode.

Conclusion

In this abstract we described how CQ-learning could be used in a multi-agent path planning context. CQ-learning is a MARL algorithm capable of exploiting independent experience in a multi-agent environment to learn a multi-agent policy using sparse interactions. This is done by means of statistical tests to determine if there are changes in the outcome of a navigation action by an agent If this is the case, the state space in which agents are planning or learning is augmented to include the location of the other agent, such that it can use a multi-agent planning technique. This techniques allows for two extensions. The first is that these conflict states can be generalized and transferred to other agents/environments (De Hauwere, Vrancx, and Nowé 2010). The second is that by performing the statistical test on long term post-conditions, conflicts can be identified several timesteps ahead of the actual problem situation (De Hauwere, Vrancx, and Nowé 2011).

References

Bennewitz, M.; Burgard, W.; and Thrun, S. 2002. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*.

De Hauwere, Y.-M.; Vrancx, P.; and Nowé, A. 2010. Learning multi-agent state space representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 715–722.

De Hauwere, Y.-M.; Vrancx, P.; and Nowé, A. 2011. Solving sparse delayed coordination problems in multi-agent reinforcement learning. In *Adaptive Agents and Multi-agent Systems V*, Lecture Notes in Artificial Intelligence. Springer-Verlag.

Hopcroft, J. E.; Schwartz, J. T.; and Sharir, M. 1984. On the complexity of motion planning for multiple independent objects: PSPACE-hardness of the “warehouseman’s problem”. *International Journal of Robotics Research* 3(4):76–88.

Kitano, H.; Tadokoro, S.; Noda, I.; Matsubara, H.; Takahashi, T.; Shinjou, A.; and Shimada, S. 1999. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE International Conference on Systems, Man and Cybernetics*, 739–746. IEEE Computer Society.

LaValle, S. M. 2006. *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press. Available at http://planning.cs.uiuc.edu/.

Melo, F., and Veloso, M. 2010. Local multiagent coordination in decentralised mdps with sparse interactions. Technical Report CMU-CS-10-133, School of Computer Science, Carnegie Mellon University.

Pallottino, L.; Scordio, V. G.; Frazzoli, E.; and Bicchi, A. 2007. Decentralized cooperative policy for conflict resolution in multi-vehicle systems. *IEEE Trans. on Robotics* 23(6):1170–1183.

Silver, D. 2005. Cooperative pathfinding. In Young, R. M., and Laird, J. E., eds., *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference, June 1-5, 2005, Marina del Rey, California, USA*, 117–122. AAAI Press.