Decentralized Partially Observable Markov Decision Processes for Planning in Smart Self-Adaptive Cyber-Physical Systems

Guided Research

Supervisor: Prof. Dr. Alexander Pretschner
Advisor: Ana Petrovska
Email: {alexander.pretschner, ana.petrovska}@tum.de
Phone: +49 (89) 289 - 17830
Starting date: October 2019

Fakultät für Informatik
Lehrstuhl 4
Software & Systems Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3
85748 Garching bei München

Tel: +49 (89) 289 - 17830
https://www4.in.tum.de

Context

Modern systems as Cyber-Physical Systems operate in dynamic, multiagent, uncertain, and partially observable environments or contexts. Hence, the state and the behavior of context where the systems are operating at run-time cannot be known a priori, during the design phase of the system. As a result, the systems themselves need to be able to observe run-time changes in their operational context and to automatically reconfigure themselves, or self-adapt. Even though there is no widely accepted definition of self-adaptive systems yet; however, it is known that the properties mentioned above are the few necessary properties that these systems need to fulfill. Therefore, for a system to be called self-adaptive, it should be able to adapt at run-time, as a response to the changes which occur in its environment, while: 1) the system continues to meet its functional specifications and performs its designated task, and 2) preserves, or even improves, its quality objectives.

These adaptations often are not foreseeable during design time, so the systems should be able to plan and re-plan during run-time while accounting for the partiality of the observation and the uncertainty of the information. A formal, mathematical framework for planning under uncertainty and partial information is Partially Observable Markov Decision Processes (POMDP).

The project in Detail

For a long time, because of their rather bad scaling, POMDPs were not well suited for solving real-time planning problems. Recent Monte-Carlo based solvers provide significant enhancement in terms of speed, allowing to plan and re-plan in real-time even for moderately sized environments.

As a use-case, a multi-robot collaborative dirt exploration and allocation task shall be solved. In a 2D map, two or more robots collaborate in order to detect, locate, and navigate to randomly spawning tasks. This is done by using sensors, which inhibit uncertainty in their sensing, and are limited by their sensor radius. POMDPs, as a complete module realizes both the management (allocation) and the path planning tasks together.

In a preceding lab course, we used centralized POMDPs solved via Partially Observable Monte Carlo Planning (POMCP) combined with a naive machine learning aspect to tackle the multi-robot cleaning use-case. Although it was successful as a proof of concept, there still is room for improvement, both in terms of the specific choice in modeling the POMDP and the implementation. Choosing specific macro-actions instead of atomic actions is shown to be suitable for warehouse planning with Dec-POMDPs [1]. By using macro actions, together with Dec-POMDPs and state-of-the-art solvers, we believe that the overall performance for our use-case will be significantly increased.

As part of this Guided Research, a decentralized version of POMDPs will be modeled and implemented. Also, a well-performing solver for the resulting model has to be chosen and implemented. Two of the candidates for solvers are POMCP [2] and occupancy MDP [3].

Goal

Design and implementation of a Dec-POMDP based solution for a multi-robot use-case with added naive machine learning aspect. We will evaluate the implemented approach considering different parameters, and different sets of macro-actions.

Working Plan

1. Study literature of POMDPs, Dec-POMDPs, and related solvers
2. Create a model for the components of the POMDP
    - State space
    - Action space

    - Different reward functions
    - Environment black-box generator
3. Implement Dec-POMDP structures
4. Implement POMDP solvers
5. Evaluate approach
    - Tuning of hyper-parameters
    - Comparison of different sets of macro-actions
    - Bench-marking with different environment configurations
6. Technical Report
7. Final Report

Fakultät für Informatik
Lehrstuhl 4
Software & Systems Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3
85748 Garching bei München

Tel: +49 (89) 289 - 17830
https://www4.in.tum.de

Deliverables

- Source code of the implementation.
- Technical report with comprehensive documentation of the implementation, i.e. design decision, architecture description, API description and usage instructions.
- Final report written in conformance with TUM guidelines.

Pre-requisite

- Good Python and C/C++ skills
- Previous knowledge and experience with ROS
- Knowledge and good understanding of Markov Decision Processes
- Ideally be familiar with the technical implementation of the multi-robot use-case

References

[1] C. Amato, G. Konidaris, G. Cruz, C. A. Maynor, J. P. How, and L. P. Kaelbling, "Planning for decentralized control of multiple robots under uncertainty," in *In IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[2] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in Neural Information Processing Systems 23* (J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, eds.), pp. 2164–2172, Curran Associates, Inc., 2010.

[3] J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet, "Optimally solving dec-pomdps as continuous-state mdps," *J. Artif. Int. Res.*, vol. 55, pp. 443–497, Jan. 2016.