Multi-Agent Programming Contest 2013

—

The GOAL-DTU Team

Jørgen Villadsen, Andreas Schmidt Jensen, Nicolai Christian Christensen,
Andreas Viktor Hess, Jannick Boese Johnsen, Øyvind Grønland Woller, and
Philip Bratt Ørum

Algorithms, Logic and Graphs Section
Department of Applied Mathematics and Computer Science
Technical University of Denmark
Matematiktorvet, Building 303B, DK-2800 Kongens Lyngby, Denmark

Abstract. We provide a brief description of the GOAL-DTU system, including the overall design, the tools and the algorithms that we plan to use in the agent contest.

Introduction

1. The name of our team is GOAL-DTU. We participated in the contest in 2009 and 2010 as the Jason-DTU team [1,2] and in 2011 and 2012 as the Python-DTU team [3,4].
2. The members of the team are as follows:

   - Jørgen Villadsen, PhD
   - Andreas Schmidt Jensen, PhD student
   - Nicolai Christian Christensen, MSc student
   - Andreas Viktor Hess, BSc student
   - Jannick Boese Johnsen, MSc student
   - Øyvind Grønland Woller, BSc student
   - Philip Bratt Ørum, MSc student

   We are affiliated with DTU Compute (short for Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU) and located in the greater Copenhagen area).
3. The main contact is associate professor Jørgen Villadsen, DTU Compute, email: `jovi@dtu.dk`
4. We expect that we will have invested approximately 500 man hours when the tournament starts.

System Analysis and Design

1. The team's main strategy is for the agents to determine the location of high value nodes and position themselves on them, while trying to disrupt the opponent from doing the same.
2. We will not use any existing multi-agent system methodology.
3. We do not plan to distribute the agents on several machines.
4. We plan to use a decentralized approach for decision making, where each agent attempts to determine what action is best for it and the team. We plan to use a centralized solution for passing messages regarding percepts of nodes, where one agent receives messages from everyone else and distributes beliefs about nodes.
5. Our agents use limited communication, only sharing map structure and team and enemy positions.
6. Our agents are largely independent, but they do coordinate their positions when trying to control an area. We also ensure that agents on the same nodes do not attempt the same action if doing so is redundant, e.g. attacking or repairing the same target.
7. Each agent acts on its own behalf based on its local view of the world which is updated through percepts, and is thus autonomous and reactive. This is implemented as an agent-control-loop in which the agents decide which actions to execute based on their current view of the world. Disabled agents will declare that they are disabled, allowing repairers to move towards them as quickly as possible. Our repairers sometimes try to anticipate when our saboteurs will be attacked and repair them in advance, allowing us to gain the upper hand.

Software Architecture

1. We use the GOAL agent programming language.
2. We use the GOAL IDE as a development platform as well as GEdit and Eclipse as code editors/IDEs.
3. As the runtime platform for the competition we use a suitable Linux system with the newest version of GOAL.
4. We use our own implementation of Dijkstra's algorithm for pathfinding. Since there is no usable heuristic, we consider this to be the best option.

References

1. Niklas Skamriis Boss, Andreas Schmidt Jensen, and Jørgen Villadsen. *Building Multi-Agent Systems Using Jason.* Annals of Mathematics and Artificial Intelligence, 59:373-388, Springer 2010.
2. Steen Vester, Niklas Skamriis Boss, Andreas Schmidt Jensen, and Jørgen Villadsen. *Improving Multi-Agent Systems Using Jason.* Annals of Mathematics and Artificial Intelligence, 61:297-307, Springer 2011.
3. Mikko Berggren Ettienne, Steen Vester, and Jørgen Villadsen. *Implementing a Multi-Agent System in Python with an Auction-Based Agreement Approach.* Lecture Notes in Computer Science, 7217:185-196, Springer 2012.
4. Jørgen Villadsen, Andreas Schmidt Jensen, Mikko Berggren Ettienne, Steen Vester, Kenneth Balsiger Andersen, and Andreas Frøsig. *Reimplementing a Multi-Agent System in Python.* Lecture Notes in Computer Science, 7837:205-216, Springer 2013.

Acknowledgements

Thanks to Per Friis for IT support.

Thanks to Koen Hindriks for prompt GOAL support and also for the HactarV2 system which we have used as a starting point.

More information about the GOAL-DTU team is available here:

`http://www2.compute.dtu.dk/~jovi/MAS/`