Multi-Agent Programming Contest 2017
Participation Registration

E.I.Sarmas

Introduction

1. What is the name of your team?

Flisvos 2017 ("Hot Stuff" edition)

2. Who are the members of your team? Please provide names, academic degrees and institutions.

E.I.Sarmas, M.Sc. Computer Science Rutgers University, M.B.A. NTUA, B.Sc. Physics UoA, not affiliated with any institution

3. Who is the main-contact? Please also provide an Email address.

E.I.Sarmas, eis@sarmas.com

4. How much time (man hours) will you have invested (approximately) until the tournament?

A rough estimation is 60-90 man-hours.

System Analysis and Design

1. Briefly, what is the main strategy of the team?

To produce a good and dynamic global plan that completes selected jobs as fast and efficiently as possible.

2. Will you use any existing multi-agent system methodology such as Prometheus, O-MaSE, or Tropos?

No existing multi-agent system methodology is used.

3. Do you plan to distribute your agents on several machines?

No, the agents will be threads on same machine.

4. Is your solution based on the centralisation of coordination/information on a specific agent? Conversely if you plan a decentralised solution, which strategy do you plan to use?

Solution is decentralized but exactly same algorithms providing global action planning are used by all agents. For efficiency in this implementation, at any time a random agent does the calculations and planning (instead of it getting repeated by 16 agents) and updates a single shared data structure that tracks the actions and progress of the team.

5. Describe the communication strategy in the agent team. Can you estimate the communication complexity of your approach?

Unfortunately due to limited available time it is not possible to implement a real communication strategy, but instead shared data structures are updated. This strategy has been used frequently in past MAPC.

6. Describe the team coordination strategy (if any).

Due to the nature of this MAPC scenario, a common planning algorithm was adopted that is executed by all agents, so there is no coordination strategy.

7. How are the following agent features implemented: autonomy, proactiveness, reactiveness?

A goal/utility-based architecture is used.

Software Architecture

1. Which programming language do you plan to use to implement the multiagent system? (e.g. 2APL, Jason, Jadex, JIAC, Goal, Java, C++, . . .)

Python

2. Which development platform and tools are you planning to use?

Notepad++ and CLI

3. Which runtime platform and tools are you planning to use? (e.g. Jade, AgentScape, simply Java, . . .)

Python

4. Which algorithms will be used?

Many ad-hoc algorithms for composite and simple tasks, mostly classified in two categories, extensive calculation ones for estimating best action parameters or results of actions and derivatives of known optimization heuristics to select best actions.