Multi-Agent Programming Contest 2012
Participation Registration

Abstract. This document describes the architecture and strategies adopted by the LTI-USP team to participate in the 2012 edition of the Multi-Agent Programming Contest (MAPC 2012). The team was developed using the JaCaMo multi-agent framework and the main strategy was to divide the agents into two groups: one to occupy the best zone in the map, and other to sabotage the opponents.

Introduction

The LTI-USP, located at the University of São Paulo, is the oldest and one of the most relevant research groups in multi-agent systems in Brazil. The group participated in the 2010 edition of the Multi-Agent Programming Contest and the cows-and-cowboys scenario was used during the last two years in the Multi-Agent course held by Jaime Simão Sichman and Anarosa Alves Franco Brandão at the Department of Computer Engineering and Digital Systems of the University of São Paulo.

For this year's tournament, the LTI-USP team is working since the beginning of May, having invested (approximately) 200 hours. The LTI-USP team if formed by:

- Team members:
  - Mariana Ramos Franco [1] - MSc. Student at University of São Paulo
  - Luciano Menasce Rosset - Undergraduate Student at University of São Paulo
- Supervisor:
  - Jaime Simão Sichman - Associated Professor at University of São Paulo

System Analysis and Design

The team's main strategy is to divide the agents into two group: one in charge of occupying the best zone in the map, and the other one in charge of sabotaging the opponents. We're also studying the possibility of subdivide the first group in two, so that we can occupy two different zones in the map.

In our team, agents can communicate with each other to:

- tell his position, role and status;

[1] team's main-contact

– tell about new visible vertices and edges;
– inform the opponent's position;
– inform the value of new probed vertices;
– ask the other agents to go to a determined vertex.

There is a strong exchange of information between the agents in the beginning of the match due to the broadcast of new percepts - vertices, edges, probed vertices and surveyed edges. However, the communication between the agents decreases as the agent's world model starts to be more complete.

The adopted solution is based on the centralisation of coordination, that is, one agent is responsible for determining which is the best zone in the map, and then conduct the other agents to occupy this zone.

The best zone is obtained by calculating for each vertex the sum of its value with the value of all direct and second degree neighbours. The vertex with the greatest sum of values is the center of the best zone.

The coordination algorithm works as following: all agents, apart those from the sabotage group, are asked to go occupy an empty vertex of the known best zone at time. When all agents are in the best zone, the algorithm starts to look to the neighbour vertices of the team's zone in which a agent can move to increase the zone's value.

Each agent has its own beliefs, desires, intentions and control thread. Each agent is autonomous to decides by itself its next action.

In each step, the agent must decide which plan will be executed given the state of the environment captured in its world model. The plan's priority is determined by the order in which the plans were declared, and the executed plan will be the first one to has its conditions satisfied.

For this year's contest we do not intend to distribute the agents in several machines but is our intention to work after the tournament on a distributed team.

We did not use any existing multi-agent system methodology.

Software Architecture

Out team was developed using the JaCaMo[2] framework which is a combination of: Jason[3], for programming autonomous agents; Cartago[4], for programming environment artifacts; and Moise[5], for programming multi-agent organisations.

The Java programming language was used to handle the communication to the server, the geometric problems - as find the best zone in the map - and the world modeling.

The organization structure was defined through Moise; and the core of the system, the agents decision mechanism, was programmed using Jason.

[2] http://jacamo.sourceforge.net/
[3] http://jason.sourceforge.net/wp/
[4] http://cartago.sourceforge.net/
[5] http://moise.sourceforge.net/

The development platform was the Eclipse IDE[6], and the JRE 1.6 was the runtime platform.

We used the breadth-first search and the Dijkstra's algorithms to find the minimum path - regarding respectively the number of edges or the energy cost - between two vertices in the graph.

[6] http://www.eclipse.org/