MASTER'S THESIS

Multi-agent team competitions and the implementation of a team-strategy

Wang, Tingting

Date of Award:
2006

Multi-Agent Team Competitions and the Implementation of a Team-Strategy

Wang Tingting

A thesis submitted in partial fulfillment of the requirements

for the degree of

Master of Philosophy

Principal Supervisor: Prof. LIU Jiming

Hong Kong Baptist University

September 2006

Abstract

One of the research problems in Distributed Artificial Intelligence (DAI) is how to develop coordination methods that enable agents to interact coherently as a whole, so as to pursue a common goal or perform a set of tasks. Specifically, for agents involved in team-based competitions, these methods, i.e., a team-strategy, should also help them to pursue better performance than their opponents. A team-strategy refers to a collection of decisions made by a team of agents when facing various situations. An agent makes a decision by selecting an *optimal* action from a pool of possible actions. The *optimal* here refers to the contribution of executing this action with respect to a team, rather than to a single agent.

Team-based competitions can be widely observed in multi-agent applications. For example, in E-commerce, competitive agents involved in an auction need to compete as a team for an expected price against others. Also, there are two commonly used software environments for testing competitive behaviors of agents: RoboCup and RoboCupRescue [31, 50].

In our research, we aim to design and characterize a general competitive multi-agent environment. By *general* we mean:

1. It can simulate different objects in different competitive scenarios, e.g., obstacles in a route, targets to be caught, tasks to be accomplished and competitors.

2. It can implement different rules in different competitive scenarios, e.g., if agent A sees a target of type B, agent A will move to the target.

3. It can implement different properties (e.g., stamina and vision range) and behaviors of agents, including individual and coordinative behaviors.

Next, we discuss how to design and validate an effective team-strategy in such a general environment as well as a specific benchmark environment, i.e., RoboCup. A benchmark environment is necessary because (1) it is widely recognized by other researchers; (2) we can compare with other team-strategies that have been proposed under the same situations, e.g., number of agents, parameters of an agent and rules of a competition.

Here we explain further the *effectiveness* of a team-strategy:

- We cannot determine whether or not a team-strategy is effective without comparing with others. In a competition, if agents with *TeamStrategy A* performs better than agents with *TeamStrategy B*, we can say *TeamStrategy A* is effective.

- The performance of a team-strategy is relative to a specific competitive scenario, e.g., in a target-chasing scenario, better performance corresponds to catching a target more often; in a task-allocation scenario, better performance corresponds to accomplishing more tasks than others.

- A team-strategy may not be always effective. The effectiveness depends on specific conditions, e.g., the level of opponents and the irregularity of its environment.

To describe different competitive conditions, we need to find a method to quantitatively characterize an environment. In this thesis, we characterize the *irregularity*, i.e., the dynamics of an environment. The higher the irregularity of an environment, the more difficult for a team to predict it. If a team learns faster than its opponents in predicting the dynamics of an environment, i.e., it uses a more effective team-strategy, it will be more likely to catch a target, or more rapidly to accomplish the required tasks in competitions. Thus we can test the performance of a team-strategy with respect to different degrees of the *irregularity* in a competitive environment.

Our research has been divided into the following objectives:

- To develop a general competitive multi-agent environment that embodies most of the coordinative (cooperative and competitive) behaviors among agents, and to quantitatively characterize this environment. (Chapter 3)
- To develop and validate the effectiveness of a team-strategy, i.e., the Minority Game team-strategy (MGTS) in different scenarios, and to study under what specific conditions (characterized by the *irregularity* of an environment) a team-strategy is effective. (Chapters 4 and 5)
- To apply and validate the effectiveness of MGTS in a benchmark competitive environment, i.e., RoboCup in our research. (Chapter 6)

Our main contributions are stated as follows:

1. We have designed `DynaGrid` as a general competitive environment.
   (a) It is a grid-like environment with adjustable dimensions.
   (b) Each location in the environment has a utility value, which represents the importance of this location.
   (c) The agents in this environment can be classified according to their roles, e.g., moving targets, target-chasing agents, defensive agents and free agents.
2. We have proposed an index ψ, which is derived from Recurrence Plot [29]. Using ψ, we can quantitatively measure the irregularity of an environment under the following scenarios: (1) a target-chasing scenario, in which more than one team of agents compete to catch a moving target; (2) a task-allocation scenario, in which more than one teams of agents with different roles compete to accomplish some static tasks.
3. We have designed a team-strategy, i.e., the MG team-strategy (MGTS), based on the Minority Game (MG) theory [13]. MG, which has been widely used in modeling economic problems such as purchasing and selling stocks in stock

markets, has shown similar characteristics that meet the fundamental requirements of coordination among agents in dynamic competitions.

4. We have discussed how to apply MGTS to different scenarios in `DynaGrid` and to RoboCup. Through experimentation, we have demonstrated that in different dynamic competitive scenarios, MGTS is useful for a team of agents to improve their performance under certain conditions. For example, in a target-chasing scenario, MGTS works best when the irregularity range is $\psi \in [1, 4]$; in a task-allocation scenario, MGTS works in a certain region: $\psi \in [1.3, 3]$; in RoboCup, MGTS works in the region of $\psi \in [1, 3]$. Also, the improvement validates some intrinsic characteristics, e.g., the phase transition phenomenon on the memory size, as discovered in the theoretical MG model.

Contents