Biologically Inspired Cognitive Architectures (2014) 7, 87–97

Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/bica

RESEARCH ARTICLE

Performance of heterogeneous robot teams with personality adjusted learning

Thomas Recchia *, **Jae Chung, Kishore Pochiraju**

Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, United States

Received 13 September 2013; accepted 26 October 2013

KEYWORDS
Multi-agent system;
Reinforcement learning;
Myers–Briggs Type Indicator;
Robot teaming;
Heterogeneous robot team

Abstract
This paper presents a reinforcement learning algorithm, which is inspired by human team dynamics, for autonomous robotic multi agent applications. Individual agents on the team have heterogeneous capabilities and responsibilities. The learning algorithm assigns strictly local credit assignments to individual agents promoting scalability of the team size. The Personality Adjusted Learner (PAL) algorithm is applied to heterogeneous teams of robots with reward adjustments modified from earlier work on homogeneous teams and an information-based action personality type assignment algorithm has been incorporated. The PAL algorithm was tested in a robot combat scenario against both static and learning opponent teams. The PAL team studied included distinct commander, driver, and gunner agents for each robot. The personality preferences for each agent were varied systematically to uncover team performance sensitivities to agent personality preference assignments. The results show a significant sensitivity for the commander agent. This agent selected the robot strategy, and it was noted that the better performing commander personalities were linked to team oriented actions, rather than more selfish strategies. The driver and gunner agent performance remained insensitive to personality assignment. The driver and gunner actions did not apply at the strategic level, indicating that personality preferences may be important for agents responsible for learning to cooperate intentionally with teammates.

Introduction

As robotic agents become more capable and less expensive, there is an increasing potential for teams of robots to interact with humans and each other in order to cooperatively achieve tasks. One way to compose a high performance team is to implement multi agent learning for the agents,

* Corresponding author. Tel.: +1 862 219 5193.
E-mail addresses: trecchia@stevens.edu (T. Recchia), jae.chung@stevens.edu (J. Chung), kishore.pochiraju@stevens.edu (K. Pochiraju).

2212-683X/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.bica.2013.10.003

so that the agents themselves can learn the best policies to work together. Systems in which each agent is responsible for learning its own policies are termed concurrent learning systems (Panait & Luke, 2005). These systems have the benefit of open-ended scalability when the design of individual agents is not linked to the design of a number of other agents in the team. Also, when a human based psychological model for personality preferences is incorporated into the agents, this approach applies directly to the BICA Challenge to create a real-life computational equivalent of the human mind (Samsonovich, 2012). We explore the effectiveness of human-like personality preferences to change learned behaviors and improve team performance, which is envisioned to lead to improved implicit and naturally occurring cooperation with other agents and eventually with human teammates.

In cooperating heterogenous teams, determining how the actions of individual agents influence the achievement of the team goals is quite difficult. This has been referred to as the credit assignment problem (Panait & Luke, 2005), and several approaches to solving this problem have been explored in the literature (Agogino & Tumer, 2006; Balch, 1997; Chang, Ho, & Kaelbling, 2003; Kalyanakrishnan et al., 2009; Mataric, 1994; Makar, Mahadevan, & Ghavamzadeh, 2001; Santana, Ramalho, Corruble, & Ratitch, 2004; Tangamchit, Dolan, & Khosla, 2002; Tumer, Agogino, & Wolpert, 2002; Tumer & Agogino, 2006; Wolpert & Tumer, 2001). Many of these approaches use reinforcement learning, which defines a reward scheme that enable the agents to learn cooperation. Recently, the authors (Recchia, Chung, & Pochiraju, 2013) have developed a local reward scheme inspired by human teaming concepts which was shown to promote team development in a scarce resource gathering task for a team of agents with homogeneous capabilities. This paper extends that work by adapting the algorithm to be used on a team of agents with heterogeneous capabilities in a combat scenario, and investigating its effects on team performance.

Background

In the current investigation, a team of heterogeneous agents capable of adapting through personality adjusted reinforcement learning are studied in a combat scenario against both a static and learning opponent team. The capability of the agents to learn to take the best actions to increase team performance is measured in order to evaluate the effect of assigning various personality preferences to the different types of agents. Because this investigation integrates ideas from various backgrounds, a brief overview of the relevant concepts is warranted.

There are five important aspects to the current investigation. The first aspect pertains to agent learning. Q-learning was selected for this study as a representative type of reinforcement learning. The second aspect is related to the solutions of the credit assignment problem for cooperative multi agent systems. The third is the application of personality types as inspiration for the agent teaming scheme. The Myers–Briggs Type Indicator (MBTI) (Myers & Myers, 1995), which is a human psychology tool, is explored in this investigation. The fourth aspect is related to the use of an information based model (Lowen, 1982) to classify action decisions into the MBTI structure. The fifth aspect is the implementation framework for conducting performance simulations. Each of these aspects is discussed in this section.

Q-learning

One of the widely used adaptive algorithms for robot control in dynamic environments is Q-learning (Arkin, 1998; Sutton & Barto, 1998; Watkins & Dayan, 1992). In this algorithm, a robotic agent is characterized by a state vector, $\vec{x}$, and can choose to perform an action, $a_i \in \vec{a}$, which is the set of all possible actions. The Q-function defines a value that represents the utility of the action, a_i, given the current state, $\vec{x}$. Normally, the problem is discretized, so the Q-function is represented by a table of values for all possible state-action combinations. This table is calculated recursively on-the-fly by the agent as it performs actions and evaluates a reward/punishment equation that depends on the agent's goal. The standard update equation for the Q-function is:

$$Q_{new}(\vec{x}, a_i) = Q(\vec{x}, a_i) + \alpha(r + \gamma \max(Q(\vec{y}, \vec{a})) - Q(\vec{x}, a_i)) \quad (1)$$

where α is the learning rate parameter that controls how quickly the agent learns; r is the reward or punishment received; γ is the discount factor that controls how much the agent plans for future states; $max(Q(\vec{y}, \vec{a}))$ is the utility of state $\vec{y}$, which results from taking action a_i from state $\vec{x}$. It is the maximum value of $Q(\vec{y}, \vec{a})$, over all possible actions, $\vec{a}$. Every time an agent takes an action, one entry in its Q-function is updated to reflect the utility of taking that action from the state the agent was in at the time. In this way, the Q-function represents the current estimate of the optimal policy for the agent to follow to achieve its goal (Sutton & Barto, 1998; Watkins & Dayan, 1992).

Often the agent is trying to achieve its goal even before the Q function has completely converged to an optimal policy. In this case it is necessary for it to decide if it should follow the current policy, or to try something new at any given decision point. One popular algorithm to handle this is called the ε-greedy policy. In this policy, a parameter between 0 and 1, ε, is set by the designer. During execution, a random draw between 0 and 1 is made by the agent. If the drawn number is less than ε, the agent tries a random action to explore its options. If it is greater than ε, the agent exploits the current Q function recommended action. This ensures that as the number of actions taken goes to infinity, all state action pairs are visited (Sutton & Barto, 1998).

Credit assignment

Several researchers investigated the credit assignment problem using global rewards to enable relatively small teams of agents to learn cooperative behaviors. Balch (1997) and Tangamchit et al. (2002) studied the effect of using global versus local rewards on teams of concurrently learning agents with reinforcement learning (RL). Balch investigated the emergence of teaming behavior on a globally rewarded soccer team and reported on the superior performance of the globally rewarded team versus the locally rewarded one. Tangamchit et al. compared the performance of globally versus locally rewarded agent teams in