Chapter 8.16
Evolutionary Computation as a Paradigm for Engineering Emergent Behavior in Multi-Agent Systems

Robert E. Smith
The University of the West of England, UK

Claudio Bonacina
The University of the West of England, UK

ABSTRACT

In the multi-agent system (MAS) context, the theories and practices of evolutionary computation (EC) have new implications, particularly with regard to engineering and shaping system behaviors. Thus, it is important that we consider the embodiment of EC in "real" agents, that is, agents that involve the real restrictions of time and space within MASs. In this chapter, we address these issues in three ways. First, we relate the foundations of EC theory to MAS and consider how general interactions among agents fit within this theory. Second, we introduce a platform independent agent system to assure that our EC methods work within the generic, but realistic, constraints of agents. Finally, we introduce an agent-based system of EC objects. Concluding sections discuss implications and future directions.

INTRODUCTION

With the advance of computational power and communications speed, we now live in a computational world where a large number of software agents may be acting on behalf of even the most casual user: searching for music, comparing pension schemes, purchasing goods and services, identifying chat partners, etc. Moreover, these agents may be collaborating with those of other users, while spawning and managing agents of their own. In more formal settings, a business, academic, or government user may simultaneously employ many software agents to manage workflow, trade goods or information, collaboratively solve problems, etc. In the future, even relatively simple household appliances may play a role in this churning system of interacting, computational agents.

The behavior of a multi-agent system (MAS) is a result of the repeated (usually asynchronous) action and interaction of the agents. Understanding how to engineer adaptation and self-organization is thus central to the application of agents in the computational world of the future.

Desirable self-organization is observed in many biological, social, and physical systems. However, fostering these conditions in artificial systems proves to be difficult and offers the potential for undesirable behaviors to emerge. Thus, it is vital to be able to understand and shape emergent behaviors in agent-based systems. Current mathematical and empirical tools give only partial insight into emergent behavior in large, agent-based societies. Evolutionary Computation (EC) (Back et al., 1997) provides a paradigm for addressing this need. Moreover, EC techniques are inherently based on a distributed paradigm (natural evolution), making them particularly well suited for adaptation in agents.

In the MAS context, EC theories and practices have new implications. Agents that interact according to these theories are no longer locked inside the laboratory conditions imposed by EC researchers and users. Thus, it is important that we consider the embodiment (in a sense similar to that in Brooks et al., 1998) of EC in "real" agents, that is, agents that involve the real restrictions of time and space within a MAS.

We address this issue in two ways. First, we have developed a platform independent agent system, to assure that we work within the generic, but realistic, constraints of agents. Second, we have developed an agent-based system of EC objects. The prime thrust of our research with these tools is to facilitate understanding of EC within agents and understanding of more general agent interactions in the light of EC theories.

The following sections describe the foundations of EC theory and practice and relate these foundations to MAS, both in terms of applying EC to MAS and in terms of understanding general MASs as evolving systems, through EC theory. The platform independent agent system is presented, along with the generic software framework for EC in MAS. Final sections include implications and future directions.

BACKGROUND

The key qualities exhibited by software agents are autonomy, reactivity, and proactivity (Franklin & Graesser, 1997; Wooldridge & Jennings, 1996). Moreover, agents have the possibility of mobility in complex network environments, putting software functions near the computational resources they require. Agents can also explicitly exploit the availability of distributed, parallel computation facilities.

However, these qualities ultimately depend on the potential for agent adaptation. For instance, if an agent is to operate with true autonomy in a complex, dynamic environment, it may have to react to a spectrum of circumstances that cannot be foreseen by the agent's designer. Autonomous agents may need to explore alternative reactive and proactive strategies, evaluate their performance online, and formulate new, innovative strategies without user intervention.

Areas where agents could benefit from adaptation are addressed by active research in machine learning (e.g., classification of unforeseen inputs, strategy acquisition through reinforcement learning, etc.). However, many machine learning techniques are focused on centralized processing of databases to formulate models or strategies. In contrast, EC techniques are inherently based on a distributed paradigm (natural evolution), making them particularly well suited for online, ongoing adaptation in agents.

Note that in the following description, we will intentionally avoid providing the typical description of EC, to show that the connection to agents emerges from less method-specific rationale. Also note that (while explicitly avoiding entering the endless debate on what constitutes an agent), we