Available online at www.sciencedirect.com
SCIENCE DIRECT
Electronic Notes in Theoretical Computer Science 141 (2005) 163–179
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Adaptation and Evolution in Dynamic Persistent Environments

David Keil and Dina Goldin[1]

Computer Science & Engineering Dept.
University of Connecticut
Storrs, CT USA

Abstract

Optimization (adaptation) of agents interacting with *dynamic persistent environments* (DPEs) poses a separate class of problems from those of static optimization. Such environments must be incorporated into models of interactive computation.
By the No Free Lunch Theorem (NFLT), no general-purpose function-optimization *algorithm* can exist that is superior to random search. But *interactive* adaptation in environments with persistent state falls outside the scope of the NFLT, and useful general-purpose interactive optimization *protocols* for DPEs exist, as we show.
Persistence of state supports *indirect interaction*. Based on the observation that *mutual causation* is inherent to interactive computation, and on the key role of persistent state in multiagent systems, we establish that indirect interaction is essential to multiagent systems (MASs).
This work will be useful to researchers in coordination, evolutionary computation, and design of multiagent and adaptive systems.

Keywords: adaptive systems, coordination, dynamic persistent environments, evolutionary computation, interactive computing, models of computation, multiagent systems

1 Introduction

Environments of adaptive or intelligent agents have been characterized along five dimensions: (1) accessible vs. inaccessible; (2) deterministic vs. nondeterministic; (3) episodic vs. nonepisodic; (4) static vs. dynamic; (5) discrete vs. continuous [28]. The most difficult environments for which to develop intelligent systems are those that approximate the real world, i.e. ones that are

[1] Email: {dkeil,dqg}@engr.uconn.edu

doi:10.1016/j.entcs.2005.05.021

inaccessible, non-deterministic, nonepisodic, dynamic, and continuous. For function-based computation, captured by Turing machines, the properties of an environment are of no consequence, because one execution of an algorithm simply computes a function on whatever single input arrives from the environment. (Nondeterministic Turing machines compute functions that yield *sets* of outputs.)

Some work on *evolutionary computation* has characterized real-world environments in a similar way and has pointed out that an agent's environment may be a function of the existing agent population [13]. By contrast, assumptions about the environment (e.g., *constraints* on it) are part of the interactive *problem specification*. To design cars, for example, we assume they will be driving on a paved road, with gravity and temperature conditions normal on Earth. We may further restrict the environment for research purposes by assuming that the road has lane dividers and is clear of obstacles.

Evolutionary computation emerged to address difficult optimization problems using interactive processes of selection, mutation, and recombination found in nature [15]. In accordance with the *function-based* paradigm that dominates computer science, however, EC has traditionally been conceived as the *algorithmic* search for solutions to *algorithmic* problems. Thus, the "evolutionary algorithm" is applied to static "function optimization" problems. In Section 2, we show how this traditional way of posing problems leads into paradoxical results such as the No Free Lunch Theorem that seem (falsely) to lead to pessimistic conclusions.

Models suitable for evolutionary computation in real-world-like environments must incorporate the environment [20,10]. A significant research trend redefines the computational problem from one of static optimization to one in dynamic environments [8,5]. We suggest a focus on *persistent state* (memory) in such environments. It is persistence of state that enables non-episodic behavior. Our interest is broader than evolutionary computation, because the lessons that can be learned about *dynamic persistent environments* (DPEs) generalize to any adaptive system.

By the No Free Lunch Theorem (NFLT) no function-optimization algorithm exists superior to random choice. For dynamic persistent environments, however, we show the existence of useful general-purpose optimization protocols (Section 3). For example, life forms and human societies have survived in dynamic environments by *learning* methods that have broad applicability.

Because of the power of persistence, the problem of adaptation to a persistent environment may become more challenging than for nonpersistent environments. Since persistent state supports indirect interaction among agents with access to that state, the question of multiagent systems (MASs) arises