Chapter 1

GPTP 2009: AN EXAMPLE OF EVOLVABILITY

Una-May O’Reilly¹, Trent McConaghy² and Rick Riolo³

¹Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology; ²Solido Design Automation Inc., Canada; ³Center for Study of Complex Systems, University of Michigan.

Abstract

This introductory chapter gives a brief description of genetic programming (GP); summarizes current GP algorithm aims, issues, and progress; and finally reviews the contributions of this volume, which were presented at the GP Theory and Practice (GPTP) 2009 workshop.

This year marks a transition wherein the aims of GP algorithms – reasonable resource usage, high quality results, and reliable convergence – are being consistently realized on an impressive variety of “real-world” applications by skilled practitioners in the field. These aims have been realized due to GP researchers’ growing collective understanding of the nature of GP problems which require search across spaces which are massive, multi-modal, and with poor locality, and how that relates to long-discussed GP issues such as bloat and premature convergence. New ways to use and extend GP for improved computational resource usage, quality of results, and reliability are appearing and gaining momentum. These include: reduced resource usage via rationally designed search spaces and fitness functions for specific applications such as induction of implicit functions or modeling stochastic processes arising from bio-networks; improved quality of results by explicitly targeting the interpretability or trustworthiness of the final results; and heightened reliability via consistently introducing new genetic material in a structured manner or via coevolution and teaming. These new developments highlight that GP’s challenges have changed from simply “making it work” on smaller problems, to consistently and rapidly getting high-quality results on large real-world problems. GPTP 2009 was a forum to advance GP’s state of the art and its contributions demonstrate how these aims can be met on a variety of difficult problems.
1. The Workshop

In the beautiful, springtime charm of Ann Arbor, the seventh annual Genetic Programming Theory and Practice (GPTP) workshop was held at the University of Michigan campus from May 14-16, 2009.

We are grateful to all sponsors and acknowledge the importance of their contributions to such an intellectually productive and regular event. The workshop is generously founded and sponsored by the University of Michigan Center for the Study of Complex Systems (CSCS) and receives further funding from the following people and organizations: Michael Korns of Freeman Investment Management, Ying Becker of State Street Global Advisors, John Koza of Third Millennium, Bill and Barbara Tozier of Vague Innovation, Mark Kotanchek of Evolved Analytics, Jason Moore of the Computational Genetics Laboratory of Dartmouth College and Conor Ryan of the Biocomputing and Developmental Systems Group of the University of Limerick.

To make the results of the workshop useful to even a relative novice in the field of GP, we start the chapter with a brief overview of genetic programming (GP). Sections 3 and 4 describe current GP challenges and progress in GP. Sections 5 and 6 then organize and summarize the contributions of chapters in this volume from two perspectives: according to how contributed empirical research is informing GP practice, then according to the domains of application in which success through best practices has been reported. We conclude with a discussion of observations that emerged from the workshop and potential avenues of future work.

2. A Brief Introduction to Genetic Programming

Genetic programming (GP) is a search and optimization technique for executable expressions that is modeled on natural evolution. Natural evolution is a powerful process that can be described by a few central, general mechanisms; for an introduction, see (Futuyma, 2009). A population is composed of organisms which can be distinguished in terms of how fit they are with respect to their environment. Over time, members of the population breed in frequency proportional to their fitness. The new offspring inherit the combined genetic material of their parents with some random variation, and may replace existing members of the population. The entire process is iterative, adaptive and open ended. GP and other evolutionary algorithms typically realize this central description of evolution, albeit in somewhat abstract forms. GP is a set of algorithms that mimic of survival of the fittest, genetic inheritance and variation, and that iterate over a “parent” population, selectively “breeding” them and replacing them with offspring.

Though in general evolution does not have a problem solving goal, GP is nonetheless used to solve problems arising in diverse domains ranging from en-