METHODOLOGY, PEDAGOGY, AND PHILOSOPHY
Poster Paper

Erick Cantú-Paz, chair

JEO: Java Evolving Objects

M.G. Arenas[(1)], B. Dolin[(1,2)], J.J. Merelo[(1)], P.A. Castillo[(1)], I. Fdez de Viana[(4)], M. Schoenauer[(3)]

[(1)]Arquitectura y Tecnología de los Computadores
Universidad de Granada
CP:18071 Granada - Spain
maribel@geneura.ugr.es

[(2)]Computer Science Dept.
Stanford University
Stanford, CA 94305
USA
dolin@cs.stanford.edu

[(3)]Ecole Polytechnique
91128 Palaiseaou
Cedex – France
marc.schoenauer@inria.fr

[(4)]Ciencias de la Computación
Universidad de Granada
CP:18071 Granada- Spain
ijfviana@ugr.es

1 INTRODUCTION

JEO addresses the problem of increased computational resources currently demanded from Evolutionary Computation (EC). Experiments and applications need, in many cases, not merely a single computer but the computational power of an entire network of computers. Since these networks are almost always heterogeneous, portability is an additional problem. DREAM (*Distributed Resources Evolutionary Algorithm Machine*) [1] is a European research project designed to provide the research community with a Peer-to-Peer (P2P) system for EC problems. DREAM consists of a code distribution system (the "distributed resource machine" [2]) and an evolutionary computation system (JEO), all written in platform-independent Java code. The fully-integrated package, as such, solves the distrubuted computation and portability requirements.

2 DESIGN

JEO design is based on the following principles:

1. JEO is *object-oriented* and enables easy code re-use and extensibility.
2. JEO is as *platform independent* as Java.
3. JEO allows *diverse types of evolvable objects*. Any evolvable object must complete a set of rules for mutation, crossover, etc. Not only individuals can be programmed in this way, but an operator, or any other object, could implement these rules for evolution.
4. JEO presents a seamless view of the network as a computational resources pool. A single experiment can be run in a distributed and heterogeneous virtual machine. It builds a layer over the already implemented DRM [2] layer. The EC user deals only with EC concepts, such as islands, operators, etc., as opposed to distributed computing concepts like serialization and remote method invocation. JEO task distribution is based on the Deme Model [3].

3 FEATURES

1. JEO supports various EC representations, including *vectors, trees* and *graphs*.
2. Any variable or result, even over various machines, may be recorded and processed statistically at user specified intervals.
3. Multiple competitive or cooperative populations of individuals may be co-evolved.
4. Migration functionality and topology can be specified by the user. Migration topology may be completely independent from machine topology, and can even adapt dynamically to changing run conditions and/or network failures.

Acknowledgments

This work is supported by *Distributed Resources Evolutionary Algorithm Machine* (DREAM IST-1999-12679) project. *This work is funded as part of the European commission Information Society Technologies Programme (Future and Emerging Technologies). The authors have sole responsibility for this work: it does not represent the opinion of the European Community, and the European Community is not responsible for any use that may be made of the data appearing herein.* Brad Dolin is supported by a Fulbright Grant.

References

[1] B. Paechter, T. Baech, M. Schoenauer, M. Sebag, A. E. Eiben, J. J. Merelo, T. C. Fogarty. "DREAM Distributed Resource Evolutionary Algorithm Machine". In proceedings of the Congress on Evolutionary Computation 2000. Vol II pp 951-958. (http://www.sourceforge.net/dr-ea-m)

[2] M Jelasity, M. Preuβ, M. van Steen and B. Paechter. "Maintaingin Connectivity in a Scalable and Robust distributed Environment" In 2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid2002), May 21-24 2002, Berling Germany.

[3] D. E. Golberg. E. Cantú-Paz, "Modeling idealized bounding cases of parallel genetic algorithms", In Koza J. & Co. Eds. Genetic Programming 1997: Proceedings of the Second Annual Conference (pp. 353-361), Morgan Kaufmann (San Francisco, CA).