mpEAd: Multi-Population EA Diagrams

Sebastian Lenartowicz
University of Guelph
50 Stone Road E, Guelph
Ontario, Canada N1G 2W1
slenarto@mail.uoguelph.ca

Mark Wineberg
University of Guelph
50 Stone Road E, Guelph
Ontario, Canada N1G 2W1
mwineber@uoguelph.ca

ABSTRACT

Multi-population evolutionary algorithms are, by nature, highly complex and difficult to describe. Even two populations working in concert (or opposition) present a myriad of potential configurations that are often difficult to relate using text alone. Little effort has been made, however, to depict these kinds of systems, relying solely on the simple structural connections (related using *ad hoc* diagrams) between populations and often leaving out crucial details. In this paper, we propose a notation and accompanying formalism for consistently and powerfully depicting these structures and the relationships within them in an intuitive and consistent way.

CCS Concepts

•Mathematics of computing → Evolutionary algorithms; •Computing methodologies → Genetic algorithms;

Keywords

evolutionary algorithms, genetic algorithms, genetic programming, evolutionary strategies, multipopulation, mpEAd

1. INTRODUCTION

Evolutionary algorithms (EAs) with multiple populations produce complexities that simple single-population systems do not incur. Populations may exchange both genetic and evaluative information[4], and, in more esoteric systems, other types of information as well[2]. The complexity in the structure of information flow throughout the system is in addition to the actual movement of individuals between populations, as seen in island-model migration[4].

To combat the confusion arising from all this complexity, we have developed a graphical formalism that encapsulates the different relationships that can exist between multiple populations in an EA system. The multi-population EA diagram (mpEAd) employs a concise visual grammar to depict multiple populations and the information flow between them, in a similar way to how the Unified Modelling Language (UML)[3] captures the categorization of and relationships between the component parts of object-oriented software systems.

2. ESSENTIAL STRUCTURE OF MPEAD

2.1 Basic Elements of mpEAd

The graph at the heart of mpEAd incorporates two types of nodes: population nodes and conversion nodes, as well as a number of different edge types.

2.1.1 Population Nodes

A population node corresponds to a single optimization algorithm (usually an EA) and a set of solutions (the population). It is denoted using a simple hollow rectangle with solid borders and the name of the node written inside. The population node always includes a set of parallel lines, which serve as a visual reminder of the population inside.

2.1.2 Information as Edges

Edges in the mpEAd graph are used to model the information flow between nodes. As information flow is directional, mpEAd becomes a directed graph, and, per the convention, uses arrows to indicate direction. Genetic information, as often used for evaluation, is represented using a solid edge with a closed arrowhead that is either hollow (genotypic information) or filled (phenotypic information). Non-genetic information simply encapsulates all other types of information, the most ubiquitous type being evaluative information (fitness values). There are, however, many other types of non-genetic information, a full discussion of which is outside the scope of this paper. Regardless of type, non-genetic information is depicted using a dashed edge, with different arrowheads being used to distinguish between different types of information; an example of this is the open arrowhead for evaluative information (clearly seen in Figure 2).

2.1.3 Conversion Nodes

The second type of node, the conversion node, is less obvious and is one of the elements that makes the mpEAd formalism more than simply a topological model of the connections between populations. The role of the conversion node is to take in one or more streams of information, perform processing on them, and to provide the result to another node or nodes. Conversion nodes perform a variety of information processing operations, including but not lim-

GECCO'16 Companion July 20-24, 2016, Denver,CO, USA

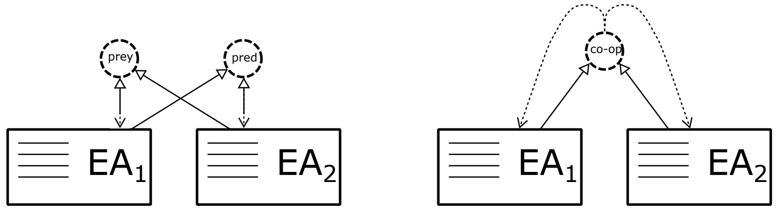
ACM ISBN 978-1-4503-4323-7/16/07.

DOI: http://dx.doi.org/10.1145/2908961.2909011

(a) A naïve approach to representing two-population co-evolutionary systems.

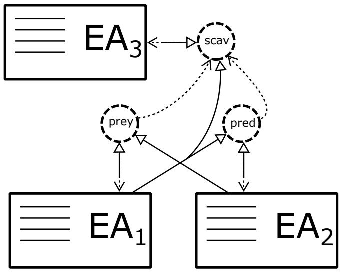
(b) An attempt to model de Boer, Folkert, and Hogeweg's predator/prey/scavenger system[1] using the naïve approach.

Figure 1: A typical naïve way to model co-evolutionary systems.

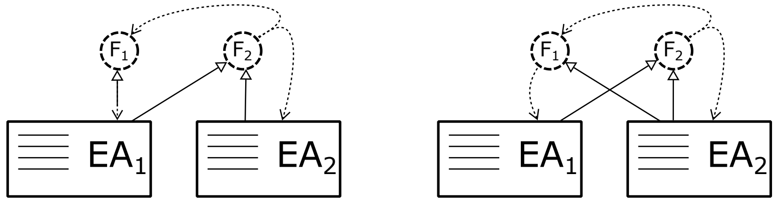


(a) A basic predator/prey co-evolutionary system.

(b) Standard co-operative co-evolution.



(c) Predator/prey extended to include scavengers[1].



(d) A hybrid of co-operative and predator/prey co-evolution with modification.

(e) A hybrid of co-operative and predator/prey co-evolution with cross-modification.

Figure 2: Examples of various co-evolutionary systems. The two complementary systems in 2d and 2e are both, to the best of our knowledge, novel.

ited to decoding genotypic information into phenotypic information, evaluating fitness, combining information from different sources, and converting statistical information into control information (used by the EA within a population node to change its behaviour). The conversion node is depicted using a large hollow circle, often labelled with a name, such as the name of the fitness function used for evaluation.

It should be noted that the fundamental difference between a conversion node and a population node is that, while both types can perform information processing, the conversion node is stateless and does not store information, only taking input and producing output based upon it.

2.2 mpEAd in Action

While efforts have been made in the past to model the interactions between populations, they are often simplistic and rely on *ad hoc* notations, similar to what is seen in Figure 1. The power of mpEAd becomes apparent in comparison to this, as it permits much more accurate and detailed modelling of how the populations interact. All of the diagrams in Figure 2 are different co-evolutionary systems that would be equivalent to the one in Figure 1a, apart from 2c, which actually models the predator/prey/scavenger system in a way that Figure 1b cannot. Many disparate types of multi-population systems (in this case, a variety of co-operative and competitive co-evolutionary systems) can be represented in a way such that their similarities, as well as their differences, become apparent.

With the co-evolutionary systems in Figures 2d and 2e, interesting possibilities begin to appear, as these systems are unknown in the literature. On examination, they appear to be a hybrid between co-operative and predator/prey co-evolution, where the co-operative output is further modified either by the individual itself or by an individual from the other population.

3. CONCLUSION

The mpEAd formalism is a graphical notation designed to permit the depiction of large, complex multi-population EA systems. Designed with the goals of being as intuitive, consistent, distinctive, and simple as possible, mpEAd is a powerful modelling tool for systems often considered too complex to describe clearly.

4. REFERENCES

[1] de Boer, F. K., and Hogeweg, P. Co-evolution and ecosystem based problem solving. *Ecological Informatics 9* (2012), 47–58.

[2] Li, Q., Liu, D., Leung, A., Zhang, N., and Luo, Q. A multilevel genetic algorithm for the optimum design of structural control systems. *Int'l Journ. for Numerical Methods in Eng. 55*, 7 (2002), 817–834.

[3] Rumbaugh, J., Jacobson, I., and Booch, G. *Unified Modeling Language Reference Manual, The.* Pearson Higher Education, 2004.

[4] Whitley, D., Rana, S., and Heckendorn, R. B. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology 7* (1999), 33–48.