Workshop on Complexity through Development and Self-Organizing Representations: Building complexity from simplicity.

Ivan I. Garibay
University of Central Florida
School of EECS & ORC
Orlando, FL 32826
igaribay@mail.ucf.edu

Julian Miller
Department of Electronics
University of York
York, UK
jfm@ohm.york.ac.uk

Sanjeev Kumar
Sibley School of Mech. and Aerospace Eng.
Cornell University
Ithaca, New York 14853
sk525@cornell.edu

Categories and Subject Descriptors

I.2 [**ARTIFICIAL INTELLIGENCE**]: General

Keywords

development, self-organizarion, modularity

1. MOTIVATION

This workshop follows on from the successful workshops on self-organization in representations in evolutionary algorithms, and scalable, evolvable, emergent developmental systems at previous GECCO conferences. This year's workshop is a unified workshop covering both closely related areas.

Evolutionary algorithms (EAs) have been applied to an ever increasing variety of problem domains, for which they have achieved human competitive results on small evolutionary design problems. The application of EAs to tasks of ever increasing difficulty is fraught with problems, namely: stagnation of search in large search spaces, negative epistatic effects, disruption of large building blocks, and scalability, amongst others. Recently, the problem of scalability has attracted much attention, and deservedly so, as its resolution is linked to other critical and demanding open research problems such as: development, evolvability, and modularity. In order to improve the scalability of such systems fundamental research must be undertaken to discover how to evolve increasingly more complex designs. For this we look at the two systems that have achieved scalability: human engineering and natural systems. Manually constructed systems have achieved such things as aircraft with over a million parts, software with tens of millions of lines of code and over a hundred million transistors in microprocessors, suggesting that we can improve the scalability of automated design by using principles of engineering. Similarly, natural evolution and developmental biology have produced adaptable and self-repairing systems of even greater complexity using principles of self-organization. Self-organization is fundamental to the developmental process at all levels: molecular, genetic, and cellular. Nature evolves "'instructions"' in the form of genes that are used to specify the construction of organisms during the process of development. However these "'instructions"' orchestrate a vast number of interactions obeying laws that are no encoded in the genotype. With reports of the number of genes in the human genome being revised downwards, the role of self-organization in complex webs of gene regulation is all the more salient. Given these new findings, perhaps self-organization controlled by genotypic instructions is an important missing ingredient from EAs. To this end, it is anticipated that models of biological cells and multicellular development represent a valuable source of knowledge that will aid us in designing EAs with emergent phenomena such as: adaptability, scale-free-ness, evolvability, and robustness. Regardless of the developmental model or generative representation chosen – cellular automata, genetic regulatory networks, L-systems, etc - we must understand exactly what gives such systems their computational power and exactly how they affect evolvability.

The objective of this workshop is to explore domain-independent methods for representing complex solutions with self-organizable building blocks, and on developmental principles for specifying the construction of complex systems.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

2. DEVELOPMENT, SELF-ORGANIZATION, AND MODULARITY

In recent years, a trend has emerged on Evolutionary Computation that attempts to increase the complexity of the represented solutions without increasing the complexity of the genomes in order to scale up evolutionary search. In order to achieve this goal some sort of complex or "intelligent" genotype to phenotype mapping is required. This mapping needs to extract key features from the problem space and incorporate these features automatically into the representation space in order to:

1. prune unpromising sections of the search space, hence reducing its size

2. evolving increasingly more complex designs, hence achieving scalability

Most approaches to this problem include to some degree development, self-organizarion and modularity. To name a few: developmental mappings [1, 2], implicit embryogenies [3, 4], generative representations [5, 6], modularity, regularity and hierarchy [7, 8, 9, 10, 11], augmenting topologies [12], molecular computing [13], and self-replicating sequences [14, 15].

3. CONTRIBUTING PAPERS

The issue of scaling in a developmental setting is analyzed by Tufle and Thomassen in the first paper of the workshop. They use a rule-based system on a cellular automata topology to model development and conclude that size matters: their results improve with the number of genes on the genome and with the available space for phenome growth.

The encoding of solutions, or in a broader sense the representation of solutions, is crucial for the success of evolutionary search. Direct encodings usually refer to simple genotype to phenotype mappings where the genomes encode a fix predetermined number of phenotypic characters. Indirect encodings are more complex mappings that usually involve the generation, interpretation or construction of the phenotype from genotypic instructions. Designing effective encodings is paramount. Stanley's contribution offers a comparison between artificial phenotypes and naturally occurring patterns in biological organisms in order to help to estimate the value and viability of artificial developmental encodings. Evolvability and adaptability are two important and desirable features of any encoding. Seys and Beer provide experimental evidence that the number of genomic parameters is not as important for evolvability and adaptability as the selected encoding method. Their results favor what the authors called indirect encoding versus direct encoding for their problem domain: evolving a neural network controller for an hexapod agent. Harding, Miller and Lasarczyk offer a comparison between a direct encoding and two cellular automata based indirect encodings targeting a bit pattern problem of variable Kolmogorov complexity. The authors conclude that the performance of the indirect encoding diminishes with increasing problem complexity.

Phenotype self-organization and the emergence of phenotypic traits from basic genotypic structures is an alternative to the generative or interpretative phenome construction found in developmental indirect encoding approaches. Eppstein, Pyne and Goodnight contribution provide insights about Charles Darwin's "mystery of mysteries": speciation. Using a computer simulation the authors show that given localized mating, interbreeding populations self-organize into patches of compatible types, and that if certain types of mild epistasis are present this process results in speciation. Finally, Bakhouya and Gaber contribution offers a framework for self-organization of a collection of agents operating in a dynamic environment. The framework is inspired on the immune system. In their setting, unexpected environmental changes are considered antigens and the collection of agents self-organize its structure in order to deliver an adequate response to eliminate the infection.

4. WORKSHOP MATERIAL ON-LINE

The materials presented during the workshop as well as other related information is available on-line at:
http://codesoar.research.ucf.edu/

5. ADDITIONAL AUTHORS

Additional authors: Gregory Hornby (NASA Ames Research Center, email: `hornby@email.arc.nasa.gov`) and Ozlem Garibay (University of Central Florida, email: `ozlem@cs.ucf.edu`) and Kivanc Oner (Univeristy of Central Florida, email: `kivanc@cs.ucf.edu`).

6. REFERENCES

[1] Gruau, F.: Neural Network Synthesis using Cellular Encoding and the Genetic Algorithm. PhD thesis, Ecole Normale Suprieure de Lyon, France (1994)

[2] Koza, J.R., III, F.H.B., Andre, D., Keane, M.A.: Genetic Programming III. Morgan Kaufmann Publishers (1999)

[3] Kumar, S., Bentley, P.J., eds.: On Growth, Form and Computers. Academic Press (2003)

[4] Bentley, P.J.: Natural design by computer. In: Proc. of the AAAI Spring Symposioum Series, AAAI Press (2003) 1–2 TR SS-03-02.

[5] Hornby, G.S., Lipson, H., Pollack, J.B.: Generative representations for the automated design of modular physical robots. IEEE Transactions on Robotics and Automation (2003) In Press.

[6] Hornby, G.S.: Measuring, enabling and comparing modularity, regularity and hierarchy in evolutionary design. In Beyer, H.G., O'Reilly, U.M., eds.: Proc. of the 2005 genetic and evolutionary computation conference, ACM Press (2005) 1729–1736

[7] Lipson, H.: Principles of modularity, regularity, and hierarchy for scalable systems. In Poli, R., et al., eds.: GECCO 2004 Workshop Proceedings, Seattle, Washington, USA (2004)

[8] de Jong, E.D., Thierens, D., Watson, R.A.: Defining modularity, hierarchy, and repetition. In Poli, R., et al., eds.: GECCO 2004 Workshop Proceedings, Seattle, Washington, USA (2004)

[9] Garibay, O.O., Garibay, I.I., Wu, A.S.: The modular genetic algorithm: exploiting regularities in the problem space. In: Proc. of ISCIS 2003. (2003) 578–585

[10] Garibay, I.I., Garibay, O.O., Wu, A.S.: Effects of module encapsulation in repetitively modular genotypes on the search space. In: Proc. of the 2004 genetic and evolutionary computation conference. (2004) 1125–1137

[11] Watson, R.A.: Hierarchical module discovery. In: 2003 AAAI Spring Symposium Series. (2003) 262–267

[12] Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation **10** (2002) 99–127

[13] Conrad, M.: Computation: Evolutionary, neural, molecular. In: 2000 IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks. (2000) 1–9

[14] Garzon, M., Blain, D., Bobba, K., Neel, A., West, M.: Self-assembly of DNA-like structures in silico. Genetic Programming and Evolvable Machines **4** (2003) 185–200

[15] Banzhaf, W., Dittrich, P., Eller, B.: Selforganization in a system of binary strings with topological interactions. Physica D **125** (1999) 85–104