Evolutionary and Swarm Design

CPSC 599.33 — Summer 2001

Christian Jacob — jacob@cpsc.ucalgary.ca

http://www.cpsc.ucalgary.ca/~jacob

03 July 2001

Course Abstract

Designing like Nature

One of the most exciting,challenging and rewarding research areas today is *biomimetics*, the science and art to incorporate principles of nature into human design processes—from system designs in engineering and computer science to "creative" designs in art.

Nature offers an amazing, yet largely unexplored, repertoir of approaches to design complex and elaborate systems. Instead of composing its entities from pre-fabricated parts, as is the standard way of our current engineering system designs, nature uses alternative approaches to find design solutions for the creation of its organisms, such as

- structure formation,
- growth,
- self-assembly,

- self-organization, and
- emergent system behaviours.

These natural design principles reflect a bottom-up approach to fabrication and system design, which we will explore in this course. It still remains a major challenge to translate nature´s pool of design ideas into terms that we—as artists, engineers, computer scientists, and system designers—can work with.

We will focus on two major paradigms of natural design, which are in fact key ingredients to unravel self-organizational principles of living systems: *swarm intelligence* and *evolution*.

Swarm Intelligence Systems

In general terms, a swarm system can be defined as “a population of interacting elements that is able to optimize some global objective through collaborative search of a space.” [Kennedy and Russel,2001]. A typical example of a natural swarm system is an ant colony. Each ant follows only simple local rules, but through the interaction of a large number of ants, the colony as a whole acts like a super-organism: it acquires food, competes for foraging areas, grows, and maintains a highly complex spatial as well as social organization.Noticably, there is no central control instance. Rather, the overall behaviour of the colony emerges from a massively parallel, decentralized system of interactions among its entities.

Similar principles apply to any multi-cellular organism and its cell-components, to any ecosystem and its collaborating and competing members, or to any society of interacting agents. Swarm intelligence systems will play a key role in developing computer models that allow the exploration of—in particular, biological— systems that show complex, emergent properties through massive interactions of relatively simple agents.Obviously,social insects do solve complex design problems through implicit self-assembly and self-organization, resulting in emergent structures, such as termites' nests or "living bridges" formed by interconnected ants.

Our *Evolutionary and Swarm Design* group is currently exploring these natural design paradigms for applications in graphical 2D and 3D design and in sculpturing (in collaboration with the Faculty of Fine Arts). In particular ,we are also interested in connections between agent-based pattern formation and developmental processes. Eventually, we would like to be able to answer questions such as: How do termites build their nests? How does self-assembly in "living architectures" of social insects work? Do similar principles apply to cluster formation and cell cleavage?

Evolutionary Computation and *Evolvica*

Evolution is the key-designer in nature. Through evolutionary processes of selection and mutation different design ideas are evaluated for their feasibility, i.e., survivability and reproductive success of the associated organisms. Consequently, we will also include evolutionary components into our design system, which will help us to explore design program spaces more efficiently than by simple user-evaluation and-interaction.

In my recent book, "*Illustrating Evolutionary Computation with Mathematica*", I discuss a wide range of evolutionary algorithms, from evolution strategies and genetic algorithms to genetic programming and show applications of these evolutionary systems to design and development. One result of the research for this book is *Evolvica*, a programming and research environment for evolutionary computing, which consists of a large collection of Mathematica packages, notebooks, and tutorials. *Evolvica* is the current major engine for my research in evolutionary systems and is also used by a growing number of other research groups around the world. It is the intention of our *Evolutionary and Swarm Design* group to extend *Evolvica* by a parallel computing environment, which will allow us to run massively parallel evolutionary experiments on large computer clusters. As our first step in this direction we have installed a Parallelization Toolkit and are currently testing several parallelization models for our evolutionary system on a cluster of five Linux workstations. Porting this smaller test system to a larger cluster will then not pose a big problem.

Evolutionary and Swarm Design

The goal of the Evolutionary and Swarm Design project is to set up a programming and research environment to explore the combination of swarm intelligence and evolutionary systems. In particular, for the initial phase of this project, we want to test this new approach of combining a massively parallel, swarm-based system with evolutionary design principles in the area of creating two- and three-dimensional forms, such as artful design patterns and sculptures. The scenario for the system, we intend to build to help us evaluate this novel design approach, can be described as follows:

• **Stigmergy and swarms**: A swarm-based system allows us to generate 3D structures through direct interaction with an initially setup object (e.g., a simulated block of clay), which is subsequently modified by a "swarm toolset," similar to the building of nests by ants, termites, or wasps where stigmergy plays an important part.

• **Self-assembly in swarms**:Alternatively, a swarm of "agents" may self-assemble into a three-dimensional structure, similar to the formation of "living bridges" by ants.

• **Breeding of swarm behaviours**: An evolutionary system will help us to "breed" particular swarm behaviours. We will use mutation and recombination operators to manipulate the contol programs of the agents that constitute the swarms.

  ○ **Interactive breeding**: The breeding or evolution of the swarm control programs can be done either interactively, by direct visual inspection of the results on the computer screen, and subsequent selection of mutation operators.

  ○ **Evolutionary optimization**: Given a specific fitness function for each swarm behaviour or its resulting designs, the evolution process can also be carried out automatically. This offline evolution will complement the interactive breeding process, and can be sped up by mapping the evolutionary algorithms to a machine cluster.

Although we initially focus on the swarm-based evolution of spatial structures,eventually this evolutionary swarm system will not only provide an experimental platform to explore new approaches in 3D structure design, but will also provide a framework to study “natural design processes” and their applicability in system design for computer science and engineering, in particular. The implemented simulation environment will also help biologists to better understand key mechanisms of natural regulation mechanisms in, for example, gene-gene interaction networks, cell development patterns, social insect behaviours, interactions in ecosystems, and even social interactions in human societies.

General Objective

Our general objective is to work towards a programming environment and computational framework to explore and utilize decentralized, massively parallel, agent-based methodologies that can be used to automatically create (evolve) computer programs and design patterns. The use of evolutionary techniques (including implementations on a large computer cluster) to design, breed, test, and optimize swarm systems is a key component of this framework.

References

C.Jacob, *Illustrating Evolutionary Computation with Mathematica*. Morgan Kaufmann Publishers, San Francisco, 2001.

J.Kennedy and Eberhart,R., *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, 2001.

Time Table Summary

□ Lectures

- **3 July**, Tuesday *Introduction to Evolutionary Computing*
  Evolution Strategies, Evolutionary Programming
- **4 July**, Wednesday *Genetic Algorithms, Genetic Programming*
- **5 July**, Thursday *Evolutionary Art and Design*
- *No lectures: 6 July to 15 July*
- **17 July**, Tuesday *Introduction to Swarm Systems*
- **18 July**, Wednesday *Swarm Systems: In-Depth Examples*
- **19 July**, Thursday *Evolutionary Swarms and Swarm Art*

□ Project Assignments

- **24 July**, Tuesday

Here we will finalize your assignments to a particular project, which is defined either by yourself or taken from the list of proposed projects.

□ Project Class Meetings

- **31 July**, Tuesday
- 7 **August**, Tuesday
  (*Note: The university is closed on 6 August for Alberta Heritage Day*.)

- **Project Presentations**
  - **13 August**, Monday
  - **14 August**, Tuesday
  - **15 August**, Wednesday

Evolutionary and Swarm Design — *Table of Contents*

- **The Robot as an Artist**
- **"Evolutionary" Computer Graphics**
- **Organic Genetic and Evolutionary Art**
- **Genetic L-System Programming**

Swarm Systems

- **Cell Replication:**
- **Competitive Cell Replication:**
- **"BlocksWorld":**

Design by Swarms

Projects in Evolutionary and Swarm Design

References

Evolutionary Computing

□ Evolutionary Computing in General

Fogel, D. B., Ed. (1998). *Evolutionary Computation: The Fossil Record*. New York, IEEE Press.

Jacob, C. *Illustrating Evolutionary Computation with Mathematica*. Morgan Kaufmann Publishers, San Francisco, 2001. (www.cpsc.ucalgary.ca/~jacob/IEC)

□ Genetic Algorithms

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, Addison-Wesley.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA, MIT Press.

□ Genetic Programming

Banzhaf, W., et al. *Introduction to Genetic Programming*. Morgan Kaufmann Publishers, San Francisco, 1999.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, MIT Press.

Koza, J. R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA, MIT Press.

Koza, J. R., D. Andre, et al. (1998). *Genetic Programming III: Automatic Programming and Automatic Circuit Synthesis*. San Francisco, CA, Morgan Kaufmann.

☐ Evolution Strategies

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*. Oxford, Oxford University Press.

Rechenberg, I. (1994). *Evolutionsstrategie '94*. Stuttgart, Frommann-Holzboog.

Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. New York, John Wiley & Sons.

☐ Evolutionary Programming

Fogel, L. J., A. J. Owens, et al. (1966). *Artificial intelligence through simulated evolution*. New York, John Wiley and Sons.

Fogel, L. J. (1999). *Intelligence Through Simulated Evolution*. John Wiley & Sons, Neww York.

Fogel, D. B. (1992). *Evolving artificial intelligence*. San Diego, University of California.

Fogel, D. B. (1995). *Evolutionary Computation. Towards a New Philosophy of Machine Intelligence*. IEEE Press, New York.

Evolutionary Design

Bentley, P., and Corne, D. W. *Creative Evolutionary Design*. Morgan Kaufmann Publishers, San Francisco, 2001.

Bentley, P. (Ed.). *Evolutionary Design by Computers*. Morgan Kaufmann Publishers, San Francisco, 2000.

Todd, L., and Latham, W. *Evolutionary Art and Computers*. Academic Press, London, 1992.

Swarm Systems

Resnick, M. (1997). Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. Cambridge, MA, MIT Press.

Kennedy, J., and Eberhart, R. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco.

Artificial Life

Adami, C. (1998). *Introduction to Artificial Life*.Springer/TELOS, Berlin.

Online References

Course web site: www.cpsc.ucalgary.ca/~jacob/

Evolvica web pages: *www.cpsc.ucalgary.ca/~jacob/Evolvica*

Illustrating Evolutionary Computation web site: *www.cpsc.ucalgary.ca/~jacob/-IEC*

MathReader and *Mathematica*

The Wolfram Research web site: www.wolfram.com

The free *MathReader* for reading *Mathematica* notebooks: www.wolfram.com/products/mathreader

Access to *Mathematica* through your CPSC account at UofC: `/usr/local/bin/mathematica`