Cooption and Catalysis in a Model of Technological Evolution

Andrews University
J.N. Andrews Honors Program
SANTA FE INSTITUTE

Eric “Siggy” Scott
Department of Engineering and Computer Science
Department of Mathematics

Mentors: J. Doyne Farmer (SFI)
Stephen Thorman (AU)

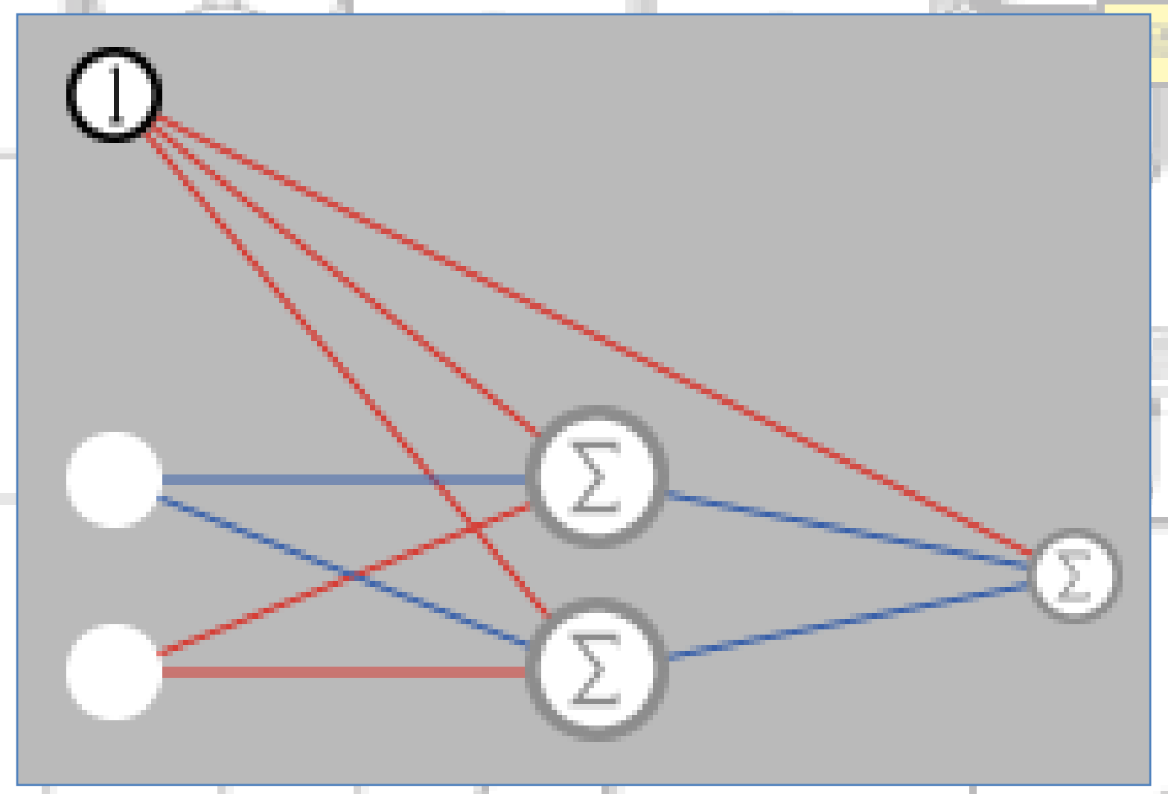
Abstract

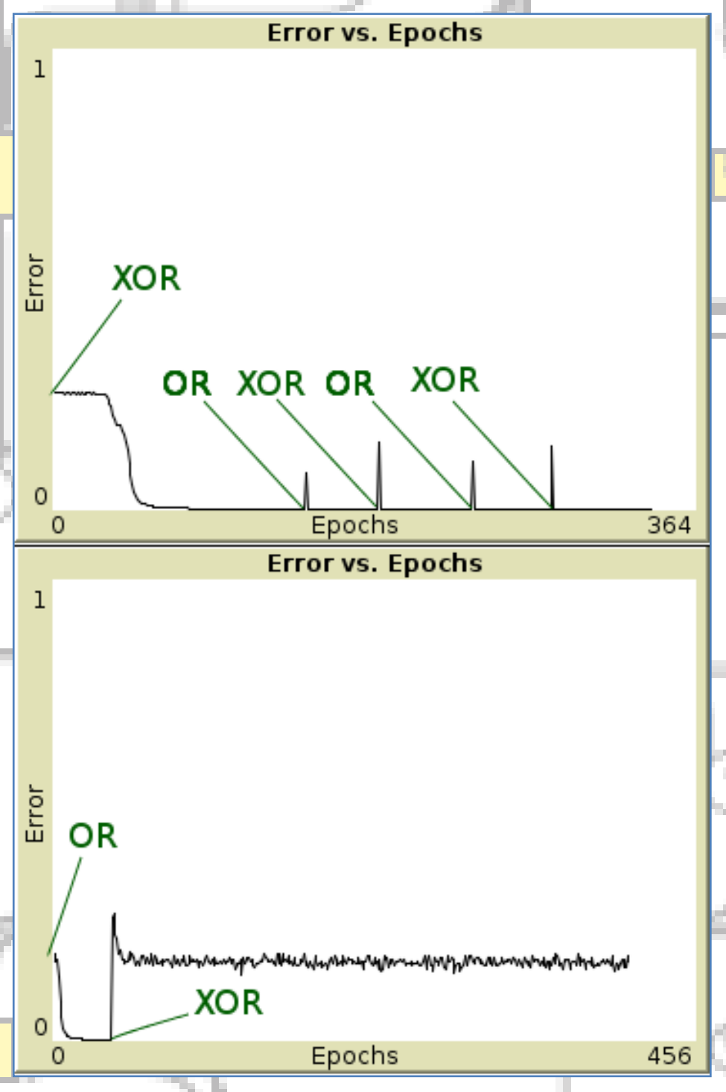
Most evolutionary algorithms ultimately focus on optimizing solutions to a single target function, coevolution and related methods notwithstanding. Cooptive phenomena between organisms adapted to distinct environmental niches, however, lie at the heart of the evolution of complex functions in nature and technology, where solutions adapted for one problem are repurposed to solve another, related problem. Boolean functions have become a popular toy model for exploring the dynamics of such processes, and provide insight into new approaches to evolutionary computation. We implemented the basics of a model of combinatorially evolving logic circuits developed by Brian Arthur and Wolfgang Polak, and began to explore the sort of cooptive and catalytic phenomena its success depends upon. We observed a significant difference in the dynamics of evolution between when the full suite of fitness functions is present, versus when only a few pieces of the selective pressure are active at a time. Future work will examine these dynamics in more detail.

Artificial Neural Network Example

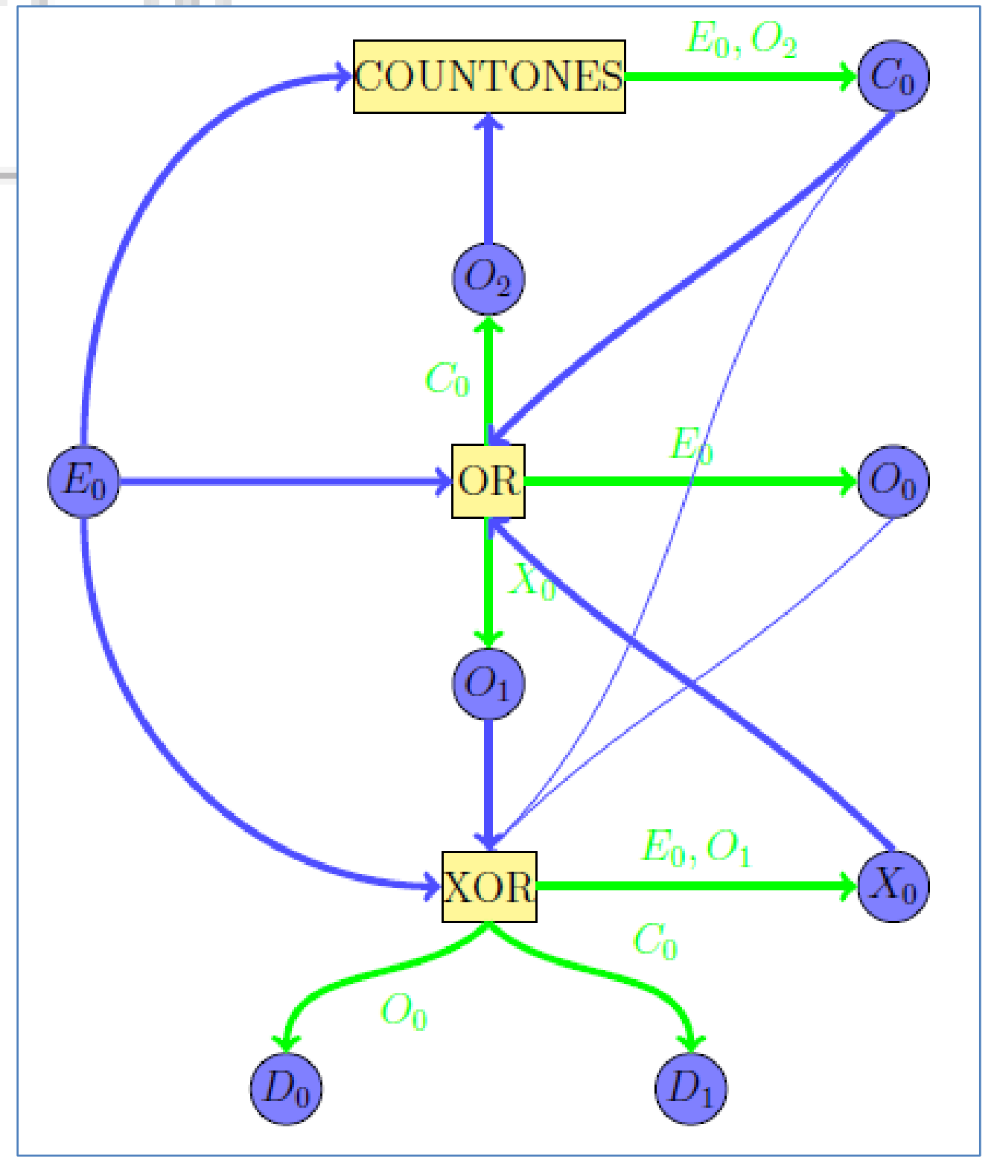
Evolutionary computation (EC) aims to exploit the complexity-generating power of adaptive processes in nature to optimize solutions for real-world, human-defined problems. To date, the state of the art is notoriously limited when compared to the algorithms' natural counterparts. It is the present author's emphasis that part of the solution to these difficulties lies in Stephen Jay Gould's concept of *exaptation*, in which functions evolved for one purpose or need are coopted and further refined for another task or environment (Gould 1982). Exaptation exploits commonalities and chance relationships between ecological niches to learn and ultimately generalize from multiple fitness functions and develop higher quality and/or more complex solutions. In this paradigm, intermediate functions play a major role.

We illustrate this perspective with a toy example using an Artificial Neural Network (ANN). A simple ANN was trained via backpropagation to simulate the OR, XOR, and COUNTONES (count the number of ones in the input signal) functions for a two-bit input string. The solution to a function is used as the initial condition to solve another problem, and cooptive effects are observed. For example, learning XOR – a more difficult problem – first makes it trivial to learn OR. Learning OR first, however, inhibits the network’s ability to learn XOR.





The cooptive relationships between three objective functions are summarized in the diagram below. Green edges point from functions to resulting solutions, and are labeled with the initial condition that was used. Oi’s solve OR, Ci’s solve COUNTONES, Xi’s solve XOR, and Di’s fail to solve any of the functions. The initial condition E0 corresponds to random weights on (-0.1, 0.1).

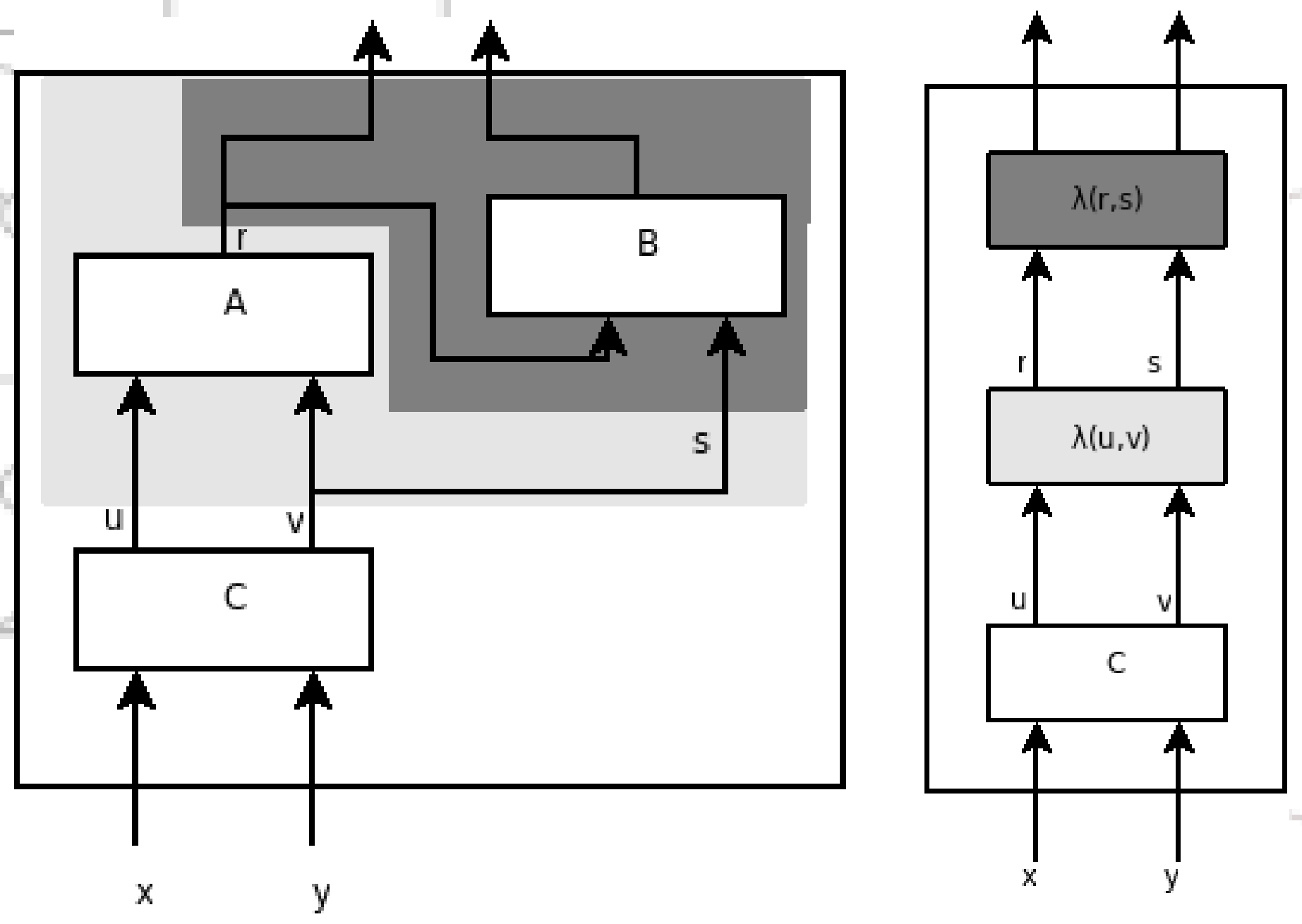


Technological Evolution

Cooptive effects are not limited to the biological domain, being the focus of the study of path-dependence and increasing returns in economics. In particular, the process of technological evolution by combination of existing technologies into new solutions relies heavily on the development of intermediate technologies that are retooled for myriad purposes.

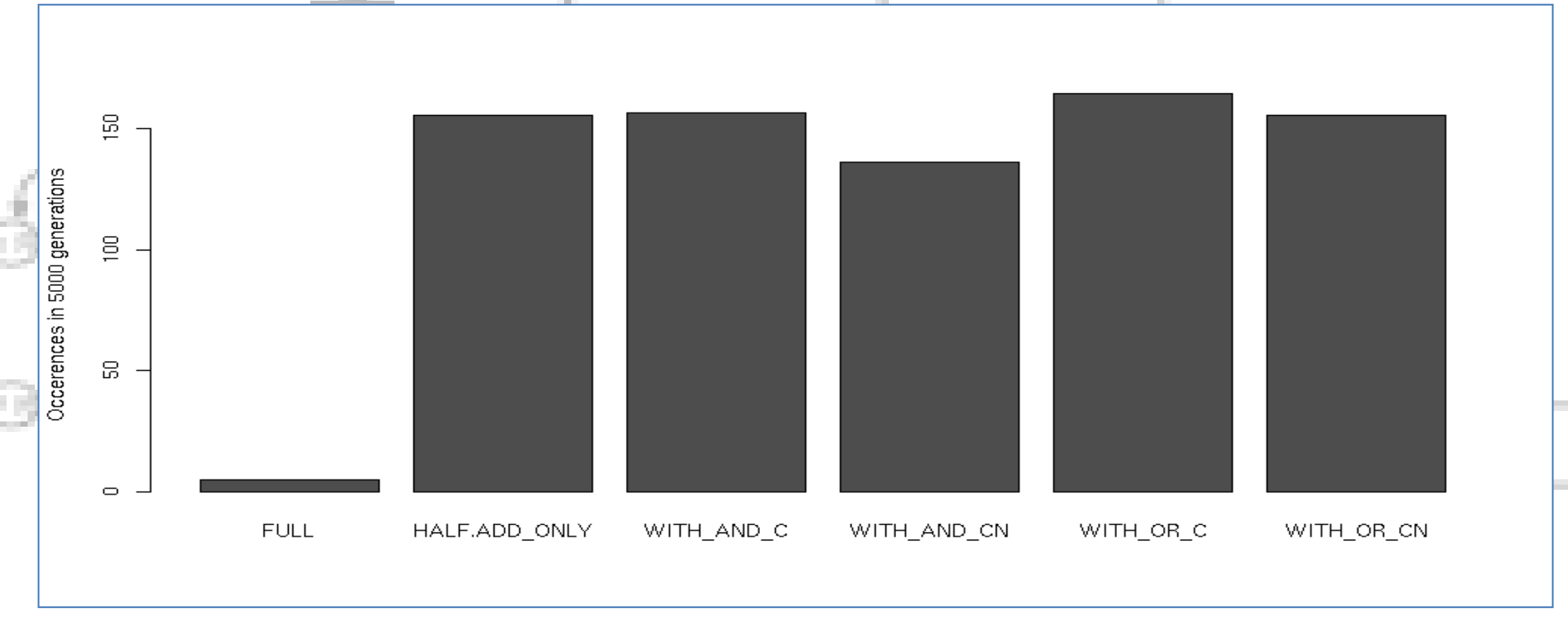
We analyze a model of evolving logic circuits developed by Arthur and Polak (2006). New circuits are generated via random compositions of circuits already in the pool, and evaluated against a battery of fitness functions, its mean fitness score defining its probably of being incorporated into a new circuit via tournament selection.

The model was implemented in Common Lisp, an adjacency matrix used to remember circuit designs, a topological sort on its components used to order parallel modules for execution, and a Reduced Ordered Binary Decision Diagram (ROBDD) used to cache the truth tables of each circuit for fast processing.



As a first step toward creating a map of cooptive relationships between initial conditions and fitness functions similar to the one we created for the ANN, we graph the number of times the HALF-ADDER function was evolved in 5,000 generations under different combinations of selective pressures.

We found no statistically significant change in evolution rate for the HALF-ADDER between different combinations of initial conditions (i.e. circuits available in the initial pool) and selective pressures (fitness functions in the battery) involving the AND and OR functions. But when several dozen fitness pressure were active, the search process diverged from the HALF-ADDER, paralyzing its evolution.



Future Work

The goal of this research is to address questions like

- Does the presence of an AND circuit in the primitives set facilitate the evolution of a HALF-ADDER circuit?
- Does the presence of both AND and HALF-ADDER fitness functions as a multiobjective selection pressure facilitate the evolution of a HALFADDER?
- Are the catalytic effects of fitness functions greater than the sum of their parts?
- What functions must evolve, and in what order, to allow more complex functions to be solved?
- What models besides logic circuits (ex. Biological neural networks) might be amenable to A) simulation of cooptive evolution, and/or B) combinatorial evolution.

We aim to map the dependencies in question in more detail, and to develop a framework for systematically studying and representing cooptive phenomena in arbitrary optimization problems.

Selected Biography

- W. Brian Arthur and Wolfgang Polak. The evolution of technology within a simple computer model. *Complexity*, 11(5):23–32, May/June 2006.
- Cristoph Flamm, Alexander Ullrich, Heinz Ekker, Hartin Mann, and Daniel Högerl et al. Evolution of metabolic networks: A computational framework. *Journal of Systems Chemistry* 1:4, August 2010.
- J. Gerhart and M. Kirschner. The theory of facilitated variation. *PNAS*, 104, May 2007.
- Stephen Jay Gould and Elisabeth S. Vrba. Exaptation–a missing term in the science of form. *Paleobiology*, 8(1):4–15, 1982.
- J. R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA, 1992.
- Richard E. Lenski, Charles Ofria, Robert T. Pennock, and Christoph Adami. The evolutionary origin of complex features. *Nature*, 423:139–144, May 2003.
- Javier Macia and Richard Sole. Distributed robustness in cellular networks: insights from synthetic evolved circuits. *J. R. Soc. Interface*, 6:393–400, September 2008.
- Merav Parter, Nadav Kashtan, and Uri Alon. Facilitated variation: How evolution learns from past environments to generalize to new environments. *PLoS Computational Biology*, 4(11), November 2008.
- W. Rand and U. Wilensky. Netlogo articial neural net model.
- Thomas Ray. Evolution, ecology, and optimization of digital organisms. Santa Fe Institute working paper. August 1992.
- David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125{147, 2000.