Complex Adaptive Systems and Tierra

Vaibhav Mallya

March 4, 2009

1 Abstract

Artificial life is one of the most sought-after yet most-unfulfilled dreams of the computing era. What makes the search for A-life, and open-ended evolution in particular, such an important part of the search for a general theory of complex adaptive systems? In this paper, we dissect the well-known A-life simulation Tierra, and try to determine how and why it failed. We pose and attempt to answer the question: what might be necessary for a good solution to the problem to finally succeed?

2 Introduction

As the computer sciences have advanced, we have seen an increase in numerous predictive and adaptive technologies. Regardless, the hard A-life problem of open-ended evolution has remained relatively unfazed since most of these advances have been in information retrieval, data mining, and machine learning.

The study of complex adaptive systems is intimately linked to the search for artificial life in no small part because biological ecosystems are a classic case study in chaos, adaptation, and nonlinearity. One of the problems in studying real-world cas (indeed, in studying any chaotic system) is that making accurate predictions beyond the short or medium term is impossible simply because the state of the system cannot be perfectly observed. The small errors in measurement snowball rapidly and cause any predictive model to quickly spiral to inaccuracy. If we could simulate certain cas in their entirety *in silico*, we could gain perfect knowledge of state, and use this to validate potential unifying theories with a much higher degree of certainity. Additionally, we can exploit the fact that since we have total control over the simulated environment, we retain the ability to perturb the simulation in a way that allows us to test specific facets of the theory. This is obviously non-trivial to accomplish in, for example, financial [3] or biological systems.

Therefore, A-life is important because we can have total knowledge of the system at any given point in time. By observing interactions in a totally controlled digital system, we can eliminate the uncertainties in measurement that frustrate so many real-world models.

3 Description of Tierra

Tierra was an open-ended digital experiment conducted by ecologist Thomas Ray. It was derived from a modestly popular two-person programmer's game created during the 1980s called Core Wars. The premise of Core Wars is as follows:

- The playing field is a two-dimensional grid that represents a virtual machine's memory space
- Every square of the grid contains a Redcode instruction
- Players write a "warrior-program" in Redcode
- The grid is randomly populated with each player's warrior at once to begin the game
- The simulation then produces automatically and completely deterministically
- Warrior-programs are deterministically executed one after another, with players alternating
- The player whose warriors dominate the virtual memory space after some number of cycles wins.

Redcode [1] is a simple abstract RISC assembly language:

- Every instruction is a three-tuple of opcode, source address, destination address
- Minimal opcode set (10 total)
- In addition to add/subtract, notable opcodes include copy and move
- Impossible to create an invalid instruction in the underlying numeric representation
- Instructions are indivisible - no partial overwriting or copying

Although very basic, there is a surprising amount of complexity that can emerge very quickly when playing Core Wars. For instance, as a result of the instruction set and the rules of the game, polymorphism is the rule rather than the exception. Additionally, there have even been several implementations of the genetic algorithm to optimize Core Wars warriors. What Thomas Ray did with Tierra [2] was the following:

- Instead of being perfect, cloning operations randomly mutate the genotype
- A background "reaver" killed off programs that were deemed too old
- No players and no explicit notion of winning or losing

The initial grid was populated with a very simple 80-byte program that just duplicated itself. After a few thousand iterations, the small copy mutation rate caused small variants to appear that were largely identical. However, after several more generations, drastically different programs "evolved", including parasitic programs that were only 45 bytes long, extremely shorter programs that were parasites of the parasites, and so forth. This was clearly very exciting behavior, and seemed to point the way forward as far as A-life enthusiasts and researchers were concerned.

4 Discussion

Although Tierra's general approach may be promising, Tierra creatures themselves were found to have low overall complexity - the curve of interesting entities flattened out quickly with the parameters used by Ray. We focus primarily on Russel Standish's analyses of Tierra. [4]

There seem to be no local influence constraints in Tierra due to the fact that Redcode did not have them. That is, any cell can modify any other cell on any part of the board it wants. There is little sense of locality, and this obviously presents a massive problem for scaling the simulation. Any potential distributed algorithm suddenly has to communicate global state, eliminating whole classes of optimizations and slowing down potential traces significantly. Additionally, since there are clearly locality constraints in the real world, we lose significant structural simulation power.

Additionally, Standish puts forth a great deal of effort to determine the phenotypes of all the organisms in several trial runs of his simulation. Once a new genotype is found, there is a pairwise tournament with every other existing phenotype.

Phenotype, then, is equivalent to performance and behavior. This is in some ways quite thorough, but it is computationally difficult for even the relatively small arenas that Standish uses, despite the fact that some clever algorithms are employed. It would be outright intractable for more interesting numbers of organisms.

Even more disturbing is that even if there do exist "superorganisms" that are developed from unusual symbioses of existing organisms, they will never be detected by this method. There must surely be a better way to have a cheaper on-line complexity metric. [5] Perhaps some kind of simulated energy could be introduced into the system, and its status monitored on every step? If we cannot easily know the system's complexity, even if we do have a good A-life simulation, we might not know. We want to avoid the generalized pattern-recognition problem of "find the complexity in the haystack" at all costs.

How faithful of a structural simulation will we have to have before we make significant additional progress in this area? Our primary concern is that most such simulations have been conducted in only two dimensions, and with little idea of physics or external influence other than other organisms. Exceptions exist, but they have not had the power to scale and breed organisms at the same level of complexity as Tierra or some of the other simple simulations.

One point to note: Tierra and its cousins have an *explicit* way to reproduce or breed. Would it be fruitful to devise a system in which the breeding process itself was "emerged"? This would mean a much lower-level simulation of "molecules" or "forces". Might we be able to emerge a more novel set of breeding methods than we currently know about in the real world? Dealing with the problem of where to draw the boundaries for the simulation of the system is persistent in cas. Naturally, this is no different.

Finally, as has been hinted at, we continually face problems of scale. Thankfully, we might finally be able to chip away at this problem with advances in distributed and cloud computing. However, it would require a non-trivial amount of engineering. Thomas Ray has been attempting for several years to create a distributed version of Tierra called Network Tierra, but progress has been stagnant for some time now; he simply does not have the domain-specific knowledge that a distributed systems engineer or large-scale game designer might have.

5 Conclusion

When the Tierra results were released, there was a flurry of press coverage proclaiming the radical second coming of artificial life - and indeed, the behavior exhibited by the modified Core Wars engine is really quite fascinating. Upon closer empirical inspection however, the creatures produced by Tierra at any reasonable scale seem to have a fairly limited information-theoretic complexity. However, the incremental, yet subtle, progress in the area has been good. A-life's importance to cas as a theoretical tool is significant, and with any luck, one of the numerous projects hoping to improve on the approaches of Tierra and its cousins will yield even more interesting results in the near future.

References

[1] Ilmari Karonen. Beginners guide to redcode. `http://vyznev.net/corewar/guide.html`.

[2] mathew. Get a-life. `http://www.infidels.org/library/modern/meta/getalife/coretierra.html`.

[3] Felix Salmon. Recipe for disaster: The formula that killed wall street. `http://wired.com/techbiz/it/magazine/17-03/wp_quant`.

[4] Russell K. Standish. Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3:2003, 2003.

[5] Hubert P. Yockey. Information theory, evolution and the origin of life. *Inf. Sci. Inf. Comput. Sci.*, 141(3-4), 2002.