A Success Story or an Old Wives' Tale? On Judging Experiments in Evolutionary Computation

In- or Out-of-Species Dilemma

BY MOSHE SIPPER

Moshe Sipper is a Senior Researcher in the Logic Systems Laboratory at the Swiss Federal Institute of Technology in Lausanne, Switzerland.

Consider the following two hypothetical endings of a paper in the field of evolutionary computation:

> Ending I: In this paper we presented a novel evolutionary algorithm, applying it to the problem of evolving flying robotic pigs. We performed a total of 1000 evolutionary runs, 853 of which were successful in that they gave rise to high-fitness solutions, that is, pigs that fly.
>
> Ending II: In this paper we presented a novel evolutionary algorithm, applying it to the problem of evolving flying robotic pigs. We performed a total of 1000 evolutionary runs, one of which was successful in that it gave rise to a high-fitness solution, that is, a pig that flies.

I believe most referees would tend to eye with suspicion Ending II, leaning toward rejection of the paper on the grounds of "statistical insignificance." Ending I, on the other hand, would probably garner the referees' approval: the approach put forward by the authors has been tested on an interesting problem, with *most runs* producing good solutions.

This little tale of mine epitomizes a common attitude among practitioners who use evolutionary-computation techniques as an *engineering tool*. Though such an attitude is not altogether mistaken, I argue that there is more here than meets the eye.

Generally speaking, engineering involves the design and manufacture of products that are useful to people (NB: I use the term "engineering" in the broad sense, that is, encompassing related fields such as computing science and applied mathematics). Faced with a problem to solve—such as designing and constructing robotic pigs that fly—the engineer must come up with a *good solution*. This latter term actually caches two distinct criteria, which involve two separate questions: (1) engineer a solution (does the product work?), (2) that is good (does the product work well?).

Toward these two ends, an engineer seeks to employ a methodology that is *robust*; by this I mean that he can use the given approach to solve a wide range of problems, whose parameters can be tweaked, twisted, turned, and tuned. I should like to bring this point home by way of a classical example from civil engineering, that of designing a bridge. The objective (and object) in this case is quite clear: erecting a structure between point A and point B. The constraints—some implicit, some explicit (these latter are usually called specifications)—are multitudinous: the bridge must be stable, it must support a given maximal weight, it must have a certain width, it must not exceed a certain cost, and so on (the wish list can often be quite long).

Civil engineers have developed over the years well-honed methodologies for designing bridges that meet specifications. And these methodologies are robust in that they apply to a wide range of bridges: short, long, narrow, wide, cheap, expensive, footbridges, drawbridges, suspension bridges, and truss bridges. Thus, a civil engineer need not reinvent the wheel every time he is faced with a new bridge-design problem; when asked to design a bridge, he *has little doubt* that the task is doable.

As evolutionary-computation practitioners we seek nature's boons, but will not hear of her banes; we want to eat the cake and have it too, using a (simulation of a) messy and failure-prone process as a reliable and robust engineering methodology.

But just how little is his doubt? To wit, out of 1000 bridges he is asked to design, how many are feasible using current wisdom: 1000? 853? 1? Engineering practice dictates that a valid design approach must be able to handle a majority of the cases thrown at it—one does not wish to constantly reinvent the wheel. This dictum is quite sensible and often works beautifully (cf. modern civilization). Nature's evolutionary engineering, on the other hand, fails more often than not; as succinctly put by Richard Dawkins: "however many ways there may be of being alive, it is certain that there are vastly more ways of being dead, or rather not alive." [1] (page 9).

As evolutionary-computation practitioners we seek nature's boons, but will not hear of her banes; we want to eat the cake and have it too, using a (simulation of a) messy and failure-prone process as a reliable and robust engineering methodology. How does one bridge this seemingly abyssal nature-to-engineering gap? Have we been too stringent all along, demanding of our ersatz simulacra to outdo the grand mother (nature) herself? The answer, I argue, is both yes and no.

In *The Structure of Scientific Revolutions* Thomas Kuhn famously put forward the thesis that scientists work mostly within a given *paradigm*, generally devoting themselves to solving "puzzles"—problems whose solutions reinforce and extend the scope of the paradigm rather than challenge it [2]. But there are always anomalies, phenomena that the paradigm cannot account for or that directly contradict it. These anomalies are often swept aside, but if they accumulate, they may trigger a revolution—a paradigm shift in which scientists abandon the old paradigm for a new one.

Engineers, like scientists, mostly work within a given paradigm, and in doing so they expect the paradigm to be robust.[1] From time to time technological breakthroughs are achieved, which bring about engineering paradigm shifts (e.g., transistors replacing vacuum tubes); indeed, to the extent that technological breakthroughs are harbingers of engineering revolutions, they could be considered as analogues of anomalies in science.

When applying evolutionary computation as an engineering tool within a given paradigm, where a lore of (good) practice already exists (e.g., evolving bridges \cite funesPollack98) [3]—then we expect this tool of ours to work most of the time and not just once in a blue moon. However, as we stray off the beaten paradigm track, success becomes less a straight-A student and more a one-off Einstein. Can this be the solution to our nature-to-engineering problem? More precisely, does Ending I typify an in-paradigm problem, where we seek high success rates (straight As), whereas Ending II typifies an out-of-paradigm problem, where we aim for that rare, one-in-a-thousand success? While the notion of "paradigm" does shed light on the nature-to-engineering gap, it does not close it. The problem is that in-paradigm problems may rightfully give rise to one-off success stories. I will demonstrate this with an example in the area of electronic-circuit evolution.

Working within the paradigm of classical, digital-circuit design, Miller and Thomson [4] evolved arithmetic circuits, such as two- and three-bit binary multipliers. They wrote that "Arithmetic

[1] *In a postscript written in 1969, Kuhn cites one of his readers who concluded that the term "paradigm" is used in at least 22 different ways in* The Structure of Scientific Revolutions. *I shall thus avoid providing a 23rd definition of my own, leaving the reader to draw his own nebulous understanding of the term.*

circuits are interesting . . . since there are well-known conventional designs with which the evolved solutions can be compared" [4]. Thompson and his colleagues [5], on the other hand, wished to transcend the boundaries of the digital paradigm, exploring unconventional and unconstrained evolution. They showed that "Evolutionary algorithms can explore some of the regions in design space that are beyond the scope of conventional methods" [5].

The crux of the matter, I argue, is that in both cases a one-in-a-thousand success rate would be entirely justifiable (in fact, these rates were higher in both works). In Thompson's out-of-paradigm work this statement of mine is obvious: When one boldly goes where no electronic-circuit designer has gone before, one success—an astonishingly novel circuit, in this case—would serve to justify the effort. Yet this is equally true in Miller and Thomson's in-paradigm work: One novel circuit design is all it takes; finding a single two-bit multiplier, which—while respecting the conventional digital-paradigm constraints—is nonetheless better in some respect (e.g., faster, smaller, more energy-efficient), could mean riches beyond imagination (or at least a delightful publication . . .). The in-paradigm/out-of-paradigm criterion does not, then, resolve the issue—it does not differentiate the one-off problems from the straight-A ones.

Taking my cue from nature, I believe that the dividing line has to do with *species* rather than paradigms; we should distinguish between *in-species* problems and *out-of-species* ones. Though my use of the term "species" as related to artificial objects is stripped of most of its original biological meaning, there are certain abstract analogies that still hold.

When designing conventional bridges, we are interested in finding a good *individual* within a well-known "species": bridges; this is what I refer to as an in-species problem. In this case, the evolutionary algorithm (or any other algorithm for that matter) must exhibit high success rates, that is, a majority of the runs must end with good solutions. Moreover, we must be able to repeatedly apply the same methodology, with little or no tuning of the algorithm (robustness).

What happens, however, if one is asked to design a totally new kind of bridge—a new *species* of man-made objects? More generally, what happens when one is faced with the design of entirely new lines (or lineages) of products—new genera, families, orders, classes, phyla, or kingdoms? In this case, we shall be content even with a one-off success—after all, we are seeking an object the likes of which no one has designed before.

There is an interesting analogy with nature's in-species and out-of-species success rates. By definition, a large percentage of individuals within a given species survive (else the species would die out). Much lower success rates are exhibited at the out-of-species level: Most species, in fact, die out, nipped in the bud of their incipient, precarious existence. On this issue Darwin wrote:

> And of the species now living, very few will transmit progeny of any kind to a far distant futurity; for the manner in which all organic beings are grouped, shows that the greater number of species of each genus, and all the species of many genera, have left no descendants, but have become utterly extinct. We can so far take a prophetic glance into futurity as to foretell that it will be the common and widely spread species, belonging to the larger and dominant groups, which will ultimately prevail and procreate new and dominant species. [6, Chapter 14]

The line between in-species problems and out-of-species ones is obviously not dichotomous but continuous. It must be understood that being in-species does not at all imply that the problem is easy to solve, as evidenced by the vast corpus of works in the field (an excellent online repository of information can be found at www.genetic-programming.org). My contention is, rather, that both types of problems appear in the practice of evolutionary computation, and we often confound the in-species with the out-of-species, asking for straight As where a one-off would be (almost) miraculous.

Let me recap by way of another example: evolving behaviors such as obstacle avoidance and homing navigation in Khepera robots (e.g, Ref. 7) would, in my mind, be considered in-species, thus befitting of Ending I. Evolving from scratch a new phylum of flying pigs within the kingdom of robots would unarguably be an out-of-species experiment, where even a one-in-a-million success rate would be an amazing feat; Ending II, in this case, would be entirely appropriate (and publishable).

Acting as referee, how would you go about establishing whether the paper in question is in-species or out-of-species? As I mentioned earlier, the dividing line is quite fluid, so it is time I invoked the magical "beyond-the-scope-of-this-paper" incantation—my aim herein has not been to peruse the art of refereeing but to point out a fundamental issue that merits attention by evolutionary-computation practitioners. Alas, I must leave the (refereeing) reader to his or her own devices when it comes to applying my distinction judiciously upon sitting in judgment.

REFERENCES

1. Dawkins, R. The Blind Watchmaker; W.W. Norton: New York, 1986.
2. Kuhn, T.S. The Structure of Scientific Revolutions; University of Chicago Press: Chicago, 1962.
3. Funes, P.; Pollack, J. Artificial Life 1998, 4, 337–357.
4. Miller, J.F.; Thomson, P. Aspects of Digital Evolution: Geometry and Learning, in Proceedings of the Second International Conference on Evolvable Systems: Sipper, M.; Mange, D.; Pérez-Uribe, A. (Eds.). Lecture Notes in Computer Science 1998, 1478, 25–35.
5. Thompson, A.; Layzell, P.; Zebulum, R.S. IEEE Transactions on Evolutionary Computation 1999, 3, 167–196.
6. Darwin, C. On the Origin of Species; John Murray: London, 1859.
7. Floreano, D.; Mondada, F. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 1996, 26, 396–407.