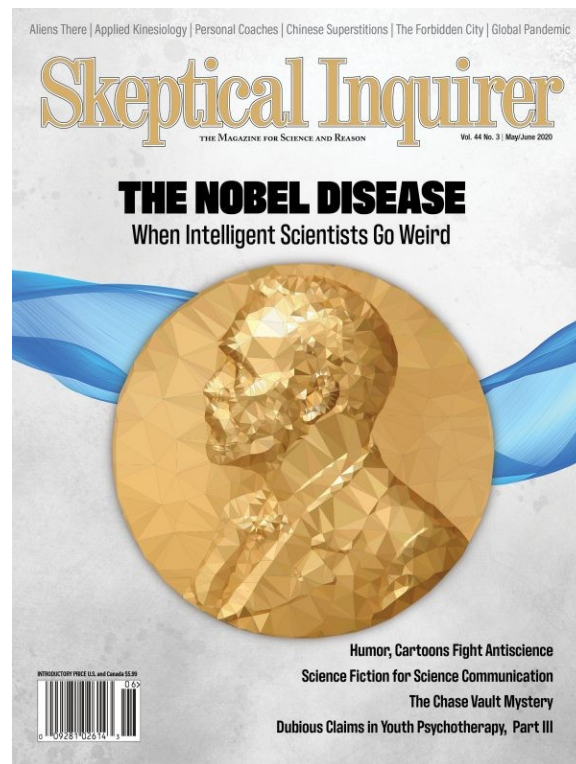


ID Movement Still Stymied By Genetic Algorithms: War Of The Weasels
Update
David E. Thomas
From: Skeptical Inquirer
Volume 44, No. 3
May / June 2020



I last wrote about the “War of the Weasels,” the ongoing Creationist attacks on Evolutionary (Genetic) Algorithms, in the May/June 2010 issue of Skeptical Inquirer (Thomas 2010). Genetic Algorithms (GAs) are computerized simulations of evolution, and are used to study evolutionary processes, and also to solve difficult engineering or math problems. Intelligent Design (ID) Creationists often criticize these algorithms for not generating true novelty, and routinely claim that the “answer” is surreptitiously introduced into the program via the algorithm's fitness testing functions, or via “active information” surreptitiously added by the programmer. The example creationists always cite to prove that GAs must be spoon-fed their solutions is Richard Dawkins's “Weasel” simulation from *The Blind Watchmaker* (1986). This algorithm, used to show the power of cumulative selection, does include a precise description of its intended “Target,” the phrase “METHINKS IT IS LIKE A WEASEL” (Hamlet), which is used all through the execution of the procedure. However, Dawkins himself stated in his exposition that evolution itself does not have a goal or a target – unlike his “Weasel.”

In 2001, I set about to challenge this strawman argument by developing a Genetic Algorithm that solved problems without any knowledge of the answers in advance. I chose “Steiner's Problem”: given a two-dimensional set of points, what is the most compact network of straight-line segments that connects the points? (Additional “Steiner Points” besides the fixed points are allowed.) I presented this

work at a lecture following a talk by Intelligent Design founder Phillip Johnson at UNM, and continued working on this problem thereafter.

In the summer of 2006, I posted several articles on this topic on the Panda's Thumb blog (Thomas 2006a). The series of posts culminated in a public "Design Challenge" (Thomas 2006b), in which readers were given a week to submit answers for a tricky six-point Steiner system. Because the ID crowd was saying the solutions were known beforehand, I posted all my source code, and challenged them to find the actual Steiner solution for the 6-point problem; it was an open-book test. Ironically, the ID "theorist" who was complaining the loudest, Salvador Cordova, was unable to derive the actual answer to the problem, even after many days of effort. The actual answers (two were possible) were found by my genetic algorithm in a few out of hundreds of runs, and also by dozens of independent fans of math and evolution. I was very surprised by the answers myself; I had expected a simpler solution, but this turned out to be inefficient. That's why I chose this particular problem for the public "Design Challenge" in the first place: I didn't know the actual answer, and could not possibly have "smuggled it in" (see Figure 1).

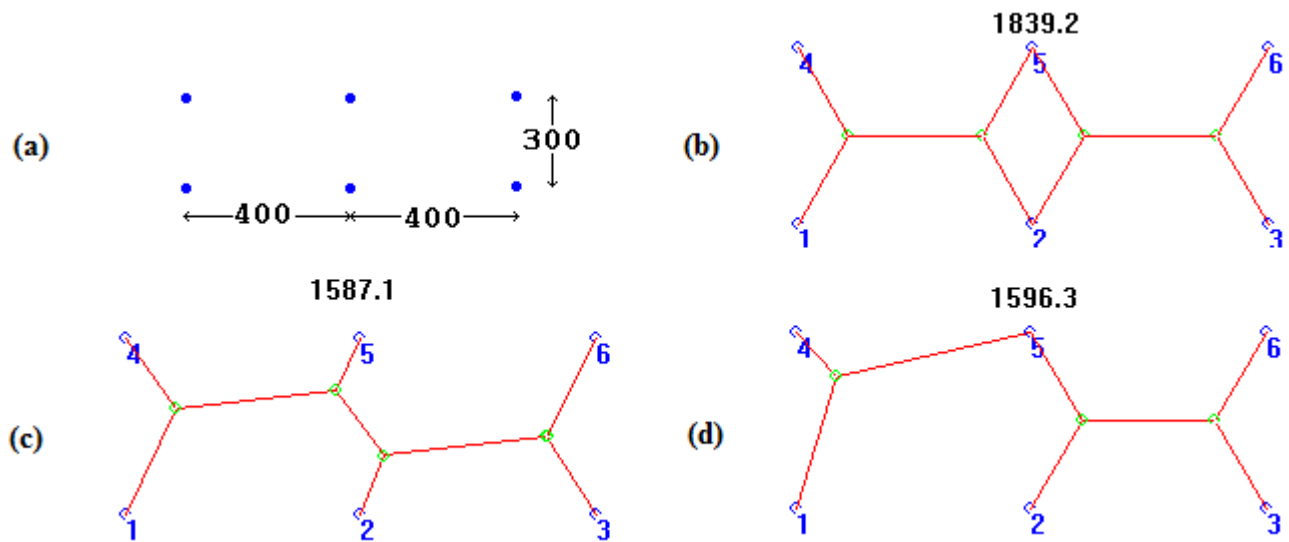


Figure 1: (a) The Design Challenge as posted on Panda's Thumb; (b) the "Double Bowtie," the shape the author expected as the solution to the challenge; (c) the actual best (Steiner) Solution, with a slightly twisted, but more minimal, arrangement of segments, and (d) a second-best "quasi" solution, which was the best that could be attained by ID theorists after weeks of effort. Because the actual Steiner shape was unknown to the programmer, there was no way the solution could have been added via "Active Information."

So, after all these years, how has the ID community responded? The ID community has by and large ignored the point of the Steiner Challenge, which was simply that most Genetic Algorithms (Dawkins' "Weasel" excluded) do not require the explicit answers it is hoped the algorithm will provide.

The mathematical stalwarts of the ID movement, William Dembski, Robert Marks and Winston Ewert (both still at Baylor, unlike Dembski, who was let go), have since responded to my article on the Steiner GA. Amazingly, they are all still painting all Genetic Algorithms with the "Weasel" brush, but any relevance to Dawkins or evolution science is becoming harder and harder to perceive.

The gist of the Ewert, Dembski and Marks response to the Steiner GA (Ewert et. al. 2012) is that “active information” is being supplied by the programmer to derive the answer. They say “The Darwinist claim is that no such assistance is required. Rather, natural selection is innately capable of solving any biological problem that it faces.” But that is clearly a strawman argument: have the stalwarts of ID never heard of ... extinction?

In another paper published in 2014 in Biocomplexity, (Ewert 2014), Ewert attacks several GAs, including my Steiner algorithm. Ewert makes two huge errors in this part of his paper. He declares that the Steiner solutions are not “Irreducibly Complex”, because a different method of connecting points, the much simpler Minimum Spanning Tree (MST) algorithm, can easily connect the dots. He says “A connected network can be achieved by random chance alone. The difficulty in the Steiner tree problem is in trying to minimize the amount of road used, not in getting a connected network.” But the Steiner networks are not simple, and are clearly Irreducibly Complex: remove or alter any segment, and the network is no longer connected, as shown in Figure 2.

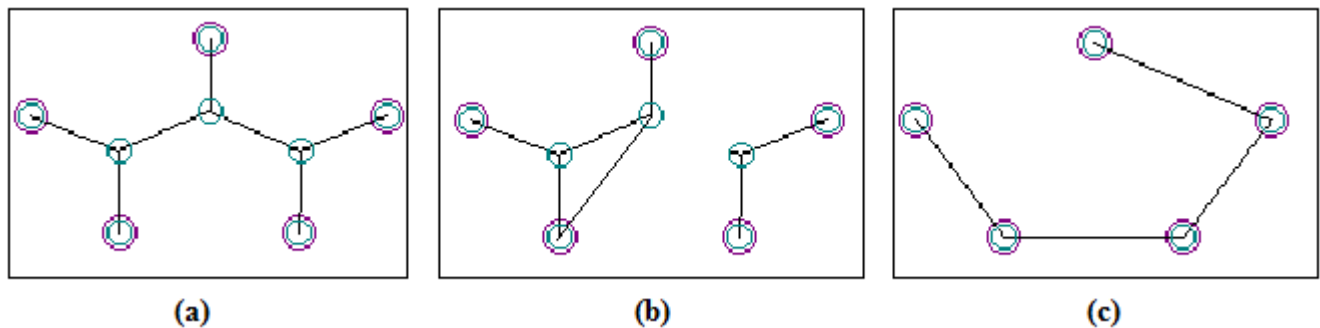


Figure 2. (a) This Steiner Solution for 5 Points is the actual Minimal Length Network. (b) This solution is Irreducibly Complex: if any segment is removed, or even re-routed, the connectivity becomes invalid. (c) A Minimum Spanning Tree for the same 5 Points; while this connects the points, it clearly does not have the complexity of the actual shortest network.

The second gaffe is use of the ID concept of “Specified Complex Information” (CSI). While the CSI concept is usually presented as “obvious” (the solutions must be complex, and specified as well), the mathematical definition of CSI has been carefully crafted so as to make the success of either evolution or GAs absolutely impossible. Dembski et. al. have defined CSI as the property of having “500 bits of complexity”. If the Genetic Algorithm under consideration always gets the answer to the posed problem, it thus has ZERO CSI (Dembski 2002). Even Dawkins’ “Methinks it is like a Weasel” has obvious complexity, but Dembski scores it as zero CSI, because the Dawkins algorithm always converges. If, like my Steiner algorithm, the GA does get the correct answer, say, only once in about 200 trials, it has less than 8 Bits of CSI ($2^8 = 256$). Only if the Genetic Algorithm gets the solution rarely (literally, once in $2^{500} \sim 10^{150}$ trials), does it finally achieve the honor of possessing “500 Bits of CSI.” This is as rare as tossing a fair coin 500 times, and getting heads every time. The Game is RIGGED: Neither Genetic Algorithms nor Evolution can ever create CSI! The creationist assault on GAs is interesting, but falls far, far short of the mark.

Dawkins, Richard. 1986. The Blind Watchmaker. New York: W.W. Norton and Company.

Dembski, W. 2002: No Free Lunch: Why Specified Complexity Cannot be Purchased Without Intelligence, Lanham, MD: Rowman & Littlefield.

Ewert, W. 2014: Digital irreducible complexity: A survey of irreducible complexity in computer simulations. BIO-Complexity 1:1–10. doi:10.5048/BIO-C.2014.1. Available online [here](#)

Ewert, W., W. Dembski, Marks and R. J. Marks II, 2012. Climbing the Steiner tree— Sources of active information in a genetic algorithm for solving the Euclidean Steiner tree problem. BIO-Complexity 1: 1-14. doi:10.5048/BIO-C.2012.1. Available online [here](#).

Thomas, David. 2006a. Target? TARGET? We don't need no stinkin' Target! Panda's Thumb Blog (July 5). Available online [here](#).

----- 2006b. Design Challenge Results: Evolution is Smarter than You Are. Panda's Thumb Blog (August 21). Available online [here](#).

----- 2010. The War Of The Weasels: Or How an Intelligent Design Theorist was Bested in a Public Math Competition by a Genetic Algorithm! Skeptical Inquirer 34(3) (May/June)