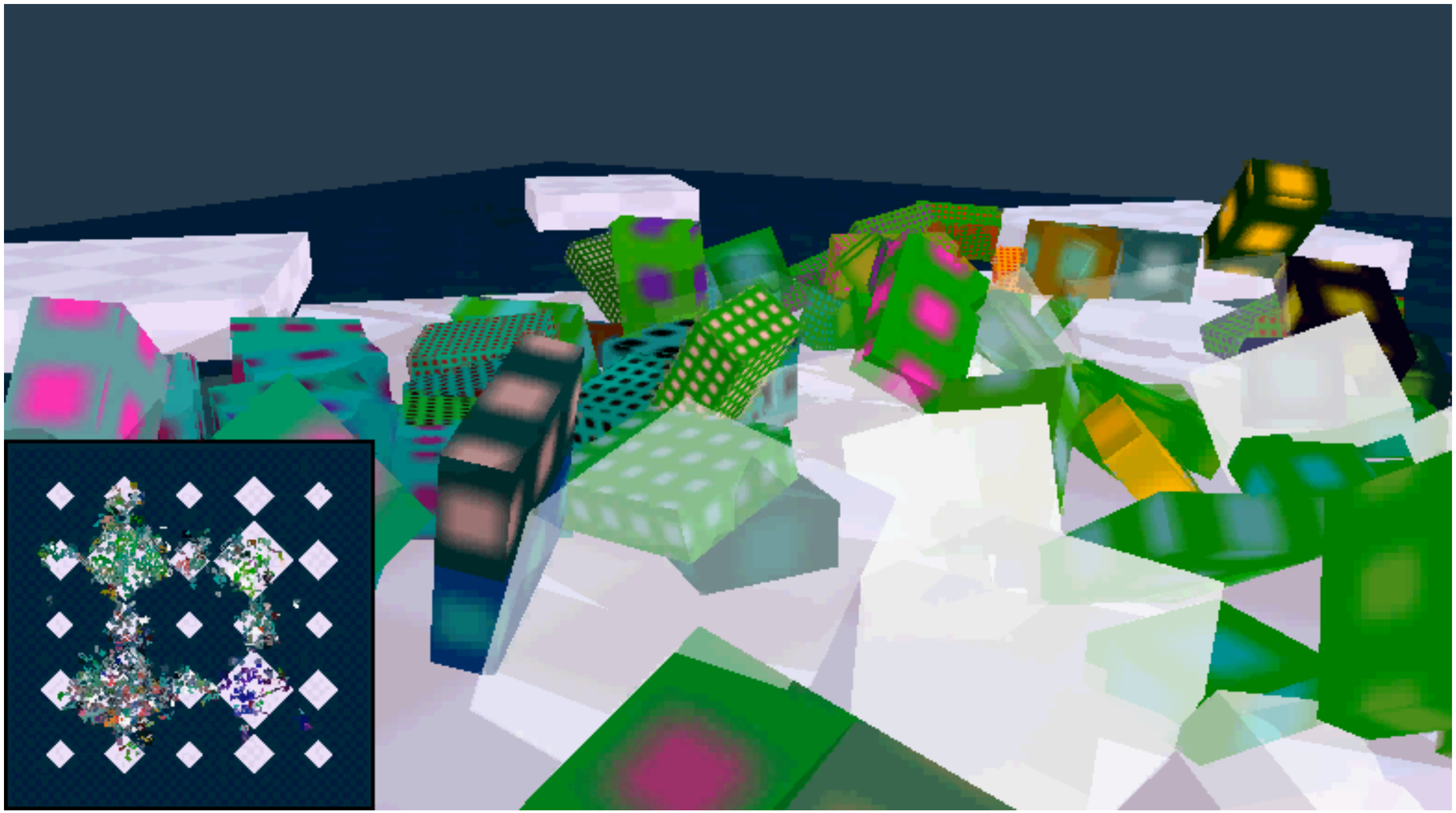
Intelligence Evolving

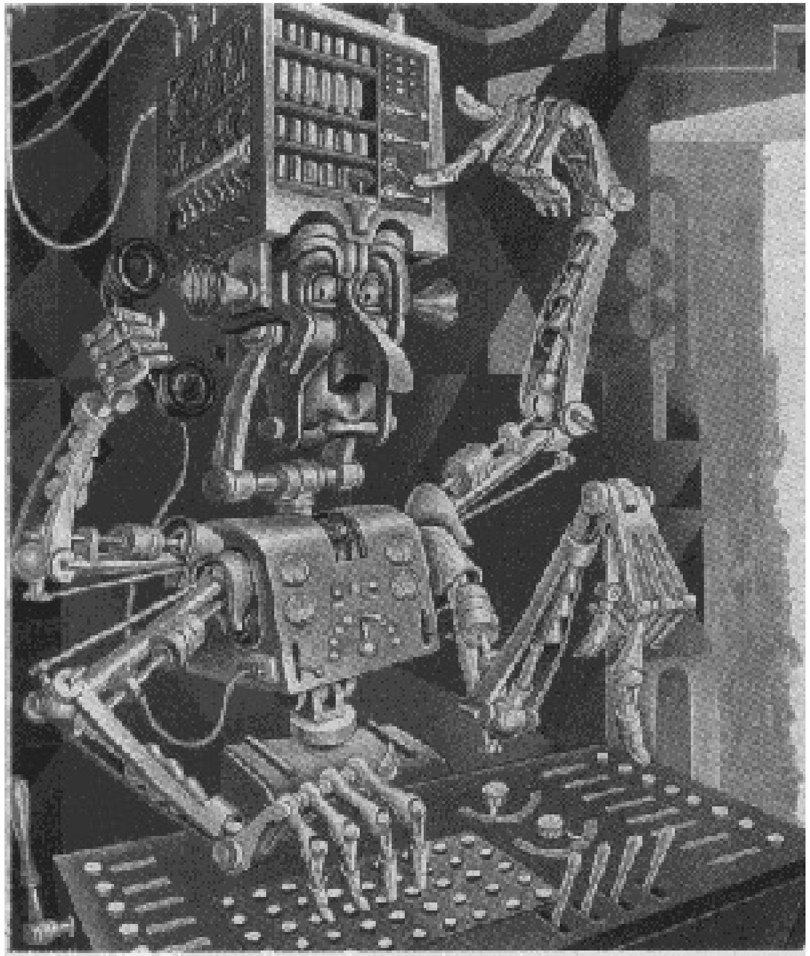


Lee Spector

Professor of Computer Science, School of Cognitive Science, Hampshire College

This work was supported by NSF Grant No. 0308540

Outline



Executive of the Future, 1952

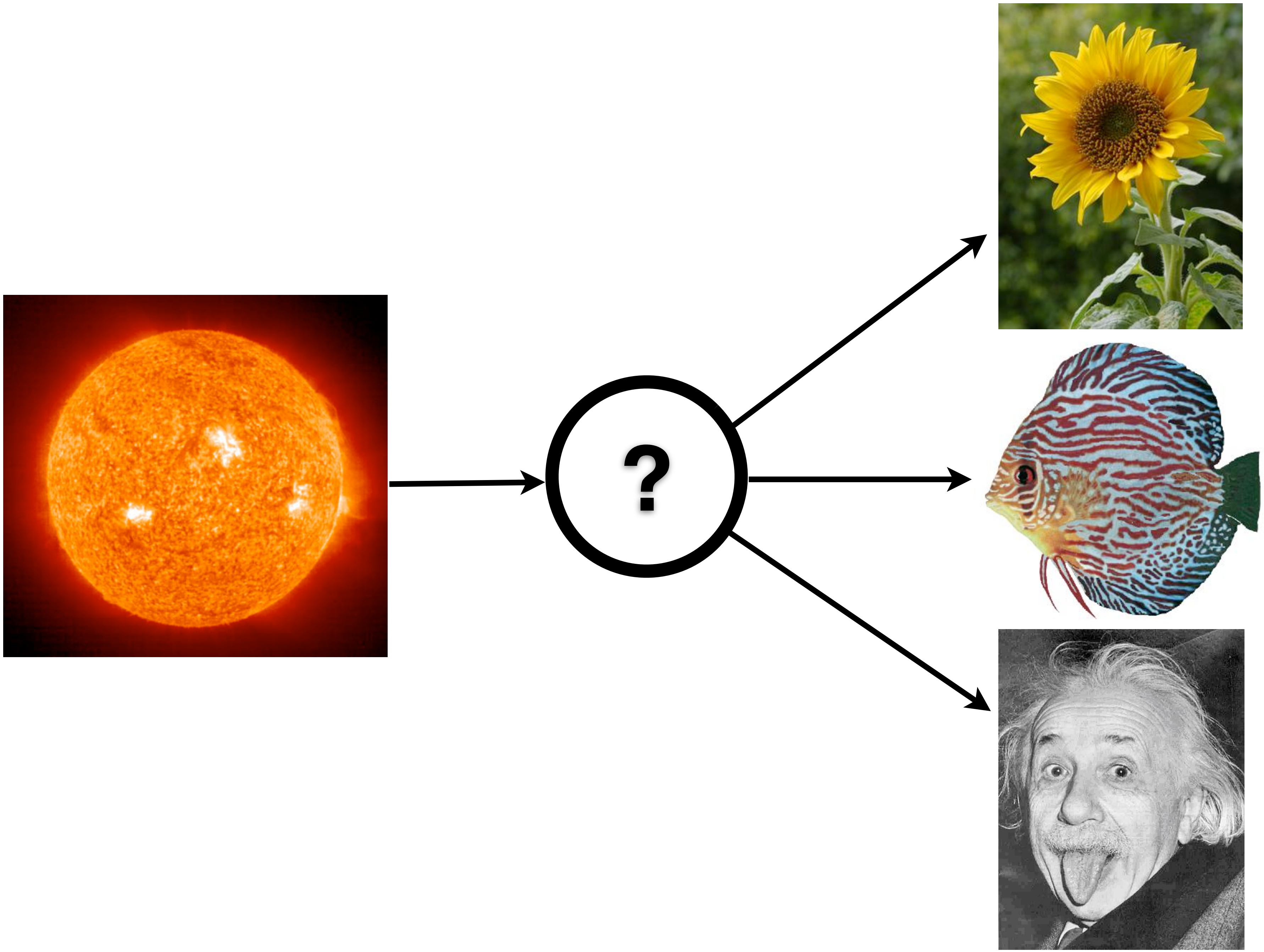
- Artificial intelligence
- Evolutionary computing
- Science, engineering, and art
- Artificial life
- Evolution, the designer

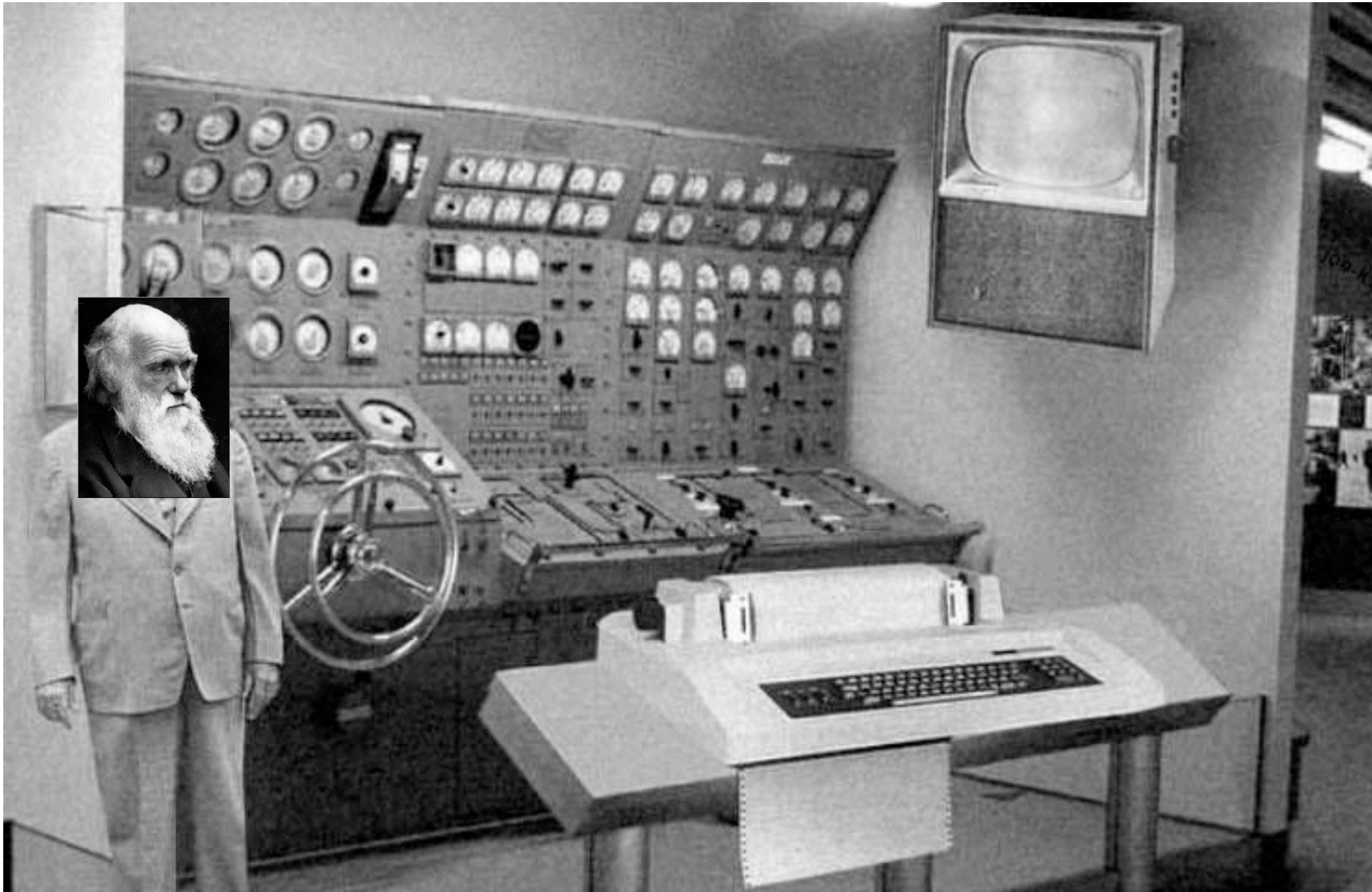
What can make a mind?

- Socrates: explicit **rules** to govern behavior
- Aristotle: **categories**, **inference**
- Leibniz: reasoning as **calculation**
- McCulloch & Pitts: artificial **neurons**
- Selfridge: neural **networks**
- Wiener: **feedback** loops
- Shannon: **search**
- Newell, Shaw & Simon: **means/ends** analysis
- Samuel: learning **strategy**
- McCarthy: **common sense** knowledge
- Quillian: semantic **networks**
- Shortliffe: **rules**
- Shank: **scripts**, **plans**, & **goals**
- Brooks: **reflexes**

What can make what can make a mind?

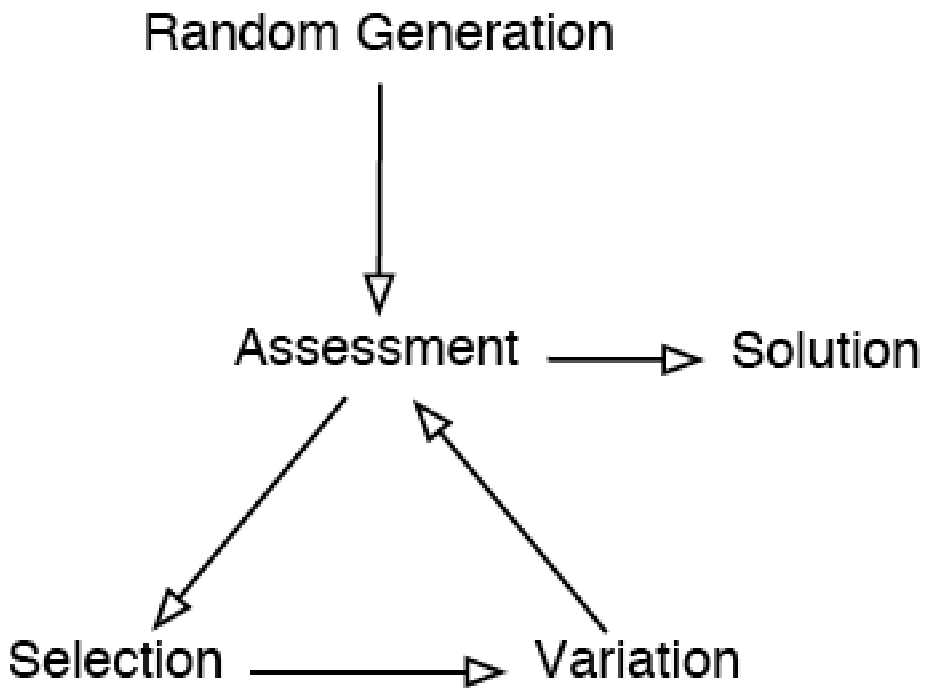
- Learning
- Development
- Evolution





Scientists from the RAND Corporation have created this model to illustrate how a "home computer" could look like in the year 2004. However the needed technology will not be economically feasible for the average home. Also the scientists readily admit that the computer will require not yet invented technology to actually work, but 50 years from now scientific progress is expected to solve these problems. With teletype interface and the Fortran language, the computer will be easy to use.

Evolutionary Algorithms



Genetic Programming

- Evolutionary algorithm in which the candidate solutions are executable computer programs.
- Candidate solutions are assessed, at least in part, by executing them.

Mutating Lisp

```
(+ (* X Y)
   (+ 4 (- Z 23)))
```

```
(+ (* X Y)
   (+ 4 (- Z 23)))
```

```
(+ (- (+ 2 2) Z)
   (+ 4 (- Z 23)))
```

Recombining Lisp

Parent 1: `(+ (* X Y)`
`(+ 4 (- Z 23)))`

Parent 2: `(- (* 17 (+ 2 X))`
`(* (- (* 2 Z) 1)`
`(+ 14 (/ Y X))))`

Child 1: `(+ (- (* 2 Z) 1)`
`(+ 4 (- Z 23)))`

Child 2: `(- (* 17 (+ 2 X))`
`(* (* X Y)`
`(+ 14 (/ Y X))))`

Symbolic Regression

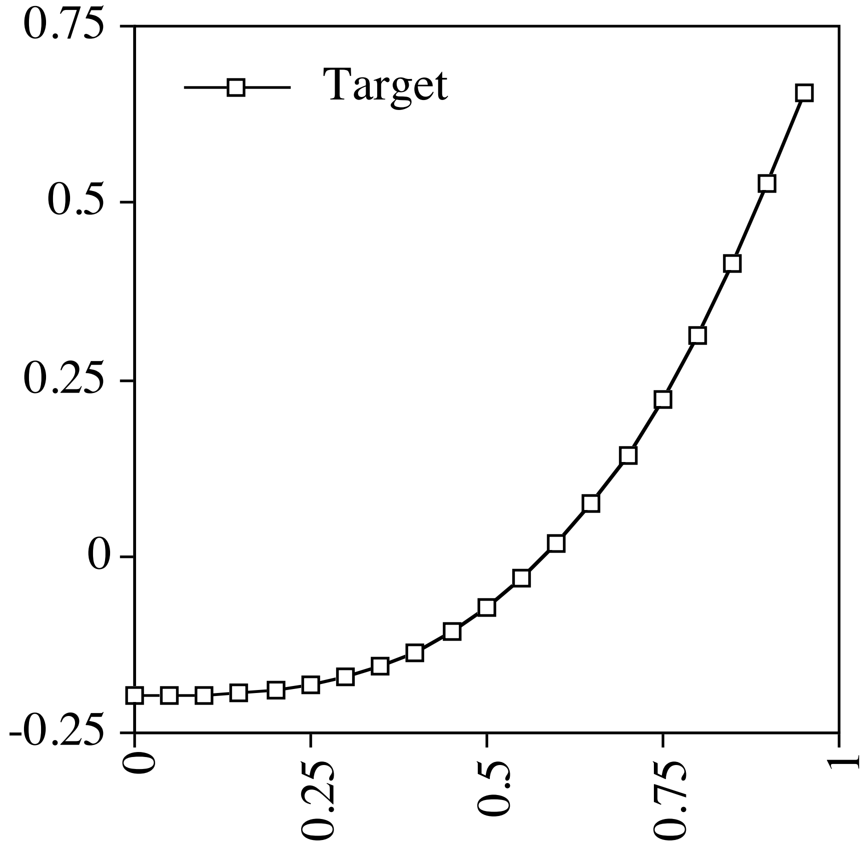
Goal: given a data set of (x,y) pairs, produce a program that takes an x value as input and produces the appropriate y value as output.

Function set: {+, -, *, %}

Terminal set: {X, 0.1}

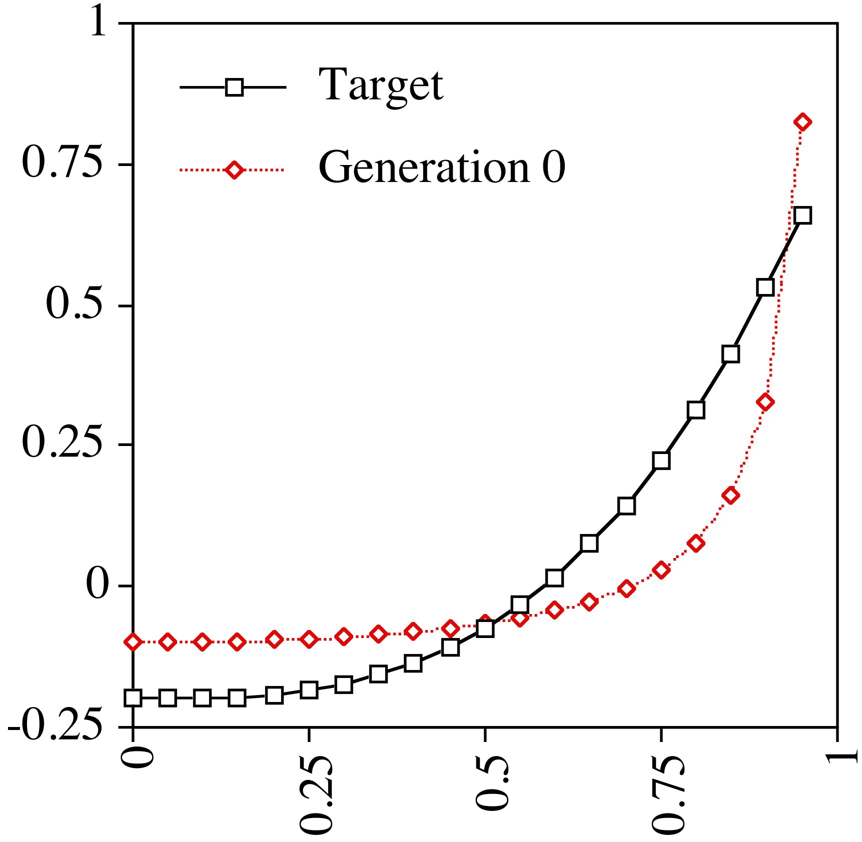
Fitness function: sum the error for X values 0.0, 0.1, ..., 0.9

Evolving $y = x^3 - 0.2$



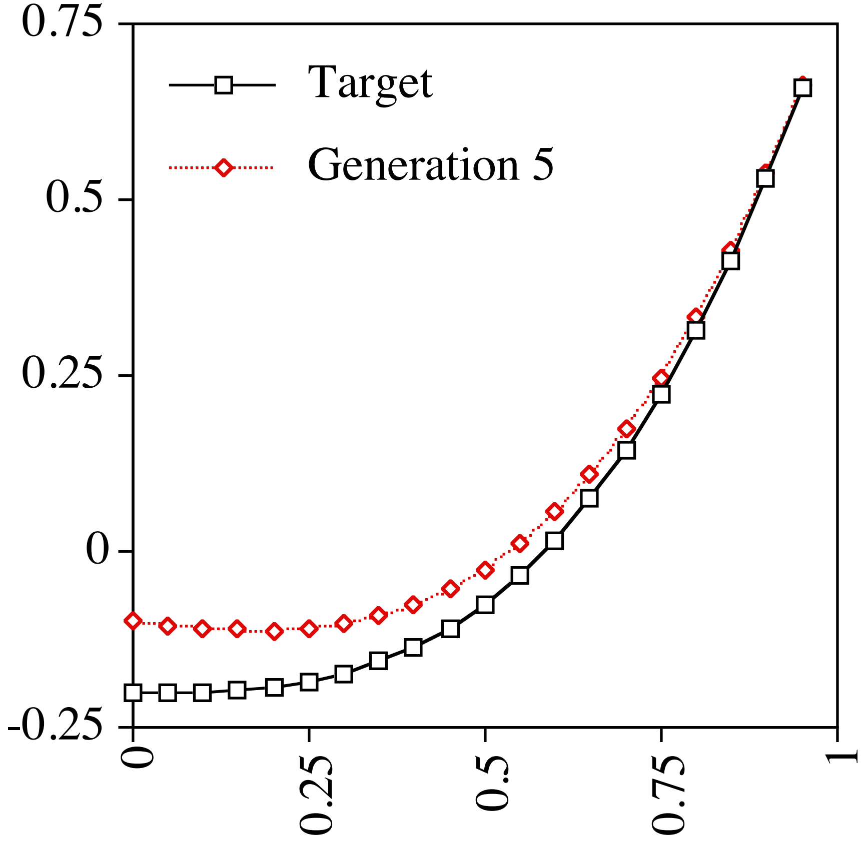
Best Program, Gen 0

```
(- (% (* 0.1
         (* X X))
      (- (% 0.1 0.1)
         (* X X)))
   0.1)
```



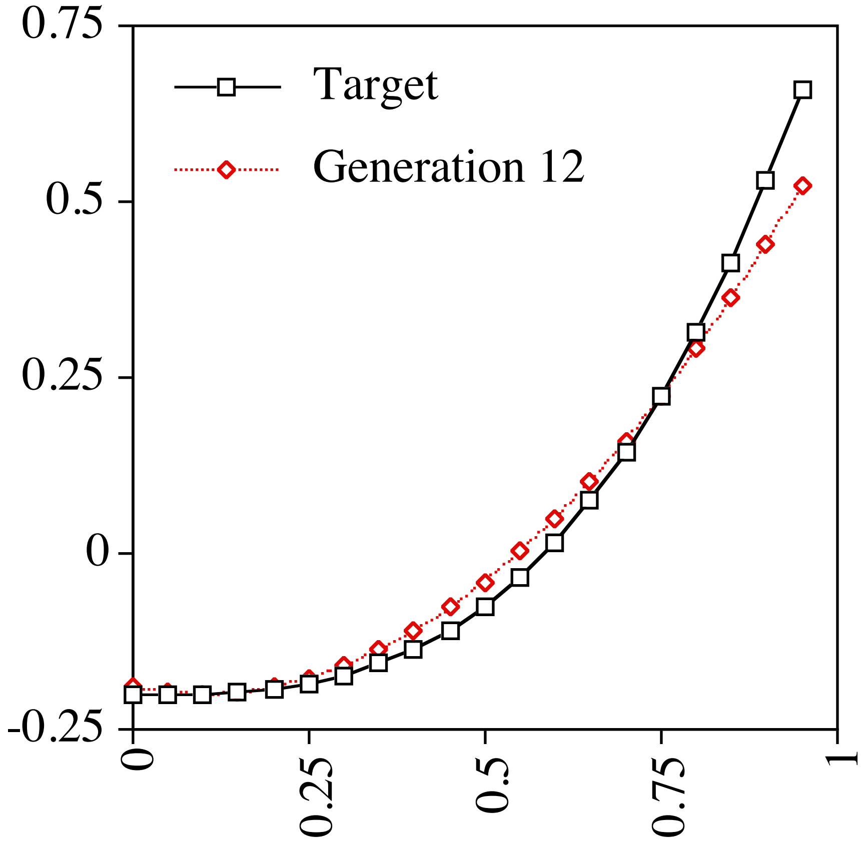
Best Program, Gen 5

```
(- (* (* (% X 0.1)
         (* 0.1 X))
      (- X
         (% 0.1 X)))
   0.1)
```



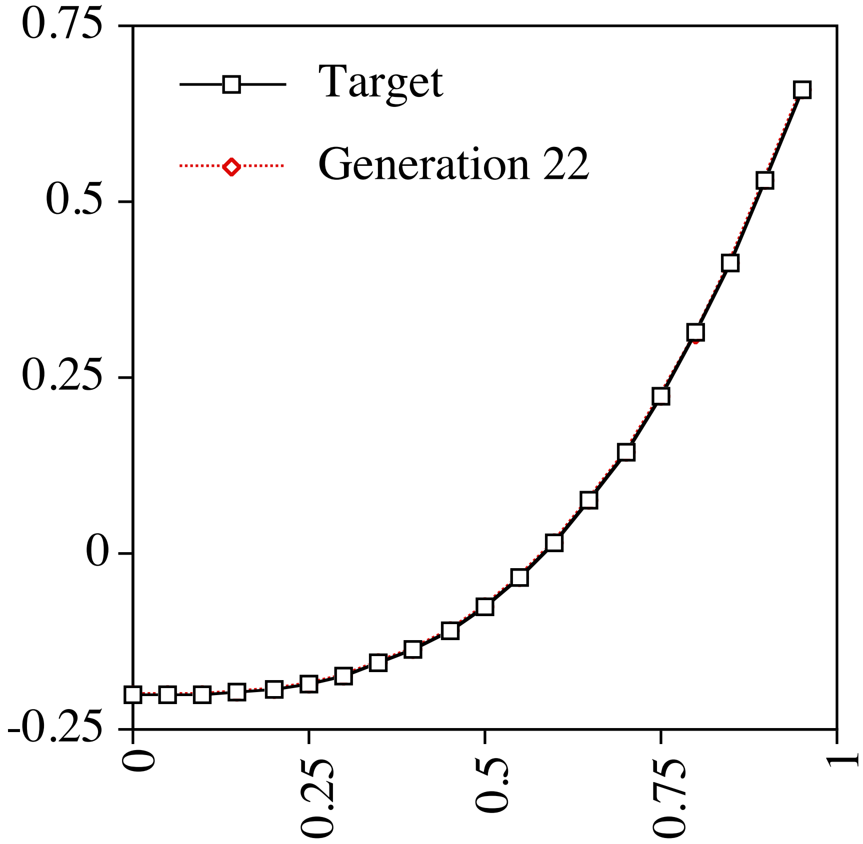
Best Program, Gen 12

```
(+ (- (- 0.1
         (- 0.1
            (- (* X X)
               (+ 0.1
                  (- 0.1
                     (* 0.1
                        0.1))))))
      (* X
         (* (% 0.1
               (% (* (* (- 0.1 0.1)
                        (+ X
                           (- 0.1 0.1)))
                     X)
                  (+ X (+ (- X 0.1)
                          (* X X)))))
            (+ 0.1 (+ 0.1 X))))))
   (* X X))
```



Best Program, Gen 22

```
(- (- (* X (* X X))
      0.1)
   0.1)
```



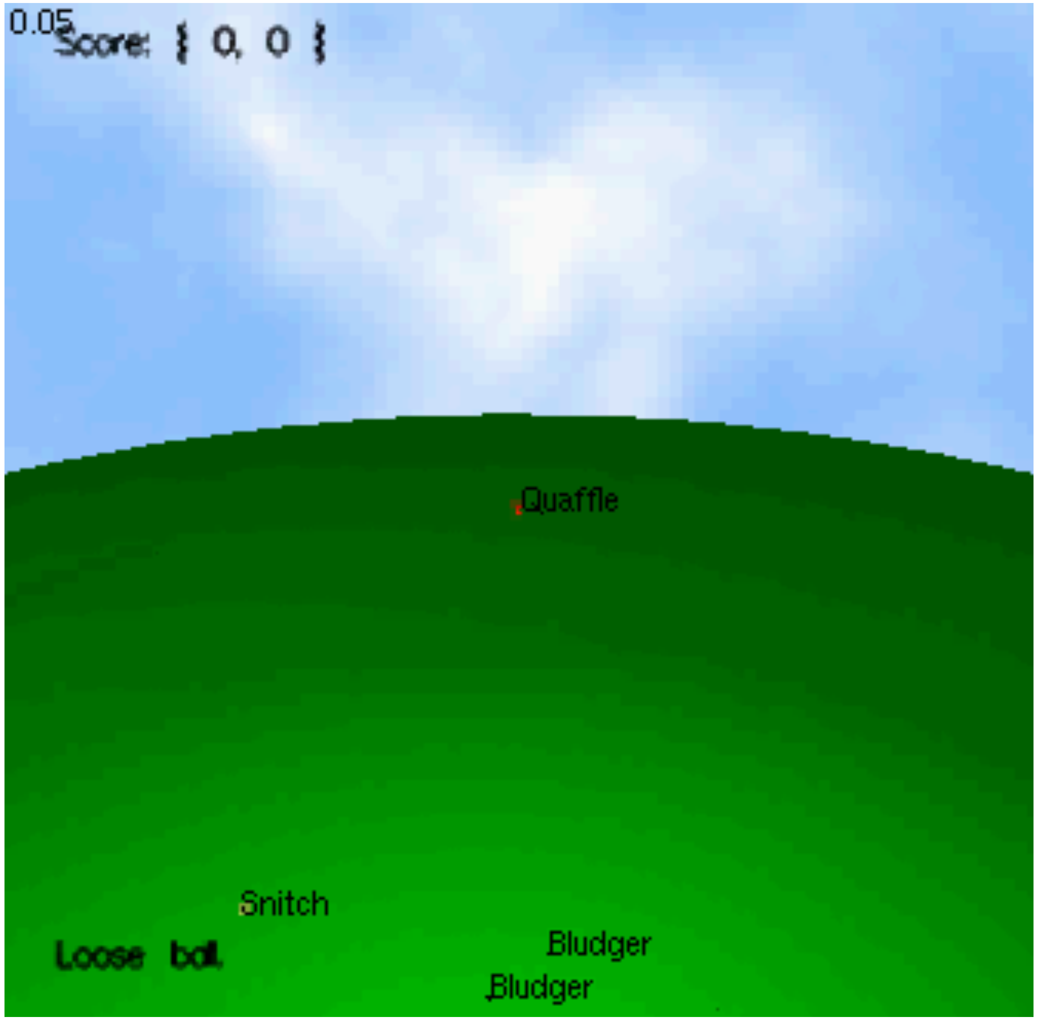
Applications

From the first 80 of more than 499 matches for the query {application and "genetic programming"} in the GP bibliography:

low-resolution character recognition algorithms, solutions to system identification problems, minimum cost topology optimisation of optical telecommunication networks, control for a four-legged robot, predictors for chaotic time series, estimating thermal profiles, producing equations from hydraulic data, data mining and knowledge discovery in sediment transport, discovery of technical trading rules, analog circuits that perform digital functions, scalable distributed controllers for a novel self-reconfigurable modular robotic application, automatic target detection within SAR imagery, image analysis for scientific inquiry, intelligent control of biotechnological processes, modeling of a biotechnological fed-batch fermentation, learning composite operators for object detection, modeling electricity demand prediction, knowledge discovery in chest-pain diagnosis, data mining of medical data sets, predicting survival of patients, mobile robot sensor fusion, corporate failure prediction, bond-issuer credit rating, trading restrictions, speculative trades and price volatility, speculative trades and financial regulations, hedging derivative securities, model learning in a coordination game, document classification, design broadcasting algorithms for Manhattan street

Dynamic Game Control

- RoboCup teams (Luke, Andre, Teller, Adorni, ...).
- Quidditch (Crawford-Marks, Spector)



Line-Drawing Mechanism

Without reference to an existing straight line.

Human-competitive results; challenged world's greatest inventors for a century (spanning 18th and 19th).

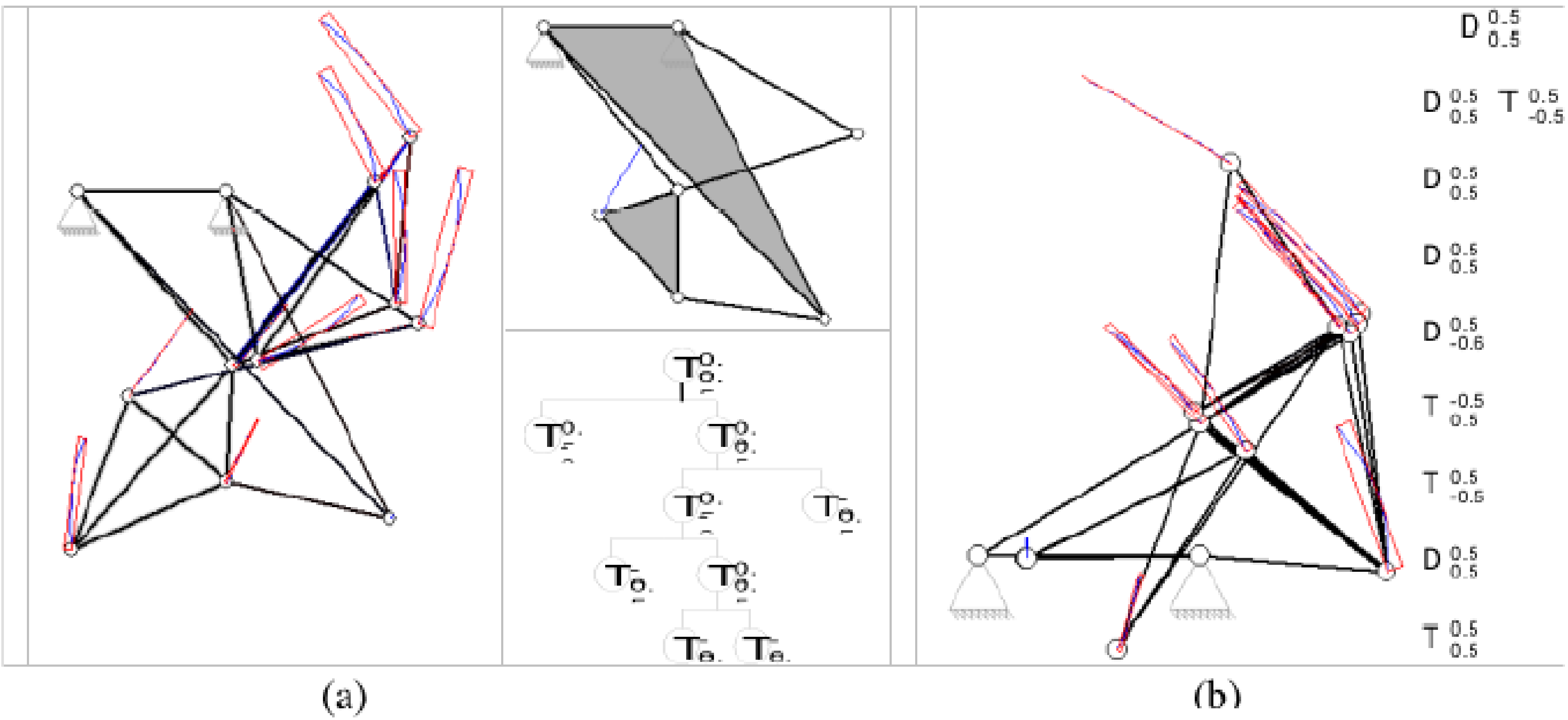
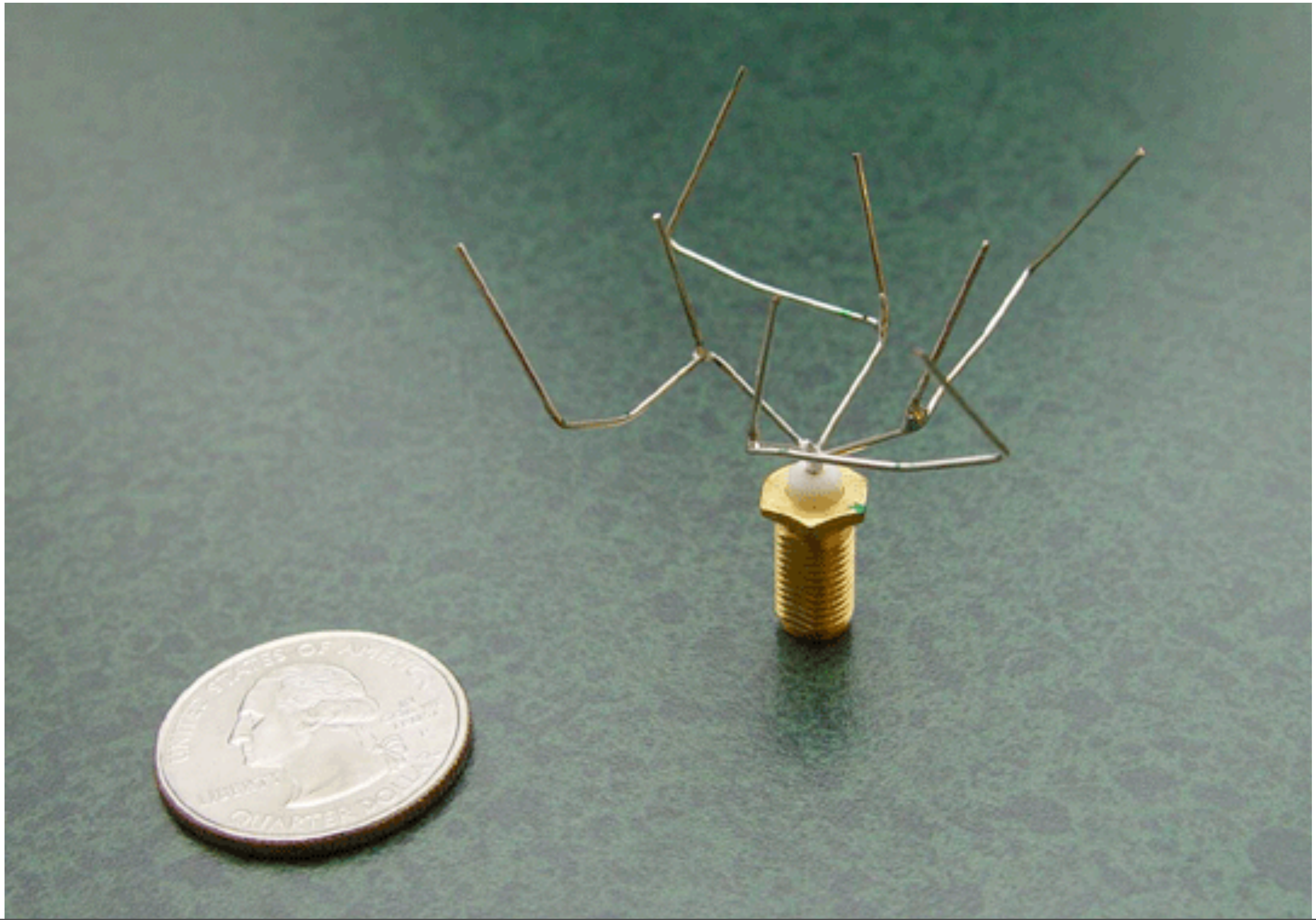


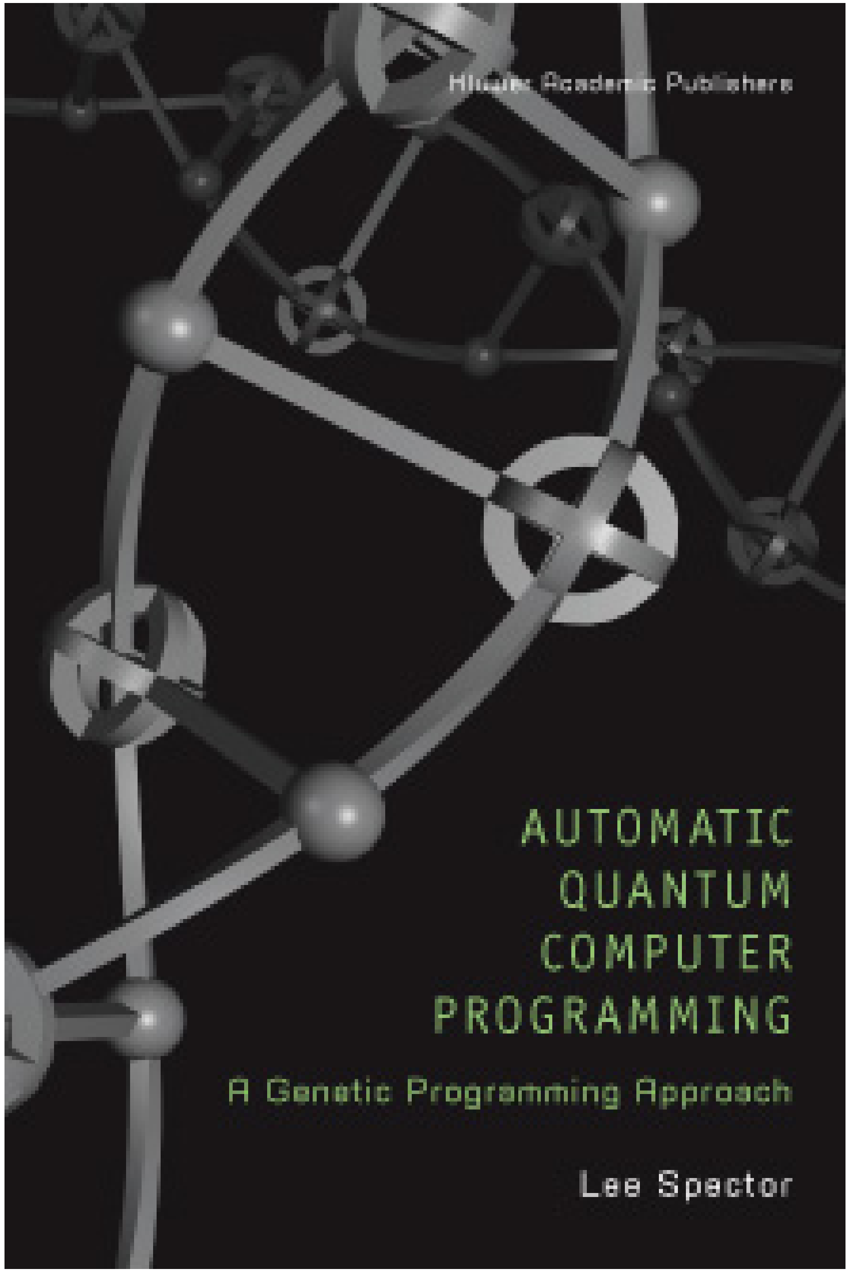
Fig. 10. Two Evolved mechanisms

Lipson, H. 2004. How to Draw a Straight Line Using a GP: Benchmarking Evolutionary Design Against 19th Century Kinematic Synthesis. GECCO-2004.

GP-Designed Antenna

- Human-competitive result.
- For NASA Space Technology 5 Mission.
- Lohn, Hornby, and Linden, 2004.

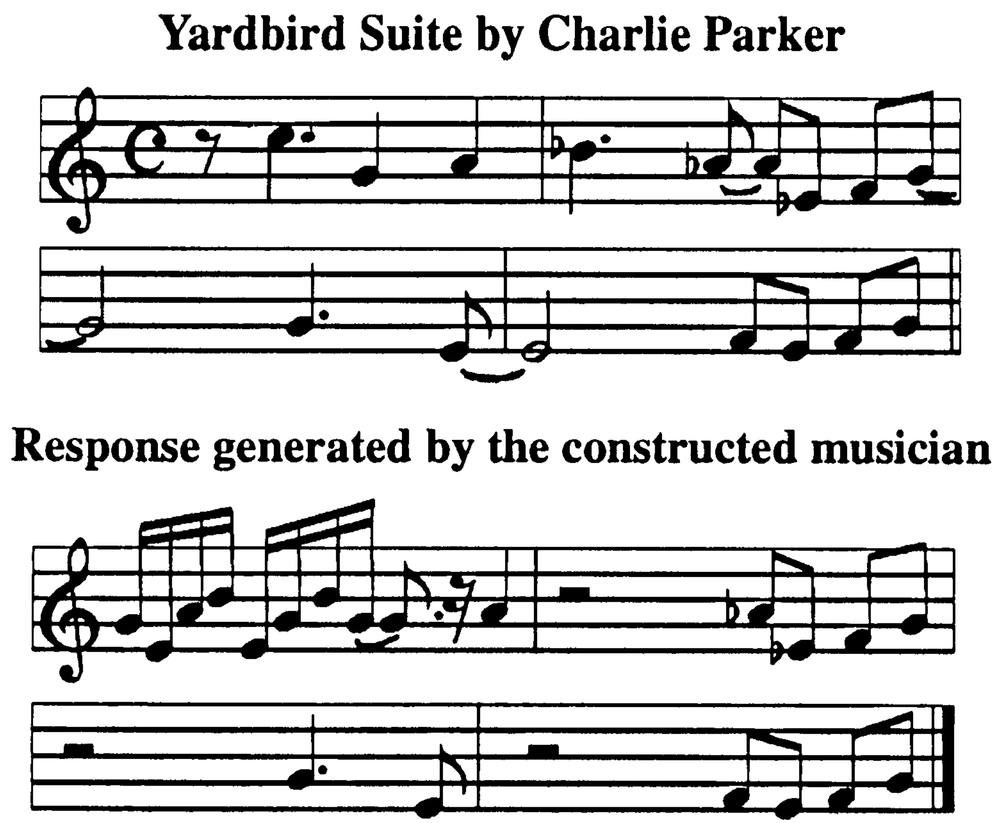




2007/2004, Springer/Kluwer Academic Publishers

http://hampshire.edu/lspector/aqcp/

Evolving a Musician



- Subsequent work used a neural network critic for better results.
- See http://hampshire.edu/lspector/genbebop.html

Example

“Irreducible Complexity”

By *irreducibly complex* I mean a single system composed of several well-matched, interacting parts that contribute to the basic function, wherein the removal of any one of the parts causes the system to effectively cease functioning. An irreducibly complex system cannot be produced directly (that is, by continuously improving the initial function, which continues to work by the same mechanism) by slight, successive modifications of a precursor system, because any precursor to an irreducibly complex system that is missing a part is by definition nonfunctional. An irreducibly complex biological system, if there is such a thing, would be a powerful challenge to Darwinian evolution.

Michael Behe. 1996. Darwin's Black Box: The Biochemical Challenge to Evolution. p. 39.

Fitness Landscapes

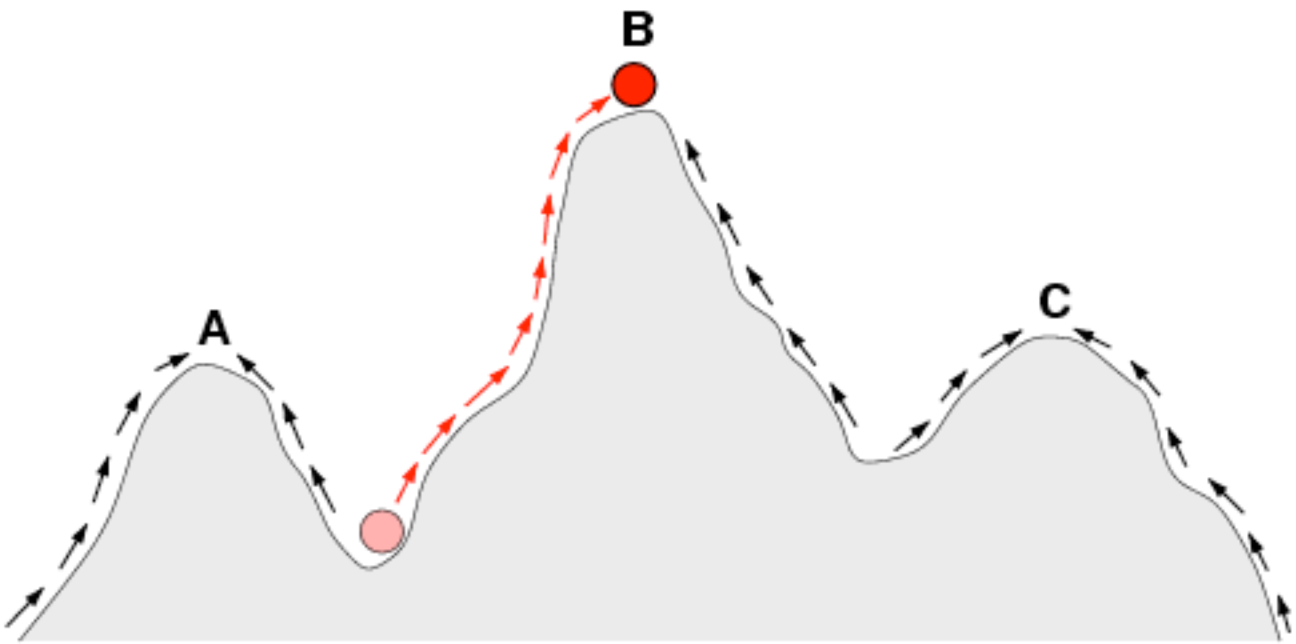
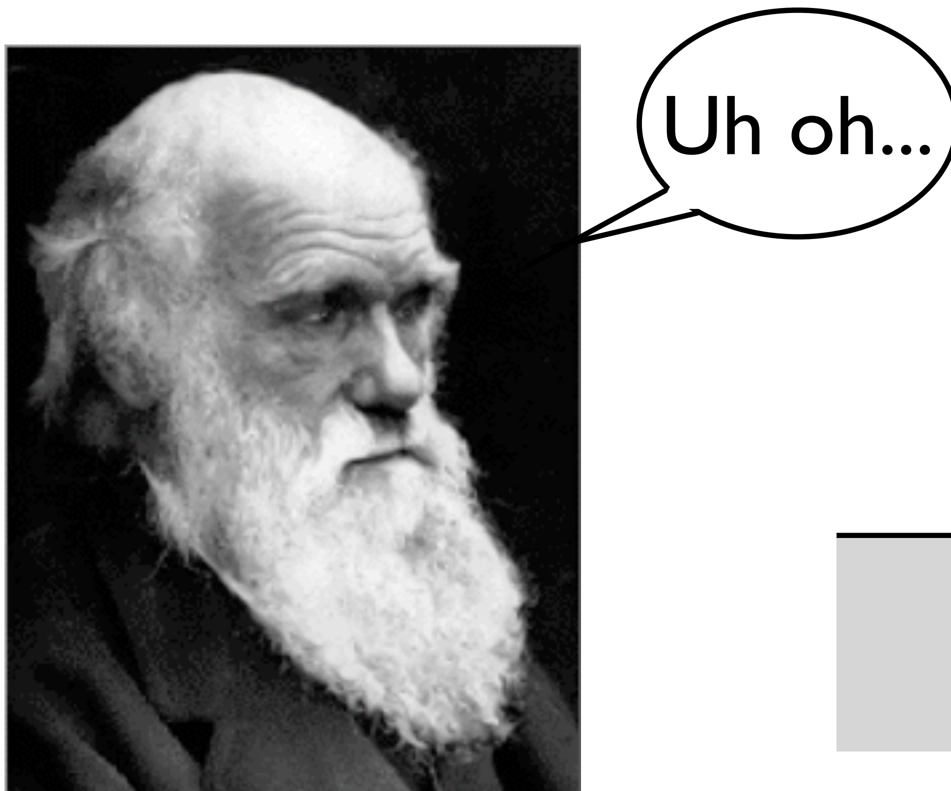


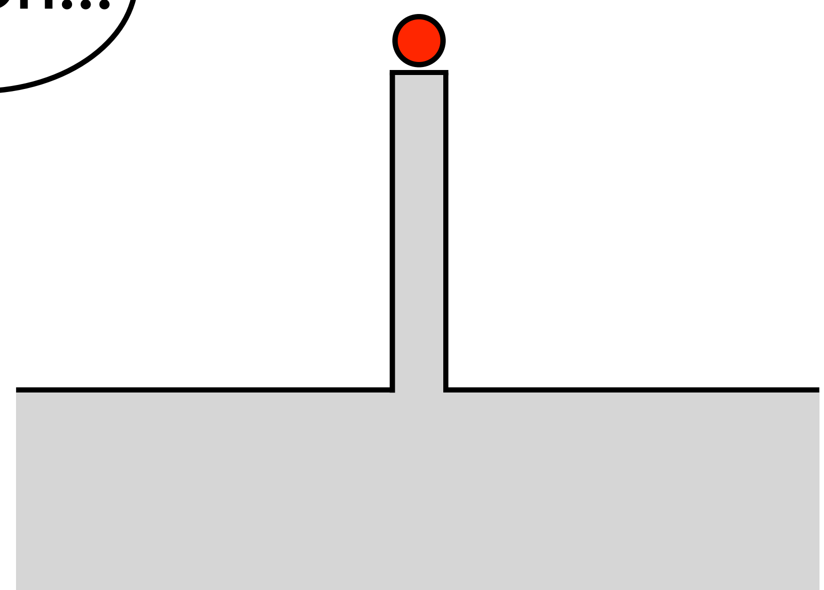
Figure 1: Sketch of a fitness landscape. The arrows indicate the preferred flow of a population on the landscape, and the points A, B, and C are local optima. The red ball indicates a population that moves from a very low fitness value to the top of a peak. Illustration by C.O. Wilke, 2001.

From the Wikipedia entry for "Fitness Landscape."

Intuitive but false

Points (organisms) that appear to be on steep fitness spikes can't be reached by Darwinian evolution.





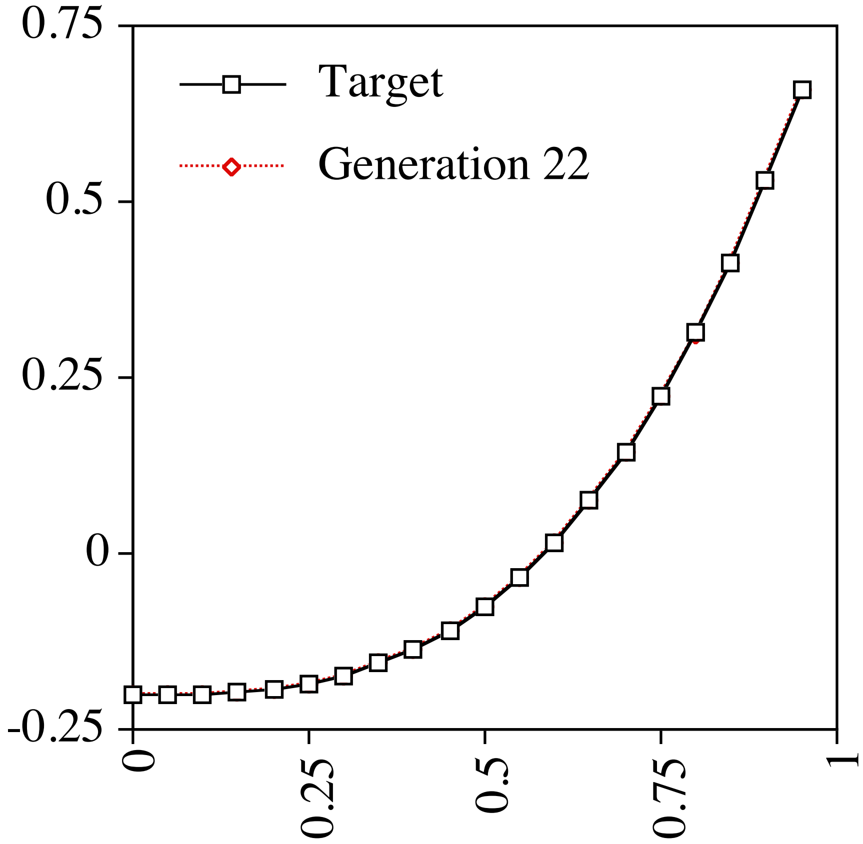
Showing it's false

- *From the fossil record and biology:*
  For each purported spike fill in the missing data and analysis to show how the fitness landscape was traversed. (*They're usually not really spikes*.)
- *From computer science:*
  Experimentally demonstrate the Darwinian evolution of novel artifacts that appear to be irreducibly complex. (*Darwinian processes can produce things that look like spikes*.)

Best Program, Gen 22

"... removal of any one of the parts causes the system to effectively cease functioning."

```
(- (- (* X (* X X))
      0.1)
   0.1)
```



Evolution, the Designer

Apparent “irreducible complexity” is actually an expected product of Darwinian mechanisms, not evidence for a non-Darwinian “designer.”

“Darwinian evolution is itself a designer worthy of significant respect, if not religious devotion.” *Boston Globe* OpEd, Aug 29, 2005

The Future of AI

"The logical extension of [AI's historical trajectory] is to model not only the products of natural evolution but also its processes."

Artificial Intelligence, Vol. 170, Issue 18, pp. 1251-1253.