Evolutionary Algorithms:

Harnessing Natural Selection in Software

Maxwell Metzner

Akhilesh Reddy

Ava Ragonese

Bailey McNamara

Stevens Institute of Technology EN 250

Introduction:

Evolutionary algorithms are algorithms that use concepts inspired by nature in order to solve problems through processes that mimic the behaviors of living organisms. Typically, evolutionary algorithms aim to solve problems that can not be solved easily or in polynomial time. They are usually used in combinatorial problems to find a combination of elements that maximize some fitness function or as an optimization technique to approximate a solution relatively quickly.

An implementation of an evolutionary algorithm requires four steps: initialization, selection, genetic operators, and termination. First, initialization is where an initial population of individuals/solutions is created. Here the population is often created randomly within the constraints of the problem. Second, selection is where the individuals/solutions are evaluated according to the fitness function that was chosen. The fitness function is often difficult to create and must accurately represent the data. Third, genetic operators are the most fit individuals/solutions that are then chosen for reproduction, creating which will become the next generation of the algorithm. The characteristics of the parents are reproduced to create children of mixed parent qualities. In addition to crossover mutations also must be created. Without this, an algorithm would constantly receive solutions in a local extrema. Mutations are completed by changing the portion of the children so they are not a perfect subset of the parent genes. With this, it is important that they are creating the necessary diversity and thereby facilitating novelty. Finally, termination is where the algorithm is completed either by hitting the maximum number of iterations or the algorithm hits a threshold of performance. A final solution is also returned and the algorithm is complete.

History:

Evolutionary programming is one of the four major evolutionary algorithm paradigms. While it is most similar to genetic programming, the structure tends to be a little different—the structure of the program to be optimized is fixed, while the numerical parameters are permitted to evolve. The roots of evolutionary programming were laid by Lawrence J. Fogel. Fogel is the man who is also credited with pioneering this concept/idea in the early 60's. Dr. Fogel is best described by his colleagues as, "a father of computational intelligence" (ETHW). Originally Fogel did not intend to make ground-breaking discoveries, rather he was using EA as an aid in his learning process to generate artificial intelligence. In 1964, Fogel published his first book, "Artificial Intelligence through Simulated Evolution" with co-authors Alvin Owens and Michael Walsh.

Approach:

In order to understand the uses and effectiveness of evolutionary algorithms, we will be coding examples of evolutionary algorithms. Our initial idea for an evolutionary algorithm is to simulate throwing an object and show how an evolutionary algorithm can be used to approximate an efficient solution for the independent variables angle and power in order to maximize the dependent variable, the total rightwards distance. The project was coded in Python, and utilizes initialization, selection, genetic operators, and termination in order to find its solution.

Works Cited

De Jong, Kenneth & Fogel, David & Schwefel, Hans-Paul. (1997). A history of evolutionary computation.

“Lawrence J. Fogel.” *Lawrence J. Fogel - Engineering and Technology History Wiki*, 13 Sept. 2011, https://ethw.org/Lawrence_J._Fogel.