

# Q1.2: What's Evolutionary Programming (EP)?

## Introduction

EVOLUTIONARY PROGRAMMING, originally conceived by Lawrence J. Fogel in 1960, is a stochastic OPTIMIZATION strategy similar to GENETIC ALGORITHMS, but instead places emphasis on the behavioral linkage between PARENTS and their OFFSPRING, rather than seeking to emulate specific GENETIC OPERATORS as observed in nature. EVOLUTIONARY PROGRAMMING is similar to EVOLUTION STRATEGIES, although the two approaches developed independently (see below).

Like both ES and GAs, EP is a useful method of OPTIMIZATION when other techniques such as gradient descent or direct, analytical discovery are not possible. Combinatoric and real-valued FUNCTION OPTIMIZATION in which the OPTIMIZATION surface or FITNESS landscape is "rugged", possessing many locally optimal solutions, are well suited for EVOLUTIONARY PROGRAMMING.

## History

The 1966 book, "Artificial Intelligence Through Simulated Evolution" by Fogel, Owens and Walsh is the landmark publication for EP applications, although many other papers appear earlier in the literature. In the book, finite state automata were evolved to predict symbol strings generated from Markov processes and non-stationary time series. Such evolutionary prediction was motivated by a recognition that prediction is a keystone to intelligent behavior (defined in terms of adaptive behavior, in that the intelligent organism must anticipate events in order to adapt behavior in light of a goal).

In 1992, the First Annual Conference on EVOLUTIONARY PROGRAMMING was held in La Jolla, CA. Further conferences have been held annually (See Q12). The conferences attract a diverse group of academic, commercial and military researchers engaged in both developing the theory of the EP technique and in applying EP to a wide range of OPTIMIZATION problems, both in engineering and biology.

Rather than list and analyze the sources in detail, several fundamental sources are listed below which should serve as good pointers to the entire body of work in the field.

## The Process

For EP, like GAs, there is an underlying assumption that a FITNESS landscape can be characterized in terms of variables, and that there is an optimum solution (or multiple such optima) in terms of those variables. For example, if one were trying to find the shortest path in a Traveling Salesman Problem, each solution would be a path. The length of the path could be expressed as a number, which would serve as the solution's fitness. The fitness landscape for this problem could be characterized as a hypersurface proportional to the path lengths in a space of possible paths. The goal would be to find the globally shortest path in that space, or more practically, to find very short tours very quickly.

The basic EP method involves 3 steps (Repeat until a threshold for iteration is exceeded or an adequate solution is obtained):

- (1) Choose an initial POPULATION of trial solutions at random. The number of solutions in a population is highly relevant to the speed of OPTIMIZATION, but no definite answers are available as to how many solutions are appropriate (other than >1) and how many solutions are just wasteful.

- (2) Each solution is replicated into a new POPULATION. Each of these OFFSPRING solutions are mutated according to a distribution of MUTATION types, ranging from minor to extreme with a continuum of mutation types between. The severity of MUTATION is judged on the basis of the functional change imposed on the PARENTS.
- (3) Each OFFSPRING solution is assessed by computing it's FITNESS. Typically, a stochastic tournament is held to determine N solutions to be retained for the POPULATION of solutions, although this is occasionally performed deterministically. There is no requirement that the POPULATION SIZE be held constant, however, nor that only a single OFFSPRING be generated from each PARENT.

It should be pointed out that EP typically does not use any CROSSOVER as a GENETIC OPERATOR.

#### EP and GAs

There are two important ways in which EP differs from GAs.

First, there is no constraint on the representation. The typical GA approach involves encoding the problem solutions as a string of representative tokens, the GENOME. In EP, the representation follows from the problem. A neural network can be represented in the same manner as it is implemented, for example, because the MUTATION operation does not demand a linear encoding. (In this case, for a fixed topology, real-valued weights could be coded directly as their real values and mutation operates by perturbing a weight vector with a zero mean multivariate Gaussian perturbation. For variable topologies, the architecture is also perturbed, often using Poisson distributed additions and deletions.)

Second, the MUTATION operation simply changes aspects of the solution according to a statistical distribution which weights minor variations in the behavior of the OFFSPRING as highly probable and substantial variations as increasingly unlikely. Further, the severity of MUTATIONS is often reduced as the global optimum is approached. There is a certain tautology here: if the global optimum is not already known, how can the spread of the mutation operation be damped as the solutions approach it? Several techniques have been proposed and implemented which address this difficulty, the most widely studied being the "Meta-Evolutionary" technique in which the variance of the mutation distribution is subject to mutation by a fixed variance mutation operator and evolves along with the solution.

#### EP and ES

The first communication between the EVOLUTIONARY PROGRAMMING and EVOLUTION STRATEGY groups occurred in early 1992, just prior to the first annual EP conference. Despite their independent development over 30 years, they share many similarities. When implemented to solve real-valued FUNCTION OPTIMIZATION problems, both typically operate on the real values themselves (rather than any coding of the real values as is often done in GAs). Multivariate zero mean Gaussian MUTATIONS are applied to each PARENT in a POPULATION and a SELECTION mechanism is applied to determine which solutions to remove (i.e., "cull") from the population. The similarities extend to the use of self-adaptive methods for determining the appropriate mutations to use -- methods in which each PARENT carries not only a potential solution to the problem at hand, but also information on how it will distribute new trials (OFFSPRING). Most of the theoretical results on CONVERGENCE (both asymptotic and velocity) developed for ES or EP also apply directly to the other.

The main differences between ES and EP are:

1. **SELECTION:** EP typically uses STOCHASTIC SELECTION via a tournament. Each trial SOLUTION in the POPULATION faces competition against a preselected number of opponents and receives a "win" if it is at least as good as its opponent in each encounter. SELECTION then eliminates those SOLUTIONS with the least wins. In contrast, ES typically uses deterministic SELECTION in which the worst SOLUTIONS are purged from the POPULATION based directly on their function evaluation.
2. **RECOMBINATION:** EP is an abstraction of EVOLUTION at the level of reproductive POPULATIONS (i.e., SPECIES) and thus no RECOMBINATION mechanisms are typically used because RECOMBINATION does not occur between SPECIES (by definition: see Mayr's biological species concept). In contrast, ES is an abstraction of EVOLUTION at the level of INDIVIDUAL behavior. When self-adaptive information is incorporated this is purely GENETIC information (as opposed to PHENOTYPIC) and thus some forms of RECOMBINATION are reasonable and many forms of RECOMBINATION have been implemented within ES. Again, the effectiveness of such operators depends on the problem at hand.

#### References

Some references which provide an excellent introduction (by no means extensive) to the field, include:

ARTIFICIAL INTELLIGENCE Through Simulated EVOLUTION [Fogel66] (primary)

Fogel DB (1995) "Evolutionary Computation: Toward a New Philosophy of Machine Intelligence," IEEE Press, Piscataway, NJ, forthcoming.

Proceeding of the first [EP92], second [EP93] and third [EP94] Annual Conference on EVOLUTIONARY PROGRAMMING (primary) (See Q12).

#### PSEUDO CODE

Algorithm EP is

```
// start with an initial time
t := 0;

// initialize a usually random population of individuals
initpopulation P (t);

// evaluate fitness of all initial individuals of population
evaluate P (t);

// test for termination criterion (time, fitness, etc.)
while not done do

    // perturb the whole population stochastically
    P'(t) := mutate P (t);

    // evaluate it's new fitness
    evaluate P' (t);

    // stochastically select the survivors from actual fitness
    P(t+1) := survive P(t),P'(t);

    // increase the time counter
    t := t + 1;
od
end EP.
```

[Eds note: An extended version of this introduction is available from  
ENCORE (see Q15.3) in /FAQ/supplements/what-is-ep ]

[Go Back Up](#)

[Go To Previous](#)

[Go To Next](#)