BUILDING BLOCKS OF EVOLUTIONARY ALGORITHMS

K. KrishnaKumar

Department of Aerospace Engineering and Mechanics
The University of Alabama, Tuscaloosa, AL 35487-0280
kkumar@coe.eng.ua.edu

Synthesis and analysis of complex systems can be viewed as a puzzle that needs to be solved using several individual pieces. These individual pieces can be studied, tested, understood, and built better than the complete system. The pieces, labeled in this article as *Building Blocks* of Evolutionary Algorithms (EA), when put together in the required forms, result in many different Evolutionary Algorithms.

We introduce these building blocks as either local or global building blocks. Finding the optimal parametric set for a given problem can be approached in a manner similar to the determination of the best marathon runner in the world. To do this efficiently, one resorts to competitions at local (country level) and global (world) levels. Similarly, genetic search can be made efficient by adopting global-local aspects to the search. We highlight this view at the end of the article with an example.

EA work with a coding of the parameter set (not the parameters themselves), search from a population of points (not a single point), and use probabilistic (not deterministic) transition rules. Learning capabilities, near-optimal solutions, robustness, generic code structure, capability to incorporate a prior knowledge, adaptability, memory, domain independence, hybridization, and advanced operators are some of the features of EA that make them ideal candidates for search. In the next several pages we present several building blocks of evolutionary algorithms under the broad categories of:

1) Coding
2) Population
3) Recombination
4) Mutation
5) Selection

For each of the building block presented, we present the algorithm or equations as appropriate.

1.1. Coding

One advantage of EA is that the encoding structure permits diverse solution optimization. However there are some initial decisions that have to be made, which mostly depend upon the number of parameters we want EA to optimize, and whether we want continuous or binary representation. Encoding schemes include Binary (example: 1001010), K-ary (example: 1ff0000c0), real (example: 43.76), stochastic, permutation, lisp, etc. The most common method is the binary integer representation. Here, each variable (parameter) is first linearly mapped to an integer defined in a specific range, and then encoded using a fixed number of binary bits. The population members are formed by concatenation of these binary bits.

1.1.1. Binary to Decimal (GLOBAL)

Given a binary representation, we convert it to the required decimal equivalent in two steps. In step 1, we convert the binary to an integer representation, and in step 2, we convert the integer representation to decimal equivalent or any other preferred mapping chosen by the user. It is emphasized here that step 2 is problem dependent, and it is up to the user to arrive at the necessary definition.

Step 1 Algorithm:

$$I_i = \sum_{j=1}^{n\_bits(i)} a_{i,j} 2^{j-1}$$

where $a_{i,j}$ for $j = 1, \ldots, n\_bits(i)$ represents the bit string segment of length *n_bits(i)* for encoding the i[th] parameter integer value.

Example:

$$a_i = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} a_{i,1} & a_{i,2} & a_{i,3} & a_{i,4} \end{bmatrix}$$

$$I_i = a_{i,i} 2^0 + a_{i,2} 2^1 + a_{i,3} 2^2 + a_{i,4} 2^3 = 1 * 2^0 + 0 * 2^1 + 1 * 2^2 + 0 * 2^3 = 5$$

Step 2 Algorithm:

In step 2, several options exist. We present two of them here.

Option 1 (GLOBAL): Explicit upper and lower bounds are given as $L_i \leq x_i \leq U_i$ and from L_i computed in step 1, the parameter value x_i can be computed as

$$x_i = L_i + \frac{U_i - L_i}{2^{n\_bits(i)} - 1} I_i$$

Examples:

$$I_1 = 5; \quad U_1 = 10; \quad L_1 = -5; \quad n\_bits(1) = 4; \quad \Rightarrow \quad x_1 = 0.0$$

$$I_2 = 2; \quad U_2 = 15; \quad L_2 = -5; \quad n\_bits(2) = 5; \quad \Rightarrow \quad x_2 = -2.3333$$

Option 2 (LOCAL): Given mean *(M)* and standard deviation *(Σ)* matrices, the value for the parameter x_i is randomly drawn using a Gaussian probability density function defined as:

$$N(M, \Sigma) : p(x_i) = \frac{1}{\sqrt{2\pi} \quad \Sigma(i, I_i)} \exp \quad \frac{-(M(i, I_i) - x_i)^2}{2\Sigma(i, I_i)^2}$$