Emergence of Genomic Self-Similarity in a Proteome-Based Representation

Ivan I. Garibay[1,2] and Annie S. Wu[1]
[1]University of Central Florida, School of Computer Science,
P.O. Box 162362, Orlando, FL 32816-2362
[2]University of Central Florida, Office of Research
12443 Research Parkway, Orlando, FL 32826
{igaribay,aswu}@cs.ucf.edu,
WWW home page: `http://ivan.research.ucf.edu`

Abstract

We report the emergence of genomic self-similarity with respect to fitness in a Genetic Algorithm representation with no selective pressure for any particular genomic ordering.

Introduction

The idea of representations that self-organize for evolutionary algorithms (Garibay and Wu, 2004b) bring together a collection of issues such as simple genomes that represent complex solutions, complexity building by means of evolved self-organizable building blocks, innovativeness, scalability, evolvability, and development as a source of complexity. All this ideas deal with the complexity problem: scaling evolutionary algorithms to high complexities. While evolutionary algorithms constantly struggle with complexity, natural evolution deals with extremely complex organisms with ease. The quest, therefore, is to localize and extract the principles that nature uses to deal with complexity and apply them to build better algorithms. Recently, various approaches to self-organizable representations have been proposed including methods based on developmental biology and morphogenesis (Bentley, 2004; Kumar, 2004; Koza et al., 1999), and cellular automata variations (Miller and Thomson, 2004; Rocha, 2004) among others. We propose a proteome analogy approach (Garibay and Wu, 2003) where functional structures analogous to proteins act as complexity builders for a genome. In this paper, we use a simple abstraction that focuses only on the string to multiset nature of the genome to proteome mapping: The Proportional Genetic Algorithm (PGA) (Wu and Garibay, 2002). We use this algorithm to study the emergent ordering of genomic symbols in the complete absence of selective pressure for a particular order. The PGA is a Genetic Algorithm (GA) (Holland, 1975; Goldberg, 1989) with a representation based on protein concentrations rather than on the usual gene ordering. A PGA translates strings of genes into multisets of proteins prior to fitness evaluation. As a result, there is no fitness pressure for any particular gene ordering and the order of the genes is free to evolve along with the candidate solutions they encode. Previous studies show that genomic symbols under these circumstances are evenly distributed throughout the genome (Wu and Garibay, 2002) and that their equal-symbol correlation resembles white noise behavior (Garibay and Wu, 2004a). In this paper, we provide experimental evidence of the emergence of genomic self-similarity in the PGA proteome-based representation.

Proportional Genetic Algorithm

The Proportional Genetic Algorithm is a GA with a representation inspired by the genome to proteome mapping. In a GA an individual is represented typically as a binary string. A population of these strings undergoes continuous genetic variation and fitness-based selection to produce more fit individuals. In the PGA, individuals are strings over a multi-character alphabet and they undergo the same genetic variation. The difference is that fitness is not evaluated on these strings but on their associated multisets. The multiset associated to a string of symbols is simply the multiset containing all and only the elements in the corresponding string. For instance, the associated multiset of the string "aab" is $\{|a,a,b|\}$. Note that the strings "aab", "aba", and "baa" all have the same associated multiset and are indistinguishable with respect to fitness. Consequently, there is no pressure to select any particular ordering of two "a"s and one "b" in a genome. For a full description of the PGA and its applications, we refer the reader to (Wu and Garibay, 2002; Wu and Garibay, 2004).

Self-Similarity Metric for Genomes

We use fitness as the metric for genomic self-similarity. A genome is self-similar if its fitness is approximately equal to the average fitness obtained by evaluating all of its genomic segments of a given segment length. More formally, let us introduce the following notation. A genome $g_{<1,L>} = g_1 g_2 g_3 \ldots g_L$, or simply g, is a string of length L over the genome alphabet Σ. A segment $s_{<i,j>} = s_i s_{i+1} s_{i+2} \ldots s_j$ of $g_{<1,L>}$ is defined as the substring $g_i g_{i+1} g_{i+2} \ldots g_j$, where $1 \leq i \leq j \leq L$. A fitness function $\mathsf{F}\left(g_{<1,L>}\right)$ maps strings into real numbers and is defined for any genome or segment.

Using the notation above, we define the average fitness of all segments of size r over genome $g_{<1,L>}$ as follows:

$$\hat{f}_r(g_{<1,L>}) = \frac{\sum_{i=1}^{L-r+1} \mathsf{F}\left(s_{<i,i+r-1>}\right)}{(L-r+1)} \qquad (1)$$

Note that for a segment equal to the whole genome ($r = L$), the expression above reduces simply to a fitness evaluation:

$$\hat{f}_r(g_{<1,L>}) = \mathsf{F}\left(s_{<1,L>}\right) = \mathsf{F}\left(g_{<1,L>}\right)$$

Finally, we say that a genome is self-similar if the following expression is true:

$$\forall_r \left[\hat{f}_r(g_{<1,L>}) \approxeq \mathsf{F}\left(g_{<1,L>}\right)\right] \qquad (2)$$

where r is the size of the genome segments used to analyze self-similarity.

The above equation implies the following. For an ideal case of genomic self-similarity, Equation 2 will hold indefinitely for any segment size r. In this case, genomic segments of any size will have the same fitness as the whole genome, as shown in Figure 1(A). This case is analogous to perfect fractal behavior. On the other hand, if there is no self-similarity, Equation 2 will not hold even for large segments. In this case, the fitness of the segments will not resemble the fitness of the whole genome. We thus expect random segments with average fitness equal to the median fitness of the problem as demonstrated in Figure 1(B).

Experimental Analysis

Objectives

The objective of this empirical study is to determine whether or not self-similarity with respect to fitness emerges in the PGA proteomic-based representation. We use a GA as a baseline case where the genomic order is dictated by the selection pressure. Based on previous PGA studies that suggest that PGA genome segments are coarse grained versions of the whole PGA genome, we expect the following behavior. Large genome segments will approximate the fitness of the whole genome very closely. The smaller the genome segments, the less they will resemble the fitness of the whole genome. There will be a cut-off point where the segment is too small to represent the required information and self-similarity is lost. As a result, we expect a gradual decrease in fitness as the segments get smaller until a cut-off point is reached at which point fitness will show a significant drop, as shown in Figure 1(C). Exceeding our expectations, and as we will see shortly in the results section, PGA genomes resemble ideal self-similarity until the cut off point is reached.

Fitness evaluation

For all experiments, we set the algorithms to solve a very simple problem: *number matching*. Number matching is a simple hamming distance type of problem but for real numbers. The goal is to match, as closely as possible, a vector of target numbers. We use vectors of size two. For convenience of exposition, let us define the following function:

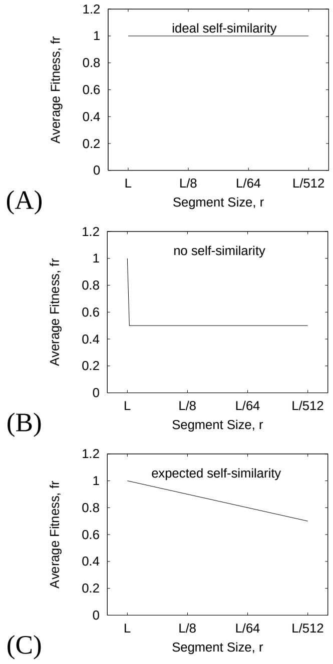

Figure 1: Predicted genomic self-similarity in populations that have converged to the optimal fitness of 1. (A) Ideal self-similar case, all genomic segments have also fitness of 1 regardless of their size. (B) No self-similarity case, segments have random fitness with average equal to the fitness median. (C) PGA expected self-similarity, segments gradually reduce in fitness as they get smaller. Experimental results resemble the ideal case (see Figure 2).

$$\delta(x,y) = \begin{cases} x/y & \text{if } x < y \\ y/x & \text{otherwise} \end{cases} \qquad \forall x,y \in \Re^+.$$

Fitness is determined by the average difference, as measured by δ, between the values encoded in the individual's genome (or segment) and the target values:

$$\mathsf{F}\left(g_{<1,L>}\right) = \frac{\delta(v^1_{target}, v^1_{encoded}) + \delta(v^2_{target}, v^2_{encoded})}{2}$$

Where, $V_{target} = \{v^1_{target}, v^2_{target}\}$ is the vector of target values, and $V_{encoded} = \{v^1_{encoded}, v^2_{encoded}\}$ is the vector of values encoded in g. For the PGA, the encoded values are given by:

$$V_{encoded}(g) = \left[\frac{|g|_a}{|g|_a + |g|_b}, \frac{|g|_c}{|g|_c + |g|_d}\right]$$

where the genomic alphabet is $\Sigma_{PGA} = \{a,b,c,d\}$, and $|X|_y$ returns the number of times symbol y appears in the multiset associated with string X. Note that this definition is also

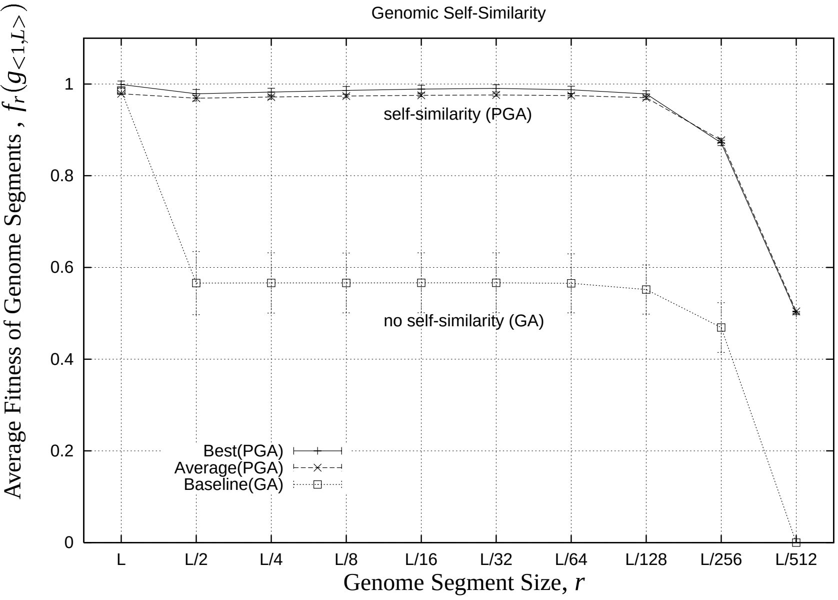

Figure 2: PGA best, PGA average, and GA baseline comparison of average fitness of genome segments($\hat{f}_r(g_{<1,L>})$) of sizes(r) L(whole genome), L/2(half genome), L/4, ... , and L/512 on a simple number matching problem averaged over 20 runs with 95% confidence intervals. (Top: self-similarity) PGA segment fitness remains approximately equal to the whole genome fitness for segment sizes of L/2, L/4, ... until L/128. (Middle: no self-similarity) No GA segment has average fitness close to the whole genome fitness for any segment size. Baseline note: for clarity, GA average is shown but GA best is not since it has similar behavior.

valid for genomic segments. Note also that only the associated multisets are used to calculate the values. The first value is encoded simply as a proportion of the concentrations between symbols *a* and *b*, and the second value, between symbols *c* and *d*. The proportions are numbers between zero and one, but they can be easily scaled to represent any parameter range of values.

For the GA, the values encoded in the genome are interpreted in the usual way. The genomic alphabet is binary: $\Sigma_{GA} = \{0,1\}$. The first half of the genome represents our first encoded value and the second half, our second encoded value. Note that this definition is also valid for genome segments.

Settings

For our experiments, we use a GA with proportional representation (PGA) and a standard GA with binary representation as a baseline. The GA uses a binary alphabet; the PGA uses a multicharacter alphabet. Mutation, in the GA, is bit-flip mutation, while in the PGA, randomly change one alphabet symbol for another. The following parameter settings are common for all experiments: the genome length is L=1000 and segment sizes are L/2, L/4, L/8, L/16, L/32, L/64, L/128, L/256, and L/512, the crossover type is two-point, the crossover rate is 0.8, the mutation rate is 0.005, selection type is tournament of size 4, population size of 250, and the number of generations is 500. We perform 20 trials for all experiments using new randomly generated targets for each run, and report average values with their 95% confidence intervals.

Results

Figure 2 shows plots of the average measured fitness for decreasing segment length. These results reveal genomic self-similarity with respect to fitness for the PGA and no self-similarity for the baseline case. The self-similarity is almost ideal (see Figure 1 (A)) for segments as small as $L/128 = 1000/128 = 7.81$ symbols. After this critical point, self-similarity is lost and fitness of the segments decreases significantly. Figure 3 shows the raw fitness of segments of size L/2 for the two encoded values over the 500 generations. (A) The variance for the self-similar case is very small. This result indicates that all segments, not just their averages, must have fitness similar to the whole genome. (B) The variance for the non-self-similar case is very high, indicating that the segments have a wide range of fitness values and are not similar to the whole genome.

Conclusions

In this paper, we offer experimental evidence of the emergence of genomic self-similarity with respect to fitness in genomes that are free to self-organize. We use a GA that

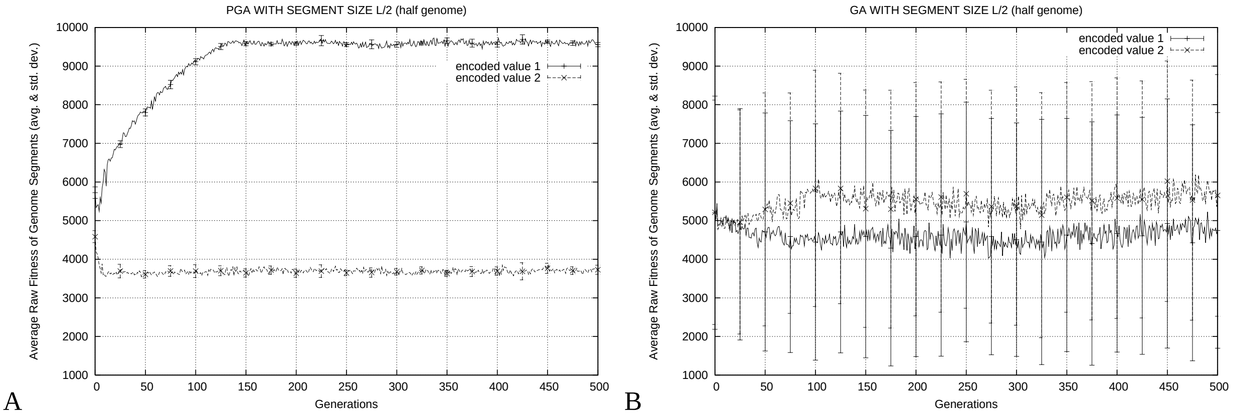

Figure 3: Average raw fitness of genomic segments of size L/2 (half-genome) for encoded values 1 and 2 over 500 generations. The plots show averages and standard deviations over 20 runs. (A) PGA encoded values 1 and 2 tight standard deviations indicate that the fitness of PGA genomic segments of size L/2 converge. From Figure 2 we observe that their fitness converge to a near optimal solution, similar to the fitness of the whole genome L. (B) GA encoded values 1 and 2 wide standard deviations indicate that GA genomic segments of size L/2 do not converge. Furthermore, the fitness of these segments appear to be randomly distributed. From Figure 2 we observe that their average normalized fitness is not similar to the whole genome, but near to the median 0.5 instead. We observe similar behavior for segments of sizes L/4 to L/128.

allows this genomic self-organization: the PGA. The key property of the PGA is that it encodes information not in the genomic order, as is typical in evolutionary computation, but in the protein concentrations. The PGA representation is implemented as a simple mapping from an individual's string of genomic symbols to a multiset of protein symbols. Our experiments show that when the genome is free to evolve genomic ordering, it self-organizes into a fractal-like structure in which genomic segments have approximately the same fitness as the whole genome. We also offer baseline experiments that show that no such self-similarity occurs in typical GAs.

References

Bentley, P. J. (2004). Fractal proteins. *Genetic Programming and Evolvable Machines Journal*, 5:71–101.

Garibay, I. I. and Wu, A. S. (2003). Cross-fertilization between proteomics and computational synthesis. In *Computational Synthesis: From Basic Building Blocks to High Level Functionality. Papers from the 2003 AAAI Spring Symposium*, pages 67–74.

Garibay, I. I. and Wu, A. S. (2004a). Emergent white noise behavior in location independent representations. In *Proc. GECCO 2004 Workshop on Self-organization in Representations for Evolutionary Algorithms: Building complexity from simplicity*.

Garibay, I. I. and Wu, A. S. (2004b). Preface. In *Proc. GECCO 2004 Workshop on Self-organization in Representations for Evolutionary Algorithms: Building complexity from simplicity*.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

Koza, J. R., III, F. H. B., Andre, D., and Keane, M. A. (1999). *Genetic Programming III*. Morgan Kaufmann Publishers.

Kumar, S. (2004). Multicellular development, self-organization, and differentiation. In *Proc. GECCO 2004 Workshop on Self-organization in Representations for Evolutionary Algorithms: Building complexity from simplicity*.

Miller, J. and Thomson, P. (2004). Beyond the complexity ceiling: Evolution, emergence and regeneration. In *Proc. GECCO 2004 Workshop on Regeneration and Learning in Developmental Systems*.

Rocha, L. M. (2004). Evolving memory: Logical tasks for cellular automata. In *Proc. Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*. In Press.

Wu, A. S. and Garibay, I. (2002). The proportional genetic algorithm: Gene expression in a genetic algorithm. *Genetic Programming and Evolvable Hardware Journal*, 3(2):157–192.

Wu, A. S. and Garibay, I. (2004). Intelligent automated control of life support systems using proportional representations. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 34(3):1423–1434.