Schemata Theory for the Real Coding and Arithmetical Operators

Diego F. Nehab
Computer Science Department
Princeton University
diego@cs.princeton.edu

Marco Aurélio C. Pacheco
Applied Computational Intelligence Laboratory
Electrical Engineering Department, PUC-Rio
marco@ele.puc-rio.br

ABSTRACT

The Schemata Theory analyzes the effect of the selection process, mutation and crossover over the number of individuals that belong to a given schema, within generations. This analysis considers, in its original form, the binary coding and operators. In this article, we present an analogous study, focusing on the real number coding and arithmetical operators. Unfortunately, the conventional schema definition is tightly dependent on discrete alphabets. Therefore, following a generalization of the concept of schema, we present a particular definition that suits better the continuous domain. Using this new definition, we reach an expression similar to the Fundamental Theorem of Genetic Algorithms [6] valid for the real coding of chromosomes.

Keywords

Genetic Algorithms, Schemata Theory, Real Coding

1. INTRODUCTION

Despite empirical evidence showing the good performance of Genetic Algorithms (GAs), the method has been criticized for the lack of theoretical foundations supporting the analytical study of its convergence. John Holland created the Schemata Theory [6] in an attempt to provide such foundations. Although his framework fails to explain the long term behavior of GAs, it provides an overall picture of how the population of chromosomes evolves from one generation to the following. Markov chains analysis has been applied later to study the long term behavior of GAs [3].

Holland's theory culminates in a mathematical expression known as the Fundamental Theorem of Genetic Algorithms. The theorem describes the variation on the population of chromosomes belonging to a given schema, from generation to generation, taking into account the selection process, the binary crossover and mutation.

The analysis of this formula leads to the conclusion that low specificity schemata, with short length and above average fitness tend to grow exponentially in number during the first generations. These schemata, thus, quickly dominate the population, and constitute the material over which the algorithm will operate.

An equivalent of the Schemata Theory does not follow immediately for the real coding because this representation and its operators are not related to those commonly used over discrete codings. Furthermore, the conventional schema formulation, on which the entire theory is based, defines concepts such as *order* and *length* which are intuitive for discrete alphabets but have no obvious meaning for the real coding because the floating-point alphabet is virtually non-enumerable.

In fact, Holland argued that low cardinality alphabets should be preferred to high cardinality alphabets. Not only has this claim been challenged theoretically [1], but empirical results [7] show that the floating-point representation actually works better for continuous domain problems.

This article attempts to conciliate theoretical and empirical results by proposing a new study of the Real Coding. Recent work [5] has studied non-linear representations of chromosomes. We do not discuss the validity of the Schema Theory. Instead, we focus on the development of a Fundamental Theorem for the Real Coding and, by analyzing the new theorem formulation, draw the appropriate conclusions.

1.1 New schema definition

The performance analysis of the real representation seems, at first, hopeless. The difficulty comes from the fact that the usual schema definition depends on the use of discrete alphabets for chromosome representation. Although one can state that any representation used by a computer will be discrete to some extent, it is not intuitive to consider a floating-point number as being discrete. Therefore, we move to a new definition of schema, one that is abstract enough so as not to tie us to discrete alphabets.

DEFINITION 1. *Let $\mathcal{U}$ be the search space of a problem, so that every chromosome* u *is an element of $\mathcal{U}$. A Schema is a subspace $\mathcal{H} \subset \mathcal{U}$. If* $\mathrm{h} \in \mathcal{H}$*, we say that $\mathcal{H}$ represents* h.

To Appear: SAC'04, March 14–17, 2004, Nicosia, Cyprus

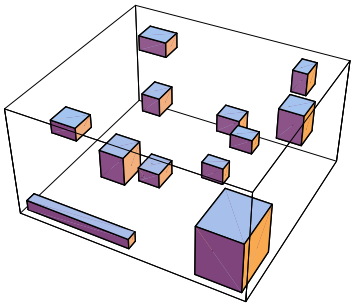


Figure 1: Examples of Schemata in $\mathbb{R}^3$.

Note that Definition 1 accepts the usual schema definition as a special case. Moreover, it allows for a new definition, more useful for the analysis of the real representation; a definition we will use throughout the text. Since, in the real representation, $\mathcal{U}$ is a subspace of $\mathbb{R}^n$, it is natural to define $\mathcal{H}$ as a region inside $\mathcal{U}$. Some examples are seen in Figure 1. More formally:

DEFINITION 2. *For the real representation, $\mathcal{H}$ is a hypercube in $\mathcal{U}$, defined by the coordinates $\mathrm{s_i}$ of one of its vertexes and the widths $\mathrm{l_i}$ through which it runs in each dimension* i *of $\mathcal{U}$, starting at $\mathrm{s_i}$. Moreover, we define $\mathcal{H}_i$ as the interval $[\mathrm{s_i}, \mathrm{s_i} + \mathrm{l_i}]$.*

The term *width* will be used throughout the text to refer to the meaning given by Definition 2. The word *length* seems more appropriate, but we will not use it to avoid confusion with the meaning the original Schema Theory gives for it, which makes no sense for the new definition of schema.

Although the concept of *order* used by the original theory also loses its meaning with the new definition, the property of specificity remains valid. We define low specificity schemata as those for which the l_i are close to the domain limits, since they cover a big part of the search space. Conversely, if the l_i are small values, we say that $\mathcal{H}$ is highly specific.

Therefore, instead of having two parameters to characterize schemata, *order* and *length*, our study will be based on a single parameter: the schema's width.

2. SCHEMATA EVOLUTION ANALYSIS

With the new definition in hand, we proceed to the analysis of the effect that the operations performed over the population $\mathcal{P}$ have on the number of elements in a schema. In the following analysis, all genes are considered to be independent. Therefore, we can freely change the relative widths of their domains and work with intervals normalized to $[0, 1]$.

2.1 The effect of the selection process

Suppose that, in generation t, there are $\mathrm{m}(\mathcal{H}, t)$ individuals of schema $\mathcal{H}$ in population $\mathcal{P}$. Using the traditional roulette wheel selection method, a chromosome h is selected for reproduction with probability $\mathrm{P}_h = f_h / \sum_{p\in\mathcal{P}} f_p$, where f_i is the fitness of element i. For each element being selected, the probability that it belongs to schema $\mathcal{H}$ is given by:

$$\mathrm{P}(\mathcal{H}) = \frac{\sum_{h\in\mathcal{H}} f_h}{\sum_{p\in\mathcal{P}} f_p}$$

Hence, for a population with n individuals, the predicted number of elements of a given schema in generation $t+1$ can be written as:

$$\mathrm{m}(\mathcal{H}, t{+}1) = n{\cdot}\mathrm{P}(\mathcal{H}) = n{\cdot}\frac{\sum_{h\in\mathcal{H}} f_h}{\sum_{p\in\mathcal{P}} f_p} = \mathrm{m}(\mathcal{H}, t){\cdot}\frac{\sum_{h\in\mathcal{H}} \frac{f_h}{\mathrm{m}(\mathcal{H},t)}}{\sum_{p\in\mathcal{P}} \frac{f_p}{n}}$$
$$= \mathrm{m}(\mathcal{H}, t) \cdot \frac{\overline{f}_{\mathcal{H}}}{\overline{f}_{\mathcal{P}}} \quad (1)$$

As we expected, Equation 1 is exactly the same as that reached for the binary representation [4]. This is no coincidence, since the only properties used in the development of Equation 1 hold for both the new and the old definitions of schema.

The expressions describing the effects of the arithmetical operations, as well as the conventional crossover effect, however, are affected by the change in the definition of schema, and require new analysis.

2.2 Arithmetical crossover analysis

The real representation of chromosomes is commonly used in conjunction with the *arithmetical crossover* [2, 7]. This operator generates as the offspring of two selected chromosomes a new chromosome corresponding to the average of the values of its parents. We proceed to the analysis of this operator over the population of a schema. Initially, we will be considering the special case of one single variable in $\mathcal{U}$. Later, in Section 2.4, the results are extended to any number of variables.

We want to determine the probability that the result of the arithmetical crossover of an element of $\mathcal{H}$ with an element of $\mathcal{U}$ remains in $\mathcal{H}$. That is, taken $h \in \mathcal{H}$ and $u \in \mathcal{U}$ at random, we are looking for a function $c(l) = \mathrm{P}[\frac{h+u}{2} \in \mathcal{H}]$, where l is the width of $\mathcal{H}$.

Assuming that the populations of $\mathcal{H}$ and $\mathcal{U}$ are uniformly distributed[1], the mathematical expression for $c(l)$ can be obtained from the following equation:

$$c(l) = \frac{1}{1-l}\int_0^{1-l} \frac{1}{l}\int_s^{s+l} \Big\{ \min\big[2(s+l)-h, 1\big] - \max\big[2s-h, 0\big] \Big\}\, \mathrm{d}h\, \mathrm{d}s \quad (2)$$

[1]A simplifying assumption that only holds for the initial population but that was also followed during the development of the original study.

Program 1 Numeric simulation of $c(l)$. For `NL` values of l equally spaced in $[0,1]$, we choose `TS` random values for s in $[0,1-l]$. Then, after choosing h in $[s,s+l]$ and u in $[0,1]$, we test if the average lies in $[s,s+l]$.

```
#define uniform(s_, l_) (s_ + \
    (((double)l_)*random())/RAND_MAX)
#define in(a, b, x) (((x) > (a)) && ((x) < (b)))
#define NS 1000
#define TS 100000
#define NL 50
int main(void)
{
    double l, s, x;
    long c, i, t;
    for (l = 0.0; l <= 1.0; l += 1.0/NL) {
        c = 0;
        t = 0;
        for (i = 0; i < TS; i++) {
            s = uniform(0.0, 1.0-l);
            x = (uniform(s, l)
                + uniform(0.0, 1.0))/2.0;
            c += in(s, s+l, x);
            t++;
        }
        printf("%f %f\n", l, ((double) c)/t);
    }
}
```

Equation 2 can be better understood with the help of Figure 2. Initially, we choose $h \in \mathcal{H}$, or in other words, $s \leq h \leq s+l$. As we see in Figure 2, if we want to choose $u \in \mathcal{U}$ such that $s \leq \frac{h+u}{2} \leq s+l$, we must have $a \leq u \leq b$. Since $u \in [0,1]$, the intersection of the two conditions leads to $\max(0,a) \leq u \leq \min(1,b)$, where $a = 2s-h$ and $b = 2(s+l)-h$. It remains to integrate over all possible values of h and all possible values of s, as seen in Equation 2.

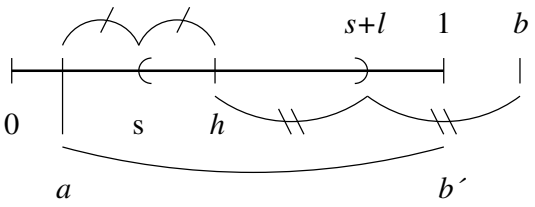


Figure 2: Interpretation of Equation 2. In the figure, $\min(1,b) = b' = 1$**, so that** u **is allowed to vary in the interval** $[a,b']$**.**

Determining an analytical solution for Equation 2, however, involves considerable work, since there are many different cases to be considered separately. Program 1 generates a sequence of points that approximates $c(l)$ for several values of l. The program simulates the random choice of elements from $\mathcal{H}$ and $\mathcal{U}$, determining if their average lies in $\mathcal{H}$. Figure 3 shows the result of the simulation, and gives the first picture of $c(l)$, plotted along with the analytical solution obtained below.

Fortunately, there is an equivalent formulation for Equation 2 that is easier to solve. Let H and U be random variables describing the distributions of the chromosome populations in $\mathcal{H}$ and $\mathcal{U}$, respectively. Let $\mathrm{Z} = \mathrm{H} + \mathrm{U}$ be another

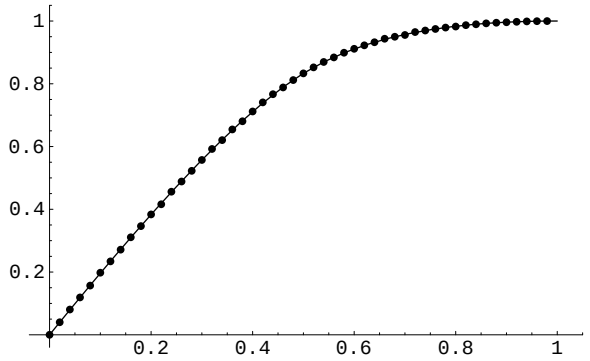


Figure 3: $c(l)$**. The dots represent the sequence of values generated by Program 1. The continuous line represents the analytic solution given by Equation 5.**

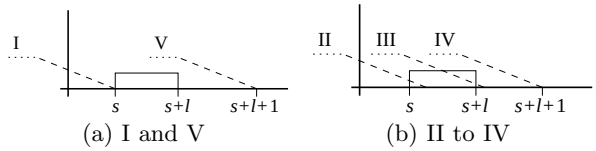
random variable, describing the sum distribution. We look for an expression describing the probability:

$$\begin{aligned} c(l) &= \mathrm{P}[s \leq \tfrac{\mathrm{Z}}{2} \leq s+l] \\ &= \mathrm{P}[2s \leq \mathrm{Z} \leq 2(s+l)] \\ &= \mathrm{P}[\mathrm{Z} \leq 2(s+l)] - \mathrm{P}[\mathrm{Z} \leq 2s] \end{aligned} \tag{3}$$

In order to solve Equation 3, we need to determine an expression for $\mathrm{P}[\mathrm{Z} \leq z]$. This expression is given by the solution of Equation 4, where $\mathrm{P}[\mathrm{H} = x]$ is the probability density function of H and $\mathrm{P}[\mathrm{U} \leq x]$ is the cumulative probability function of U.

$$\begin{aligned} \mathrm{P}[\mathrm{Z} \leq z] &= \mathrm{P}[\mathrm{H} + \mathrm{U} \leq z] \\ &= \int_{-\infty}^{\infty} \mathrm{P}[\mathrm{H} = x]\mathrm{P}[\mathrm{U} \leq z - x]\,\mathrm{d}x \end{aligned} \tag{4}$$

Equation 4 is clearly a convolution integral and can be solved by parts, with the division of the integration interval into simpler subintervals. Figure 4 shows the appropriate divisions.



(a) I and V (b) II to IV

Figure 4: Integration subintervals for Equation 4. For case I, $z < s$**, for case V,** $z > s+l+1$**. For cases II, III and IV,** z **lies in** $[s,s+l]$**,** $[s+l,s+1]$ **and** $[s+1,s+l+1]$**, respectively.**

Equations I to V give the different expressions for the convolution integral $\mathrm{P}[\mathrm{Z} \leq z]$, as a function of s and l, for each of the different subintervals of the variable z, as shown in Figure 4.

$$\mathrm{P}[\mathrm{Z} \leq z](s,l) = 0 \tag{I}$$

$$\mathrm{P}[\mathrm{Z} \leq z](s,l) = \int_s^z \frac{1}{l}(z-x)\,\mathrm{d}x \qquad \text{(II)}$$

$$\mathrm{P}[\mathrm{Z} \leq z](s,l) = \int_s^{s+l} \frac{1}{l}(z-x)\,\mathrm{d}x \qquad \text{(III)}$$

$$\mathrm{P}[\mathrm{Z} \leq z](s,l) = \int_s^{z-1} \frac{1}{l}\,\mathrm{d}x + \int_{z-1}^{s+l} \frac{1}{l}(z-x)\,\mathrm{d}x \qquad \text{(IV)}$$

$$\mathrm{P}[\mathrm{Z} \leq z](s,l) = 1 \qquad \text{(V)}$$

After solving all integrals in Equations I to V, we need to analyze the value of the expression $\mathrm{P}[\mathrm{Z} \leq z]$ for $z = 2s$ and $z = 2(s+l)$ before substitution in Equation 3. In each case, we integrate over all values of s, to reach one-variable functions of the width l of the schema. Since the integration limits are themselves functions of l, the choice of which of the cases I to V will be used for $z = 2s$ and $z = 2(s+l)$ also depends on l.

$$0 \leq l < \frac{1}{2} \Longrightarrow$$

$$\begin{cases} \mathrm{P}[\mathrm{Z} \leq 2s] = \frac{1}{1-l}\int_0^l \{\mathrm{II}\}\,\mathrm{d}s + \int_l^{1-l}\{\mathrm{III}\}\,\mathrm{d}s \\ \mathrm{P}[\mathrm{Z} \leq 2(s+l)] = \frac{1}{1-l}\int_0^{1-2l}\{\mathrm{III}\}\,\mathrm{d}s + \int_{1-2l}^{1-l}\{\mathrm{IV}\}\,\mathrm{d}s \end{cases}$$

$$\frac{1}{2} \leq l < 1 \Longrightarrow$$

$$\begin{cases} \mathrm{P}[\mathrm{Z} \leq 2s] = \frac{1}{1-l}\int_0^{1-l}\{\mathrm{II}\}\,\mathrm{d}s \\ \mathrm{P}[\mathrm{Z} \leq 2(s+l)] = \frac{1}{1-l}\int_0^{1-l}\{\mathrm{V}\}\,\mathrm{d}s \end{cases}$$

After substitution in Equation 3, we reach the final expression for $c(l)$. The graph for the analytical solution of $c(l)$ is shown as the continuous line of Figure 3.

$$\mathrm{c}(l) = \begin{cases} \dfrac{6l - 7l^2}{3(1-l)}, & 0 \leq l < \frac{1}{2}, \\ \dfrac{5l - l^2 - 1}{3l}, & \frac{1}{2} \leq l < 1. \end{cases} \qquad (5)$$

The graph of Equation 5 as seen in Figure 3 matches the results obtained with the simulation. As we could expect, the function is increasing on the width of the schema, ranging from zero to one as the width of the schema grows. In other words, low specificity schemata (with large widths) have a better chance surviving the arithmetical crossover.

2.3 One-point crossover

Whenever a problem involves several variables, a chromosome is defined as a string of genes, each gene representing the value of one of the variables considered in the problem. In these cases, it is common to use another operator, known as *one point crossover*. The operator acts as shown in Figure 5.

We have to analyze the probability that a or b, the chromosomes generated from a chromosome h of $\mathcal{H}$ and a random partner u taken from $\mathcal{U}$, will be a member of $\mathcal{H}$. For that, we must ensure that all genes a_i, for instance, fall within

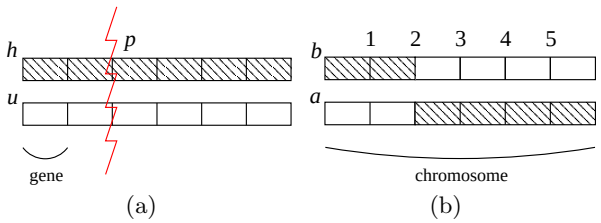


Figure 5: One-point crossover. (a) A random site p is chosen and the parent chromosomes h and u are cut at that point. (b) The recombination of the parent chromosomes yields two new chromosomes, a and b.

their respective intervals $[s_i, s_i + l_i]$, as defined by $\mathcal{H}$. Assuming u_i is a random value uniformly distributed in $[0, 1]$, the probability that it lies in $\mathcal{H}_i$ is given by l_i. It follows that, if p is the site chosen for the cut operation and n is the number of variables in $\mathcal{U}$:

$$\mathrm{P}[a \in \mathcal{H}](p) = \prod_{i=1}^{p} l_i$$

from which we deduce the final relation, taking into consideration all possible values for p:

$$\mathrm{P}[a \in \mathcal{H}] = \frac{1}{n-1}\sum_{p=1}^{n-1}\prod_{i=1}^{p} l_i \qquad (6)$$

To reach a simpler expression for Equation 6, we can restrict the analysis to schemata for which all l_i are equal. In that case, when $l_i = l$, we reach the following expression:

$$\begin{aligned} \mathrm{P}[a \in \mathcal{H}] &= \frac{1}{n-1}\sum_{p=1}^{n-1}\prod_{i=1}^{p} l \\ &= \frac{1}{n-1}\sum_{p=1}^{n-1} l^p = \frac{(l^n - l)}{(n-1)(l-1)} \end{aligned} \qquad (7)$$

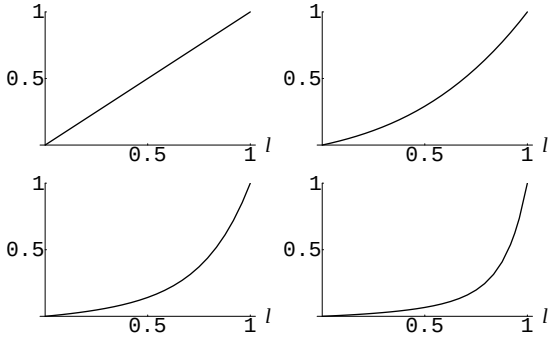


Figure 6: The probability that an individual will survive one-point crossover as a function of the schema width l, given by Equation 7, for 2, 4, 8 and 16 genes.

Figure 6 shows the graph for Equation 7 for several different values of n. Once again, we conclude that low specificity schemata are more likely to survive the crossover operation.

2.4 Arithmetical crossover revisited

In Section 2.2, we restricted our analysis of the arithmetical crossover to one gene chromosomes. However, the generalization of these results is simple, since the operator acts independently on each gene. We simply change the condition $s \leq \frac{h+u}{2} \leq s+l$ to $s_i \leq \frac{h_i+u_i}{2} \leq s_i+l_i$. That way we can make sure that each gene i of the result belongs to the interval $\mathcal{H}_i$ of the schema.

Using the expression for $\mathrm{c}(l)$ given by Equation 5, we reach the general expression for the arithmetical crossover, as follows:

$$\mathrm{c}(\mathcal{H}) = \mathrm{P}[\tfrac{h+u}{2} \in \mathcal{H}] = \prod_{i=1}^{n} \mathrm{c}(l_i) \tag{8}$$

Once again, the expression can be better visualized if we force $l_i = l$. In that case, it reduces to Equation 9, the graph of which is seen in Figure 7:

$$\mathrm{c}(\mathcal{H}) = \mathrm{c}^n(l) \tag{9}$$

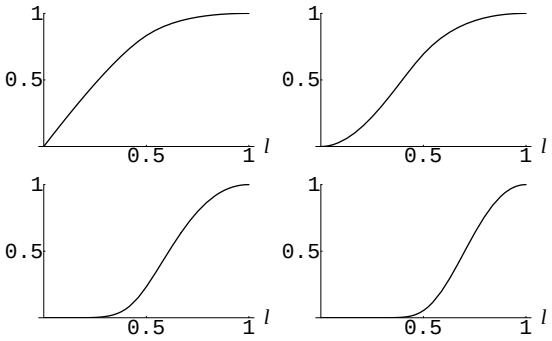


Figure 7: The probability that an individual will survive arithmetical crossover as a function of the schema width l, given by Equation 9, for 1, 2, 8 and 16 genes.

2.5 Creep mutation

Another commonly used operator is known as *creep mutation* [2, 7], and consists of the addition of a random value to a selected chromosome. In general, the added value is restricted to a known interval $[-d, d]$. We wish to analyze the effect of this operator over the number of individuals of a schema in the population. The problem is very similar to that of Section 2.2. Therefore, we will use the same technique to solve it.

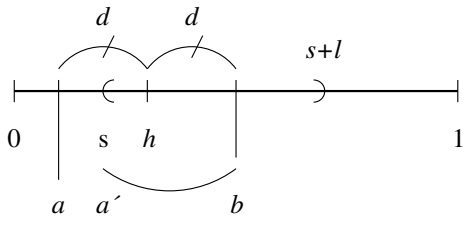


Figure 8: Interpretation of Equation 10. For the figure, $\max(s, p-d) = a' = s$, so that $p+m$ can vary in [a',b].

Program 2 Numeric simulation of $u(t)$. For l in $[0, 20d]$, `TD` random values of p and d are chosen. The number of sums falling within $[0, l]$ is then determined.

```
#define uniform(s_, l_) (s_ + \
    (((double)l_)*random())/RAND_MAX)
#define in(a, b, x) (((x) > (a)) && ((x) < (b)))
#define TD 50000
int main(void) {
    double l, p, x, dl;
    long c, i, t;
    for (dl = 0.1; l <= 10.0; l += dl) {
        dl += 0.005;
        c = 0; t = 0;
        for (i = 0; i < TD; i++) {
            p = uniform(0.0, l);
            x = p + uniform(-1.0, 2.0);
            c += in(0.0, l, x); t++;
        }
        printf("%f %f\n", l, ((double) c)/t);
    }
}
```

Given h in $\mathcal{H}$ and a random value m in $[-d, d]$, we look for an expression giving the probability that $h+m$ belongs to $\mathcal{H}$. Figure 8 shows the situation to be analyzed.

We notice that the sum $p+m$ spawns the interval $[p-d, p+d]$. We want a sum such that $s \leq p+m \leq s+l$. Therefore, the intersection of the two conditions leads to the interval $[\max(s, p-d), \min(s+l, p+d)]$. The length of this interval is then integrated over all values of p and s.

$$\mathrm{u}(l,d) = \int_0^{1-l} \frac{1}{1-l} \int_s^{s+l} \frac{1}{l}\frac{1}{2d} \big\{ \min(s+l, p+d) - \max(s, p-d) \big\}\, \mathrm{d}p\, \mathrm{d}s \tag{10}$$

A numeric approximation for the solution of Equation 10 can be seen in Figure 9, plotted along with the analytical solution obtained below. The dots represent the values obtained by the simulation performed by Program 2.

Again, finding an analytic solution by direct integration involves a considerable amount of work, so we prefer the solution by the alternative convolution formulation. This time, let D be the random variable for the uniform distribution in the interval $[-d, d]$. Since the origin s of the schema makes no difference for the result, we define H as the random variable uniformly distributed in the interval $[0, l]$. Now, let $\mathrm{W} = \mathrm{H} + \mathrm{D}$. We are want a function describing the probability

$$\mathrm{u}(l,d) = \mathrm{P}[0 \leq \mathrm{W} \leq l] = \mathrm{P}[\mathrm{W} \leq l] - \mathrm{P}[\mathrm{W} \leq 0] \tag{11}$$

The procedure used to find $\mathrm{u}(l,d)$ from an expression of $\mathrm{P}[\mathrm{W} \leq w]$ is the same as that followed for the arithmetical crossover analysis, at the end of which we reach

$$\mathrm{u}(l,d) = \begin{cases} \dfrac{l}{2d}, & 0 \leq l < d, \\ \dfrac{2l-d}{2l}, & l \geq d. \end{cases} \tag{12}$$

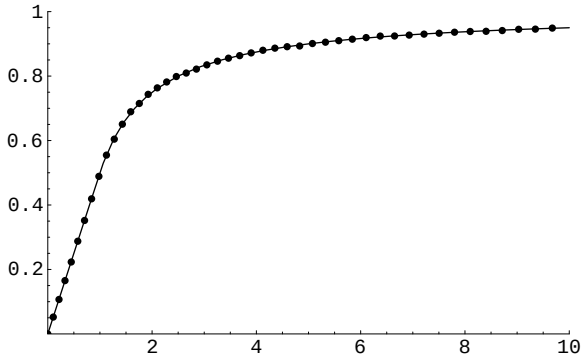


Figure 9: $u(t)$. Parameter t represents the ratio l/d. The dots are the values obtained by the simulation performed by Program 2. The continuous line is a graph of the analytical solution given by Equation 13.

Observing Equation 12, we realize that it is actually a function of the ratio $t = l/d$ between the schema width l and the maximum mutation value d. Rewriting the equation to replace l/d with t, we reach Equation 13. The graph for this new formulation can be seen in Figure 9.

$$u(t) = \begin{cases} \dfrac{t}{2}, & 0 \leq t < 1, \\ \dfrac{2t-1}{2t}, & l \geq 1. \end{cases} \tag{13}$$

The graph of Equation 13 in Figure 9 perfectly matches the numeric simulation. We see that $u(t)$ tends to one as t tends to infinity ($l \gg d$). Moreover, as expected, the function goes to zero when t tends to zero ($d \gg l$). Yet again, we conclude that low specificity schemata have better chances of survival.

3. TRACKING SCHEMATA

In order to better visualize the population of a given schema throughout several generations, a interactive program was constructed. The program runs a real-coded genetic algorithm to find the maximum value of the Shaffer's F6 function:

$$\mathrm{F6}(x,y) = 0.5 + \frac{\sin^2\left(\sqrt{x^2+y^2}\right) - 0.5}{\left(1 + 0.001(x^2+y^2)\right)^2}$$

The F6 function is a a two-dimensional multimodal function with its global maximum at the origin, commonly used to test genetic algorithms [2].

The program allows one to choose the schema to be inspected, and tracks the percentage of individuals that belong to that schema, in all generations of the population. By inspecting the evolution of the number of individuals of several schemata, we can confirm, in practice, the theory that we have developed.

Figure 10 shows the evolution of the population of some schemata throughout 40 generations. Highest fitness chromosomes lie close to the origin, where the function has its global maximum. We clearly see that the population is

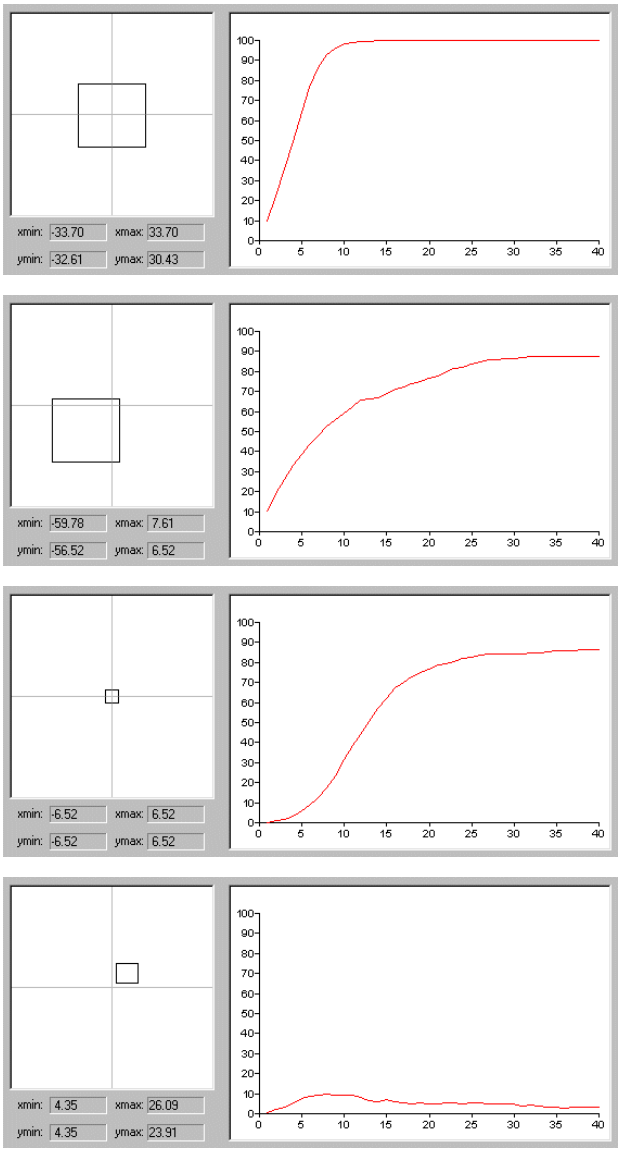


Figure 10: Percentage of total population belonging to low specificity schemata (top two) and high specificity schemata (bottom two), within several generations.

quickly dominated by schemata representing these chromosomes. Comparing the plots in Figures 10, we also note that, as expected, low specificity schemata grow much faster in population than high specificity schemata.

4. CONCLUSIONS

The analytical solution to all integrals have been double-checked with graphs of their numerical solution using the software *Mathematica* to make sure they match. The results seen in Equations 1, 8 and 13, give us a model similar to that which the Schema Theory provides for the binary coding of chromosomes. The new model holds for the real coding and the arithmetical operators. All that is left is to join all results into a single equation.

Let μ be the probability that a selected chromosome will mutate, and let α be the probability that there will be crossover. Then, the following equation holds:

$$\mathrm{m}(\mathcal{H}, t+1) \geq \mathrm{m}(\mathcal{H}, t) \cdot \frac{\overline{f}_{\mathcal{H}}}{\overline{f}_{\mathcal{P}}} \cdot [1 - \alpha + \alpha \mathrm{c}(\mathcal{H})] \cdot [1 - \mu + \mu \mathrm{u}(\mathcal{H})] \quad (14)$$

The qualitative analysis of Equation 14 shows that the conclusions reached by the Fundamental Theorem of Genetic Algorithms also hold for the real representation. That is, schemata with above average fitness with low specificity tend to proliferate quickly within generations.

5. REFERENCES

[1] Jim Antonisse. A new interpretation of schema notation that overturns the binary encoding constraint. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers, 1989.

[2] Lawrence Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

[3] Kenneth A. De Jong, William M. Spears, and Diana F. Gordon. Using markov chains to analyze GAFOs. In L. Darrell Whitley and Michael D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 115–137. Morgan Kaufmann, San Francisco, CA, 1995.

[4] David Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Adison-Wesley, 1989.

[5] William A. Greene. A non-linear schema theorem for genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 189–194. Morgan Kaufmann, July 2000.

[6] John Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[7] Zbigniew Michalewicz, editor. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1996.