

# THE STATE EQUATION OF A GENETIC ALGORITHM

Radu BELEA

*Department of Electronics and Telecommunications  
"DUNĂREA DE JOS" University of Galați  
1 Științei street 800146 Galați Romania  
e-mail: radu.belea@ugal.ro*

**Abstract:** *The sense of evolution in a genetic algorithm is explained by the schema theorem. This paper proposes a new construction named "the state equation of the genetic algorithm" that is a generalisation of schema theorem and incorporates all previous results. Every genetic algorithm is a dynamic process whose dynamics is given by the state equation. The experimental results were acquired on the dominant individual problem, by integrating the state equation.*

**Keywords:** *genetic algorithm, schema, fitness function, selection pressure, pattern of the schema, kernel of the schema, the state equation of a genetic algorithm.*

## 1. INTRODUCTION

A genetic algorithm (GA) is an optimisation method with two empirical sources: the Darwinian evolutionism and the random search optimisation methods. Goldberg [3] states: "there is a pre-history of genetic algorithms, but the true history begins with the schema theorem given by Holland in 1985".

The schema theorem, also known as the fundamental theorem of genetic algorithms is the moment to go beyond the empirical state of art and make the first steps in structuring the genetic algorithm field. Today, in genetic algorithms, on one hand, there is still empiricism, and, on the other hand, there is an axiomatic base still incomplete.

The schema theorem has not the rigorous style of a geometrical theorem, because it only explains the sense of evolution in a genetic algorithm. E.g. in [5] the schema theorem is presented in 3<sup>rd</sup> chapter entitle "GAs Why Do They Work?"

A mention should be done. The schema theorem has one enounce and three proofs: Holland 1985 admits three simplifying hypothesis (see section

2.3); Schaffer 1987 reduces one of the simplifying hypothesis; Stephens and Waelbroek 1997 reduce the entire simplifying hypothesis. This proof is named the *exact schema theorem*.

The paper is structured as in follows: Chapter 2 is a short introduction of the most important elements of GA theory and the schema theorem is presented. Also the dynamic system sense of final equation given by the schema theorem is highlighted. Chapter 3 presents the state equation construction method, and the algorithms for the calculus of the equation coefficients. Chapter 4 presents an application of schema theorem: the dominant individual problem, a simple theoretical problem that permits the study of the population dynamics. Chapter 5 contains the conclusions.

## 2. BACKGROUND IN GA THEORY

The genetic algorithm theory is made up two fundamental principles: the schema theorem and the building block hypothesis. This chapter is a short presentation of the theorem and its consequences while the building block hypothesis is presented in section 3.3.

### 3.1. Introduction in artificial genetics

A genetic algorithm operates with three entities: the *artificial environment*, the *individual* and the *population*.

The relation between an individual and the environment is measured by the value of its fitness function value  $f(x_i)$ , where  $x_i$  is the fitness function parameter vector acquired by decoding the individual genetic information. The fitness function and all the rules that generate the competitive relations between individuals are the artificial environment.

The individuals are rudimentary beings with one chromosome. The individual behaviour is strictly genetic determined and is unchanged in its life. In other words, the individuals are unteachable. Each individual is a possible solution of the given optimisation problem.

A set whose elements can repeat is a metaset. For example the population is a metaset of individuals because it is possible that two individuals have identical chromosomes. The elements of a metaset are called instances. We note  $P(t)$  the population, because its composition changes in time. The time is measured in generations.

Let  $N$  be the population size. The population has an average fitness function.

$$f(P(t)) = \frac{1}{N} \sum_{i=1}^N f(x_i) \quad (1)$$

Also, the average value of fitness function  $f(P(t))$  is time dependent.

The probability that an individual will be selected for reproduction proportionally with its fitness function value is:

$$p_i = \frac{f(x_i)}{N \cdot f(P(t))} \quad (2)$$

### 3.2. The schema concept

The chromosome is a string of bits, characters of the alphabet  $V = \{0, 1\}$ , as an example '11010' is a five-bit string. The string set is  $\{0, 1\}^L$ , and there are  $2^L$  distinct strings of  $L$  length. We must specify, the terms: individual, chromosome and string have the same meaning.

The classical genetic algorithm theory operates with the schema concept. A schema is a string of characters of the alphabet  $V_+ = \{*, 0, 1\}$ , where the character '\*' is a meta-symbol that indicates any '0' or '1' symbols. The schema set is

$\{*, 0, 1\}^L$ , and there are  $3^L$  distinct schemas of  $\{0, 1, *\}$  length. Also a schema is a set of strings. E.g. the schema '110\*\*' is the set

$$\{'11000', '11001', '11010', '11011'\} \quad (3)$$

In the schema theorem another two dimensions are used: the size of the schema  $\delta(H)$ , that is the distance between the first and the last fixed characters and the order of the schema  $o(H)$ , that is the number of fixed characters. For example, the schema  $H = '*01**1**'$  has  $\delta(H) = 4$  and  $o(H) = 3$ .

The intersection operator in  $H \cap P(t)$  is used to filter the instances of  $H$  schema in the population  $P(t)$ . The  $|\cdot|$  operator calculates the cardinal of a set or metaset. For example  $|H \cap P(t)|$  is the number of instances of the  $H$  schema in the population at the  $t$  generation.

### 3.3. The schema theorem

The whole theory of genetic algorithms is developed on the SGA (Simple Genetic Algorithm). This algorithm uses the binary coded genes; one point crossover and bit defined mutation probability. At each generation the entire the population is replaced.

In [5] pp. 51 the next statement of the schema theorem can be found:

"Short, low order, above-average schemata receive exponentially increasing trials in subsequent generations of a genetic algorithm."

Let us consider  $H$ , a schema existent in the population at the  $t$  generation. In the first proof of the theorem, Holland allows three simplifying hypothesis:

- (a) it is a small probability to mate two individuals of  $H$  schema;
- (b) it is a small surviving probability of the  $H$  schema if the crossover point occurs inner the schema;
- (c) it is a small probability to generate the  $H$  schema by mate two different individuals.

The Holland's proof is presented in [3], [5] or in the appendix of [2]. Schaffer in 1987 gives a new proof in which the hypothesis (a) was eliminated. The Schaffer proof is presented in [7] or in the appendix of [2]. All the demonstrations of the schema theorem states the equation:

$$\begin{aligned} \langle |H \cap P(t+1)| \rangle &\geq \\ &\geq \varphi_H(t) \alpha(H) \beta(H) |H \cap P(t)| \end{aligned} \quad (4)$$

where  $\langle \cdot \rangle$  is the expected value operator hence  $\langle |H \cap P(t+1)| \rangle$  is the expected value of the schema instances number at the  $t+1$  generation. In (1)  $\varphi_H(t)$  is the selection pressure of the schema,  $\alpha(H)$  is the probability with that the crossover operator will not disrupt the  $H$  schema,  $\beta(H)$  is the probability with that the mutation operator will not modify a fixed bit of  $H$  schema and  $|H \cap P(t)|$  is the number of instances of the schema at the  $t$  generation.

In all the previous deductions was established the equation of the  $H$  schema selection pressure

$$\varphi_H(t) = \frac{f(H \cap P(t))}{f(P(t))}, \quad (5)$$

where:  $f(H \cap P(t))$  is the average fitness function value of the  $H$  schema in the population  $P(t)$  and  $f(P(t))$  is the population average fitness value. The selection pressure of  $H$  schema must be evaluated at each generation.

In the Holland's demonstration  $\alpha(H)$  is a static coefficient:

$$\alpha(H) = 1 - p_c \frac{\delta(H)}{L-1} \quad (6)$$

where,  $p_c$  is the crossover probability,  $\delta(H)$  is the length of the schema and  $L$  is the length of the chromosome. In the Schaffer demonstration  $\alpha(H, t)$  is a time depending coefficient:

$$\alpha(H, t) = 1 - p_c \frac{\delta(H)}{L-1} \left( 1 - \frac{|H \cap P(t)|}{N} \varphi_H(t) \right) \quad (7)$$

and all the new coefficients were before explained. We note that in first generations

$$\frac{|H \cap P(t)|}{N} \rightarrow 0 \quad (8)$$

the low probability to mate two genetic identical individuals (8) allows the deduction of equation (6) from (7).

In all the deductions the probability that the mutation operator will not modify a fixed bit of  $H$  schema is:

$$\beta(H) = (1 - p_m)^{o(H)} \quad (9)$$

where  $p_m$  is the mutation probability and  $o(H)$  is the order of the schema.

### 3.4. The schema theorem results

A GA is also a dynamic process. E.g. Liepins and Vose, in [4] notice that the equation (4) has the aspect of a discrete differential equation in the expected values of the schema instances.

In a GA with complete substitution of population at each generation, the time is a discrete quantity. If in (4) we consider:

$$\begin{aligned} y_H(t+1) &= \langle |H \cap P(t+1)| \rangle, \\ y_H(t) &= |H \cap P(t)|, \end{aligned} \quad (10)$$

and after the substitution of the operator " $\geq$ " with the operator " $=$ " we get the discrete differential equation:

$$y_H(t+1) = \varphi_H(t) \alpha(H) \beta(H) y_H(t) \quad (11)$$

if in Holland model (11) we note:

$$r = \varphi_H(t) \alpha(H) \beta(H) \quad (12)$$

we get the schema evolution exponential model:

$$y_H(t+1) = r^t y_H(t). \quad (13)$$

That explains why in a few generations the feat schemas have an exponential increasing evolution and the weak schemas are fastly disappearing from the population.

## 3. THE STATE EQUATION

Specific for the all variants of the schema theorem is the constructive style of the proof. In this chapter a new construction that incorporates all the Holland and Schaffer results is presented. It does not need the simplifying hypothesis and it is equivalent with the Stephens and Waelbroek construction, [6], but is more general.

### 3.1. The state equation construction

Let us consider a genetic algorithm with complete substitution of population at each generation. If the length of chromosome is  $L$ , there are  $3^L$  distinct schemas. After choosing an indexing rule the state variable vector is

$$y(t) = \parallel |H \cap P(t)| \parallel \quad (14)$$

The selection process is the application  $P(t) \mapsto P'(t)$  that selects the mating pool  $P'(t)$

from the population  $P(t)$ . The discrete model of the selection process is

$$\| \langle H \cap P'(t) \rangle \| = \| \varphi(t) \| \times y(t) \quad (15)$$

where:  $\| \langle H \cap P'(t) \rangle \|$  is the expected values vector of instance number of schemas selected for reproduction  $P'(t)$  is the matting pool. We can notice that a schema that is not present in the population  $P(t)$  cannot be selected for reproduction. As a consequence,  $\| \varphi(t) \|$ , the selection matrix is a diagonal matrix:

$$\varphi_{j,k}(t) = \begin{cases} \frac{f(H_j \cap P_t)}{f(P_t)} & \text{for } j = k \\ 0 & \text{for } j \neq k \end{cases} \quad (16)$$

The  $\varphi_{j,k}(t)$  coefficient must be calculated at each generation because the population composition is time changing.

The reproduction process is the application  $P(t) \mapsto P''(t)$ , where  $P'(t)$  is the matting pool and  $P''(t)$  is the new generation. The actions of the crossover and mutation operators can be described as:

$$\begin{aligned} \| \langle H \cap P''(t) \rangle \| &= \\ &= \| \beta \| \times \| \alpha \| \times \| \langle H \cap P'(t) \rangle \| \end{aligned} \quad (17)$$

where:  $\| \langle H \cap P''(t) \rangle \|$  is the expected values vector of instance number of the  $H$  schema in the new generation,  $\| \alpha \|$  is the crossover matrix and  $\| \beta \|$  is the mutation matrix.

After  $P(t+1) = P''(t)$ , the replacement of the whole population with the new generation, we can note:

$$\| \langle H \cap P(t+1) \rangle \| = y(t+1) \quad (18)$$

the expected values vector of instance number of the  $H$  schema in the new generation.

Substituting equation (14) and (18) in (17) we obtain the GA state equation:

$$y(t+1) = \| \beta \| \times \| \alpha \| \times \| \varphi(t) \| \times y(t) \quad (19)$$

This is the generalisation of equation (11) and if the coefficients of the crossover and mutation matrices are calculated with equations (6) and (9), the equation (19) incorporates all the previ-

ous results. The calculus method of the  $\alpha_{j,k}$ , and  $\beta_{j,k}$  coefficients will be detailed later.

### 3.2. The pattern of the schema

The following development of GA theory was generated by the question “what is happening when the crossover operator disrupts a schema or the mutation operator modifies a fixed bit of the schema?”

**Definition 1:** Let  $H_1$  and  $H_2$  be two  $L$  length schema. The two schemas have the same pattern if they have the same number of fixed characters and the fixed characters are in the same positions.

The pattern of the schema is a string made of alphabet characters  $\{*, \#\}$ . The character  $*$  has the same signification as in the schema definition: unfixed character. The character  $\#$  has the signification: fixed character but which is not specified yet. Obviously if a schema is a set of strings, a pattern is a set of schemas. The set  $\{*, \#\}^L$  is the set of all  $L$  length patterns and there are  $2^L$  distinct patterns. The patterns are noted with Greek lowercase letters.

**Theorem 1:** Let  $\pi$  a pattern and  $H \in \pi$  a schema. If a genetic operator, crossover or mutation modifies one or more fixed bits of the schema, consequently the  $H$  schema modifies in the  $H' \in \pi$  schema.

**Proof:** If the genetic operator modifies one or more fixed bits of the schema the fixed bit positions are the same and the pattern preserves.  $\square$

The definition 1 also induces a new theorem:

**Theorem 2:** Let a  $L$  length kind of strings and a population  $P(t)$  be. All the  $L$  length patterns are present in the  $P(t)$  population.

**Proof:** In [2] there is a well-structured proofing using the schema operations properties. It can be stated as follows: if we get an arbitrary individual from the population  $P(t)$  and we get an arbitrary pattern from pattern set  $\{*, \#\}^L$  and we compare them, we find a unique schema that has all the fixed characters from the string and all the unfixed characters from the pattern.  $\square$

The theorem proofs the principle known as *the implicit parallelism of genetic algorithms*. The principle states:

“At any time the genetic algorithm works on more schemas than individuals”.

The proof is immediate: at any time, all the patterns are present in the population. At least one of the schemas of an arbitrary chosen pattern is populated. In other words  $2^L$  (number of patterns), is greater than  $N$ , size of population.

The previous enounce is true, even if the population is degenerated (a quasi-uniform population with small differences between individuals).

The pattern set  $\pi_i \in \{\#, * \}^L$  and the schema set of a given pattern are indexed sets. There are two simple rules for indexing the patterns and the respectively schemas of a given pattern.

**Rule 1:** Making the substitutions  $\# \rightarrow '1'$  and  $* \rightarrow '0'$ , we get an index  $i = 0 \dots 2^L - 1$  that uniquely identifies each pattern.

**Rule 2:** The unsigned integer number, built with all the fixed characters of a schema, taken in the same order, uniquely identifies the schemas of a given pattern.

Let us consider  $\pi$  a pattern taken arbitrary. All the schemas of the  $\pi$  pattern have the same dimension and the same order, so that we can write  $\delta(\pi)$  and  $o(\pi)$ . That means that the span of the indexing rule 2 is  $0 \dots 2^{o(\pi)} - 1$ .

If the  $3^L$  state equations are first ordered with rule 1 and applying the rule 2 within a pattern we get the ordered form of state equation. In this way the crossover operator is:

$$\|\alpha\| = \begin{pmatrix} \|\alpha\|_0 & & & 0 \\ & \|\alpha\|_1 & & \\ & & \ddots & \\ 0 & & & \|\alpha\|_{2^L-1} \end{pmatrix}, \quad (20)$$

where the square blocks  $\|\alpha\|_i$  are corresponding to the patterns  $\pi_i$ ,  $i = 0 \dots 2^L - 1$ . The mutation matrix has the same form.

### 3.3. The state equation size reducing method

As specified above, the second principle of classical GA theory is “the building block hypothesis”. In [5] pp. 51 there are the next statement of the hypothesis:

“A genetic algorithm seeks near-optimal performance through the juxtaposition of short,

low-order, high performance schemata, called building blocks”

Also, paper [1] is a survey on the role of the building block hypothesis in a GA. Briefly, when an experimenter solves a problem with a GA, he assumes that the GA can find the optimal solution. There are problems especially build to encumber a genetic algorithm. These artificial constructions are known as “fully deceptive problems”.

If the GA solves a problem in a way that the building block hypothesis is true, the next theorem suggests the way to reduce the state equation size.

**Theorem 3:** Let a GA be in which the building block hypothesis is true, thus the study of GA dynamics is sufficient to study the dynamics of the schemas from a single pattern arbitrary taken.

**Proof:** Theorem 1 states: “the pattern preserves indifferent of the way in which the genetic operators modifies the fixed bits of the schema of the same pattern”. Theorem 2 states: at any time all the patterns are present in population.

If the building block hypothesis is true, the solution of the genetic algorithm is the “population winning individual” also, in each pattern competition result is the “winning schema”. The keyword of the previous sentence is “each”. In the deceptive problems (exceptions of this theory), the second affirmation is not true.

The competition in a genetic algorithm is generating by individual-medium interaction specified by the individual fitness value. The selection process reflects this competition.

The competition process is not determinate by schemas. On the contrarily, the competition between schemas reflects the competition between individuals. All the schemas of a pattern are competitive, and the competition is taken simultaneously in all the patterns. This competition reflects the changes in the population and does not depend on the pattern parameters.

Let  $\pi$  be a pattern taken arbitrary. Due to the theorem 3 the size of the state equation reduces from  $3^L$  to  $2^{o(\pi)}$ . In this case we can write

$$y_\pi(t+1) = \|\beta\|_\pi \times \|\alpha(t)\|_\pi \times \|\varphi(t)\|_\pi \times y_\pi(t), \quad (21)$$

The coefficients of  $\|\varphi(t)\|_\pi$  square block are directly calculated with equation (5), but the calculus of  $\|\alpha(t)\|_i$  and  $\|\beta\|_i$  coefficients are more difficult if we want to eliminate the entire simplifying hypothesis.

### 3.4. The $\alpha_{j,k}$ and $\beta_{j,k}$ coefficients

Let us consider  $\pi$ , the pattern used in equation (21). For the calculus of the crossover matrix we can separate the matrix in two terms as it follows:

$$\|\alpha(t)\|_\pi = \|\alpha'(\pi, t)\|_\pi + \left( I_\pi - \|\alpha'(\pi, t)\|_\pi \right) \times \|\alpha''\|_\pi \quad (22)$$

In (22) the first term is a diagonal matrix with the probabilities that the crossover point does not occur inside the schemas. The  $\alpha'(\pi, t)$  coefficients are

$$\alpha'(\pi, t) = \begin{cases} \alpha(H_j, t) & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases} \quad (23)$$

where  $\alpha(H_j, t)$  are calculated with equation (7).

In (22) the second term is a square matrix where  $\alpha''_{j,k}$  are the probabilities to generate the  $H_j$  schema from the  $H_k$  schema. This matrix is weighted with the probabilities that the crossover point occur inner the schemas (the round parenthesis).

For the calculus of the  $\alpha''_{j,k}$  coefficients it is necessary to introduce two new entities: the *kernel of the schema* and the *kernel of the pattern*.

**Definition 2:** Let  $H \in \pi$  be a  $L$  length schema. The kernel is a function that returns a sub-string of the schema or a pattern that begins with the first fixed character and ends with the last fixed character.

As an example, let the schema  $H = '*10**0**'$  and its pattern  $\pi = '*\#\#\#\#\#*'$  be. Their kernels are:

$$\begin{aligned} \ker(H) &= \ker('*10**0**') = '10**0' \\ \ker(\pi) &= \ker('*\#\#\#\#\#*') = '\#\#\#\#' \end{aligned} \quad (24)$$

In (24) we notice that is possible to exist many schemas that have the same kernel, this notice remaining true for the patterns.

For a given pattern kernel the probabilities  $\alpha''_{j,k}$  can be generated with a program that does a systematic exploration of the all the schemas kernel. In table 1 there are presented the square block coefficients  $\|\alpha''_{j,k}\|_{\ker(\pi)}$ , where  $\ker(\pi)$  is the string ' $\#\#\#\#$ '.

**Table 1.** The crossover matrix for the kernel ' $\#\#\#\#$ '

|         |       |       |       |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|
| '00**0' | 0.394 | 0.194 | 0.116 | 0.039 | 0.142 | 0.013 | 0.090 | 0.013 |
| '00**1' | 0.194 | 0.394 | 0.039 | 0.116 | 0.013 | 0.142 | 0.013 | 0.090 |
| '01**0' | 0.116 | 0.039 | 0.394 | 0.194 | 0.090 | 0.013 | 0.142 | 0.013 |
| '01**1' | 0.039 | 0.116 | 0.194 | 0.394 | 0.013 | 0.090 | 0.013 | 0.142 |
| '10**0' | 0.142 | 0.013 | 0.090 | 0.013 | 0.394 | 0.194 | 0.116 | 0.039 |
| '10**1' | 0.013 | 0.142 | 0.013 | 0.090 | 0.194 | 0.394 | 0.039 | 0.116 |
| '11**0' | 0.090 | 0.013 | 0.142 | 0.013 | 0.116 | 0.039 | 0.394 | 0.194 |
| '11**1' | 0.013 | 0.090 | 0.013 | 0.142 | 0.039 | 0.116 | 0.194 | 0.394 |

In Table 1 there are 8 rows and 8 columns, corresponding to schemas kernel indexed with rule no. 2. We notice that  $\alpha''_{j,k} = \alpha''_{k,j}$  and the sum of probabilities of each row or column is 1.

The  $\beta_{j,k}$  coefficients are simpler to calculate because the probability to generate the  $H_j$  schema from the  $H_k$  schema depends on the Hamming distance between  $j$  and  $k$  indexes build with the indexing rule 2. The Hamming distance between two unsigned integer number in binary representation is:

$$n = bc(j \oplus k) \quad (25)$$

where  $bc(\cdot)$  is the function “bit count” which returns the number of '1' bits of the argument and the operator  $\oplus$  is the bit-wise exclusive-or operator. The  $\beta_{j,k}$  coefficients are calculated with equation:

$$\beta_{j,k} = p_m^n (1 - p_m)^{o(\pi) - n} \quad (26)$$

where  $p_m$  is the bit mutation probability,  $n$  is calculated with the equation (24) and  $o(\pi)$  is the order of the pattern.

## 4. EXPERIMENTAL RESULTS

Goldberg, in [3] p. 55-63, presents one of the first studies on the GA-dynamics on the problem known as “the prisoner’s dilemma”. In fact there are two prisoners who, at each iteration, independently takes the decision to cooperate or to defect. In this situation the GA individual chromosome has two bits and the landscape of

the fitness function is given by the values of the four points  $f(00)$ ,  $f(01)$ ,  $f(10)$  and  $f(11)$ .

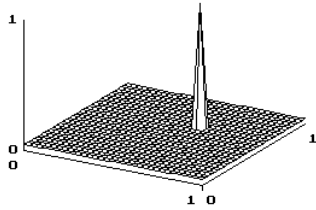
In this problem there are four discrete differential equations that can be integrated with an initial condition on diverse landscapes.

This chapter introduces the problem named “the dominant individual problem” that permits the algebraic evaluation of the selection pressure of a given schema. The following experimental results were acquired on this problem.

#### 4.1. The dominant individual problem

Let a SGA algorithm be. In the initial population all the individuals have the same the fitness function value, with the exception of a single individual who has greater fitness function value in comparison with the rest of population. It is required to study the dynamic evolution of the population. In Fig. 1 the landscape of the dominant individual problem is presented.

It is obviously that the individual who has the best fitness value dominates the selection process. The dominant individual problem looks to be very simple, but this problem is a good model for the situation in which an individual arrives in an ecological niche in which there are better conditions. The arrival moment starts the dynamical process for moving the entire population from the present positions to the new ecological niche.



**Fig. 1.** The landscape in the dominant individual problem

It is obviously that the individual who has the best fitness value dominates the selection process. The dominant individual problem looks to be very simple, but this problem is a good model for the situation in which an individual arrives in an ecological niche in which there are better conditions. The arrival moment starts the dynamical process for moving the entire population from the present positions to the new ecological niche.

#### 4.2. The solution of the dominant individual problem with the Holland model

Let us consider  $H$  an arbitrary given schema of the dominant individual. In the population there are two disjunctive groups:

- $\{H\}$  the group of individuals that exists in the new ecological niche;
- $\{\bar{H}\}$  the groups of individuals that exists in the rest of landscape.

If  $N$  is the size of population then

$$|H \cap P(t)| + |\bar{H} \cap P(t)| = N \quad (27)$$

If  $|H \cap P(t)| = y_H(t)$ , the  $H$  schema instance number in population  $P(t)$  is known (see equation (10));  $\phi_H(t)$ , the selection pressure of  $H$  schema is calculated with (5) equation then

$$\phi_H(t) = \frac{y_H(t) f(H)}{y_H(t) f(H) + (N - y_H(t)) f(\bar{H})} \quad (28)$$

where  $f(H)$  is the fitness function value in the new ecological niche and  $f(\bar{H})$  is the fitness function value the rest of landscape. Substituting (28) in (11) we get:

$$y_H(t+1) = \frac{\alpha(H) \beta(H)}{1 + \frac{N - y_H(t)}{y_H(t)} \cdot \frac{f(\bar{H})}{f(H)}} y_H(t) \quad (29)$$

The  $f(H)/f(\bar{H})$  factor is a modelling parameter that quantifies the dominance degree of the dominant individual.

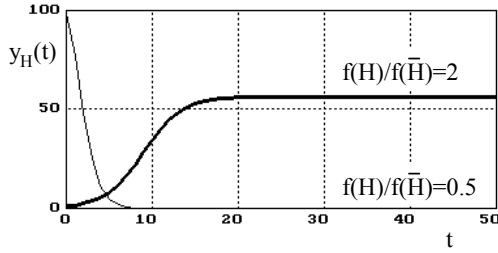
Equation (29) is integrated with the SGA algorithm in the next conditions:

- the length of the chromosome  $L = 16$  ;
- the population size  $N = 100$  ;
- the length of the  $H$  schema  $\delta(H) = 5$  ;
- the order of the  $H$  schema  $o(H) = 3$  ,
- the bit mutation probability  $p_m = 0.05$  ;
- the crossover probability  $p_c = 1$  ;
- $\alpha(H)$ , the probability that the crossover operator will not disrupt the  $H$  schema is calculated with equation (6);
- $\beta(H)$ , the probability that the mutation operator will not modify a fixed bit of the  $H$  schema is calculated with equation (9).

For the modelling parameter  $f(H)/f(\bar{H})$  and the initial population  $P(0)$  the simulation is made for two condition sets:

$$\begin{cases} f(H)/f(\bar{H}) = 2 \\ |H \cap P(0)| = 1 \end{cases}, \quad (30)$$

$$\begin{cases} f(H)/f(\bar{H}) = 0.5 \\ |H \cap P(0)| = 99 \end{cases}. \quad (31)$$



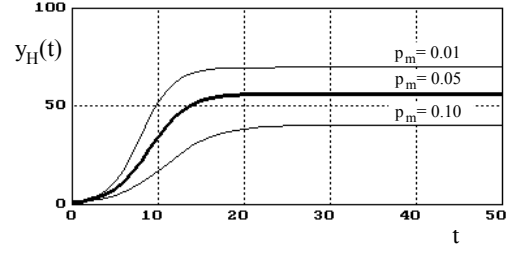
**Fig. 2.** Influence of  $f(H)/f(\bar{H})$  modelling parameter

In Fig. 2, on the abscissa is represented the time expressed in generations, and in the ordinate is represented  $y_H(t)$  expressed in instances. The time evolution of the  $H$  schema instance number in population is plotted with thick line for the first condition set, and is plotted with thin line for the second condition set. We can notice:

- When  $f(H)/f(\bar{H}) = 1$ , the feat  $H$  schema has an initially exponential evolution. After a few generations, composition of the population changes and the process evolves to a final stable state;
- When  $f(H)/f(\bar{H}) < 1$ , the  $H$  schema is weak and disappears fastly from the population.

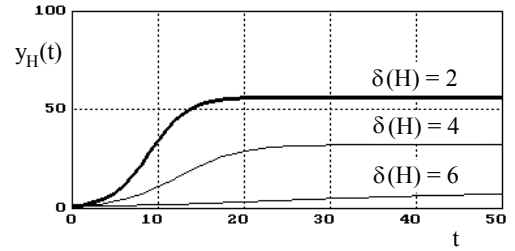
More tests were made to study the influence of mutation probability  $p_m$  and  $H$  schema parameter  $\delta(H)$  on the evolution speed and the final stable state. Further on the tests are made in condition set (30).

Fig. 3. presents the effect of the mutation probability on the speed of the evolution and the stable final value. We notice that the evolution speed and the final stable value depend on the decreasing of a great mutation probability. In GA practice the fact is very well known that a mutation probability of 0.1 is very great.



**Fig. 3.** Influence of the mutation probability in the population dynamics

Fig. 4 presents the effect of the length of the chromosome on the speed of the evolution.



**Fig. 4.** Influence of the length of the schema in the population dynamics

The result from Fig. 4 proofs that the short schemas evolve more rapidly than the long ones. Also similar tests to study the influence  $H$  schema parameter  $\phi(H)$  on the evolution speed were made, but the results are similar with those presented in Fig. 4.

#### 4.3. The solution of the dominant individual problem with the state equation

The goal of this section is to compare the results of the state equation with those given by the Holland model or Schaffer model on the dominant individual problem.

Further on is used the  $\pi$  pattern, which contain the kernel '###'. Because  $\delta(\pi) = 5$  and  $\phi(\pi) = 3$ , the parameters of the  $\pi$  pattern are the same the pre considered  $H$  schema. The pervious experimental conditions are preserved.

For all the tree models the fitness function is:

$$f(H_j) = \begin{cases} 1.0 & \text{if } H_j = H \\ 0.6 & \text{if } H_j \neq H \end{cases} \quad (32)$$

The population size is 100. The population  $P(0)$  was initialised as follows:  $y_H(t) = 1$  and  $y_{\bar{H}}(t) = 99$  for the Holland's and Schaffer's models. For the state equation  $y_H(t) = 1$  and the 99 remained instances are uniform distributed to the  $\pi$  pattern schemas.



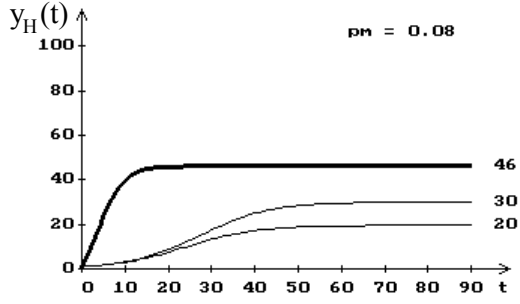


Fig. 5. Comparison between the state equation, the Schaffer's model and the Holland's model

In Fig. 5, on the abscissa is the time expressed in generations and in the ordinate is  $y_H(t)$  in instances. In figure there are three time evolution plots calculated with the models:

- the state equation (19) where the coefficients  $\alpha_{j,k}$  and  $b_{j,k}$  are calculated with the chapter 8 method (thick line);
- the Schaffer's model (29) where  $\alpha(H)$  is calculated with equation (7) (upper thin line);
- the Holland's model (29) where  $\alpha(H)$  is calculated with equation (6) (lower thin line).

In Fig. 5, for each model, in the right of the plots is written the stable final value,  $y_H(\infty)$ , towards evolve the H schema instance number in population.

Analysing these values we find that the mutation probability  $p_m = 0.08$  scatters too much the population. More similar tests than those presented in Fig. 5 established that optimum mutation probability is  $p_m = 0.005 \dots 0.05$ .

We note that the Holland's model is the minimal estimation of the  $y_H(\infty)$  value. The Schaffer's model eliminates the simplifying hypothesis (a) and from the 15<sup>th</sup> generation it gives a bigger estimation.

The state equation eliminates the entire simplifying hypothesis and gives the higher estimation. Also we note that the initial evolution speed given by the state equation is faster than the others models are.

#### 4.4. Test for scattering effect of crossover

In a genetic algorithm the crossover operator performs the exploration action while the mutation operator is used to scatter the population to postpone the convergence moment. Also, the

crossover operator has a secondary role in scattering the population.

To study the scattering effect of the crossover we consider  $p_m = 0$ . All the test conditions are similar to those presented in Fig. 5. The results are presented in Fig. 6.

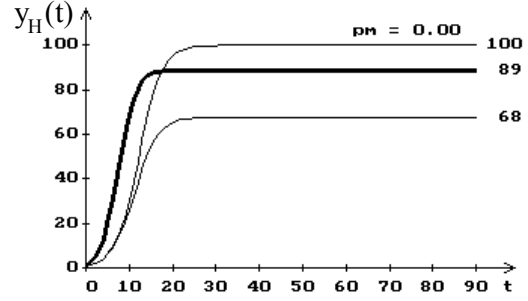


Fig. 6. Comparison between the three models for  $p_m = 0$

The state equation result is plotted with thick line, and the Schaffer model result is plotted with the upper thin line.

We note that the Schaffer model makes an exaggerate evaluation of crowding of the population in the new ecological niche. Only the state equation makes an accurate evaluation of the additional role of crossover in scattering action of the population.

#### 4.5. Test for scattering effect of mutation

The state equation can be used to analyse the scattering action of mutation. If we consider:

$$\|\alpha_{j,k}(t)\|_{\pi} = \|\phi_{j,k}(t)\|_{\pi} = \|I\|_{\pi}, \quad (33)$$

then the state equation (19) reduces at:

$$y(t+1) = \|\beta_{j,k}\|_{\pi} \times y(t) \quad (34)$$

Equation (34) is a dynamic model for a genetic algorithm whose population evolves only due to mutations.

For a separate study of influence of the mutation probability on the population dynamics we propose to integrate the equation (34) in following conditions:

- Let us consider  $H \in \pi$ , an arbitrary chosen schema taken from the pattern  $\pi$ ;
- The coefficients  $\beta_{j,k}$  are calculated with equation (26);
- The population  $P(0)$  is initialised only with strings of H schema.

The theorem 1, section 3.1, states that the pattern of the schema is preserved. So, we try to

assume that in the population  $P(\infty)$  the instances of the schemas  $H_i \in \pi$ ,  $i = 1, \dots, 2^{o(\pi)}$  are uniform scattered.

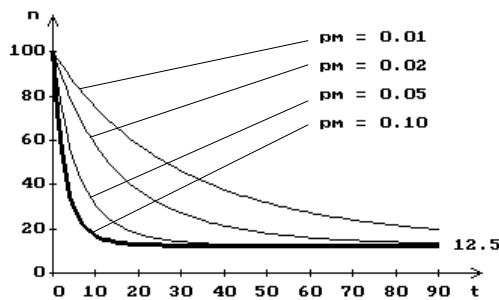


Fig. 8. Influence of the mutation probability on the spread speed of population

Fig. 8 presents the time evolution of a schema that contains the kernel '###\*#'. The simulation was performing in following mutation probabilities: 0.01, 0.02, 0.05 and 0.1. We can notice that a great mutation probability increases the scattering speed performed by mutation operator.

## 5. CONCLUSIONS

All the arguments used in the schema theorem proof and all results have a “statistic sense” so the exceptions are allowed. The schema theorem interpretation is not a finished action because there are always new points of view supported or rejected by the theorem.

A genetic algorithm is a discrete dynamic system, that has no entries and the initial state is randomly generated. In the early generations the state equation has constant coefficients and above-average schema has exponentially increasing trials.

After few generations the coefficients of state equation  $\varphi_{j,k}(t)$  become time dependent, because the population composition is changing in time and the weak schema are loosing from population.

If we consider all the schemas in a usual genetic algorithm, the state equation has a huge dimension ( $3^L$  differential equations). Only in simplest examples, this equation can be integrated.

The classical GA theory is based on two principles: the schema theorem and the building block hypothesis. In the proof of the theorem are used the concepts of string, schema and fitness function. The state equation is an extension of the schema theorem and incorporates all the

previous results of the proofs of Holland and Schaffer. In addition, the state equation uses two new concepts: the pattern of the schema and the kernel of the schema. The concept *pattern of the schema* is important and explains two other empirical observations: *the implicit parallelism of the genetic algorithms* and the *competitive schemas*.

## REFERENCES

- [1] Belea R. and L. Beldiman - “A Rigorous Wording for the Building Bloc Hypothesis”. PERIODICA POLITEHNICA, Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE, Vol. 47 (61), 2002, p. 145-150.
- [2] Belea R. - “Contribution to the system identification using genetic algorithms”. PhD Thesis, “DUNĂREA DE JOS” University of Galați, 2004; P. 176.
- [3] Goldberg D. E. - “Genetic Algorithms”. Addison-Wesley USA, (15767), 1991. France translation, Copyright © Juin 1994 Editions Addison-Wesley France, S. A.
- [4] Liepins G. E. and M. D. Vose - “Deceptiveness and Genetic Algorithm Dynamics”. “Foundations of genetic Algorithms”. Edited by Gregory Rowllins Morgan Kaufmann Publishers, Inc. 1991.
- [5] Michalewicz Z. - “Genetic Algorithms + Data Structures = Evolution Programs”. Springer-Verlag Berlin Heidelberg 1994.
- [6] Stephens C. and H. Waelbroeck - “Schemata Evolution and Building Blocks”. Evolutionary Computation, Vol. 7 No. 2, p. 109-124, © 1999 Massachusetts Institute of Technology.
- [7] Rojas R. - “Theorie de Neuronen Netze: Eine Systematische Einführung”. Cap. 17. Genetische Algorithmen. Springer 1993.