

Grammar-based model of Hierarchical Genetic Strategy - preliminary study

Joanna Kołodziej

University of Bielsko - Biala, Department of Mathematics and Computer Science,
Bielsko-Biala, Poland
email: jkolodziej@ath.bielsko.pl

Abstract The scope of this paper is the construction of Hierarchical Genetic Strategy theoretical model based on the L-systems framework. The model was defined in the case of inactive prefix comparison procedure of HGS. We applied Vose's theory in a short formal analysis of basic search mechanisms implemented in the strategy.

1 Introduction

Theoretical properties of evolutionary algorithms have been studied since their beginning. Early works of Holland and Goldberg (see [2]) contain the theory of schemata and building-blocks hypothesis for genetic algorithms. Vose and Liepins (see [14]) constructed the rigorous mathematical model of the Simple Genetic Algorithm (SGA). In this model the trajectory of the evolving populations is a discrete-time, homogeneous Markov chain with states in the set of all possible populations of the fixed size.

Markov model of SGA has been extended to the evolutionary strategies (see [13], [10]). It can be also useful in the formal description of basic mechanisms in parallel and hierarchical evolutionary strategies like Island SGA or Hierarchical Genetic Strategy (HGS) (see [12] for example). However, in this case, it is very hard to define the relations among all parallel processes using Markov chain states and its transition probabilities. The model of a family of the dependent Markov processes constructed for HGS (see [7]) was too complicated for the estimation of the population distributions. In consequence, no probability distribution was defined for the whole strategy.

The goal of our current research is to simplify the Markov model of HGS. In this paper we propose a simple grammar-based model of HGS with SGA engine in the case of inactive prefix comparison operation. We applied a framework of the stochastic Lindenmayer system to simulate the growth of HGS tree structure. All populations evolving in HGS branches are interpreted as the letters in this grammar. The words of grammar are the sequences of populations generated in HGS metaepochs. New sequences are created by the grammar parallel rewriting rules executed for each letter in the given word. All HGS procedures are defined in the terms of grammar productions. This method can be very useful in the analysis of the dependency relation among HGS branches as well as in the estimation of the population probability distributions, what is demonstrated in the

Section 4. Markov model of SGA is there applied for the calculation of the production probabilities.

The paper is organized as follows. The background information about stochastic L-systems is given in Section 2. Section 3 contains a short description of the main HGS procedures together with the general definition of Vose's model of SGA. In Section 4 we defined the new model of HGS in the case of inactive prefix comparison procedure. We calculated the probabilities of productions and we defined the probability distribution for the whole strategy. The paper ends with some final remarks.

2 Stochastic Lindenmayer systems

Lindenmayer-systems (or *L-systems*) are named after the biologist Aristid Lindenmayer (1925-1989). The emphasis of his study was on modeling the plant topology, i.e. the neighborhood relations between cells or larger plant modules (see [9]). Koza (see [8]) and Jacob (see [6]) applied the deterministic L-systems in genetic programming to produce the genotype patterns. Jacob and Rehder (see [5]) have constructed the simple theoretical model of neural networks based on hierarchical grammar system.

The central idea of L-systems is that of rewriting. The concept of L-systems rewriting rules is based on Noam Chomsky's idea of the description of the syntactic features of natural languages.

According to Chomsky's theory each grammar can be defined as a triple $G = (\Sigma, A, P)$, where

- Σ - called *an alphabet* - is a finite set of basic symbols called *letters*,
- A - is a finite set of start strings called *axioms*,
- P - is a finite set of rewriting rules called *productions*.

Let us denote by Σ^* a set of all finite words (strings) over the alphabet Σ . All axioms are the elements of Σ^* and each production p from the set $P \subset \Sigma \times \Sigma^*$ is a homomorphism $p : \Sigma \rightarrow \Sigma^*$. The production p is written as $a \rightarrow \omega$, that is the letter a , called *the predecessor*, is replaced by the word ω , called *the successor* of the production.

In contrast to Chomsky grammars, in L-systems all letters in the given word are replaced in parallel and simultaneously. The simplest L-systems are deterministic and context-free (0L-systems), where for any letter $a \in \Sigma$ there exists at least one word $\omega \in \Sigma^*$ such that $a \rightarrow \omega$. If for the predecessor a no production is explicitly given, it is assumed that the identity map $a \rightarrow a$ belongs to the set of productions.

A 0L-system is deterministic (D0L-system) if and only if for each $a \in \Sigma$ there is exactly one word $\omega \in \Sigma^*$ such that $a \rightarrow \omega$. In stochastic 0L-systems each predecessor may have more than one successor. The productions are executed with the additional parameters called production probabilities. For any predecessor a the sum of all the probabilities of all productions of a must be equal to 1.

Many interesting examples of L-systems can be found in [9]. We applied the stochastic Lindenmayer systems framework to model the structure and the dependency relations between evolutionary processes activated in HGS branches.

3 HGS basics

The efficiency of Hierarchical Genetic Strategy (HGS) comes from a concurrent search in the optimization landscape performed by many small populations. The creation of these

populations is governed by the dependent genetic processes with low complexity. The processes of low order represent chaotic search. They detect the promising region on the optimization landscape in which more accurate processes are activated.

We assume in this paper that SGA is the basic engine of HGS. The general description of HGS procedures is introduced by a background information on the SGA model [15].

3.1 Mathematical model of SGA

Let us denote by $\mathcal{D} \subset \mathbb{R}^n$ the admissible set for global optimization problem. We assume that its subset $\mathcal{D}_r \subset \mathcal{D}$ contains all individual phenotypes encoded into binary strings of the fixed length $s \in \mathbb{N}$ stored in a genetic space Ω_s . Ω_s is usually identified with the set of integers from the interval $[0, \dots, r - 1]$, where $r = 2^s$.

A collection of n elements (not necessarily different) of Ω_s is called a *population* of size n and it is represented by a vector:

$$p = [p_0, \dots, p_{r-1}]^T : \quad p_t \geq 0, \quad \sum_{t=0}^{r-1} p_t = 1 \quad (1)$$

where p_t is the proportion of element t in the population.

The coordinates of the vector p are identical with the barycentric coordinates of some point in the standard unit $(r - 1)$ -dimensional simplex Λ .

Simple Genetic Algorithm (SGA) may be described as an instance of *random heuristic search* procedure (see [15]) generating a sequence of populations:

$$p^0, p^1 = \tau(p^0), p^2 = \tau(p^1), \dots; p^i \in \Lambda, i = 0, 1, 2, \dots \quad (2)$$

with some transition rule τ , where p^0 is referred to as the initial population vector. Operator τ is stochastic in case of finite populations ($n < +\infty$).

Vose and Liepins (see [14]) introduced a *heuristic function* $G : \Lambda \rightarrow \Lambda$ that, given the current population vector p , produces a vector $G(p)$ whose i -th component is defined as the probability that the element i is the result of some stochastic procedure, given the population p .

It is proved in [15] (see Theorem 3.3) that $G(p)$ is the expected next population vector for p .

Function G for SGA is defined as the superposition $G = M \circ F$, where M - denotes recombination operator and F - selection operator (see [14]).

Let us denote by X_n^r the finite subset of points from Λ corresponding to all possible populations of the fixed size n . Vose (see [15]) proved that for populations of fixed size $n < +\infty$ SGA forms a Markov chain with states in the set X_n^r and transition matrix $Q = [Q_{i,j}]$, where $Q_{i,j}$ are defined as follows:

$$Q_{i,j} = r! \prod_{y=0}^{r-1} \frac{(G(p^i)_y)^{n(p^j)_y}}{(n(p^j)_y)!}, \quad i, j = 0, \dots, N. \quad (3)$$

$(p^j)_y$ is the y -th coordinate of vector p^j , $G(p^i)_y$ denotes the probability of producing string y , ($y = 0, 1, \dots, r - 1$) in the population p^i and N is the number of all possible populations of the fixed size n . N is calculated from the following formula:

$$N = \binom{n+r-1}{r-1}. \quad (4)$$

Applying Chapman-Kolmogorov's equations (see [3]) we can calculate the transition matrix

$$Q^k = [Q_{i,j}^k], \quad (5)$$

where $Q_{i,j}^k$ is the probability that population p^i will change into population p^j in k steps (generations).

We will apply the above formulas for the calculation of the production probabilities in our grammar-based HGS model in Section 4.

3.2 HGS procedures

To start HGS we have to fix the lengths $1 \leq s_1 \leq \dots \leq s_m = s_{max}$ of genotypes of the individuals in the populations of the sizes n_1, n_2, \dots, n_m respectively. We also have to specify the period $k \in \mathbb{N}$ of metaepoch.

The algorithm starts from a randomly selected population of the shortest binary genotypes (s_1 -length) and the root of the HGS structure is built. The branch has degree $j \in \{1, \dots, m\}$ if it is created by populations containing chromosomes of the length $s_j \in \{s_1, \dots, s_m\}$. The root is the unique branch of the lowest degree 1.

The populations evolving in branches of the different degrees are denoted by:

$$p_{i,j}^e \quad (6)$$

where:

- $e \in \mathbb{N}$ - is the global metaepoch counter,
- $j, j \in \{1, \dots, m\}$ - degree of the branch containing individuals of the length s_j ,
- i - is the unambiguous branch identifier, $i = (i_1, \dots, i_m)$, $i_h = 0$ for $h > j$.

The unambiguous branch identifier $i = (i_1, \dots, i_m)$ describes the "history of creation" of the given branch. For example, let i_1, \dots, i_{j-1} be the numbers of predecessors (counted from the root) of the branch of degree j . The current branch is the i_j -th consecutive child of the branch indexed by $(i_1, \dots, i_{j-1}, 0, 0, \dots, 0)$.

A new branch could be created after executing a metaepoch in the parental branch. A k -periodic metaepoch $M_k, (k \in \mathbb{N})$ is a discrete evolution process which starts from the given population and terminates after at most k generations by selection of the best adapted individual. The outcome of the metaepoch started from the population p^d may be denoted by :

$$M_k(p^d) = (p^{d+l}, \widehat{x(k)}, stop); \quad l \leq k, \quad (7)$$

where p^{d+l} is the resulting population and $\widehat{x(k)}$ is the best adapted individual in the metaepoch. *stop* is the branch stop criterion flag (see [7] for details).

New branches of higher degree are created by the *sprouting procedure*. Let $\widehat{x(k)}$ be the best adapted individual in some metaepoch of the period k . An operator given by the formula:

$$SO(p_{i,j}^{e+1}) = (p_{i,j}^{e+1}, p_{i',j+1}^0) \quad (8)$$

where $i' = (i_1, \dots, i_{j-1}, 1, 0, \dots, 0)$, defines the process of construction of an initial population for a new branch of degree $j+1$ represented by the vector $p_{i',j+1}^0$. This population is a multisubset of $\Omega_{s_{j+1}}$. The individual genotypes are defined by $x = \widehat{x(k)}y$, where:

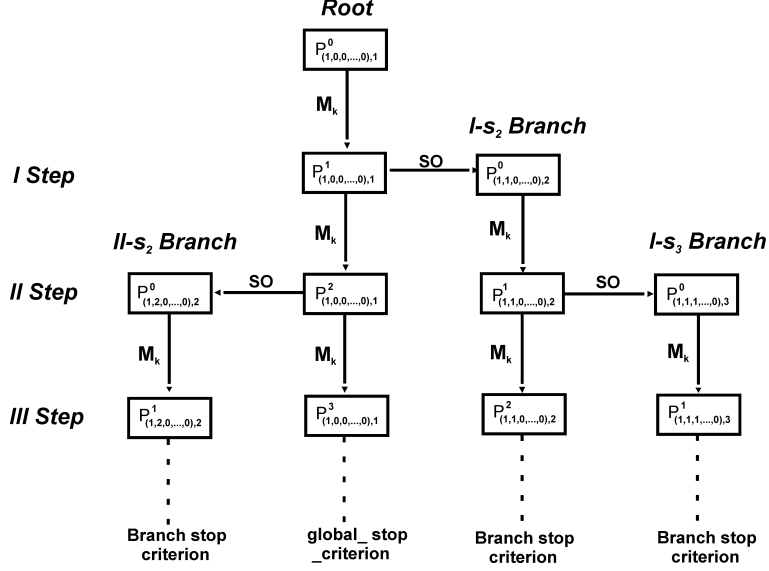


Figure 1. The general schema of the HGS structure after 3 metaepoch

- $\widehat{x(k)}$ - is the s_j - length prefix of x and
- $y \in \Omega_{(s_{j+1}-s_j)}$ is the $s_{j+1} - s_j$ - length suffix of x selected according to the uniform probability distribution over $\Omega_{(s_{j+1}-s_j)}$.

The sprouting procedure can be activated or not, depending on the outcome of the prefix comparison operator. The codes of all HGS procedures can be found in [4].

4 Grammar-based HGS model

Let us assume that the prefix comparison operator is not active in HGS mechanism. It means that each branch of degree j ; ($j \in \{1, \dots, m-1\}$) after each metaepoch produces a "child" of degree $j+1$. All branches of maximal degree m have to be just elongated. The general schema of the HGS structure in this special case is presented in Figure 1. Let us denote by $X_{s_j}^{n_j}$; $j \in \{1, \dots, m\}$ the set of all possible populations of the size n_j consisting of genotypes of the length s_j . It contains N_j elements, where N_j can be calculated from the formula (4) applied for s_j and $r_j = 2^{s_j}$.

HGS in this case can be modeled by a context-free L-system $G = (\Sigma, S, P)$, where:

- $\Sigma = \bigcup_{j=1}^m X_{s_j}^{n_j}$ is the alphabet,
- $S = X_{s_1}^{n_1}$ is the set of start symbols,
- $P = \{\widehat{P}_1, \widehat{P}_2\}$ is the set of stochastic productions.

Productions from the set P are defined in the following way:

$$\widehat{P}_1 : p_{i(j),j}^e \xrightarrow{(P_1(j))} p_{i(j),j}^{e+1} p_{i(j)+1,j+1}^0; \quad j = 1, \dots, m-1,$$

$$\widehat{P}_2 : p_{i(m),m}^e \xrightarrow{(P_2)} p_{i(m),m}^{e+1}; \quad e > 1$$

where $p_{i(j),j}^e \in X_{s_j}^{n_j}; j = 1, \dots, m; i(j) = (i_1, \dots, i_j, 0, 0, \dots, 0)$ and $i(j) + 1 = (i_1, \dots, i_j, 1, 0, \dots, 0)$.

Production (\widehat{P}_1) defines the process of the execution of the single k -periodic metaepoch in the parental branch of degree j . This procedure is at the end combined with the sprouting of a new branch of degree $j + 1$. The successors of (\widehat{P}_1) are the words from Σ^* consisted of two letters: one is a resulting population for the metaepoch executed in the parental branch and the second one is an initial population for the new process activated by the operator SO defined in the previous section. It means that:

$$\begin{aligned} p_{i(j),j}^e &\xrightarrow{M_k} p_{i(j),j}^{e+1} \\ p_{i(j),j}^{e+1} &\xrightarrow{SO} p_{i(j)+1,j}^0 \end{aligned}$$

We assumed that SGA is the basic engine for HGS. The evolution process in each branch can be then modeled as a Markov chain according to Vose's theory. Probability $(P_1(j))$ is calculated for the populations $p_{i(j),j}^e$ and $p_{i(j)+1,j}^{e+1}$ in the following way:

$$(P_1(j)) = Prob(p_{i(j)+1,j}^{e+1} | p_{i(j),j}^e) \cdot Prob(p_{i(j)+1,j}^0). \quad (9)$$

where:

- $Prob(p_{i(j)+1,j}^{e+1} | p_{i(j),j}^e)$ is an element of the matrix $Q(k, s_j) = Q^k(s_j)$ (see formula(8)) and $Q^k(s_j)$ denotes the Markov transition matrix defined for SGA in the branch of degree j ,

$$Prob(p_{i(j)+1,j}^0) = \frac{1}{\binom{2^{s_{j+1}-s_j} + n_j - 1}{n_j}}. \quad (10)$$

The last formula is a simple consequence of the definition 3.2. The number of possible choices of an $s_{j+1} - s_j$ - length suffix of any genotype in $p_{i(j)+1,j}^0$ is $t_j = 2^{s_{j+1}-s_j}$. Thus the probability of creation of the new population of the size n_{j+1} is $\frac{1}{c}$, where $c = C_{t_j+n_j-1}^{n_j}$ is the number of choices of n_j samples of the suffixes (with replacement).

Probability P_2 is calculated in the same way as $Prob(p_{i(j)+1,j}^{e+1} | p_{i(j),j}^e)$.

The words produced by the presented L-system, are the sequences of populations generated in HGS metaepoch. For example, the structure of HGS presented on the Figure 1 can be defined as a following sequence of words produced by the grammar-based model of HGS:

- **start symbol:** $p_{(1,0,\dots,0),1}^0$
- **step I:** $p_{(1,0,\dots,0),1}^1 \quad p_{(1,1,0,\dots,0),2}^0$
- **step II:** $p_{(1,0,\dots,0),1}^2 \quad p_{(1,2,\dots,0),2}^0 \quad p_{(1,1,\dots,0),2}^1 \quad p_{(1,1,1,0,\dots,0),3}^0$

The production probabilities from the set P determine the probability distribution for the strategy HGS (as well as for the L-system defined above). We can reconstruct the creation process of the given population $p_{i(j),j}^e$ by the analysis of the unambiguous branch identifier $i(j) = i = (i_1, \dots, i_{j-1}, \dots)$. The values of i_1, \dots, i_{j-1} are determined by the grammar productions. In the above example population $p_{(1,2,\dots,0),2}^0$ of degree $j = 2$ was created from the root ($i_1 = 1$) after two metaepochs ($i_2 = 2$). The probability of its production from the start symbol $p_{(1,0,\dots,0),1}^0$ can be calculated as follows (see formula (9)):

$$Prob_{e,i,j} = Prob(p_{(1,0,\dots,0),1}^1 | p_{(1,0,\dots,0),1}^0) \cdot Prob(p_{(1,0,\dots,0),1}^2 | p_{(1,0,\dots,0),1}^1) \cdot Prob(p_{(1,0,\dots,0),1}^0 | p_{(1,0,\dots,0),1}^1). \quad (11)$$

The main task in our current research is to generalize the above formula for populations in all types of branches. It could help to define the formal criterion for the procedures of the empirical estimation of the population distributions, like those presented in [11].

Let us observe that the numbers of branches of the different degrees in several metaepochs form the Pascal triangle. We start from the single population in the branch of degree 1, after the first metaepoch we have one population in the branch of degree 1 and one population in the branch of degree 2, then L-system produces 1 population in branch of degree 1, two populations in the branch of degree 2 and one population in the branch of degree 3, etc. The process is continued till the creation of the initial population in the first branch of the maximal degree. It happens after running of m metaepochs. We have to wait another m metaepochs for the next branch of the maximal degree.

This simple observation confirms a basic property of the multitype Galton-Watson branching process (see [1]), which also can be used to model HGS in this case. However, the branching processes theory cannot be easily adapted for the analysis of structure and properties of HGS with any additional procedure like prefix comparison or branch reduction (see [4]). In that case we plan to extend the L-systems framework by the definition of some logical mechanism governing the grammar stochastic productions.

5 Conclusions and future work

- Lindenmayer systems (L-systems), originally designed for the simulation of the natural growth processes, can be also applied in genetic programming to produce the genotype patterns. We used them in modeling the growth of the tree structure of hierarchical evolutionary multipopulation algorithm.
- All populations evolving in HGS branches are interpreted as the letters in our grammar-based model. The words of grammar are the sequences of populations generated in HGS metaepochs and HGS procedures are defined in terms of grammar productions.
- We demonstrated in Section 4 that the method is very useful in the analysis of the dependency relation among HGS branches as well as in the estimation of the population probability distributions for the HGS in the case of inactive prefix comparison operator.
- The next research aim is the extension of the current model to the case of HGS with the active prefix comparison and branch reduction procedures. We want to

combine the presented L-system framework with some additionally defined logical procedure, what can give us an example of logical L-system.

Bibliography

- [1] K.B. Athreya and P.E. Ney. *Branching processes*. Springer Verlag, 1972.
- [2] D.E Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.
- [3] M. Iosifescu. *Finite Markov processes and its applications*. Polish ed. PWN, 1988.
- [4] R. Gwizdała J. Kołodziej and J. Wojtusiak. Hierarchical genetic strategy as a method of improving search efficiency,. *Advances in Multi-Agent Systems, R. Schaefer and S. Sędziwy (Eds.), UJ Press, Cracow*, pages 149–161, 2001.
- [5] C. Jacob. Genetic l-system programming. In *PPSN III - Parallel Problem Solving from Nature, International Conference on Evolutionary Computation, LNCS 866, Springer-Vlg, Berlin*, pages 334 – 343, 1994.
- [6] C. Jacob and J. Rehder. Evolution of neutral net architectures by a hierarchical grammar- based genetic system. In *ICANNGA '93, Innsbruck , Springer Vlg., Wien*, pages 602 – 606, 1993.
- [7] J. Kołodziej. Modelling hierarchical genetic strategy as a family of markov chains. In *R. Wyrzykowski, J. Dongarra, M. Paprzycki and J. Waśniewski (eds.), Parallel Processing and Applied Mathematics, LNCS, vol. 2328, Springer Vlg, pages 595–599, 2002*.
- [8] J.R. Koza. *Genetic Programming. Part 1, Part 2*. MIT Press, 1992 (Part 1), 1994 (Part 2).
- [9] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer Vlg, 1990.
- [10] G. Rudolph. Stochastic processes. *Handbook of Evolutionary Computation B2.2*, 1997.
- [11] R. Schaefer and K. Adamska. On genetic clustering using finite mixture models - error estimation and practical tests. In *Proc. of KAEiOG 2003, Łagów Lubuski, 26-28.05.03*, pages 183–191, 2003.
- [12] R. Schaefer and J. Kołodziej. Genetic search reinforced by the population hierarchy. *FOGA VII, Morgan Kaufmann*, pages 383–401, 2003.
- [13] H.-P. Schwefel T. Back. *Evolution strategies I, in J. Periaux, G. Winter (eds.) Genetic algorithms in engineering and computer science*. J. Wiley, 1995.
- [14] M. D. Vose and G.E. Liepins. Punctuated equilibria in genetic search. *Complex Systems*, 5:31–34, 1991.
- [15] M.D. Vose. *The Simple Genetic Algorithm*. MIT press, 1999.