We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

1

Limit Properties of Evolutionary Algorithms

Witold Kosiński[1,3] and Stefan Kotowski[1,2]
[1] *Faculty of Computer Science Polish-Japanese Institute of Information Technology,*
[2] *Institute of Fundamental Technological Research IPPT PAN,*
[3] *Institute of Environmental Mechanics and Applied Computer Science Kazimierz Wielki University*
Poland

1. Introduction

In the chapter limit properties of genetic algorithms and theproblem of their classification are elaborated. Recently one can observe an increasing interest in properties of genetic algorithms modelled by Markov chains (Vose, Rowe). However, the known results are mainly limited to existence theorems. They say that there exists a limit distribution for a Markov chain describing a simple genetic algorithm. In the chapter we perform the next step on this way and present a formula for this limit distribution for a Markov chain. Moreover, we claim that our convergence theorems can be extended to algorithms which admit the change in the mutation rate and others parameters.

The formula for a limit distribution requires some knowledge about the distribution of the fitness function on the whole solution space. However, it suggests the methods to control the algorithm parameters to get better convergence rate. The formula can play an important role in deriving new classification tools for genetic algorithms that use methods of the theory of dynamical systems. That tools will exploit real dynamics of the search and be independent of the taxonomic methods of classification that are used nowadays.

On the base of the knowledge of the limit distribution we construct an optimal genetic algorithm in the probabilistic sense. Generally this algorithm is impossible to describe. This is an open problem at the moment, however, its existence and its form suggest an improvement of the original algorithm by changing its parameters. Constructed in this way the optimal genetic algorithm is an answer to one of the questions stayed by famous No Free Lunch Theorem. Moreover, it is a complementary result to this theorem. On the base of this theoretical result we perform a classification of algorithms and show empirical (computational) results in getting which the entropy, fractal dimension, or its approximations: the box-counting dimension or information dimension, are used.

One of the most difficult, however, of practical importance, problems is the choice of an algorithm to given optimisation problem.

The distinguishing between an optimisation problem and the algorithm and its choice creates to the main difficulty. Consequently, the distinguishing is an artificial operation because it abstains from the idea of genetic algorithm (GA), since the fitness function, arises from the cost function (i.e. the function to be optimised) is the main object of the genetic algorithm and it emerges from the formulation of the optimisation problem and it is difficult

Source: Advances in Evolutionary Algorithms, Book edited by: Witold Kosiński, ISBN 978-953-7619-11-4, pp. 468, November 2008, I-Tech Education and Publishing, Vienna, Austria

to speak about genetic algorithm as an operator without the fitness function. However, in our consideration we will simultaneously use both notions of the genetic algorithms. The first notion as an operator acting on the cost (fitness) function, the second - as a specific (real) algorithm for which the fitness is the main component being the algorithm's parameter.

This dual meaning of the genetic algorithm is crucial for ou consideration, because our main aim is to try to classify genetic algorithms. The classification should lead to a specific choice of methodology of genetic algorithms understood as operators. It is expected that in terms of this methodology one will be able to choose the appropriate algorithm to given optimisation problem. We claim that using this classification one could improve existing heuristic methods of assortment of genetic algorithms that are based mainly on experiences and programmer intuition.

There is the so-called "No-free lunch theorem" [12] according to which it does not exist a best evolutionary algorithm and moreover, one cannot find most suitable operator between all possible mechanisms of crossover, mutation and selection without referring to the particular class of optimisation problems under investigation. Evolutionary algorithms are the methods of optimizations which use a limited knowledge about investigated problem. On the other hand, our knowledge about the algorithm in use is often limited as well [13, 14].

The "no free lunch" results indicate that matching algorithms to problems give higher average performance than those applying a fixed algorithm to all problems. In the view of these facts, the choice of the best algorithm may be correctly stated only in the context of the optimisation problem.

These facts imply the necessity of searching particular genetic algorithms suitable to the problem at hand.

The present paper is an attempt to introduce an enlarged investigation method to the theory of genetic (evolutionary) algorithms. We aim at

1. the investigation of convergence properties of genetic algorithms,
2. the formulation of a new method of analysis of evolutionary algorithms regarded as dynamical processes, and
3. the development of some tools suitable for characterization of evolutionary algorithms based on the notions of the symbolic dynamics.

Genetic algorithm (GA) performs a multi-directional search by maintaining a population of potential solutions and encourages information formation and exchange between these directions. A population undergoes a simulated evolution due to the iterative action with some probability distributions of a composition of mutation, crossover and selection operators. The action of that composition is a random operation on populations.

If we imagine that a population is a point in the space Z of (encoded) potential solutions then the efect of one iteration of this composition is to move that population to another point. In this way the action of GA is a discrete (stochastic) dynamical system. We claim that by implementing the methods and the results of the theory of dynamical systems, especially those known from the analysis of dynamics of 1D mappings, one can move towards the goal of the theory of GA, which is the explanation of the foundations of genetic algorithm's operations and their features.

In GA with the known fitness function the proportional selection can be treated as a multiplication of each component of the frequency vector by the quotient of the fitness of the

corresponding element to the average fitness of the population. This allows to write the probability distribution for the next population in the form of the multiplication of the diagonal matrix times the population (frequency) vector. Moreover, results of the mutation can also be written as a product of another matrix with the population (probability) vector. Finally the composition of both operations is a matrix, which leads to the general form of the transition operator (cf.(17)) acting on a new probability vector representing a probability distribution of appearance of all populations of the same *PopSize*. The matrix appearing there turns to be Markovian and each subsequent application of SGA is the same as the subsequent composition of that matrix with itself. (cf.(19)). Thanks to the well-developed theory of Markov operators ([18, 22, 26, 27]) new conditions for the asymptotic stability of the transition operator are formulated.

2. Genetic algorithms

In the paper we use the term *population* in two meanings; in the first it is a finite multi-set (a set with elements that can repeat) of solutions, in the second it is a frequency vector composed of fractions, i.e. the ratio of the number of copies of each element $z_k \in Z$ to the total population size *PopSize*.
In our analysis we are concerned with probability distributions of each population for a particular case of the simple genetic algorithm (SGA) in which the crossover follows the mutation and the proportional selection. In the case of a binary genetic algorithm (BGA) the mutation can be characterized by the bitwise mutation rate μ - the probability of the mutation of one bit of a chromosome. In the paper, however, we are not confined to binary operators; the present discussion and results are valid under very week assumptions concerning the mutation and selection operators.

2.1 Population and frequency vector

Let

$$Z = \{z_0, ..., z_{s-1}\},$$

be the set of individuals called *chromosomes*. [1]By a *population* we understand any multi-set of r chromosomes from Z, then r is the population size: *PopSize*.

Definition 1. *By a frequency vector of population we understand the vector*

$$\boldsymbol{p} = (p_0, ..., p_{s-1}) \ , \ \ where \ p_k = \frac{a_k}{r} \ , \tag{1}$$

where a_k is a number of copies of the element z_k.

The set of all possible populations (frequency vectors) is

$$\Lambda = \{\boldsymbol{p} \in \mathbf{R}^s : \ p_k \geq 0, \ p_k = \frac{d}{r}, \ d \in \mathbf{N}, \ \sum_{k=0}^{s-1} p_k = 1 \ \}. \tag{}$$

[1] If one considers all binary l-element sequences then after ordering them one can compose the set Z with $s = 2^l$ elements.

When a genetic algorithm is realized, then we act on populations, and new populations are generated. The transition between two subsequent populations is random and is realized by a probabilistic operator. Hence, if one starts with a frequency vector, a probabilistic vector can be obtained. It means that in some cases p_i cannot be rational any more. Hence the closure of the set Λ, namely

$$\overline{\Lambda} = \{\boldsymbol{x} \in \mathbf{R}^s : \forall k,\ x_k \geq 0,\ \text{and}\ \sum_{k=0}^{s-1} x_k = 1\ \}, \tag{3}$$

is more suitable for our analysis of such random processes acting on probabilistic vectors; they are in the set $\overline{\Lambda}$.

2.2 Selection operator

Let a fitness function $f : Z \rightarrow \mathbf{R}^+$ and population $\boldsymbol{p}$ be given. If we assume the main genetic operator is the *fitness proportional selection*, then the probability that the element z_k will appear in the next population equals

$$\frac{f(z_k)p_k}{\overline{f}(\boldsymbol{p})}, \tag{4}$$

where $\overline{f}(\boldsymbol{p})$ is the *average population fitness* denoted by

$$\overline{f}(\boldsymbol{p}) = \sum_{k=0}^{s-1} f(z_k)p_k. \tag{5}$$

We can create the matrix $\boldsymbol{S}$ of the size s, where its values on the main diagonal are

$$S_{kk} = f(z_k). \tag{6}$$

Then the transition from the population p into the new one, say q is given by

$$\boldsymbol{q} = \frac{1}{\overline{f}(\boldsymbol{p})}\boldsymbol{S}\boldsymbol{p}, \tag{7}$$

and the matrix $\boldsymbol{S}$ describes the *selection operator* [21, 23, 24].

2.3 Mutation operator

Let us define a matrix

$$\boldsymbol{U} = [U_{ij}],$$

with U_{ij} as the probability of mutation of the element z_j into the element z_i, and U_{ii} - the probability of the surviving of the element (individual) z_i. One requires that

1. $U_{ij} \geq 0$;

2. $$\sum_{i=0}^{s-1} U_{ij} = 1 \ , \ \text{for} \ \ \text{all} \ j. \tag{8}$$

In the case of the binary uniform mutation with parameter μ as the probability of changing bits 0 into 1 or vice versa, if the chromosome z_i differs from z_j at c positions then

$$U_{ij} = \mu^c (1-\mu)^{l-c} \tag{9}$$

describes the probability of mutation of the element z_j into the element z_i.

2.4 Crossover operation

In order to define the operator of crossover $\boldsymbol{C}$ one needs to introduce additional denotation. Let matrices $\boldsymbol{C}_0, \ldots, \boldsymbol{C}_{s-1}$ be such that the element (i, j) of the matrix $\boldsymbol{C}_k$ denotes the probablity that an element z_i crossovered with an element z_j will generate an element z_k.
For the presentation simplicity let us consider the case of chromosoms of the lenght $l = 2$. Then elements of the space B will be of the form

$$z_0 = 00, \ z_1 = 01, \ z_2 = 10, \ z_3 = 11. \tag{10}$$

For the uniform crossover operation when all elements may take part, the matrix $\boldsymbol{C}_0$ has the form

$$\boldsymbol{C}_0 = \begin{pmatrix} 1.0 & 0.5 & 0.5 & 0.25 \\ 0.5 & 0.0 & 0.25 & 0.0 \\ 0.5 & 0.25 & 0.0 & 0.0 \\ 0.25 & 0.0 & 0.0 & 0.0 \end{pmatrix} \tag{11}$$

One can define the remaining matrices; all matrices $\boldsymbol{C}_k$ are symmetric. Finally, the operator $\boldsymbol{C}$ in the action on a population $\boldsymbol{p}$ gives

$$\boldsymbol{C}(\boldsymbol{p}) = (\boldsymbol{p} \cdot \boldsymbol{C}_0 \boldsymbol{p}, \ldots, \boldsymbol{p} \cdot \boldsymbol{C}_{s-1} \boldsymbol{p}), \tag{12}$$

where the dot $\cdot$ denotes the formal scalar product of two vectors from s-dimentional space. Hence, from a given population (say, $\boldsymbol{p}$) to the next population (say, $\boldsymbol{q}$) the action of the simple genetic algorithm (SGA) [21, 23, 24] is described by the operator $\mathcal{G}$ being a composition of three operators: selection, mutation and crossover:

$$\mathcal{G} = \boldsymbol{C} \circ \boldsymbol{U} \circ \mathcal{F}. \tag{13}$$

The reader interested in the detailed descrition of the operators is referred to the positions [21, 23]. In what follows the crossover is not present. However, most of the results of subsequent sections hold if the crossover is present.

3. Transition operator

Let $\boldsymbol{p} = (p_0,\ldots,p_{s-1})$ be a probabilistic vector. If we consider $\boldsymbol{p} \in \overline{\Lambda}$, then transition operators should transform set $\overline{\Lambda}$ into itself. The action of the genetic algorithm at the first and at all subsequent steps is the following: if we have a given population $\boldsymbol{p}$ then we sample with returning r-elements from the set Z, and the probability of sampling the elements $z_0,\ldots, z_{s-1}$ is described by the vector $\mathcal{G}(\boldsymbol{p})$, where

$$\mathcal{G}(\boldsymbol{p}) = \frac{1}{\bar{f}(\boldsymbol{p})}\boldsymbol{USp}\ . \tag{14}$$

This r-element vector is our new population $\boldsymbol{q}$.
Let us denote by W the set of all possible r-element populations composed of elements selected from the set Z, where elements in the population could be repeated. This set is finite and let its cardinality be M: It can be proven that the number M is given by some combinatoric formula

$$M = \binom{s+r-1}{s-1} = \binom{s+r-1}{r}\ . \tag{15}$$

Let us order all populations, then we identify the set W with the list $W = \{w^1,\ldots,w^M\}$. Every w^k, $k = 1, 2,\ldots,M$, is some population for which we used the notation $\boldsymbol{p}$ in the previous section. According to what we wrote, the population will be identified with its frequency vector or probabilistic vector. This means that for the population $\boldsymbol{p} = w^k = (w_0^k,\ldots,w_{s-1}^k)$, the number w_i^k, for $i \in \{0,\ldots,s-1\}$, denotes the probability of sampling from the population w^k the individual z_i (or the fraction of the individual z_i in the population w^k).
Let us assume that we begin our implementation of SGA from an arbitrary population $\boldsymbol{p} = w^k$. In the next stage each population $w^1,\ldots,w^M$ can appear with the probability $\beta_{1k}, \beta_{lk},\ldots, \beta_{Mk}$ which can be determined from our analysis. In particular, if in the next stage the population has to be $\boldsymbol{q}$, with the position l on our list W, then this probability [23, 28, 31] is equal

$$\beta_{lk} = r!\prod_{j=0}^{s-1}\frac{(\mathcal{G}(\boldsymbol{p})_j)^{rq_j}}{(rq_j)!}\ ,\ \text{with } \boldsymbol{q} = w^l \text{ and } \boldsymbol{p} = w^k\ . \tag{16}$$

Notice that $\sum_{j=1}^{M}\beta_{jk} = 1$ for every $k = 1, 2,\ldots,M$. After two steps, every population $w^1,\ldots,w^M$ will appear with some probability, which is a double composition of this formula[2]. It will be analogously in the third step and so on. Then it is well founded to analyze the

[2] With our choice of denotations for the populations $\boldsymbol{p}$ and $\boldsymbol{q}$ in (16), the element β_{lk} of the matrix will give transition probability from the population with the number k into the population with the number l.

probability distribution of the population's realization in the next steps. This formula gives a possibility of determining all elements of a matrix $\boldsymbol{T}$ which defines the probability distribution of appearance of populations in the next steps, if we have current probability distribution of the populations.
It is important that elements of the matrix are determined once forever, independently of the number of steps. The transition between elements of different pairs of populations is described by different probabilities (16) represented by different elements of the matrix.
Let us denote by

$$\Gamma = \{\boldsymbol{y} \in \mathbf{R}^M : \forall k \; y_k \geq 0 \text{ oraz } ||\boldsymbol{y}|| = 1\},$$

where $\|\boldsymbol{y}\| = y_1 + ... + y_M$, the set of new M-dimensional probabilistic vectors. A particular component of the vector $\boldsymbol{y}$ represents the probability of the appearance of this population from the list W of all M populations. The set Γ is composed of all the possible probability distributions for M populations. Described implementation transforms at every step the set Γ into the same.
On the set Γ the basic, fundamental *transition operator*,

$$T(\cdot) : \mathbf{N} \times \Gamma \to \Gamma. \tag{17}$$

is defined. If $u \in \Gamma$, then $T(t)u = ((T(t)u)_1, \ldots, (T(t)u)_M)$ is the probability distribution for M populations in the step number t, if we have begun our implementation of SGA given by $\mathcal{G}$ ((14)) from the probability distribution $u = (u_1,\ldots,u_M) \in \Gamma$, by t – application of this method. The number $(T(t)u)_k$ for $k \in \{1 \ldots, M\}$denotes the probability of appearance of the population w_k in the step of number t. By the definition $\mathcal{G}(\boldsymbol{p})$ in (14),(16) and the remarks made at the end of the previous section the transition operator $T(t)$ is linear for all natural t.
Let us compose a nonnegative, square matrix $\boldsymbol{T}$ of dimension M, with elements β_{lk}, $l, k = 1, 2,\ldots,M$, i.e

$$\boldsymbol{T} = [\beta_{lk}]. \tag{18}$$

We will call it *the transition matrix*. Then the probability distribution of all M populations in the step t is given by the formula

$$\boldsymbol{T}^t u, \; t = 0, 1, 2, \ldots$$

Elements are independent from the number of steps of the algorithm. The above introduced transition operator $T(t)$ is linked with the transition matrix by the dependence

$$T(t) = \boldsymbol{T}^t. \tag{19}$$

Notice that though the formula (16) determining individual entries (components) of the matrix $\boldsymbol{T}$ are population dependent, and hence nonlinear, the transition operator $T(t)$ is linear thanks to the order relation introduced in the set W of all M populations. The multi-

index (l, k) of the component β_{lk} kills, in some sense, this nonlinearity, since it is responsible for a pair of populations between which the transition takes place. The matrix $\boldsymbol{T}$ in (18) is a Markovian matrix. This fact permits us to apply the theory of Markov operators to analyze the convergence of genetic algorithms [18, 22, 26, 27].

Let $e_k \in \Gamma$ be a vector which at the k-th position has one and zeroes at the other positions. Then e_k describes the probability distribution in which the population w^k is attained with the probability 1.

By the notation $T(t)w^k$ we will understand

$$T(t)w^k := T(t)e_k \tag{20}$$

which means that we begin the GA at the specific population w^k. Further on we will assume $U_{jj} > 0$ for $j \in \{0,\ldots,s-1\}$.

For a given probability distribution $u = (u_1,\ldots,u_M) \in \Gamma$ it is easy to compute that the probability of sampling the individual z_i, for $i \in \{0,\ldots,s-1\}$, is equal to

$$\sum_{k=1}^{M} w_i^k u_k \ , \tag{21}$$

where w_i^k is the probability of sampling from k-th population the chromosome z_i, and u_k - the probability of appearance of the k-th population. By an *expected population* we call the vector from $\mathbf{R}^s$ of which i-th coordinate is given by (21). Since $u_k \geq 0,\ w_i^k \geq 0$ for $k \in \{1,\ldots,M\}$, $i \in \{0,\ldots, s-1\}$ and

$$\sum_{i=0}^{s-1}\left(\sum_{k=1}^{M} u_k w_i^k\right) = \sum_{k=1}^{M} u_k \left(\sum_{i=0}^{s-1} w_i^k\right) = \sum_{k=1}^{M} u_k = 1 \ ,$$

the vector belongs to $\overline{\Lambda}$. From (21) we obtain that the expected population is given by

$$\sum_{k=1}^{M} w^k u_k \tag{22}$$

Obviously, it is possible that the expected population could not be any possible population with r-elements.

For every $u \in \Gamma$ and for every t certain probability distribution for M populations $T(t)u$ is given. Consequently the expected population in this step is known. By $R(t)u = ((R(t)u)_0,\ldots,(R(t)u)_{s-1})$ we denote the expected population at the step t, if we begun our experiment from the distribution $u \in \Gamma$; of course we have $R(t)u \in \overline{\Lambda}$.

3.1 Asymptotic stability

Definition 2. *We will say that the model is asymptotically stable if there exist* $u^* \in \Gamma$ *such that:*

$$T(t)u^* = u^* \quad \text{for} \quad t = 0, 1, \ldots \tag{23}$$

$$\lim_{t\to\infty} ||T(t)u - u^*|| = 0 \quad \text{for all } u \in \Gamma \ . \tag{24}$$

Since for $k \in \{1,\ldots, M\}$ we have

$$|(T(t)u)_k - u_k^*| \le ||T(t)u - u^*|| \ , \tag{25}$$

then (24) will give

$$\lim_{t\to\infty} (T(t)u)_k = u_k^* \ . \tag{26}$$

It means that probability of appearance of the population w^k in the step number t converges to a certain fixed number u_k^* independently of the initial distribution u. It is realized in some special case, when our implementation began at one specific population $\boldsymbol{p} = w^j$.

Theorem 1. *If the model is asymptotically stable, then*

$$\lim_{t\to\infty} ||R(t)u - \boldsymbol{p}^*|| = 0 \quad \text{for} \quad u \in \Gamma \ , \tag{27}$$

where $\boldsymbol{p}^* \in \overline{\Lambda}$ *is the expected population adequate to the distribution u*. Particularly, we have also*

$$\lim_{t\to\infty} ||R(t)\boldsymbol{p} - \boldsymbol{p}^*|| = 0 \quad \text{for} \quad \boldsymbol{p} \in W. \tag{28}$$

Proof. From (22) we have

$$R(t)u = \sum_{k=1}^{M} w^k (T(t)u)_k$$

and

$$\boldsymbol{p}^* = \sum_{k=1}^{M} w^k u_k^* \ .$$

Then

$$||R(t)u - \boldsymbol{p}^*|| = \sum_{j=0}^{s-1} |\sum_{k=1}^{M} w_j^k (T(t)u)_k - \sum_{k=1}^{M} w_j^k u_k^*|$$

$$\le \sum_{j=0}^{s-1}\sum_{k=1}^{M} w_j^k |(T(t)u)_k - u_k^*| = ||T(t)u - u^*||.$$

On the basis of (24) the equality follows (27). Taking into account our notation, given in (20), the formula (28) is the particular case of (27). △

Theorem 1 states that for the asymptotically stable case the expected population stabilizes, converging to $\boldsymbol{p}^* \in \overline{\Lambda}$ independently of initial conditions. This result has a fundamental meaning for the analysis of the convergence of genetic algorithms. This generalization will be the subject of our next paper. Moreover, this theorem is an extension of *Th.*4.2.2.4 4 from

[24] for the case when it is possible to attain any population in a finite number of steps, (not only in one step). It means that the transition operator does not need to be positively defined, but there exists such k, that the k-th power of the transition matrix possesses a column which is strongly positive. The same concerns *Th*.4.2.2 1 of [24, 25] which is true only for a positively defined transition matrix.

We shall say that from the chromosome z_a it is possible to obtain z_b in one mutation step with a positive probability if $U_{ba} > 0$. We shall say that from the chromosome z_a it is possible to get the chromosome z_b with positive probability in n-step mutation if there exists a sequence of chromosomes $z_{l_0}, \ldots, z_{l_n}$, such that $z_{l_0} = z_a$, $z_{l_n} = z_b$, and for any $k = 1, \ldots, n$ it is possible to attain the chromosome z_{l_k} from $z_{l_{k-1}}$ in one step with a positive probability.

Definition 3. *Model is pointwise asymptotically stable if there exists such a population w^j that*

$$\lim_{t\to\infty}(T(t)u)_j = 1 \text{ for } u \in \Gamma \ . \tag{29}$$

Condition (29) denotes that in successive steps the probability of appearance of a population other than w^j tends to zero. It is a special case of the asymptotic stability for which

$$u^* = e_j \ .$$

Theorem 2. *Model is pointwise asymptotically stable if and only if there exists exactly one chromosome z_a with such a property that it is possible to attain it from any chromosome in a finite number of steps with a positive probability. In this situation the population w^j is exclusively composed of the chromosomes z_a and*

$$T(t)w^j = w^j \tag{30}$$

holds. Moreover, the probability of appearance of population other than w^j tends to zero in the step number t with a geometrical rate, i.e. there exists $\lambda \in (0, 1)$, $D \in \boldsymbol{R}_+$ such that

$$\sum_{\substack{i=1\\ i\neq j}}^{M}(T(t)u)_i \leq D \cdot \lambda^t \ . \tag{31}$$

△

The proofs of our theorems and auxiliary lemmas are stated in other articles [29-31, 33].

From the formula (30) it follows, that from a population w^j we receive w^j with the probability equal 1. Moreover, if w^j becomes once, then from this moment on we shall permanently have populations w^j . Numbers λ and D could be determined for a specific model. It will be the subject of the next articles.

Theorem 2 states that the convergence to one population could occur only under specific assumptions. This justifies the investigation of the asymptotic stability which is different from that in Definition 3.

Definition 4. *By an attainable chromosome we denote $z_a \in Z$ such that it is possible to attain it from any other chromosome in a finite number of steps with a positive probability. Let us denote by Z^* the set of all z_a with this property.*

Theorem 3. *Model is asymptotically stable if and only if* $Z^* \neq 0$.

△

Theorem 4. *Let us assume that the model is asymptotically stable. Then the next relationship holds*:

(*war*) $u_k^* > 0$ *if and only if the population* w^k *is exclusively composed of chromosomes belonging to the set* Z^*. △

Corollary 1. *If* $Z^* = Z$ *then* $u_k^* > 0$ *for all* $k \in \{1,\ldots,M\}$. △

Here we set the summary of our results:

1. $Z^* = 0 \Rightarrow$ lack of asymptotic stability;
2. $Z^* \neq 0 \Rightarrow$asymptotic stability but:
3. cardinality $(Z^*) = 1 \Rightarrow$ pointwise asymptotic stability (in some sense convergence to one population);
4. cardinality $(Z^*) > 1 \Rightarrow$ asymptotic stability, but there is no pointwise asymptotic stability.

If one restricts to a binary simple genetic algorithm with a positive mutation probability, then it is possible to attain any individual (chromosome) from any other individual. Then there is more than one binary chromosome which is possible to attain from any other in a finite number of steps with a positive probability, and by Corollary 1, it is impossible to get the population composed exclusively of one type of chromosome. It could be interesting to consider non-binary cases for which the above observation does not hold.

3.2 Genetic algorithms with parameters adaptation

Genetic algorithm is realized as an adaptation process, hence it is natural to expect, that during its action its parameters are adapted on the base of some internal dynamics of the algorithm. It follows from the conjecture, that at different states, i.e. at different steps of the algorithm, values of algorithm parameters could be changed in the optimal way to accelerate the process convergence.

Till now the problem of algorithm parameters fitting is complex and not well defined, and it has an undefined structure. However, there exist many arguments for parameters adaptations that can improve action of actual genetic algorithm. There exists an opinion that by adding individual algorithm or metha-algorithm related to the actual one one can improve the solution of the problem. Such situation may be realized by an adaptation of genetic algorithms parameters on the base of the present state of the process (i.e. the actual population). It is conducted, for example, by introducing the methodology of parameters changing, which uses information on populations and values of the fitness function. The same can be proposed by a modification of the fitness function only.

In most case such adaptation is realised by increasing not only the dimension of chromosoms but also the search space, and consequently the population vector. Then, there appears an extra meta-algorithm, which runs parallel to the actual genetic one.

Even in such situations our algorithm model is conserved (16), and then the search space is enlarged (the arguments set) and in consequence the number of possible populations grow. The dimension of the Markovian matrix describing new, composed algorithm 18 grows. However, the transition operator (19) has the same properties as in the classical simple

genetic algorithm. Consequently, all theorems on convergence of genetic algorithms from the previous sections are conserved, as well as the results concerning the limit algorithm of the next Section 4.2 and the form of the optimal algorithm in probabilistic sense.

4. Classification of algorithms and its invariants

The convergence of GAs is one of the main issues of the theoretical foundations of GAs, and has been investigated by means of Markov's chains. The model of GA as a Markov's chain is relatively close to the methods known in the theory of dynamical systems.
In the analysis of GAs regarded as (stochastic) dynamical systems one can use the fact, (proven by Ornstein and Friedman [4, 10]) which states that mixing Markov's chains are Bernoulli's systems and consequently, the entropy of the systems is a complete metric invariant.
Those facts enable us to classify GAs using the entropy. The systems for which the entropies have the same value are isomorphic. Hence the entropy makes it possible to classify GAs by splitting them into equivalence classes.

4.1 Isomorphism of algorithms

The domain of research of the ergodic theory is a space with measure and mappings which preserve it. The measure space is the point set X with a measure m (when normalised to one, it is called the probability) defined on σ - algebra of its subsets $\mathcal{B}$, called measureable. To use results of the theory some defintions [16, 15] must be introduced.

Definition 5. *Let* $(X_1, \mathcal{B}_1, m_1)$, $(X_2, \mathcal{B}_2, m_2)$ *be measure spaces. We say that a mapping* $\phi : X_1 \to X_2$ *is measure preserving if: i) it is measurable, i.e.* $\phi^{-1}(A) \in \mathcal{B}_1$ *for every* $A \in \mathcal{B}_2$, *and ii)* $m_1(\phi^{-1}(A)) = m_2(A)$. *If* $X_1 = X_2$ *and* $m_1 = m_2 =: m$ *and* ϕ *preserves a measure m then we say that m is* ϕ*-invariant (or invariant under* ϕ*).*

In the example below we will say that so-called 1D *backer's transformation*[3] preserves Lebesgue measure of the line. Let $X = [0; 1)$ and consider $\phi_1(x) = 2x \pmod 1$. Notice that even though the mapping doubles the length of an interval I, its inverse image has two pieces in general, each of which has the length of I, and when we add them, the sum equals the original lenght of I. So ϕ_1 preserves Lebesgue measure.
The generalization of the above mapping to 2D is the backer' transformation defined[4] on the square $X = [0, 1] \times [0, 1]$ as

$$\phi_2(x,y) = \left\{ \begin{array}{ll} (2x, \frac{1}{2}y) \ , & 0 \le x \le \frac{1}{2} \\ (2x-1, \frac{1}{2}y + \frac{1}{2}) \ , & \frac{1}{2} \le x \le 1 \end{array} \right\} \tag{32}$$

which presereves the 2D Lebesgue measure on the unit square.

Definition 6. Probability spaces $(X_1, \mathcal{B}_1, m_1)$, $(X_2, \mathcal{B}_2, m_2)$ are said to be *isomorphic* if there exist $M_1 \in \mathcal{B}_1$, $M_2 \in \mathcal{B}_2$ with $m_1(M_1) = 1 = m_2(M_2)$ and an invertible measure preservimg transformation $\phi : M_1 \to M_2$.

---

[3] It is also called 1D Bernoulli shift.
[4] The transformation is the composition of three transformations of the unit square first, press down the square, cut in the midle and move the right half to the top of the left half.

In [16] the defintion is more general and requires the mapping ϕ to be defined on whole X_1 and be almost everywhere bijective from X_1 onto X_2, i.e. it must be bijective except for the sets of measure zero. However, in view of Definition 6 the sets $X_1 \backslash M_1$ and $X_2 \backslash M_2$ have zero measure.

In order to investigate genetic algorithms and their similarity (or even more - isomorphism) we need to consider mappings defined on probability space.

Definition 7. *Suppose probability spaces* $(X_1, \mathcal{B}_1, m_1)$, $(X_2, \mathcal{B}_2, m_2)$ *together with measure preserving transformations* $T_1 : X_1 \to X_1$; $T_2 : X_2 \to X_2$. *We say that* T_1 *is* ***isomorphic to*** T_2 *if there exist* $M_1 \in \mathcal{B}_1$, $M_2 \in \mathcal{B}_2$ *with* $m_1(M_1) = m_2(M_2) = 1$ *such that: i)* $T_1(M_1) \subseteq M_1$, $T_2(M_2) \subseteq M_2$, *and ii) there is an invertible measure-preserving transformation*

$$\phi : M_1 \to M_2 \quad with\ \phi(T_1(x)) = T_2(\phi((x)) \quad for\ all\ x \in M_1.$$

Consider infinite strings made of k symbols from [1,…, k]. Put $X = \prod_1^\infty \{1, ..., k\}$. An element x of X is denoted by $(x_1\, x_2\, x_3 \ldots)$.[5] Let a finite sequence $p_1, p_2, \ldots, p_k$, where for each i the number $p_i \in [0, 1]$ be such that $\sum_{i=1}^{k} p_i = 1$. For $t \geq 1$ define a cylinder set (or a block) of length n by

$$[a_1, a_2, ..., a_n]_{t,...,t+n-1} = \{x \in X : x_{t+1} = a_1, ..., x_{t+n} = a_n\} \tag{33}$$

With this denotation let us introduce the main definition of the Bernoulli shift which plays the main role in our approach [15, 16].

Definition 8. *Define a measure* μ *on cylinder sets by*

$$\mu([a_1, a_2, ..., a_n]_{t,...,t+n-1}) = p_{a_1} \cdots p_{a_n} \ . \tag{34}$$

A probability measure on X, again denoted by μ, *is uniquely defined on the* σ *- algebra generated by cylinder sets. We call* μ *the* $(p_1,\ldots,p_k)$*-Bernoulli measure and X is the Bernoulli shift space. The* ***one-sided Bernoulli shift transformation*** T *on X defined by*

$$(x_1 x_2 x_3 \ldots) \longmapsto (x_2 x_3 x_4 \ldots) \ , \tag{35}$$

i.e. $T(x_1 x_2 x_3 \ldots) = (x_2 x_3 x_4 \ldots)$.

Similarly, we may define the *two-sided Bernoulli shift transformation* by

$$(\ldots \overset{*}{x}_0 x_1 x_2 x_3 \ldots) \longmapsto (\ldots \overset{*}{x}_1 x_2 x_3 x_4 \ldots)$$

on $\prod_{-\infty}^{\infty} \{1, ..., k\}$ where * denotes the 0-th coordinate in a sequence. Let us notice that the shift preserves the measure μ.

In the case of a binary sequence when we have two symbols only and if each symbol has probablity $\frac{1}{2}$ the space X identified with $X = \prod_1^\infty \{0, 1\}$ is $(\frac{1}{2}, \frac{1}{2})$-Bernoulli shift space. Moreover, the space X is somorphic to [0, 1] with Lebesgue measure if each element $x = (b_1, b_2, \ldots) \in X$ and the transformation is defined by

[5] If $k = 2$ then x is said to be a **binary** sequence.

$$\phi(x) = \sum_{n=1}^{\infty} b_n 2^{-n} \ . \tag{36}$$

To see why, notice that not every $y \in [0, 1]$ has unique binary expansion, but the set of such points has measure zero, and we ignore them. Hence the transformation (36) is almost everywhere bijective (cf. remark below Def. 6) and measure preserving.
The next notions are related to Markov measure and Markov shift. As previously consider the space $X = \prod_1^{\infty}\{1, ..., k\}$. and let $\boldsymbol{P} = (P_{ij})$ be a $k \times k$ stochastic matrix with the right hand operation[6]. Suppose that $\boldsymbol{\pi} = (\pi_i)$ be the right probability eigenvector of $\boldsymbol{P}$, i.e. it satisfies $\sum_{i=1}^{k} \pi_i = 1$ and $\boldsymbol{P\pi} = \boldsymbol{\pi}$. Define ν on the cylinder sets by

$$\nu([a_1, \ldots, a_n])_{t,\ldots,t+n-1} = P_{a_n a_{n-1}} \ldots P_{a_2 a_1} \pi_{a_1} \ . \tag{37}$$

Notice that the sequence of appearance is $a_1, a_2, \ldots, a_n$.
Definition 9. *A unique shift invariant probability measure, again denoted by ν, on the σ-algebra generated by the cylinder sets, we call the Markov measure and then X is called the Markov shift space.*
Notice that the matrix $\boldsymbol{P}$ defines the transition probabilty

$$Pr(x_{n+1} = j | x_n = i) = P_{ji}$$

which is the conditional probabilty (of an event $x_{n+1} = j$ given that an event $x_n = i$ has occured). Notice that Markov shifts are Bernoulli shifts if the columns of the matrix $\boldsymbol{B}$ are identical. Moreover, the numbers $Pr(x_{n+1} = j | x_n = i)$ satisfy

$$\sum_{b=1}^{k} Pr(b|a) = 1$$

for any $a \in \{1, 2, \ldots, k\}$.
We can identify a Bernoulli measure or a Markov measure with a measure on the interval [0, 1] through the binary expansion (36) (i.e. each binary sequence $x = (b_1, b_2, \ldots)$ is identified with the sum of R.H.S. of (36)). If the probability $p \notin \{0, 1/2, 1\}$, then the $(p, 1\text{-}p)$- Bernoulli measure represented on [0, 1] is singular continuous [16].

4.2 Limit distribution

Now, after [16] we are ready to formualate main facts concerning the limit distribution of the Markov matrix.
Theorem 5. *Let $\boldsymbol{T} = (T_{ij})$ be a $M \times M$ stochastic matrix. Suppose that $\boldsymbol{\pi} = (\pi_i)$ be a right probability eigenvector of $\boldsymbol{T}$, i.e. it satisfies $\sum_{i=1}^{M} \pi_i = 1$ and*

$$\boldsymbol{T\pi} = \boldsymbol{\pi}. \tag{38}$$

Then the following relationship hold:

[6] Choe in [16] considers the left hand operation.

i. *there exists* $\boldsymbol{Q} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \boldsymbol{T}^n$.

ii. $\boldsymbol{Q}$ *is stochastic (i.e. Markovian) matrix ,*

iii. $\boldsymbol{Q}\boldsymbol{T} = \boldsymbol{T}\boldsymbol{Q} = \boldsymbol{Q}$,

iv. *If* $\boldsymbol{T}\boldsymbol{v} = \boldsymbol{v}$ *then* $\boldsymbol{Q}\boldsymbol{v} = \boldsymbol{v}$.

△

Theorem 6. *All columns of* $\boldsymbol{Q}$ *are identical and equal to the column vector* $\boldsymbol{\pi}$. △

Since each Markov shift is a Bernoulli shift if columns of the Markov matrix are identical, the limit distribution may be regarded as a Bernoulli shift. Hence the isomorphism of the limit distribution may be treated in the same way as for Bernoulli shifts, i.e. with the help of the entropy, cf. Theorem 9.

Theorem 7. *The convergence* $\|\boldsymbol{T}^n - \boldsymbol{Q}\|$ *is of exponential type, when* $n\to\infty$. △

One may ask whether it is possible to find a convergence bound in terms of the second eigenvalue of the matrix $\boldsymbol{T}$ and how it is related to the eigenvalues of the matrix $\boldsymbol{Q}$? Moreover, the limit operator $\boldsymbol{Q}$ is a projection operator $\boldsymbol{Q}\boldsymbol{Q} = \boldsymbol{Q}$. Its eigenspace is composed of one eigenvector $\boldsymbol{\pi}$ and its properties will help in finding relations to NFL. It will be the subject of the next publication [32].

Theorem 8. *If a genetic algorithm (14) is described by a transition matrix (18) that possesses the eigenvector* $\boldsymbol{\pi}$ *as a probability vector corresponding to the unit eigenvalue, i.e. the matrix satisfies Eq. (38), then there exists an optimal algorithm in the probabilistic sense. It means that the algorithm starting at an arbitrary initial distribution of populations in one step generates the limit distribution. This limit distribution is desrcibed by the matrix* $\boldsymbol{Q}$ *appearing in Theorem 5.*

Proof. Let a vector $\boldsymbol{c} = (c_i)$ describe the initial distribution of populations, with $\sum_{i=1}^{M} c_i = 1$. Let us take an arbitrary row of the matrix $\boldsymbol{Q}$, say j. Then in view of Theorem 6 all elements of this row are the same and equal to π_j. Then making the product $\boldsymbol{Q}\boldsymbol{c}$ we will get for this row

$$c_1\pi_j + c_2\pi_j + \cdots + c_M\pi_j = \pi_j(c_1 + c_2 + \cdots + c_M) = \pi_j.$$

This means that $\boldsymbol{Q}\boldsymbol{c} = \boldsymbol{\pi}$.

The recent theorem is in some sense complementary to the No Free Lunch Theorem. NFL Theorem describes the whole universe of optimization problems and algorithms used to solve them. The present theorem, on the other side, concernes on an individual algorithm dedicated to an individual optimization problem. The former theorem tells that in the mean all algorithms behave in similar way as far as all problems are concerned. The latter theorem, however, states that for allmost every genetic (evolutionary) algorithm and every single optimization problem there exists not only the better algorithm but also the best (optimal) in the probabilistic sense. This algorithm cannot be, in general, deterministic, since the assumptions concerning the pointwise asymptotic stability may not hold (cf. Definition 3 and Theorem 2). The problem of determining, even in the approximate form, the best algoritm is still open. It is hope that the pointwise asymptotic stability can be helpful here.

There is of course the question of uniquenss: two different genetic algoritms may lead to two different limit distributions. Moreover, to two different algorithms may correspond one optimal algorithm. This remark may be used in formulation new methods of classification of genetic algorithms, additional to the entropy and the fractal dimension.

5. Trajectory of BGA

Let X be a space of solutions of an optimisation problem characterized by a fitness function $f : X \rightarrow \mathbf{R}$;$X \subset \mathbf{R}^m$ for which a binary genetic algorithm (BGA) will be invented. Each element $x \in X$ will be encoded in the form of a binary chromosome of the length l (cf. Section 2.1). The coding function φ: $X \rightarrow \{0, 1\}^l = B$ maps elements of X into chromosome from the B space.

Let us assume that the genetic algorithm acts on r-element populations. Each population forms a multiset $[P^r]$ in the product space B^r. For the i-th generation we will use the denotation $[P_i^r]$, for the population and each element of this multiset can be identified with a vector

$$P_i^r = [x_1^i, x_2^i, \ldots, x_r^i] \tag{39}$$

rembering that a population is an equivalent class of points from the vector space B_r. The equivalent relation is defined by the class of all possible permutations of the set of r-th numbers {1, 2,…, r}. Notice that in view of our denotation from Sec.2.1 each x_j^i, j = 1, 2,…, r is one of elements of the set Z.

Let us notice that we can identify points from X with their encoded targets in B under the action of space X^r. By a trajectory of the genetic algorithm of the duration N we mean a set

$$\mathcal{T} = \bigcup_{i=1}^{N} [P_i^r] \ , \tag{40}$$

where N is the number of steps (generations) of the genetic algorithm which is realized.

Let p_m and p_c be the probabilities of the mutation and crossover, respectively, while p_s is the probability of selection, all independent from the generation.

Then, for such a genetic algorithm the probability of the appearance of the population $[P_{i+1}^r]$ at the generation i + 1 after the population $[P_i^r]$ at the generation i, is the conditional probability

$$P(P_{i+1}^r | P_i^r \ , f(P_i^r), p_m, p_c, p_s) \ . \tag{41}$$

Here by $f(P_i^r)$ we understand the vector{valued function of the form $[f(x_1^i), f(x_2^i),\ldots, f(x_r^i)]$. The initial population $[P_1^r]$ is generated by the use of a uniform probability distribution over the set B, i.e. each point from B has the same probability of being selected as a member

(component) of [P_1^r]. Next populations following that one, i.e. chosen in next generations, are results of the action of the GA and, hence, may have a non-uniform probability distribution.

Let us notice that in view of our assumptions it follows from (41) that the probability of the appearance of each population depends on the previous population and does not depend on the history (i.e. on earlier population; the probabilities p_m, p_c and p_s can be regarded as parameters of the function P).

Now, if we look at the trajectory of the GA defined by (40), we can see that its generation is an ergodic (mixing) process and Markov's one. Subsequent populations (i.e. points of the trajectory) are states of the process about which we can say that each state is accessible with the probability 1.

6. Entropy

Let us denote by T_i the operator which maps i-th generation (point of the trajectory) into the next one. Having the probability distribution (41) characterizing the mapping T_i from one population to another, we can define the entropy of the mapping

$$H(\mathtt{T}_i) = -\sum_{j=1}^{N} P(P_{i+1,j}^r | P_i^r, f(P_i^r), p_m, p_c, p_s) \cdot \log P(P_{i+1,j}^r | P_i^r, f(P_i^r), p_m, p_c, p_s) \tag{42}$$

where [$P_{i+1,j}^r$] is a possible population from the coding space B, j = 1, 2,…, 2^{N_r} ,…,M:

According to our previous proposition the initial population is generated by the use of a uniform probability, and the entropy may attain the maximal value generated by the GA. In the next step the probabilities of populations are not uniform and differ at each generation; this is the essence of the action of GA. Consequently the entropy of the mapping T_i decreases. In the limit case when the number of steps tends to infinity one could expect that the terminal population will be composed of r copies (exactly speaking, according to (39) { a cartesian product) of the same element (an optimal solution). However, this case will be possible only in the case of the pointwise asymptotic stability of GA. In general, the entropy will tend to minimum.

Entropy as a function of the probability of mutation and selectio grows with the growing mutation probability and decreases when the selection pressure grows. Then the entropy could realize a measure of interactions between mutations and selection operators. Entropy also depends on the number of elements in population and it is decreasing when the population grows. The entropy value of the trajectory could be linked with computational complexity of the evolutionary algorithms.

Now several questions arise. Does an optimal form of the entrop change exist? What is its limit value, if it is different from zero for the optimisation process performed by GA ? Does an optimal process of the entropy change exist along which an optimal value of the solution can be reached?

Since the determination of the probability of the mapping T_i, as well as the entropy H_i, in an analytical way is rather difficult to be performed, we are proposing to substitute them with a fractal dimension which is related to the entropy [10] and can characterize non-

deterministic features of GA. It should be mentioned that in [8] general statistical and topological methods of analysis of GAs have been introduced from another viewpoint.

Theorem 9. *(Ornstein [10]) Every two Bernoulli shifts with the same entropy are isomorphic.* △

Lemma 1. *(Choe [16])) Let* $X = \prod_1^\infty \{0,1\}$ *be the* (*p*, 1- *p*) *Bernoulli shift space that is regarded as the unit interval* [0, 1) *endowed with the Euclidean metric. Let* X_p *denote the set of all binary sequences* $x \in X$ *such that*

$$X_p = \{x \in X : \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} x_i = p\}$$

then Hausdorff dimension of the set X_p *is equal to the entropy -p* $\log_2 p$- (1 - *p*) $\log_2(1 - p)$ *of the Bernoulli shift transformation. Similar results can be obtained for a Markov shift space.* △

Moreover one can use the Hausdorff dimension or its approximation as an invariant of equivalence of algorithms.

7. Fractal dimensions

To be more evident, let us recall the notion of the *s*-dimensional[7] Hausdorff measure ([5]) of the subset $E \subset \mathbf{R}^l$, where $s \geq 0$. If $E \subset \bigcup_i U_i$ and the diameter of U_i, denoted by $\delta(U_i)$, is less than ϵ for each *i*, we say that $\{U_i\}$ is an ϵ - **cover of** *E*. For $\epsilon > 0$, let us define

$$\mathcal{H}_\epsilon^s(E) = \inf \sum_{i=1}^{\infty} [\epsilon(U_i]^s \tag{43}$$

where the in_mum is over all ϵ-covers $\{\boldsymbol{U}_i\}$ of $\boldsymbol{E}$. The limit of $\mathcal{H}_\epsilon^s$ as $\epsilon \to 0$ denoted by $\mathcal{H}^s$(E), is the s-dimensional Hausdorff measure of E.

Let us notice that in the space $\boldsymbol{R}^l$ one can prove that $\mathcal{H}^l(\boldsymbol{E}) = k^l \mathcal{L}^l(\boldsymbol{E})$, where $\mathcal{L}^l$ is the ***l***-dimensional Lebesgue measure and $\boldsymbol{k}^l$ is a ratio of volume of the ***l*** - dimensional cube to ***l*** - dimensional ball inscribed in the cube.

It is evident that $\mathcal{H}_\epsilon^s(E)$ increases as the maximal diameter ϵ of the sets U_i tends to zero, therefore, it requires to take finer and finer details, that might not be apparent in the larger scale into account. On the other hand for the Hausdorff measure the value $\mathcal{H}^s(E)$ decreases as *s* increases, and for large *s* this value becomes 0. Then the **Hausdorff dimension** of *E* is defined by

$$\dim_H(E) = \inf\{s : \mathcal{H}^s(E) = 0\}, \tag{44}$$

and it can be verified that $\dim_H(E) = \sup\{s : \mathcal{H}^s(E) = \infty\}$.

Working with compact subsets of a metric space (*X*, *d*) new dimension is introduced. This dimension is also less accurate than the Hausdorff dimension. To calculate this dimension

[7] This *s* has nothing to do with *s* introduced in Section 2.

for a set $S \subset X$ imagine this set lying on an evenly-spaced grid. Let us count how many boxes are required to cover the set. The **box-counting** dimension is calculated by observing how this number changes as we make the grid finer. Suppose that $N(\epsilon)$ is the number of boxes of the side length ϵ required to cover the set. Then the box-counting dimension is defined as:

$$\dim_{box}(S) = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)} \tag{45}$$

In Appendix more detailed presentation of properties of the Hausdorff and box-counting dimensions is included. Harrison in [5] recommends the box-counting dimension to be used only for closed sets, although even for compact sets it can differ from Hausdorff dimension and, moreover, the box dimension gives the most natural result than the measure $\mathcal{H}^s$.

8. Dimension of trajectory

By inventing the fractal (Hausdorff) dimension the trajectory of GA's or its attractor can be investigated. Algorithms could be regarded as equivalent if they have the same computational complexity while solving the same problem. As the measure of computational complexity of genetic algorithm, we propose a product of population's size and the number of steps after which an optimal solution is reached. This measure of computational complexity of genetic algorithms joins the memory and the temporal complexity.

During the execution of genetic algorithms, a trajectory is realized and should "converge" to some attraction set. It is expected that an ideal genetic algorithm produces an optimal solution which, in the term of its trajectory, leads to an attractor which is one{ element set. On the other hand, for an algorithm without selection the attractor is the whole space. Then, we could say that algorithms are equivalent when they produce similar attractors [6].

Our proposal is to use fractal dimensions to measure the similarity of attractors on the base of Lemma 1.

Definition 10. *Two genetic algorithms are* ***equivalent*** *if they realize trajectories with the same fractal dimension.*

Hence, instead of the entropy, the fractal dimension will be use as an indicator, or better to say - a measure of the classifications of GAs.

The transfer from the entropy to the new indicator can be made with the help of particular gauges. The first gauge could be the so-called ρ-entropy based dimension introduced by Pontrjagin and Schnirelman in 1932 (and repeated by Kolmogorov and Tihomirov in 1959), in the following way: among all collections of balls of radius ρ that cover a set E in $\boldsymbol{R}^l$ (or in more general case, in some metric space) is by definition one that requires the smallest number of balls. When E is bounded, this smallest number is finite and can be denoted by $\boldsymbol{N}(\rho)$ and called ρ - entropy. Their dimension, called the ***lower entropy dimension***, was defined by

$$\lim_{\rho \to 0} \inf \frac{\log \mathsf{N}(\rho)}{\log(\frac{1}{\rho})}. \tag{46}$$

The second gauge is the so-called *information dimension* of the trajectory defined by:

$$D_I(\mathcal{T}) = -\lim_{\epsilon \to 0} \frac{\sum_{i=1}^{W(\epsilon)} p_i \ln p_i}{\ln(\frac{1}{\epsilon})} \tag{47}$$

where $W(\epsilon)$ is the number of elements of the trajectory which are contained in a l - dimensional cube with the edge length equal to ϵ, and $p_i = \frac{N_i}{N}$ is the probability of finding of i - *th* element, and N_i - number of points in i-th hypercube, N - number of trajectory points. In further analysis we are going to replace (47) and (45) with its approximation, namely the *box* or *capacity dimension*.

In [6] the box counting dimension de_ned in [3] has been introduced with its approximated formula (cf. (2) in [6]).

Here we use another approach to the approximation. Let $N(\mathcal{T}, \epsilon)$ be the minimum number of r-dimensional cubes with the edge length equal to ϵ, that covers the trajectory $\mathcal{T} \subset X$, and X is a l- dimensional search space. To be more evident let us consider the case when $\epsilon = 2^{-k}$ and diminish the length of cube edges by half. Then the following ratio will approximate the box counting dimension of trajectory $\mathcal{T}$

$$D_c(\mathcal{T}) \approx \frac{\log_2 N(T, 2^{-(k+1)}) - \log_2 N(T, 2^{-k})}{\log_2 2^{k+1} - \log 2^k} = \log_2 \frac{N(T, 2^{-(k+1)})}{N(T, 2^{-k})} \tag{48}$$

due to the fact that $\log_2 x = \log_2 e \ln x$. The approximated expression (48) of the box dimension counts the increase in the number of cubes when the length of their edges is diminished by half.

8.1 Compression ratio

It is our conjecture that some characteristic feature of the trajectory of GA can be obtained by analysing the ration of the compressed trajectory to itself. We decided to investigate Lempel-Ziv compression algorithm [17] applied to populations executed by various genetic algorithms. We implemented five De Jong's functions with 10 different parameters sets. Each experiment was run 10 times. All together we obtained 500 different trajectories. The following settings of algorithms were considered

EXP	CROS	PC	PM	SEL
1	1	0.25	0.001	t
2	2	0.6	0.01	r
3	u	0.95	0.05	p
4	1	0.6	0.05	p
5	2	0.25	0.001	r
6	u	0.6	0.01	t
7	1	0.95	0.01	r
8	2	0.95	0.001	p
9	u	0.25	0.05	t
10	1	0.95	0.99	r

where EXP is the experiment number; CROS is type of crossover operator (one point, two point, uniform); PC and PM are probabilities of crossover and mutation, respectively; and SEL is type of selection operator (tournament, rank, and proportional). In each experiment the population consisted of 25 points and the genetic algorithm was realized on 100 generations (points).

We have performed numerous experiments on compressing particula generations with Lempel-Ziv algorithm of various bit resolution. We have measured number of prefixes resulting from compression process and corresponding compression ratio in scenarios of two types. The first one has considered single generations, and for each trajectory we have obtained corresponding trajectory of number of prefixes used. In the second scenario, each next generation was added to all past generations forming an ascending family of sets of generations. Compressing elements of such family gave an overall picture how number of prefixes used in the compression stabilizes over time.

8.2 Experiments with dimensions

The first experiments with attractors generated by GAs and the expression (48) have been performed by our co-worker in [6]. His results allow us to claim that the present approach can be useful in the GA's dynamics research.

In our paper we include new calculation results. 12 benchmark functions were used (cf. [13, 7]) in the analysis. Experiments were performed for different dimension: 10, 15, 20 bits with operator parameters and Popsize. Then the box counting dimension was used to calculate the trajectory dimension.

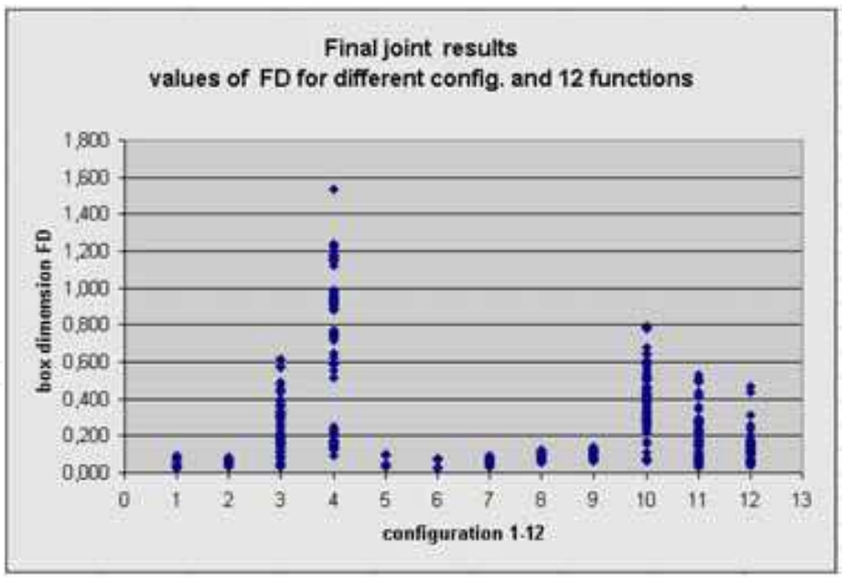

Fig. 1. Final joint results of fractal dimension

As far as the analytical approach and the formal definitions of dimensions (43) and (47) are concerned their computer implementation needs additional investigations. Computer accuracy is finite, hence all limits with e tending to zero will give unrealistic results. For example, if in (47) the calculation drops below the computing accuracy the expression value becomes zero or undefined. It means that we have to stop taking limit values in early stage.

Hence, the questions arise: to which minimal value of *e* the calculation should be performed and whether and how the relations with limits should be substituted with finite, non-asymptotic, expression? This, however, will be the subject of our further research.
The main idea of our experiments was the verification and confrontation of our theoretical considerations and conjectures with real genetic algorithms.

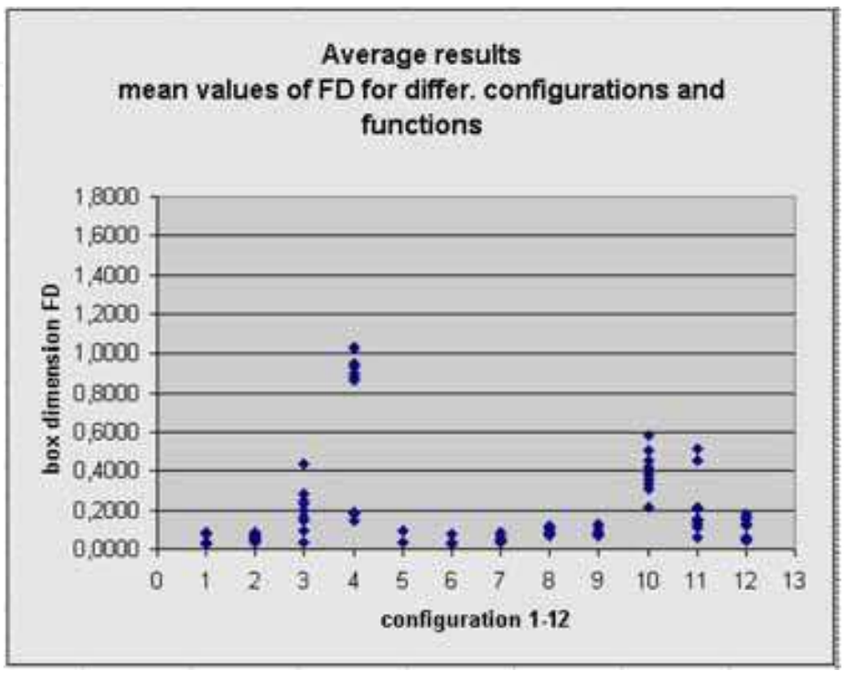

Fig. 2. Average results of fractal dimension

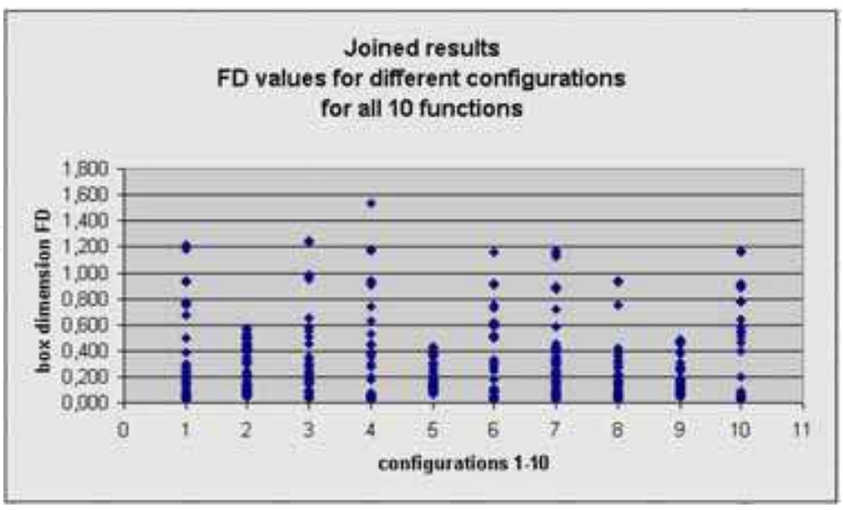

Fig. 3. Joint results of fractal dimension

On the basis of our experiments we can conclude that:
1. Selection.
Change of the selection methods while preserving the other parameters does not effect the values of fractal dimension.

2. Crossover.
When the number of crossover positions is changing the fractal dimension is growing with roulette selection method and is decreasing when selection is a tournament.
3. Populations.
Fractal dimension is growing with the number of individuals in population.
4. Mutation probability changes have small implication on the value of fractal dimension.
The analysis of the experimental result.
The value of box-counting dimension of the trajectory of genetic algorithms is not random. When we use the same fitness function and the same configurations, then the box dimensions become clustered near the same value. Whole trials of the independent running attains the same values. Moreover with the different functions but the same configuration we deal with the conservation of box-counting dimension clustering.
Average values of the box-counting dimension for the united trajectories of the algorithms from the same trial were similar to these which were calculated by averaging of the dimension of individual trajectories. This fact acknowledges the conjectures that box-counting dimension could characterize the complexity of algorithms. Box-counting dimension describes the way of evolution during search. Algorithms which attain the maximum in a wide loose set have bigger dimension than others which trajectories were narrow, with small differences between individuals.
One can say that bigger box dimension characterizes more random algorithms. The main result of the experiments states that fractal dimension is the same in the case when some boxes contains one individual as well as when these boxes contain many elements (individuals). Box dimension does not distinguish the fact that two or more elements are in the same place. They undergo counting as one element. The value of dimension should depend on the number of elements placed in each box. Our main conclusion is that good characterization is the information dimension.

9. Conclusions

One of the main results reported in this Chapter is the limiting algorithm and populations' distribution at the end of infinite steps. Theorem 5 does not tell about the form of the next population when actual population is known; it gives rather the limit distribution of all possible populations of the algorithm considered. The limiting algorithm describes globally the action of the genetic algorithm. It plays the role of the law of big numbers, known from the probability theory, however, for genetic algorithms. Knowledge the limiting algorithm could help in standard calculations: just in one step one could obtain the limit distribution. It could accelerate calculations and gives chance to omit the infinite numbers of calculation steps.
If the limiting algorithm is known an extra classification tool is for our disposal, and new hierarchial classification method can be suggested. It will base not only on entropy, fractal and dimensions of trajectory, but on transition matrix $\boldsymbol{T}$, its eigenvalues, eigenvectors and limiting matrix $\boldsymbol{Q}$. This hierarchie could be as follows:

- Two genetic algorithms are equivalent if their transition matrices are the same.
- Two genetic algorithms are equivalent if they have the same limit distribution $\boldsymbol{\pi}$.
- Two genetic algorithms are equivalent if their limiting algorithm, described by the matrix $\boldsymbol{Q}$ is the same.

- Two genetic algorithms are equivalent if the entropy of their trajectories is the same.
- Two genetic algorithms are equivalent if the fractal (box-counting, information, Hausdorff) dimensions of their trajectories are the same.
- Two genetic algorithms are equivalent if they generate the same order in populations.

We can see that the proposed scheme of classification referes to concepts known in the probability theory and the theory of dynamical systems. The open question is the role of different concepts and their importance. Is it possible to introduce the order relations in the proposed scheme? This will be investigated in the next publications.

10. Acknowledgement

The research work on the paper was partially done by W.K and S.K. in the framework of the KBN Project (State Committee for Scientific Research) No. 3 T11 C007 28. Authors thanks Professor Zbigniew Michalewicz and Dr. Jolanta Soca la for the inspiration and valuable discussions.

11. Appendix

Fractal and box - counting dimensions

To make the definitions more evident let us notice that for the graph Γ^f of a smooth, i.e. C^1, real function f of one variable we have $\dim_H(\Gamma^f) = 1$, while if the function f is C^ϵ (i.e. Hölder continuous of class ϵ) then $\dim_H(\Gamma^f) \leq 2 - \epsilon$. The Hausdorff dimension of the Peano curve has dimension 2 while the Hausdorff dimension of the Cantor middle set is log2=log3, while its topological dimension D^T is zero. In most cases Hausdorff dimension ≥ the topological one. In its classical form a **fractal** is by definition a set for which the Hausdorff dimension strictly exceeds the topological dimension.

Topological dimension takes non-negative integer values and is invariant under homeomorphism, while the Hausdorff dimension is invariant under bi-Lipschitz maps (sometimes called quasi-isometries). For self-similar sets ([5, 3]) that are built from pieces similar to the entire set but on a finer and finer scale, and can be regarded as an invariant set for a finite set of contraction maps on $\mathbf{R}^l$, the Hausdorff dimension is the same as its similarity dimension[8] It is the theory of fractal and its main object of interest, namely **iterated function systems** where fractal dimensions are commonly in use [2]. Deterministic and random algorithms are constructed for computing fractals from iterated function systems. However, such procedure are mostly implemented for 2D case, i.e. for fractals in $\mathbf{R}^2$. For genetic algorithm applications such tools are of small importance. More investigations on the similarities between genetic algorithms and iterated function systems with probabilities ([2]) are needed.

In fractal geometry, the Minkowski dimension is a way of determining the fractal dimension of a set S in a Euclidean space $\mathbf{R}^n$, or more generally of a metric space (X, d). This dimension is also, less accurately, sometimes known as the packing dimension or the box-counting

[8] Let **frig** be the contraction ratios of the family of contraction maps (S_1, S_2,…,S_m) and E be the invariant set for this family, then the unique positive number s such that $\sum_{i=1}^{m} r_i^s = 1$ is the similarity dimension of E ([5]).

dimension. To calculate this dimension for a fractal S, imagine this fractal lying on an evenly-spaced grid, and count how many boxes are required to cover the set. The **box-counting** dimension is calculated by seeing how this number changes as we make the grid finer. Suppose that $N(\epsilon)$ is the number of boxes of side length ϵ required to cover the set. Then the box-counting dimension is defined as:

$$\dim_{box}(S) = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)} \tag{49}$$

If the limit does not exist then one must talk about the **upper box dimension** and the **lower box dimension** which correspon to the upper limit and lower limit respectively in the expression above. In other words, the box-counting dimension is well defined only if the upper and lower box dimensions are equal. The upper box dimension is sometimes called the **entropy dimension**, Kolmogorov dimension, Kolmogorov capacity or upper Minkowski dimension, while the lower box dimension is also called the **lower Minkowski dimension**. Both are strongly related to the more popular Hausdorff dimension. Only in very specialized applications is it important to distinguish between the three. See below for more details. Also, another measure of fractal dimension is the correlation dimension.
Both box dimensions are finitely additive, i.e. if a finite collection of sets $\{A_1,A_2,\ldots,A_n\}$ is given then

$$\dim_{box}(\{A_1 \ldots \cup A_n\}) = \max\{\dim_{box}(A_1), \ldots, \dim_{box}(A_n)\} \ .$$

However, they are not countably additive, i.e. this equality does not hold for an infinite sequence of sets. For example, the box dimension of a single point is 0, but the box dimension of the collection of rational numbers in the interval [0, 1] has dimension 1. The Hausdorff dimension by comparison, is countably additive. An interesting property of the upper box dimension not shared with either the lower box dimension or the Hausdorff dimension is the connection to set addition. If A and B are two sets in a Euclidean space then $A + B$ is formed by taking all the couples of points a, b where a is from A and b is from B and adding $a + b$. One has

$$\dim_{upperbox}(A + B) \leq \dim_{upperbox}(A) + \dim_{upperbox}(B).$$

Relations to the Hausdorff dimension The box-counting dimension is one of a number of definitions for dimension that can be applied to fractals. For many well behaved fractals all these dimensions are equal. For example, the Hausdorff dimension, lower box dimension, and upper box dimension of the Cantor set are all equal to log(2)/ log(3). However, the definitions are not equivalent. The box dimensions and the Hausdorff dimension are related by the inequality

$$\dim_H(S) \leq \dim_{lowerbox}(S) \leq \dim_{upperbox}(S) \tag{50}$$

In general both inequalities may be strict. The upper box dimension may be bigger than the lower box dimension if the fractal has different behaviour in different scales. For example, examine the interval [0, 1], and examine the set of numbers satisfying the condition for any n, all the digits between the 2^{2n}-th digit and the 2^{2n+1}-1-th digit are zero. The digits in the

"odd places", i.e. between $2^{2n} + 1$ and $2^{2n+2} - 1$ are not restricted and may take any value. This fractal has upper box dimension 2/3 and lower box dimension 1/3, a fact which may be easily verified by calculating $N(\epsilon)$ for $\epsilon = -10^{2^n}$ and noting that their values behaves differently for n even and odd. To see that the Hausdorff dimension may be smaller than the lower box dimension, return to the example of the rational numbers in [0, 1] discussed above. The Hausdorff dimension of this set is 0.

Box counting dimension also lacks certain stability properties one would expect of a dimension. For instance, one might expect that adding a countable set would have no effect on the dimension of set. This property fails for box dimension. In fact

$$\dim_{box}(\{0, 1, 1/2, 1/3, 1/4, ...\}) = \frac{1}{2}$$

It is possible to define the box dimensions using balls, with either the covering number or the packing number. The covering number $N_{covering}(\epsilon)$ is the minimal number of open balls of radius ϵ required to cover the fractal, or in other words, such that their union contains the fractal. We can also consider the intrinsic covering number $N'_{covering}(\epsilon)$, which is defined the same way but with the additional requirement that the centers of the open balls lie inside the set S. The packing number $N_{packing}(\epsilon)$ is the maximal number of disjoint balls of radius ϵ one can situate such that their centers would be inside the fractal. While N, $N_{covering}$, $N'_{covering}$ and $N_{packing}$ are not exactly identical, they are closely related, and give rise to identical definitions of the upper and lower box dimensions. This is easy to prove once the following inequalities are proven:

$$N'_{covering}(2\epsilon) \leq N_{covering}(\epsilon), \quad \text{and } N_{packing}(\epsilon) \leq N'_{covering}(\epsilon). \tag{51}$$

The logarithm of the packing and covering numbers are sometimes referred to as **entropy numbers**, and are somewhat analogous (though not identical) to the concepts of **thermodynamic entropy** and **information-theoretic entropy**, in that they measure the amount of "disorder" in the metric space or fractal at scale ϵ, and also measure how many "bits" one would need to describe an element of the metric space or fractal to accuracy ϵ.

Sometimes it is just too hard to find the Hausdorff dimension of a set E, but possible for other definitions that have some restriction on the ϵ -covers considered in the definition. We recall here the most common alternative. It is the **box dimension**, introduced by Kolmogorov in 1961 (cf.[5]), and which is defined in the same way as Hausdorff dimension except that in the definition of measure only balls (discs) in $\mathbf{R}^l$ of the same radius ϵ are considered for covers of E. It follows that box dimension of E is always ≥ dim(E). Moreover the box dimension of the closure of E is the same as for the set E itself. Since the box-counting dimension is so often used to calculate the dimensions of fractal sets, it is sometimes referred to as "fractal dimension". We prefer the term box dimension, however, because sometimes the term "fractal dimension" might refer to box dimension, Hausdorff dimension, or even other measures of dimension such as the information dimension or capacity dimension.

Sometimes box counting dimension is referred to as "similarity dimension" in the context of self-similar sets. If a set is self-similar, there is an expansion factor r by which one can blow

up a small copy to get the whole set. If there are exactly N such small copies that make up the entire set, the box dimension is easily seen to be $\ln N / \ln r$.

Let us consider the $\mathcal{Q} = \{p/q | p, q \text{ integers}\}$ be the set of rational numbers in the interval [0, 1], that is $p \leq q$ are relatively prime integers. Since the rationals are dense in [0, 1], any interval we choose contains some. This means for every ϵ we need $N_\epsilon = \frac{1}{\epsilon}$ boxes to cover the whole $\mathcal{Q}$. Consequently $\dim_{box}(\mathcal{Q}) = \lim_{\epsilon \to 0} \frac{\frac{1}{\epsilon}}{\frac{1}{\epsilon}} = 1$. Thus the box dimension of the rational numbers is 1.

The last example will be given by the new set $\mathcal{P}$ = {$x \in$ [0, 1]} x has a decimal expansion which does not contain 4 nor 5. Notice that 0.4 has the two representations, namely .4 and .39999(9). The set $\mathcal{P}$ is disconnected: it does not contain the open interval (0.4, 0.6). We shall see that the set is closed and also self-similar: any small piece of it can be scaled up to look like the whole thing just by multiplying by an appropriate power of 10. It can be proven that

$$\dim_{box} \mathcal{P} = -\lim_{n \to \infty} \frac{\ln 8^{n-1}}{\ln (4 \cdot 10^{-n})} = \frac{\ln 8}{\ln 10} = \approx 0.903089987$$

At the same time the topological dimension of $\mathcal{P}$ is zero.

12. References

Baker G.L. and Gollub J.P.: *Chaotic Dynamics: an Introduction*, Cambridge Univ. Press, Cambridge, 1992.

Barnsley M. F.: Lecture notes on iterated function systems, in *Chaos and Fractals.The Mathematics Behind the Computer Graphics, Proc.Symp. Appl. Math.*, vol. 39, R. L. Denamney and L. Keen (eds.) American Mathematical Society, Providence, Rhode Island, pp. 127-144, 1989.

Falconer K.J.: Fractal geometry, *Math. Found. Appl.* John Wiley, Chichester, pp.15 -25, 1990.

Friedman N.A., Ornstein D.S.: On isomorphisms of weak Bernoulli transformations, *Adv. in Math.* , 5, pp. 365-394, 1970.

Harrison J.: An introduction to fractals, in *Chaos and Fractals. The Mathematics Behind the Computer Graphics, Proc.Symp. Appl. Math.*, vol. 39, R. L. Denamney and L. Keen (eds.) American Mathematical Society, Providence, Rhode Island, pp. 107-126, 1989.

Kieś P.: Dimension of attractors generated by a genetic algorithm, in *Proc. of Workshop Intelligent Information Systems IX* held in Bystra, Poland, June 12-16, pp. 40-45, 2000.

Michalewicz Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd, rev. edition, Springer, Berlin, Heidelberg *et al.*, 1996.

Ossowski A.: Statistical and topological dynamics of Statistical and topological dynamics of evolutionary algorithms, in *Proc. of Workshop Intelligent Information Systems IX* held in Bystra, Poland, June 12-16, pp. 94-103, 2000.

Ott E.: *Chaos in Dynamical Systems* Cambridge Univ. Press, Cambridge, 1996.

Ornstein D.S.: *Ergodic theory, Randomness and Dynamical Systems*, Yale Univ. Press, 1974.

Vose M.D.: Modelling Simple Genetic Algorithms, *Evolutionary Computation*, 3 (4) 453-472, 1996.

Wolpert D.H. and Macready W.G.: No Free Lunch Theorems for Optimization, *IEEE Transaction on Evolutionary Computation*, 1 (1), 67-82, 1997, http://ic.arc.nasa.gov /people/dhw/papers/78.pdf

Igel, C., and Toussaint, M.: "A No-Free-Lunch Theorem for Non-Uniform Distributions of Target Functions," *Journal of Mathematical Modelling and Algorithms* 3, 313-322, 2004.

English, T.: No More Lunch: Analysis of Sequential Search, *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pp. 227-234. 2004, http://BoundedTheoretics.com /CEC04.pdf

Szlenk W., *An Introduction to the Theory of Smooth Dynamical Systems*.,PWN, Warszawa,John Wiley&Sons, Chichester, 1984 G.H.

Choe G. H., *Computational Ergodic Theory*. Springer, Heidelber, New York 2005

G. Frizelle G., Suhov Y.M.: An entropic measurement of queueing behaviour in a class of manufacturing operations. *Proc. Royal Soc. London A* (2001) 457, 1579- 1601.

A. Lasota, Asymptotic properties of semigroups of Markov operators (in Polish), *Matematyka Stosowana. Matematyka dla Społeczeństwa*,PTM, Warszawa, 3(45), 2002, 39-51.

P. Kieś and Z. Michalewicz, Foundations of genetic algorithms (in Polish), *Matematyka Stosowana. Matematyka dla Społeczeństwa* , PTM Warszawa 1(44), 2000, 68{91.

Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin, 1996.

M. D. Vose, *The Simple Genetic Algorithm: Foundation and Theory*, MIT Press, Cambridge, MA, 1999.

A. Lasota, J. A. Yorke, Exact dynamical systems and the Frobenius-Perron operator, *Trans. Amer. Math. Soc.* 273 (1982), 375{384.

J. E. Rowe, The dynamical system models of the simple genetic algorithm, in *Theoretical Aspects of Evolutionary Computing*, Leila Kallel, Bart Naudts, Alex Rogers (Eds.), Springer, 2001, pp.31-57.

R. Schaefer, *Podstawy genetycznej optymalizacji globalnej* (in Polish), Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2002.

R. Schaefer, *Foundations of Global Genetic Optimization*, Series: Studies in Computational Intelligence, Vol. 74, Springer, Berlin, Heidelberg, 2007,

R. Rudnicki, On asymptotic stability and sweeping for Markov operators, *Bull. Polish Acad. Sci. Math.*, 43 (1995), 245-262.

J. Socała, Asymptotic behaviour of the iterates of nonnegative operators on a Banach lattice, *Ann. Polon. Math.*, 68 (1), (1998), 1-16.

J. Socała, W. Kosiński, S. Kotowski, On asymptotic behaviour of a simple genetic algorithm (in Polish: O asymptotycznym zachowaniu prostego algorytmu genetycznego), *Matematyka Stosowana. Matematyka dla Społeczeństwa* , PTM, Warszawa,6 (47), 2005, 70-86.

J. Socała, W. Kosiński, Lower-bound function method in the converegence analysis of genetic algorithms, (in Polish: Zastosowanie metody funkcji dolnej do badania zbieżności algorytmów genetycznych, *Matematyka Stosowana. Matematyka dla Spo leczeństwa*, PTM, Warszawwa, 8 (49), 2007 , 33-44.

J. Socała, W. Kosiński, On convergence of a simple genetic algorithm, *ICAICS, 9-th International Conference on Artifical Intelligence and Soft Computing, 2008*, LNAI, Springer, Berlin, Heidelberg, poz.366, in print.

J. Socała, Markovian approach to genetic algorithms, under preparation.

S.Kotowski, W. Kosiński, Z. Michalewicz, J. Nowicki, B. Przepiórkiewicz, Fractal dimension of trajectory as invariant of genetic algorithms, *ICAICS, 9-th International Conference on Artifical Intelligence and Soft Computing, 2008*, LNAI, Springer, Berlin, Heidelberg, poz.401, in print.

S. Kotowski, Analysis of genetic algorithms as dynamical systems (in Polish: Analiza algorytmów genetycznych jako układów dynamicznych, under preparation, 2008.

Advances in Evolutionary Algorithms

Edited by Xiong Zhihui

ISBN 978-953-7619-11-4
Hard cover, 284 pages
Publisher InTech
Published online 01, November, 2008
Published in print edition November, 2008

With the recent trends towards massive data sets and significant computational power, combined with evolutionary algorithmic advances evolutionary computation is becoming much more relevant to practice. Aim of the book is to present recent improvements, innovative ideas and concepts in a part of a huge EA field.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen