

On Classification Tools for Genetic Algorithms

Stefan Kotowski*[†], Witold Kosiński*[‡], Zbigniew Michalewicz*[§], Piotr Synak* and Łukasz Brocki*

*Faculty of Computer Science, Polish-Japanese Institute of Information Technology ul. Koszykowa 86 02-008 Warszawa, Poland

[†] Institute of Fundamental Technological Research

IPPT PAN, ul. Świętokrzyska 21, 00-049 Warszawa Poland

[‡]Institute of Environmental Mechanics and Applied Computer Science

Kazimierz Wielki University

ul. Chodkiewicza 30, 85-064 Bydgoszcz, Poland

[§] University of Adelaide, School of Computer Sciences,

South Australia 5005, Australia

{skot, wkos, synak, lucas}@pjwstk.edu.pl, zbyszczek@cs.adelaide.edu.au

Abstract—Some tools to measure convergence properties of genetic algorithms are introduced. A classification procedure is proposed for genetic algorithms based on a conjecture: the entropy and the fractal dimension of trajectories produced by them are quantities that characterize the classes of the algorithms. The role of these quantities as invariants of the algorithm classes is discussed together with the compression ratio of points of the genetic algorithm.

I. INTRODUCTION

THERE is the so-called “No-free lunch theorem” [10] algorithms and moreover, one cannot find most suitable operator according to which: it does not exist a best evolutionary 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 [11], [12].

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

The distinguishing between optimisation problem and the algorithm (its choice) leads 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 form the formulation of the optimisation problem and it is difficult to speak on genetic algorithm as an operator without the fitness function. However, in our consideration we will simultaneously use the 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 our consideration, because our main aim is to try to

classify genetic algorithms. The classification should lead to a specific choice 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 aim that using this classification one could improve existing heuristic methods of assortment of genetic algorithms which are based mainly on experiences and programmer intuition.

During the action of genetic algorithm several random operations are performed and each action generates a sequence of random variables, which are candidate solutions. The action of GA could be represented in the solution (or rather—search) space as a random trajectory.

The present paper is an attempt to introduce an enlarged investigation method to the theory of genetic (evolutionary) algorithms. We aim at the development of some tools suitable for characterization of evolutionary algorithms based on the notions of the symbolic dynamics as well as on compression rates.

II. CLASSIFICATION OF ALGORITHMS AND ITS TOOLS

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 [3], [8]) which state 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. Unfortunately, in many cases the determination of the entropy is very difficult or even impossible. Hence one can try to use as classification

tools the compression rate and the fractal dimension determined for trajectories of GAs.

III. GENETIC ALGORITHMS

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

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

$$P_i^K = [x_1^i, x_2^i, \dots, x_K^i] \quad (1)$$

remembering that a population is an equivalent class of points from the vector space B^K . The equivalent relation is defined by the class of all possible permutations of the set of K -th numbers $\{1, 2, \dots, K\}$.

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

$$T = \bigcup_{i=1}^M [P_i^K], \quad (2)$$

where M 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 $[P_{i+1}^K]$ at the generation $i+1$ after the population $[P_i^K]$ at the generation i , is the conditional probability

$$P(P_{i+1}^K | P_i^K, f(P_i^K), p_m, p_c, p_s). \quad (3)$$

Here by $f(P_i^K)$ we understand the vector-valued function $[f(x_1^i), f(x_2^i), \dots, f(x_K^i)]$. The initial population $[P_1^K]$ 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 $[P_1^K]$. Next populations following that one, are the 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 (3) 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 (2), 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.

IV. ENTROPY OF GA

The domain of research of the ergodic theory is a space with measure and mappings which preserve it. 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 (3) characterizing the mapping T_i from one population to another, we can define the entropy of the mapping

$$H(T_i) = - \sum_{j=1}^M P(P_{i+1,j}^K | P_i^K, f(P_i^K), p_m, p_c, p_s) \log P(P_{i+1,j}^K | P_i^K, f(P_i^K), p_m, p_c, p_s) \quad (4)$$

where $[P_{i+1,j}^K]$ is a possible population from B , $j = 1, 2, \dots, 2^{NK}, \dots, M$.

According to our previous proposition the initial population is generated by the use of a uniform probability, and attains 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 K copies (exactly speaking, according to (1) – 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 is the function of the probability of mutation and selection; it 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.

V. FRACTAL DIMENSIONS

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 [8] and can characterize non-deterministic features of GA. In [7] general statistical and topological methods of analysis of GAs have been introduced. Moreover one can use Hausdorff's dimension or its approximation as an invariant of equivalence of algorithms.

To be more evident, let us recall the notion of the s -dimensional Hausdorff measure ([4]) of the subset $E \subset \mathbf{R}^l$, where $s \geq 0$. If $E \subset \bigcup_i U_i$ and 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

$$H_\epsilon^s(E) = \inf \sum_{i=1}^{\infty} [\epsilon(U_i)]^s \quad (5)$$

where the infimum is over all ϵ -covers $\{U_i\}$ of E . The limit of H_ϵ^s as $\epsilon \rightarrow 0$ denoted by $H^s(E)$, is the s -dimensional Hausdorff measure of E .

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

It is evident that $H_\epsilon^s(E)$ 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 $H^s(E)$ decreases as s increases, and for large s this value becomes 0. Then the Hausdorff dimension of E is defined by

$$\dim(E) = \inf\{s : H^s(E) = 0\}, \quad (6)$$

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

The Hausdorff dimension is one of several fractal dimensions. To make the definitions more evident let us notice that for the Hausdorff dimension of the Peano curve has dimension 2 and of the Cantor middle set is $\log 2 / \log 3$, while its topological dimension D_T is zero. In most cases Hausdorff dimension \geq 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).

As some approximation of the fractal dimension one may consider another dimension known as the packing dimension or the **box-counting dimension** [4]. To calculate this dimension for a set S 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:

$$\dim_{box}(S) = \lim_{\epsilon \rightarrow 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)} \quad (7)$$

It is possible to define box-counting dimensions using balls, with either the covering number or the packing number. It follows that box dimension of E is always $\geq \dim(E)$.

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 [5].

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

We say that two genetic algorithms are **equivalent** if they realize trajectories with the same fractal dimension.

VI. 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 [2] 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 performed numerous experiments on compressing particular generations with Lempel-Ziv algorithm of various bit resolution. We measured number of prefixes resulting from compression process and corresponding compression ratio in scenarios of two types. The first one considered single generations, and for each trajectory we 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.

VII. EXPERIMENTS AND CONCLUSIONS

The first experiments with attractors generated by GAs and the expression (8) have been performed by our co-worker in [5]. His results allow us to claim that the present approach can be useful in the GA's dynamics research.

In the recent paper [14] we have used another approach to the approximation. Let $N(T, \epsilon)$ be the minimum number of K -dimensional cubes with the edge length equal to ϵ , that covers the trajectory $T \subset X$, 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 T

$$D_c(T) \approx \frac{\log_2 N(T, 2^{-(k+1)}) - \log_2 N(T, 2^{-k})}{\log_2 2^{k+1} - \log_2 2^k} = \log_2 \frac{N(T, 2^{-(k+1)})}{N(T, 2^{-k})}. \quad (8)$$

The approximated expression (8) of the box dimension counts the increase in the number of cubes when the length of their edges is diminished by half.

In [14] we have included new calculation results. 12 benchmark functions were used (cf. [6]) in the analysis. Experiments were performed for different dimension: 10, 15, 20 bits with operator parameters and Popsiz. Then the box counting dimension was used to calculate the trajectory dimension.

As far as the analytical approach and the formal definitions of dimensions (5) and (6) are concerned their computer implementation needs additional investigations. Computer accuracy is finite, hence all limits with ϵ tending to zero will give unrealistic results. For example, if 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 ϵ 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 made in [14] was the verification and confrontation of our theoretical considerations and conjectures with real genetic algorithms. The analysis of the experimental result has shown that 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.

ACKNOWLEDGEMENT

The research work on the paper done by W.K., S.K., Z.M. and P. S. was supported by the KBN Project No. 3 T11 C007 28 in 2005-2007. The authors are grateful to Dr. J. Socafa for valuable discussions.

REFERENCES

- [1] 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. Devaney and L. Keen (eds.) American Mathematical Society, Providence, Rhode Island, pp. 127–144, 1989.
- [2] 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.
- [3] Friedman N. A., Ornstein D. S.: On isomorphisms of weak Bernoulli transformations, *Adv. in Math.*, **5**, 365–394, 1970.
- [4] Harrison J.: An introduction to fractals, in *Chaos and Fractals. The Mathematics Behind the Computer Graphics, Proc. Symp. Appl. Math.*, vol. 39, R. L. Devaney and L. Keen (eds.) American Mathematical Society, Providence, Rhode Island, pp. 107–126, 1989.
- [5] 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*, IPI PAN, Warszawa, pp. 40–45, 2000.
- [6] Michalewicz Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd, rev. edition, Springer, Berlin, Heidelberg, 1996.
- [7] Ossowski A.: Statistical and topological dynamics of evolutionary algorithms, in *Proc. of Workshop Intelligent Information Systems IX held in Bystra, Poland, June 12–16*, IPI PAN, Warszawa, pp. 94–103, 2000.
- [8] Ornstein D. S.: *Ergodic theory, Randomness and Dynamical Systems*, Yale Univ. Press, 1974.
- [9] Vose M. D.: Modelling Simple Genetic Algorithms, *Evolutionary Computation*, **3** (4), 453–472, 1996.
- [10] 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>
- [11] 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.
- [12] 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>
- [13] Szlenk W., *An Introduction to the Theory of Smooth Dynamical Systems.*, PWN, Warszawa, John Wiley & Sons, Chichester, 1984 G. H.
- [14] Kotowski S., Kosiński W., Michalewicz Z., Nowicki J., Przepiórkiewicz B., Fractal dimension of trajectory as invariant of genetic algorithms *ICAICS, 9-th International Conference on Artificial Intelligence and Soft Computing, 2008*, LNAI 5097, Springer, Berlin, Heidelberg, New York, pp. 414–425.