ISSN 1995-0802, Lobachevskii Journal of Mathematics, 2019, Vol. 40, No. 11, pp. 1799–1808. © Pleiades Publishing, Ltd., 2019.

Conformance Evaluation of Genetic Algorithm for Evolutionary Area Search of Canonical Model

V. K. Ivanov[1*], B. V. Palyukh[1], and A. N. Sotnikov[2***]**

(Submitted by A. M. Elizarov)

[1] Tver State Technical University, Tver, 170026 Russia

[2] Joint Supercomputer Centre, Russian Academy of Sciences, Moscow, 119991 Russia

Received May 25, 2019; revised June 14, 2019; accepted July 15, 2019

Abstract—The theory and practice of genetic algorithms is largely based on the Schema Theorem. It was formulated for a canonical genetic algorithm and proves its ability to generate a sufficient number of effective schemata of individuals. Genetic algorithms to solve specific problems and to be different from canonical ones have to be checked to find out whether the Schema Theorem evaluates the algorithm fitness. The article validates the way of testing the algorithm developed as a technique of an area search. The methodology and research results are stated consistently. Coding specifics of the search queries are noted, a criterion of the coding method applicability is substantiated. A variant of the genotype geometric coding is proposed. In comparison with other methods of binary search coding, it provides a short code length and uniqueness as well as conforms the formulated criterion of applicability. Supporting experimental results are given. The Schema Theorem is shown to hold with the iterative execution of the genetic algorithm being tested.

DOI: 10.1134/S1995080219110155

Keywords and phrases: *genetic algorithm, genotype, query, coding, crossover, defining length, order, scheme, subject search, Holland's schema theorem, fitness function.*

1. INTRODUCTION

Genetic algorithms (GA) are best understood with the fundamental Holland's Schema Theorem. It was formulated for a canonical GA and proves its power to generate a sufficient number of effective schemata of individuals to achieve the vicinity of the fitness function optimum in a finite number of generations.

Hence it follows that any GA modifications created to solve a specific real-world problem and different from a canonical algorithm should be checked for the Schema Theorem evaluating the GA modification convergence.

This article substantiates the verification approach to the GA developed as a technique of an area search. The methodology and results of the conducted research are described consistently. This way, input data for research are described, coding specifics of search queries are noted, a criterion of the encoding method applicability is substantiated. We suggest a variant of geometric coding of individuals' genotype which, in comparison with other methods of binary coding of search queries, provides a short length and uniqueness of the code, and also meets the formulated criterion of applicability. This is confirmed by the results of experiments provided in the article. Further, we show that the iterative execution of the GA being checked with the individuals' schemata obtained as our experimental results proves the Schema Theorem to hold. In conclusion, the research results are discussed.

It should be noted that the article continues the authors' earlier studies on the subject [1].

[*] E-mail: mtivk@tstu.tver.ru

[**] E-mail: pboris@tstu.tver.ru

[***] E-mail: asotnikov@jscc.ru

2. SOURCE DATA AND PROBLEM DEFINITION

2.1. The Schema Theorem

Holland's Schema Theorem [2] used in GA theory says that individuals' short, low-order schemata with above-average fitness function increase exponentially in frequency in successive generations. It is expressed as the following equation:

$$\mathbb{N}(h,t+1) \geq \mathbb{N}(h,t)\frac{f(h,t)}{f(t)}\left[1-\frac{\delta(h)}{l-1}p_c - o(h)p_m\right], \tag{1}$$

where $\mathbb{N}(h,t)$ is the number of individuals belonging to schema h at generation t; $\mathbb{N}(h,t+1)$ is the same at the next generation; $f(h,t)$ is the observed average fitness of schema h at generation t; $f(t)$ is the observed average fitness of the whole population at the same generation; $\delta(h)$ is the defining length of a schema; l is the length of an individual; $o(h)$ is the order of a schema; p_c is the probability of crossover; p_m is the probability of mutation.

Despite a number of doubts and limitations, the Schema Theorem is one of the most common tools of analyzing GA. It should be noted that the requirements of small values of $\delta(h)$ and $o(h)$ follow from the specifics of crossover and mutation relating to genetic operations (GO). In particular, a canonical GA uses a binary coding of an individuals' genotype.

In general, the Schema Theorem makes it possible to identify and use GA modifications increasing its adaptability. The obvious usefulness of checking the applied algorithms differing from canonical ones for the observance of the Schema Theorem is justified by the necessity of substantiating analytically the convergence of GA with a fixed set of parameters. As [3] states the Schema Theorem "provides some tools to check whether a given representation is well suited to a GA". It is also possible to predict *a*) the structure of effective schemata reproduced by a GA; *b*) average local improvements of fitness functions; *c*) the GO influence on the values of a given fitness function. Examples of research on this subject are given in Section 3 below.

2.2. Source Data

The basic concepts of the approach used in the development of the GA proposed by the authors (hereinafter referred to as the GAP) are described in [1]. Thus, the initial interpretations are as follows: *a query is an individual (chromosome), an encoded set of query terms* represents *a genotype*, the replacement of a query term with another term is defined as *crossover*, and the replacement of a query term with its synonym is *mutation*. The procedure of a fitness function calculation consists in executing a query by a search engine and using parameters of the set of relevant documents found—*a phenotype*.

The GAP search pattern K for documents is a set of terms related to a certain subject area. Each search query is represented by a vector $\overline{q} = (c_1, c_2, \ldots, c_n, \ldots, c_m)$, where $c_n = \{k_n, w_n, S_n\}$, $k_n \in K$ is a term, w_n is a term weight, S_n is a set of term synonyms k_n, m is the number of terms in a query. The result of the query is a set of documents R. The initial population of N queries is a set Q_0, where $|Q_0| = N$, $N < |K|/2$, $q \in Q_0$.

The fitness function for the query population is calculated as follows:

$$\overline{W}(Q) = \frac{1}{N}\sum_{j=1}^{N}\overline{w}(q_j), \tag{2}$$

where $Q = (q_1, q_2, \ldots q_N)$ is the population of N queries; $\overline{w}(q_j)$ is the fitness function of j-th query

$$\overline{w}(q_i) = \frac{1}{R}\sum_{i=1}^{R} w_i(g,f,s), \tag{3}$$

where w_i is the fitness function of i-th results of j-th query. Here, w_i depends on position g in the search engine results list, frequency f of this search result in all N query result lists, similarity measure s of the short result text and search pattern K.

In order to verify the Schema Theorem suitability, the binary coding of the genotype should provide the individuals' short schemata $\delta(h)$ and $o(h)$. This means the need for a minimum code length.

2.3. Coding Specifics of Search Queries

It should be noted that when the GAP is running and the GO are performed bit operations are not used (see GO interpretations above), and elements c_n of vector $\overline{q}$ are localized by the decimal index $n \in [1, |K|]$. Numeric identifiers n are integer nominal variables and define an unordered set of objects.

The binary representation of vector $\overline{q}$ for checking the Schema Theorem in accordance with expression (1) is made possible based on the following remarks expressed in [4]:

Remark 1. Small changes of the genotype (e.g., the inversion of a single bit) must lead to small changes of function $\overline{w}(q_j)$. In other words, fitness function $\overline{w}(q_j)$ should not have such a genotype property when a small change of the phenotype requires an enormous change of the genotype (Hamming cliffs).

Remark 2. Individuals are similar not because their genotypes look similar, but because they are located close to each other in the solution space after performing the selected GO. In other words, similar individuals (similar genotypes) have similar behavior (similar phenotypes), i.e. they have nearest fitness functions. In general, genotype $\overline{q}_A$ is similar to genotype $\overline{q}_B$, if there is a high probability of conversion $\overline{q}_A$ in $\overline{q}_B$ (and vice versa) after performing GO.

For example, suppose there are binary codes of individuals (genotypes) '010110110101' and '110110110001' used in the GAP. They represent two queries of seven terms each. The term sets differ by only one element (term) which makes it possible to assume a high similarity of search results (phenotype). However, the numerical values of the individuals' codes differ significantly (four times).

2.4. Criterion of Encoding Method Applicability

The above said allows us to introduce a criterion of the encoding method applicability based on the concept of uniform continuity of query fitness function $\overline{w}(q_j)$, where q_j is a binary number representing the query (individual) in the given encoding.

Definition 1. *Function $\overline{w}(q_j)$ is called uniformly continuous on the set Q, if $\forall \epsilon > 0\ \exists \lambda > 0$, such that $\forall q', q'' \in Q$ satisfying the condition $|q'' - q'| < \lambda$, the inequality $|\overline{w}(q'') - \overline{w}(q')| < \epsilon$ is valid.*

The concept of continuity means that small changes of argument q_j lead to small changes of function $\overline{w}(q_j)$. This statement is consistent with *Remark 1*.

Also, the property of uniform continuity means that the value of λ limiting the deviation of argument q_j only depends on the value ϵ of the deviation of function $\overline{w}(q_j)$ and does not depend on the value of argument q_j, i.e. it is constant on the whole domain of the function. This statement is consistent with *Remark 2*.

Thus, a coding method is considered to be valid if the fitness function is consistent with *Definition 1*.

2.5. Problem Description

Thus, we define the following problem:

1. To propose a coding method for a vector $\overline{q}$ with a binary sequence having the following properties:

- A short code length to ensure that the conditions for $\delta(h)$ and $o(h)$ are met.
- The uniqueness of the code for individuals.
- The adherence to criterion of the encoding method applicability as stated in *Definition 1*.

2. To show that with iterative GAP execution with the individuals' schemata obtained in the experimental results, in expression (1) is valid.

It should be noted that we do not prove the validity of the Schema Theorem. We are trying to show that the GAP has properties of a canonical GA, which convergence is grounded analytically [2, 17].

3. RELATED WORKS

Questions concerning the application of the Schema Theorem and related concepts in the GA research and development remain relevant. The analysis of nature of the 'good' schemata gives few ideas on the efficiency of genetic algorithm [3]. Based on the materials from the ACM Digital Library (https://ieeexplore.ieee.org) and IEEE Xplore Library (https://ieeexplore.ieee.org) databases, the number of articles on this subject has remained fairly stable over the past 25 years. At the same time, the number of publications reached the peak within 2006–2013. The main reason for such interest is that GA applications are often based on non-canonical models requiring verification and mathematical substantiation of their applicability, in particular, convergence. For example, [5] describes the adaption of the theorem for various other crossover and mutation operators focusing on the application of GA to a music segmentation problem.

Analyzing the publications of recent years we highlight the following aspects with relevant examples:

1. *The mathematical substantiation of GA modifications.* To confirm the applicability of GA instead of simulation methods, the [6] proposes a mathematical rationale of the tournament-roulette selection as an analogue of the Schema Theorem. The [7] tests a relationship between crossover probability, mutation probability, and selection pressure using two problems. This relationship is based on the Schema Theorem. The [8] demonstrates theoretically and experimentally that GAs may be considered as an impracticable technique for variable selection in multivariate calibration problems, especially due to building block disruptions.

2. *New ways of coding a genotype.* The [9] considers the possibility of guided randomly deterministic search methods to solve NP-complete combinatorial problems. The Schema Theorem and parameter coding to solve optimization problems in which the objective function can take negative values is applied. The [10] suggests a new GA to solve the job shop scheduling (JSS) problem which uses a new coding for scheduling of jobs and machine distribution. The paper [11] proposes a new phenotype-to-genotype encoding representation technique that makes it possible to eliminate the problem of illegal gene values, can increase the variability of offspring, and increases the GA performance. Cloning creates new individuals that have gene values identical to the hypotheses from which they were cloned, but with dramatically different gene representations. Also, they can be very different from their parents. The [12] introduces a class of grammar that can represent a hierarchical schema structure in a problem space. Unlike conventional sequence-based grammars this grammar represents set-membership relationships, not strings, and is, therefore, insensitive to gene-ordering and physical linkage. This provides a robust method of building-block recombination that is linkage-invariant and not restricted to low-order schemata.

3. *Patterns used to solve special problems.* The [13] presents GA with the penalty function for the FMS scheduling problem. The proposed algorithm and operators effectively makes use of the Schema Theorem and the building block hypothesis. In [14], a heuristic GA is proposed to solve the 0-1 knapsack problem. In each generation, populations are divided into two sections: a superior clan and an inferior clan, and an excellent schema in a superior clan are picked up to replace the chromosomes in an inferior clan. Simulations show that the proposed method can obtain the best solution and convergence faster than conventional GA and Greedy algorithms.

4. *Concepts of the Schema Theorem used in other algorithms.* The [15] presents a universal method of using patterns in ant algorithms solving the traveling salesman problem as an example. This has reduced the time to search for quasi-optimal solutions and improve their quality. The [16] states the weak theoretical foundations of GP building blocks cause their role in GP evolutionary dynamics to remain still somewhat of a mystery. The paper presents a problem-independent methodology for identifying GP building blocks based on their respective sample counts in the population.

We hope that our work contributes to the mathematical rationale of the GA to solve a specialized problem of information retrieval.

4. GENOTYPE CODING

4.1. Options of Binary Coding

Let us consider several possible options of a genotype's binary coding.

4.1.1. Position coding (PC). An individual's binary code $\overline{q}_{01}$ is a bit string with a locus of nonzero value $n \in [1, |K|]$. Here, an individual's code length $L_q = |K|$, a gene's code length $L_n = 1$, the interval of coding genes' values $GInt = (0; |K| - 1)$. PC example: $|K| = 12$, $\overline{q} = (4, 7, 10)$, $m = 3$, a binary code $\overline{q}_{01} = 000100100100$, $L_q = 12$.

4.1.2. Number coding (NC). An individual's binary code $\overline{q}_0 1 = n_1^{01} \cdot n_2^{01} \cdot \ldots \cdot n_i^{01} \cdot n_m^{01}$, where n_i^{01} is a binary code of a decimal value n with length L_n. Here $L_q = m * L_n$, $L_n \geq \log_2(|K| - 1)$, $GInt = (0; 2^{L_n} - 1)$. NC example: $|K| = 12$, $\overline{q} = (4, 7, 10)$, $m = 3$, $\overline{q}_{01} = 010001111010$, $L_q = 12$. In the general case, when $|K| = 12$ and $m = 3$, an individual's length is equal to $L_q = m * \log_2(|K| - 1) = 3 \times 3.70044 = 11.10132 \approx 12$ bits.

4.1.3. Logarithmic coding (LC). An individual's binary code $\overline{q}_{01} = n_1^{01} \cdot n_2^{01} \cdot \ldots \cdot n_i^{01} \cdot n_m^{01}$, where $n_i^{01} = \alpha\beta bin$ is a binary code of a decimal value n (a gene) with length L_n, with the following correlation holding [17]:

$$[\alpha\beta bin]_{10} = (-1)^{\beta} e^{(-1)^{\alpha}[bin]_{10}}, \tag{4}$$

where $[\alpha\beta bin]_{10}$ is a decimal value of binary code $\alpha\beta bin$, α is the first code bit, β is the second code bit, bin are the rest code bits, $[bin]_{10}$ is a decimal value of binary number bin. In addition, $L_q = m * L_n$, $L_n = \lceil \log_2(|K| - 1) \rceil$, $GInt = (-e^{L_n}; e^{L_n})$. LC example: $|K| = 12$, $\overline{q} = (4, 7, 10)$, $m = 3$, $\overline{q}_{01} = 001101101001$, $L_q = 12$.

4.1.4. Geometric coding (GC). To code individuals, we suggest using distance $Dist(\overline{q}_i, \overline{q}_0)$ between vector $\overline{q}_i$ and initial vector $\overline{q}_0$ (the first individual of the initial population). Distance $Dist(\overline{q}_i, \overline{q}_0)$ can generally be calculated with any metric such as Euclidean distance, cosine measure, Manhattan distance, Chebyshev distance, etc. In the case of a cosine measure, which does not require valuation per interval (0; 1), we have

$$Dist(\overline{q}_i, \overline{q}_0) = \frac{\overline{q}_i * \overline{q}_0}{||\overline{q}_i|| \cdot ||\overline{q}_0||}. \tag{5}$$

An individual's code $\overline{q}_{01}$ is a binary code of a decimal value $q_{10} = Dist(\overline{q}_i, \overline{q}_0) \times 10^p$, where p is calculation precision (the number of digits after the point). Also, $L_q = \lceil \log_2(10^p - 1) \rceil$ does not depend on m, L_n is not used, $GInt[0; |K| - 1]$. GC example: $|K| = 12$, $\overline{q}_0 = (3, 8, 11)$, $\overline{q}_i = (4, 7, 10)$, $p = 3$, $\overline{q}_{01} = 1111100011$, $L_q = 10$.

4.2. Evaluation of Coding Options

The initial positions in the evaluation of coding options follow from expression (1) and consist in the fact that the promising properties of the phenotype must be represented in the genotype in the form of bit sections being as short as possible, with the uniqueness of individual codes having to be ensured. The criterion of uniform continuity of the fitness function should also be provided in accordance with *Definition 1*.

4.2.1. Individuals' code length. Individuals' GC is the best option in terms of code length. Fig. 1 shows the dependence of the code length L_q on $|K|$ for various values of m. Note that with GC L_q does not depend on m and $|K|$, but is determined by the calculation precision which, in turn, ensures the uniqueness of the codes. For example, $L_q = 17$ for $p = 5$.

4.2.2. Individuals' code uniqueness. There are following remarks:

1. The uniqueness of individuals' codes is obvious for PC and NC.

2. LC is used for large numbers or in case of coding non-integer values. The disadvantage of this method is that when coding numbers from an interval significantly smaller than interval $GInt = (-e^{L_n}; e^{L_n})$, we get a large number of identical sequences. This is our case: even with typical real values of $L_q = 50$ and $L_n = 7$, we have interval $GInt = (-1096; 1096)$.

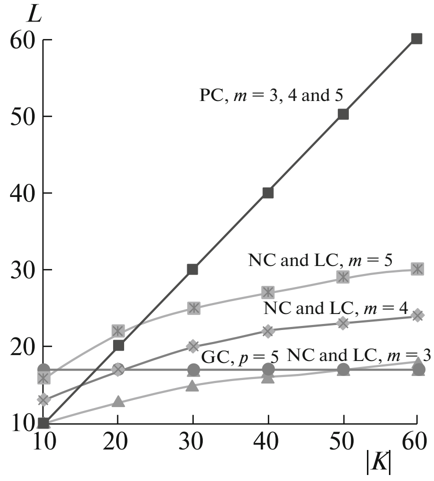

Fig. 1. The code length of various coding procedures.

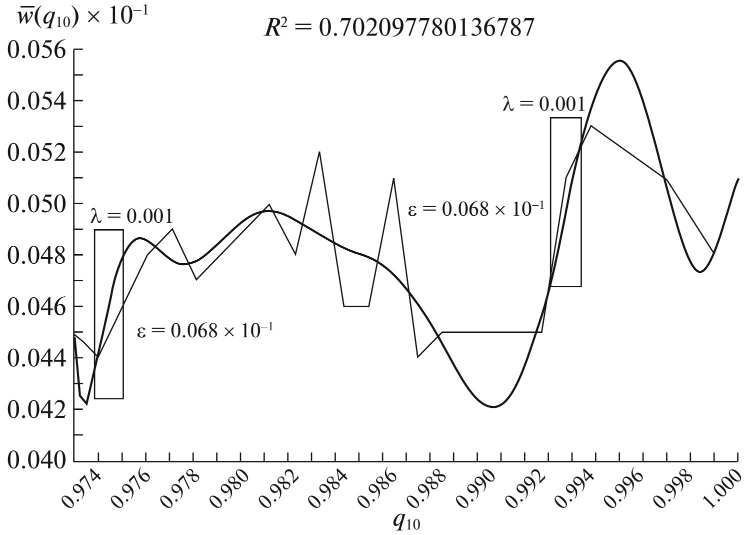

Fig. 2. Dependence of the fitness function on decimal values of individuals' binary code in genotype GC.

3. The uniqueness of individuals' codes in GC is ensured by the calculation precision. Table 1 presents the analysis results of the number of duplicate codes in the GAP experiments. The following input data were used $|K| = 50$, $p = 3$, $p = 3, 4$ and 5. It can be seen that at $p = 5$ we achieved 100% uniqueness of the generated individuals' codes.

4.2.3. Criterion of fitness function uniform continuity. As shown above, according to the criteria of individuals' code length and uniqueness the most effective option is GC. Now the GC procedure should be checked for fitness for the criterion of fitness function uniform continuity.

As an example, let us consider the dependence of the fitness function $\overline{w}(q)$ on decimal values $q_{10} \in [0; 1]$ of the individuals' binary code in GC. The graph is presented in Fig. 2. Note that scale factor 10^{-1} is used for $\overline{w}(q)$.

The input data for constructing this graph were obtained after the GAP experiments with the following parameters: the search engine Bing, the coding procedure of vector $\overline{q}_{01}$ is geometric coding (cosine measure), $|K| = 50$, $m = 8$, $p = 5$, $L_q = 17$, the crossover probability $P_c = 1$, the mutation probability $P_m = 0.01$, the number of GAP iterations is 20.

An example of the data obtained in the GAP experiments is presented in Table 2. The evolution of

Table 1. The number of unique codes in individuals' geometric coding (%)

Precision p	Vector coding procedure $\overline{q}_{01}$		
	euclidean distance	vector length	cosine measure
3	89.4	87.6	85.8
4	96.5	98.2	98.2
5	96.5	98.2	100.0

Table 2. Data obtained in GAP experiments

Iteration no.	Individual ID	q	q_{01} (GC)	q_{10}	$w(q_{10})$
1	128705	5, 7, 24, 29, 31, 36, 39, 44	10100101101111011	0.973019	0.474621724
2	128714	5, 7, 14, 29, 31, 36, 39, 44	10110001011001111	0.955444	0.481147112
3	128718	5, 7, 24, 29, 31, 36, 43, 44	10110001011001111	0.943910	0.481147112
4	128719	5, 7, 29, 31, 36, 39, 43, 44	10011110000101011	0.955444	0.482562009
...	...	...	...	...	...
1	128707	1, 13, 17, 19, 22, 32, 34, 40	10101011100010010	0.963923	0.481147112
2	128711	1, 13, 17, 19, 32, 34, 40, 43	10101110110010100	0.964174	0.481147112
3	128716	1, 13, 17, 19, 22, 32, 34, 40	10101011100010010	0.963923	0.481147112
4	128717	1, 13, 17, 19, 24, 32, 34, 40	10110001011001111	0.964174	0.482562009
...	...	...	...	...	...
1	128589	1, 2, 5, 17, 21, 26, 41, 42	11000011010100000	0.973145	0.455153172
2	128621	1, 2, 5, 17, 21, 26, 41, 49	11000001011011011	0.964174	0.482562009
3	128624	1, 2, 5, 17, 21, 26, 37, 41	11000010101100101	0.963923	0.480935913
4	128625	1, 2, 5, 7, 17, 21, 26, 41	10110011111100000	0.964174	0.480935913
...	...	...	...	...	...

three individuals (queries) after four GAP iterations and the change of the query vector $\overline{q}$ and its binary representation $\overline{q}_{01}$ is shown.

The Fig. 2 also shows the trend line and its polynomial approximation function with the coefficient of determination $R^2 \approx 0.702098$:

$$\begin{aligned} \overline{w}(q) &= 2.36404 \times 10^{-14}q^{12} - 4.50231 \times 10^{-12}q^{11} + 3.76263 \times 10^{-10}q^{10} \\ &- 1.81669 \times 10^{-8}q^{9} + 5.61547 \times 10^{-7}q^{8} - 1.16363 \times 10^{-5}q^{7} \\ &+ 0.00164527q^{6} - 0.00158434q^{5} + 0.0101851q^{4} \\ &- 0,041906q^{3} + 0.102174q^{2} - 0.126967q + 0.102964. \end{aligned} \tag{6}$$

In order to show that function $\overline{w}(q)$ satisfies the criterion of *Definition 1*, we must select such $\epsilon > 0$ and $\lambda > 0$ on some interval of q_{10} so that, when $|q_{10} - q'_{10}| < \lambda$, the inequality $|\overline{w}(q'') - \overline{w}(q')| < \epsilon$ is valid.

Obviously, the coefficients of the function polynomial $\overline{w}(q)$ are different for different search patterns K, vector components $\overline{q}$. Therefore, now it is not possible to find a general solution for the problem of ϵ and λ selection. Consider a numerical solution for segment $\overline{q}_{10}[0.973; 1]$ as shown in the graph of Fig. 2. A simple solution is obtained if we take $\epsilon = \max|\overline{w}(q'') - \overline{w}(q')|$ and $\lambda = \min|q'' - q'|$ (in the GAP experiments). Then $\epsilon = 0.068$, $\lambda = 0.001$.

A geometric interpretation of this result in the form of two rectangles is shown in Fig. 2. If the function $\overline{w}(q_j)$ is uniformly continuous on set Q, a rectangle with $\lambda(\epsilon)$ and ϵ sides, parallel axes q_{10} and $\overline{w}(q)$ can be moved along the graph of function $\overline{w}(q_j)$ keeping the sides parallel to the axes so that the graph does not intersect the rectangle sides parallel to axis q_{10} and only intersects the sides parallel to axis $\overline{w}(q)$.

5. VERIFICATION OF THE SCHEMA THEOREM

5.1. Initial Schemata

We should note some features of *the schema* notion and inequality (1) which were taken into account in generating the initial schemata:

- An individual in a population is a binary string $q \in 0, 1$ with a fixed length l.
- A schema is a ternary string $h \in \{0, 1, *\}$ with a fixed length l where $*$ has the meaning of any of the symbols $0, 1$.
- A schema h determines the set of individuals $q \in h$ if $\forall i \in \{0 \dots l\}, q_i = h_i \vee h_i = *$.
- A schema order $o(h) = ||\{\forall i \in \{1 \dots l\}, h_i \neq *\}||$ is the number of positions in the schema not equal to $*$.
- The schema defining length $\delta(h) = \max|i - j|, \forall i, j \in \{0 \dots l\}, h_i \neq * \wedge h_j \neq *)$ the maximum distance between positions not equal to $*$.
- The maximum number of possible schemata is 2^l.
- The maximum number of individuals for a schema is 2^a, where a is the number of symbols $*$ in a schema.

To generate the schemata in our experiments we used the results of the GAP work with search engine Bing. 117 experiments were performed with $|K| = 50$, $m = 8$ and various combinations of l, p and coding procedures for $\overline{q}_{01}$. We used individuals' geometric coding with calculation precision $p = 3, 4$ and 5. The metrics for calculating the vector code $\overline{q}_{01}$ were taken as follows: the Euclidean distance (Euc), the length of the vector (Len), and the cosine measure (Cos). The results of performing GO for pairs of consecutive populations were recorded.

With the help of a specially developed algorithm 3342 schemata with the length of binary strings of individuals $l = 10, 14$ and 17 were generated. It should be noted that the number of schemata obtained is much less than the possible 2^l. We generated the actual schemata, i.e. those corresponding to population individuals being the result of the GAP work. For example, if there were no codes with bits 2 and 8 equal to 1 among the 10-bit codes of individuals, then schemata of $' * 1 * * * * * 1 * *'$ form were not generated.

5.2. Suitability of the Schema Theorem for GAP

For each pair of consecutive populations and for each of the generated schemata included in this pair of populations, the right side of inequality (1) was calculated. The following initial data were taken: the probability of crossover $P_c = 1$, the probability of mutation $P_m = 0$. Order $o(h)$ and defining length $\delta(h)$ of the schemata were calculated for each scheme individually. The fitness function $f(h, t)$ of each scheme h at generation t and the average fitness function of the entire population at the same generation $f(t)$ were calculated by equations (3) and (2) respectively.

The summary results of experiments and calculations are presented in diagrams in Fig. 3. The following values to be compared are shown:

- $\overline{\mathbb{N}}(h, t)$ is the average number of individuals belonging to schema h at generation t when the GAP is performed;
- $\overline{\mathbb{N}}'(h, t)$ is the average value of the right side of inequality (1);
- $\overline{\mathbb{N}}(h, t + 1)$ is the average number of individuals belonging to schema h at the next generation when the GAP is performed.

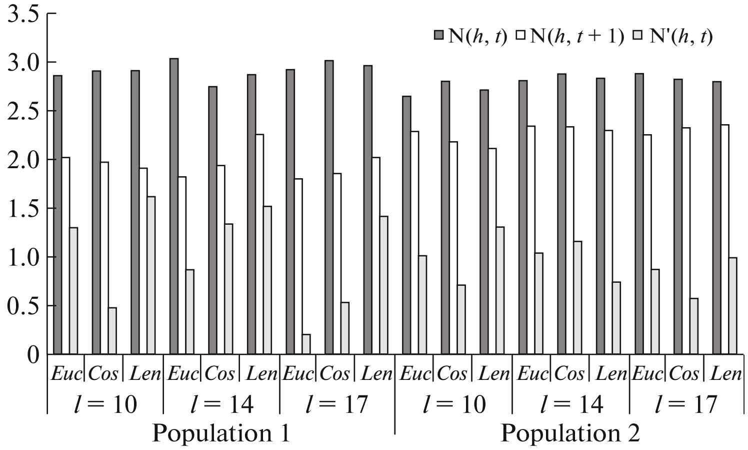

Fig. 3. The results of inequality (1) validation.

6. DISCUSSION OF RESULTS

1. Based on the analysis of the publications related to the subject of the article (see Sections 2 and 3) the Schema Theorem does not lose its relevance. It still serves as a mathematical rationale for GA modifications including new ways of the genotype encoding. The concept of patterns is used to solve specialized problems including algorithms other than GA.

2. The requirements for small values of order $o(h)$ and defining length $\delta(h)$ of schemata are critical for inequality (1) to be valid. The proposed coding of the genotype (geometric coding) allows this requirement to be ensured. For generated schemata of length $l = 10$, average value $\overline{o}_{10}(h) = 4.76$, average value $\overline{\delta}_{10}(h) = 6.35$. If $l = 14$, $\overline{o}_{14}(h) = 5.80$ and $\overline{\delta}_{14}(h) = 9.29$, and if $l = 17$, $\overline{o}_{17}(h) = 6.89$ and $\overline{\delta}_{17}(h) = 12.72$. It can be seen that $\overline{\delta}(h) < l - 1 \Rightarrow \overline{\delta}(h)/(l - 1) < 1$.

3. Another requirement for the coding method—the uniqueness of the individual's code—is also provided by geometric coding. Although the use of specified metrics to calculate the code of the vector $\overline{q}_{01}$ does not generally give the complete uniqueness of the individuals' codes, the uniqueness of individuals reaches 100% for the current set of individuals and the adopted precision of calculations.

4. A criterion of the coding method applicability has been formulated in terms of the property of uniform continuity of fitness function $\overline{w}(q)$. In order to show that function $\overline{w}(q)$ satisfies this criterion, a particular numerical solution has been found (see Section 4.2.3 above). It is not yet possible to find a common solution. This is due to the calculation specifics of function (3). The values of arguments $\overline{w}(q)$ depend on some external factors (a search engine environment) and search patterns used. They are poorly formalizable and, therefore, for the time being, we are forced to apply the experimental data approximation.

5. The graph of Fig. 3 shows that for all initial data options inequality $\mathbb{N}(h, t + 1) > \mathbb{N}'(h, t)$ is valid which corresponds to our assumption that inequality (1) holds. Moreover, Fig. 3 also shows that inequality (1) is satisfied on any pair of successive populations generated during the GAP. Since the procedure of the Schema Theorem checking has been tested, it may be necessary to repeat the calculations for a more significant number of the GAP iterations.

7. CONCLUSION

The approaches and results of applying the developed methodology for conducting experiments outlined in the article allow us to conclude that the goal of this study has been achieved. The mathematical substantiation of the GAP application in information retrieval problems has been obtained. An appropriate method for encoding a genotype has been proposed and its effectiveness has been tested. The validity of the Schema Theorem for the GAP is shown.

Thus, there is a great potential to control the structure of effective search queries.

ACKNOWLEDGMENTS

The authors express their thanks the colleagues who have been taking part in this study, namely, P. I. Meskin and N. V. Vinogradova as well as to translator of the article O. A. Gumenyuk.

FUNDING

This work was done at the Tver State Technical University with supporting of the Russian Foundation of Basic Research (project no. 18-07-00358) and at the Joint Supercomputer Center of the Russian Academy of Sciences—Branch of NIISI RAS within the framework of the State assignment (research topic 065-2019-0014).

REFERENCES

1. V. K. Ivanov, B. V. Palyukh, and A. N. Sotnikov, "Efficiency of genetic algorithm for subject search queries," Lobachevskii J. Math. **12** (3), 244–254 (2016).
2. J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (MIT Press, Cambridge, MA, 1992).
3. S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms* (Springer, Berlin, Heidelberg, New York, 2007).
4. S. Luke, *Essentials of Metaheuristics,* 2nd ed. (Lulu, 2013). http://cs.gmu.edu/ sean/book/metaheuristics. Accessed 2018.
5. B. Rafael, M. Affenzeller, and S. Wagner, in *Proceedings of the GECCO'12 Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, 2012,* pp. 469–476.
6. A. B. Glushak, V. A. Lomazov, and D. A. Petrosov, "Analysis of modified genetic algorithm based on the theory of schemes," Nauch. Vedom. BelGU, Mat. Fiz. **27** (248), 121–126 (2016).
7. P. A. Diaz-Gomez and D. F. Hougen, in *Proceedings of the GECCO'09 Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, 2009*, pp. 763–770.
8. L. C. M. de Paula, A. S. Soares, T. W. de Lima, A. R. G. Filho, and C. J. Coelho, in *Proceedings of the GECCO'16 Companion of the 2016 on Genetic and Evolutionary Computation Conference Companion, 2016,* pp. 1031–1034.
9. O. A. Melikhova, "Genetic algorithms application for the artificial intellect systems construction," Izv. SFedU, Inzhen. Nauki **7** (144), 53–58 (2013).
10. S. Noor, M. I. Lali, and M. S. Nawaz, "Solving job shop scheduling problem with genetic algorithm," Sci. Int. **27**, 3367–3371 (2015).
11. S. L. Keast, "A simple representation technique to improve GA performance," Aauburn Univ. Tech. Rep. CSSE03-11 (2003), pp. 1–21. ftp://ftp.eng.auburn.edu/pub/techreports/csse/03/CSSE03-11.pdf. Accessed 2019.
12. C. R. Cox and R. A. Watson, in *Proceedings of the GECCO'14 Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, 2014,* pp. 341–348.
13. L. Sun, X. Cheng, and Y. Liang, "Solving job shop scheduling problem using genetic algorithm with penalty function," Int. J. Intell. Inform. Process. **1** (2), 65–77 (2010).
14. Chen Lin, in *Proceedings of the 4th International Conference on Genetic and Evolutionary Computing, 2010,* pp. 301–304.
15. A. A. Kazharov and V. M. Kureichik, "Template using for ant colony algorithms," Izv. SFedU, Inzhen. Nauki **7** (144), 17–22 (2013).
16. B. Burlacu, M. Kommenda, and M. Affenzeller, in *Proceedings of the Asia-Pacific Conference on Computer Aided System Engineering, 2015,* pp. 152–157.
17. D. Rutkowska, M. Pilinski, and L. Rutkowski, *Neural Networks, Genetic Algorithms and Fuzzy Systems* (PWN, Warsaw, 1997; Hotline-Telekom, Moscow, 2013).