UDC 007.52+004.896:656.052.48:519.876.5 DOI: 10.18698/0236-3933-2021-2-52-65

BINARY DECISION TREE CONSTRUCTION USING THE HYBRID SWARM INTELLIGENCE

B.K. Lebedev lebedev.b.k@gmail.com
O.B. Lebedev lebedev.ob@mail.ru
A.A. Zhiglaty artemiy.zhiglaty@gmail.com

Academy for Engineering and Technologies, Southern Federal University, Taganrog, Russian Federation

Abstract

Solving the problem of a classification model construction is presented in the form of a sequence of considered attributes and values thereof included in the M_k route from the root to the dangling vertex. Decision tree developed interpretation is presented as a pair of chromosomes (S_k, W_k). The S_k chromosome list of genes corresponds to the list of all attributes included in the M_k route in the decision tree. The W_k chromosome gene values correspond to the attribute values included in the M_k route. Unification of data structures, search space and modernization of integrable algorithms was carried out for hybridization. Hybrid algorithm operators are using the integer parameters and synthesize new integer parameter values. Method was developed to account for simultaneous attraction of the α_i particle to three $x_i\,(t)$, $x^*_i\,(t)$, $x^*(t)$ attractors dislocating from the $x_i\,(t)$ position to the $x_i\,(t+1)$ position. Modified hybrid metaheuristic of the search algorithm is proposed for constructing a classification model using recombination of swarm and genetic search algorithms. The first approach uses genetic algorithm initially and then the particle swarm algorithm. The second approach uses the high-level nesting hybridization method based on combination of genetic algorithm and particle swarm algorithm. The proposed approach to constructing a modified paradigm uses chromosomes with integer parameter values in the indicated hybrid algorithm and operators, which assist chromosomes to evolve according to the rules of particle swarm and genetic search

Keywords

Classification, decision tree, particle swarm, genetic evolution, hybridization, integer parameter values, chromosome structures, directed mutation operator

Received 29.05.2020
Accepted 26.06.2020

This work was performed with financial support provided by the Russian Foundation for Basic Research (grant no. 20-07-00260 a)

Introduction. The most common methods for solving classification problems are using the D = (X, U} decision tree as a qualification model, where $X = \{x_i \mid i = 1, 2, \ldots, n\}$ is the set of vertices and $U = \{u_i \mid i = 1, 2, \ldots, m\}$ [1–3]. The X set includes the X_1 set of internal vertices and the X_2 set of end vertices. Inner vertices of the decision tree correspond to features characterizing the object. End vertices correspond to the values of categorical variables (specific class, grade, etc.) [4, 5]. All edges are oriented. In order to classify a new object, it is necessary in the decision tree to build an oriented route from the root to one of the end vertices. The order of vertices in the oriented route determines the order of features consideration. Thus, the edge emerging from the x_i vertex corresponds to the x_i feature value [4].

The purpose of building a decision tree is to determine the value of categorical dependent variable (class). If the target variable takes discrete values, then classification problem is being solved.

Binary decision trees are the most common and simplest case [3, 4]. The decision tree efficiency significantly depends on correct selection of the branching criterion.

Most of the known algorithms (CART, C4.5, NewId, ITrule, CHAID, CN2, etc.) [1–4] are the “greedy” algorithms of sequential type. With this approach, the decision tree is built from top to bottom. At each step of the greedy algorithm, partition of a set of objects is performed according to a feature ensuring maximum difference and distinction between subsets. Sequential algorithms are characterized by lesser labor intensity, but provide the lowest quality.

An effective way to improve the quality of solutions is using the stochastic population algorithms [5], which, as a rule, are iterative and operate in the complete solution area. Swarm and genetic algorithms are widely employed. Studies of the population algorithms efficiency demonstrated that their hybridization is a powerful means in increasing the new algorithm efficiency [6, 7]. Recombination of the population algorithms metaheuristics provides uniform and reasonable scanning of the search space and high efficiency of the integrated algorithms [8].

Solution to the problem of constructing a classification model in this work is the sequence of considered attributes and values thereof included in the route from the root vertex to the dangling vertex. Search algorithm modified hybrid metaheuristics is proposed by recombination of swarm and genetic search algorithms.

The first approach uses initially genetic algorithm and then the particle swarm algorithm.

The second approach uses the high-level nesting hybridization method based on combining genetic and particle swarm algorithms [9–11]. Hybridization usually implies unification of data structures, search space and modernization of integrated algorithms in connection with unification.

This concerns primarily types of the parameter values. Most algorithms in solving the combinatorial logical problems are using the integer parameter values [12, 13]. These algorithm operators introducing parameters with the integer values synthesize new integer parameter values. Classical paradigm of a particle swarm operates with real parameter values, while the particle swarm operators generate solutions with real values even on the basis of integer parameter values. In the proposed approach to constructing a modified paradigm in the indicated hybrid algorithm, chromosomes with integer parameter values and operators are used assisting chromosomes to evolve according to the rules of particle swarm and genetic search.

Search for the particle swarm algorithm solutions. Swarm algorithm is based on the process of step-by-step particles displacement to new positions in the search space [14, 15] determined as:

$$x_i(t+1) = x_i(t) + v_i(t+1),$$

where $v_i(t+1)$ is the vector (interval) of a particle displacement from the $x_i(t)$ position to the $x_i(t+1)$ position. The $v_i(t+1)$ vector shows the particle attraction to three attractors: $x_i(t)$ is the α_i particle current position; $x_i^*(t)$ is the α_i particle best position visited since the first iteration start; $x^*(t)$ is the α_i particle position in the particle swarm at the t time moment.

Approaches to particle swarm modification and hybridization considered below do not depend on the type of neighborhood topology.

The $v_i(t+1)$ vector is considered as a means of changing the decision and could take real values.

Target variable in the classification problem takes discrete values. Therefore, the work uses a solution search space with integer coordinate values. Particle descriptions and particle positions are presented in the form of a chromosome with integer values of genes, which in the genetic algorithm is the code of solution. Therefore, distance between positions corresponds to the degree of proximity between decisions. In this case, the search space could be considered as the affine search space.

In our case, chromosome that encodes such position is used as a position. The $x_i(t)$, $x_i^*(t)$, $x^*(t)$ positions correspond to the $H_i(t) = \{g_{il}(t) \mid l =$

$= 1, 2, \ldots, n_l\}$, $H_i^*(t) = \{g_{il}^*(t) \mid l = 1, 2, \ldots, n_l\}$, $H^*(t) = \{g_l^*(t) \| l = 1, 2, \ldots, n_l\}$ chromosomes.

Value of the pair of positions connection affinity with each other is determined by the distance between them. The smaller the distance between two positions, the more they are similar (close) to each other, and the greater is the affinity of connection between them.

At each step, the α_i particle passes in the affine space to a new H_i position, where the weight of the H_i position affine connection with the best position in the particle swarm is increasing. The distance between positions continuously decreases (the weight of affine connections between particles increases) in the process of particle swarm displacement.

Particle and position, where it is displaced, correspond to the same chromosome; therefore, two decoders $D1$ and $D2$ are used to obtain the particle and position phenotypes. As a result of applying the $D1$ decoder to the $H_i(t)$ chromosome, decision interpretation is being formed. When using the $D2$ decoder, a set of position coordinates is formed.

For each particle located in the $x_i\,(t)$ position at the t iteration, the $x_i^*(t)$ and $x^*\,(t)$ positions are determined, which are declared to be its attractors (centers of attraction).

Simultaneous attraction of the α_i particle to the three $x_i\,(t)$, $x_i^*(t)$, $x^*(t)$ attractors, when passing from the $x_i\,(t)$ position to the $x_i\,(t+1)$ position, is accounted as follows. Let us introduce the following notations: δ_1 is the weight of affine connection between $x_i\,(t)$ and $x_i^*(t)$; δ_2 is the weight of affine connection between $x_i\,(t)$ and $x^*\,(t)$; δ_3 is the weight of affine connection between $x_i\,(t+1)$ and $x_i\,(t)$; δ_4 is the weight of affine connection between $x_i\,(t+1)$ and $x_i^*(t)$; δ_5 is the weight of affine connection between $x_i\,(t+1)$ и $x^*\,(t)$. Displacement of the α_i particle from the $x_i\,(t)$ position to the $x_i\,(t+1)$ position is carried out using the directed mutation operator subjected to the condition: $\delta_1 + \delta_2 \leq$ $\leq \delta_3 + \delta_4 + \delta_5$. In other words, total connection affinity of the α_i particle with the three $x_i\,(t)$, $x_i^*(t)$, $x^*\,(t)$ attractors is not decreasing after displacement to a new position.

The α_i particle displacement means transition from the $H_i\,(t)$ chromosome to the $H_i\,(t+1)$ chromosome.

Purpose of the α_i particle displacement is to maximize the total weight of affine connections between the $H_i\,(t)$ position and the attractors.

Statement of problem in constructing a binary decision tree by methods of hybrid swarm intelligence. There is a set of objects $O = \{O_i \mid i = 1, 2, \ldots, n_o\}$, each is characterized by n_i features $A = \{A_i \mid i = 1, 2, \ldots, n_i\}$. A certain learning set of examples $P = \{P_i \mid i = 1, 2, \ldots, n_p\}$ is provided for objects with description of feature values and indication of the object class. Each A_i feature has two distinct values $Z_i^1,\ Z_i^2$.

It is necessary to elaborate an algorithm for constructing a binary classification model in the form of a decision tree, which would make it possible to classify new data coming from the outside. The goal of constructing a decision tree is to determine the categorical dependent variable values.

As an assessment of the classification quality, the $F_o = (n_o - n_o^*) / n_o$ value is chosen, where n_o is the total number of objects, and n_o^* is the number of correctly classified objects.

At the model construction stage, an ordered sequence of attributes is formed that are part of the route on the decision tree from the root vertex to the dangling vertex. Route construction ends, if the F_o minimum value (zero value) is reached or the C search depth (number of attributes in the sequence) reaches the $C_{\max}$ limiting value. In this case, the dangling vertex is declared as a leaf. The C parameter is the route estimate in the first case. In the second case, the F parameter is the route estimate:

$$F = \alpha F_o + \beta C,$$

where α, β are the proportionality coefficients.

Optimization goal is to minimize the F criterion.

Principles of binary decision tree coding. In this work, solution to the problem of constructing a qualification model lies in building a sequence of considered attributes and values thereof that are part of the M_k oriented route from the root vertex to the leaf. In general, the M_k route on a decision tree includes n_i vertices and n_i edges. The x_i and x_{i+1} vertices of the M_k route correspond to the A_i, A_{i+1} attributes. Each u_{ij} edge corresponds to the A_i attribute value, leaves out of x_i and enters x_{i+1}. The last edge in the route leaves the last vertex of the S_k list of vertices and enters the L vertex with the *-list* mark, which value corresponds to the number in the recognized class. States of the edges entering the M_k route are set by the W_k vector.

Elaborated interpretation of the decision tree is represented as a pair of chromosomes (H_k, W_k) [8, 9]. The $H_k = \{h_{ki} \mid i = 1, 2, \ldots, n_i\}$ chromosome is an ordered set of h_{ki} genes with integer values corresponding to the S_k ordered list all attributes included in the M_k route in the decision tree from the root vertex to the dangling vertex.

The $W_k = \{g_{ki} \mid i = 1, 2, \ldots, n_i\}$ chromosome is an ordered set of g_{ki} genes with integer values corresponding to the states (attribute values) entering the M_k route. Each g_{ki} gene corresponds to the w_{ki} edge index (A_i attribute value) connecting vertices x_i and x_{i+1} corresponding to genes h_{ki}, h_{ki+1}. The last route vertex is connected by an edge with the additional L vertex. The L vertex is intended for storing the classification result.

Example. To construct a classifier of rice variety, the $P = \{p_k \mid k = 1, 2, \ldots, n_t\}$ learning sample is set and presented in the Table. Each A_i feature has two values Z_i^1, Z_i^2.

Learning sample

Features					
Number	A_1, humidity, % (not more)	A_2, black dockage, % (not more)	A_3, yellowed cores, % (not more)	A_4, unpeeled cores, % (not more)	Variety
p_1	$Z_1^1 \le 10$	$Z_2^1 \le 0.2$	$Z_3^1 \le 2$	Z_4^1 are none	2
p_2	$Z_1^1 \le 10$	$Z_2^1 \le 0.2$	$Z_3^1 \le 2$	$Z_4^2 \le 2$	1
p_3	$Z_1^1 \le 10$	$Z_2^1 \le 0.2$	Z_3^2 are none	$Z_4^2 \le 2$	1
p_4	$Z_1^2 \le 18$	$Z_2^1 \le 0.2$	Z_3^2 are none	$Z_4^2 \le 2$	1
p_5	$Z_1^2 \le 18$	$Z_2^2 \le 0.3$	Z_3^2 are none	$Z_4^2 \le 2$	2
p_6	$Z_1^2 \le 18$	$Z_2^1 \le 0.2$	Z_3^2 are none	Z_4^1 are none	1
p_7	$Z_1^1 \le 10$	$Z_2^2 \le 0.3$	$Z_3^1 \le 2$	Z_4^1 are none	2

Let us consider the process of building a classification tree. Let the solution be set by the $H_1 = <x_2, x_4, x_3, x_1>$ chromosome, where x_i corresponds to A_i and the W_1 vector defining the states of edges connecting the vertices: $w_{24} = 1$, $w_{43} = 1$, $w_{31} = 1$, $w_{1L} = 1$. A pair of chromosomes corresponds to the route $M_1 = x_2, u_{24}, x_4, u_{43}, x_3, u_{31}, x_1, u_{1L}, L$. The route is supplemented with the L leaf. Here L is the class type determined after the tree is built (processing the M_1 route).

Decision tree is being formed sequentially. At each t step of constructing a tree in the M_1 route, the next x_i vertex and the u_{ij} outgoing edge are selected, for which the w_{ij} parameter value specifying the Z_i^1 or Z_i^2 values of the A_i attribute are determined in the W_1 vector.

In the first step, the A_2 attribute is selected in M_1. The set of P examples (see the Table) is divided into two subsets P_2^1 and P_2^2: $P_2^1 \in P$ contains n_2^1 examples with the first Z_2^1 value of the A_2 attribute, and $P_2^2 \in P$ contains n_2^2 examples with the second Z_2^2 value of the A_2 attribute: $P_2^1 \cup P_2^2 = P$. In our example: $n = |P| = 7$, $n_2^1 = |P_2^1| = 5$, $n_2^2 = |P_2^2| = 2$, $P_2^1 = (p_1, p_2, p_3, p_4, p_6)$, $P_2^2 = (p_5, p_7)$. In P_2^1, four examples correspond to the first variety, one to the second.

Further, in accordance with M_1, the Z_2^1 value is selected for A_2, and for further branching, P_2^1 is selected in accordance with the Z_4^1 and Z_4^2 values of the A_4 attribute. Further, the Z_4^2 value is selected for A_4 in accordance with M_1, and so on.

Fig. 1 shows a classifier that includes the M_1 route with given states of the W_1 edges. The $\theta = (\pi_1 : \pi_2)$ parameter fixes the ratio of the π_1 examples number of the first variety to the π_2 examples number of the second variety.

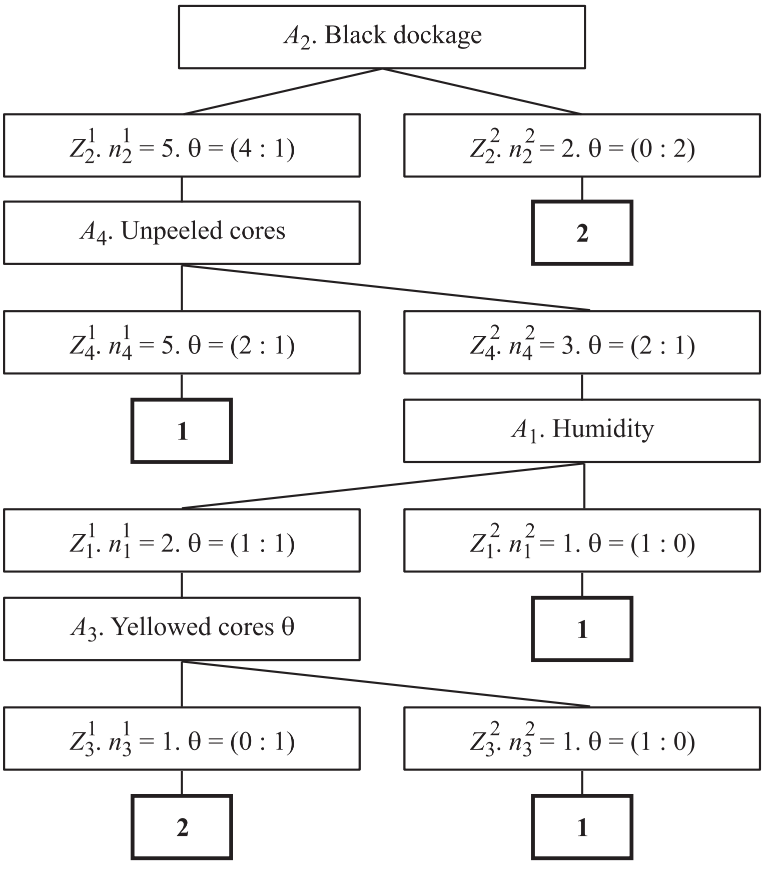

A_2. Black dockage
Z_2^1. $n_2^1 = 5$. $\theta = (4 : 1)$
Z_2^2. $n_2^2 = 2$. $\theta = (0 : 2)$
A_4. Unpeeled cores
2
Z_4^1. $n_4^1 = 5$. $\theta = (2 : 1)$
Z_4^2. $n_4^2 = 3$. $\theta = (2 : 1)$
1
A_1. Humidity
Z_1^1. $n_1^1 = 2$. $\theta = (1 : 1)$
Z_1^2. $n_1^2 = 1$. $\theta = (1 : 0)$
A_3. Yellowed cores θ
1
Z_3^1. $n_3^1 = 1$. $\theta = (0 : 1)$
Z_3^2. $n_3^2 = 1$. $\theta = (1 : 0)$
2
1

Fig. 1. Rice variety classifier

In our example (see Fig. 1), possible routes on the graph are as follows:

$$(M_2 = x_2, u_{2L}, L.\ w_{2L} = 0.\ L = 2);$$

$$(M_3 = x_2, u_{24}, x_4, u_{41}, x_1, u_{1L}, L.\ w_{24} = 1, w_{41} = 0, w_{1L} = 0.\ L = 1);$$

$$(M_4 = x_2, u_{24}, x_4, u_{41}, x_1, u_{13}, x_3, u_{3L}, L.\ w_{24} = 1, w_{41} = 0, w_{13} = 1, w_{3L} = 1.\ L = 2);$$

$$(M_5 = x_2, u_{24}, x_4, u_{41}, x_1, u_{13}, x_3, u_{3L}, L.\ w_{24} = 1, w_{41} = 0, w_{13} = 1, w_{3L} = 0.\ L = 1).$$

The M_2 route has the shortest length, while the M_4 and M_5 routes are having the maximum length.

Fig. 2 provides the process of reconstructing a decision tree from the found solution interpretation set by the M_1 and $M_1 W_1$ pair.

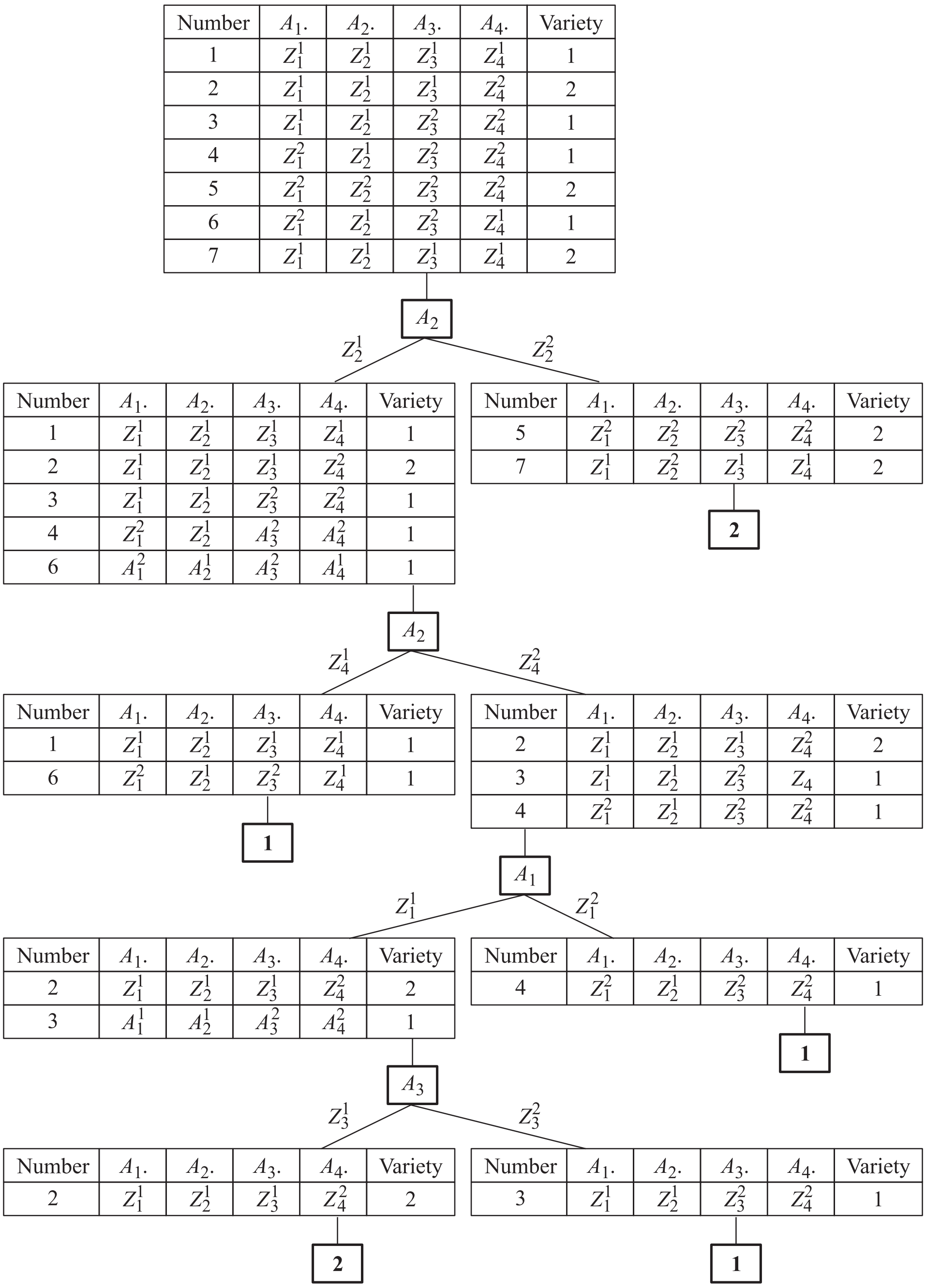
Procedures for forming positions and displacing particles in the decision search affine space. Current decision population is exposed to changes using genetic operators in the genetic algorithm at each iteration. When particles are displaced in the search space, a pair of chromosomes (H_k and W_k) is considered as a single object; however, mechanisms of these operators are different, independent and correspond to the H_k and W_k chromosome structures.

In a general case, the ξ single search space could be considered, where position of each α_k particle is determined by a pair of chromosomes (H_k, W_k). This work applies an approach, where the k-th population solution corresponds to a pair of α_{hk} and α_{wk} particles being synchronously displaced respectively in the ξ_h and ξ_w search subspaces, $\xi_h \cup \xi_w = \xi$.

The number of axes in the ξ_h search subspace of the α_{hk} particle described by the H_k chromosome is equal to the number of genes in the H_k chromosome. The $H_k(t) = < h_{ki}(t) \mid i = 1, 2, \ldots, n_i >$ chromosome corresponds to the ordered list of attributes $M_k(t) = < m_{ki}(t) \mid i = 1, 2, \ldots, n_i >$. Each i locus in the $H_k(t)$ chromosome corresponds to an axis in the ξ_h search subspace. Each axis hosts n_i reference points corresponding to the possible gene values. Note that each $h_{ki}(t)$ value is plotted only on a single axis and only once. For example, the $H_k(t) = < 1, 5, 6, 4, 2, 3 >$, $n_i = 6$ chromosome is located in a subspace with 6 coordinate axes.

At each t step, the α_{hk} particle exposed to attraction to the attractor moves from $H_k(t)$ to $H_k(t + 1)$ with the new mutual arrangement of genes.

To assess the affine relationship (distance estimation) between two positions, an indicator is used, i.e., the $\delta 1_{kz}(t)$ difference degree. The $\delta 1_{kz}(t)$ difference degree between two chromosomes of the same length is the number of loci,



Number	A_1.	A_2.	A_3.	A_4.	Variety
1	Z_1^1	Z_2^1	Z_3^1	Z_4^1	1
2	Z_1^1	Z_2^1	Z_3^1	Z_4^2	2
3	Z_1^1	Z_2^1	Z_3^2	Z_4^2	1
4	Z_1^2	Z_2^1	Z_3^2	Z_4^2	1
5	Z_1^2	Z_2^2	Z_3^2	Z_4^2	2
6	Z_1^2	Z_2^1	Z_3^2	Z_4^1	1
7	Z_1^1	Z_2^1	Z_3^1	Z_4^1	2

Number	A_1.	A_2.	A_3.	A_4.	Variety
1	Z_1^1	Z_2^1	Z_3^1	Z_4^1	1
2	Z_1^1	Z_2^1	Z_3^1	Z_4^2	2
3	Z_1^1	Z_2^1	Z_3^2	Z_4^2	1
4	Z_1^2	Z_2^1	A_3^2	A_4^2	1
6	A_1^2	A_2^1	A_3^2	A_4^1	1

Number	A_1.	A_2.	A_3.	A_4.	Variety
5	Z_1^2	Z_2^2	Z_3^2	Z_4^2	2
7	Z_1^1	Z_2^2	Z_3^1	Z_4^1	2

Number	A_1.	A_2.	A_3.	A_4.	Variety
1	Z_1^1	Z_2^1	Z_3^1	Z_4^1	1
6	Z_1^2	Z_2^1	Z_3^2	Z_4^1	1

Number	A_1.	A_2.	A_3.	A_4.	Variety
2	Z_1^1	Z_2^1	Z_3^1	Z_4^2	2
3	Z_1^1	Z_2^1	Z_3^2	Z_4	1
4	Z_1^2	Z_2^1	Z_3^2	Z_4^2	1

Number	A_1.	A_2.	A_3.	A_4.	Variety
2	Z_1^1	Z_2^1	Z_3^1	Z_4^2	2
3	A_1^1	A_2^1	A_3^2	A_4^2	1

Number	A_1.	A_2.	A_3.	A_4.	Variety
4	Z_1^2	Z_2^1	Z_3^2	Z_4^2	1

Number	A_1.	A_2.	A_3.	A_4.	Variety
2	Z_1^1	Z_2^1	Z_3^1	Z_4^2	2

Number	A_1.	A_2.	A_3.	A_4.	Variety
3	Z_1^1	Z_2^1	Z_3^2	Z_4^2	1

Fig. 2. Decision tree construction process

and in each of them the $h_{ki}(t) \in H_k(t)$ and $h_{zi}(t) \in H_z(t)$ genes do not coincide. Let $\delta 1_{kz}(t)$ be the initial difference degree between positions $H_k(t)$ and $H_z(t)$. By modifying $H_k(t)$, the α_{hk} particle is displaced to the new $H_k(t+1)$ position with a lower value of the difference degree: $\delta 1_{kz}(t+1) \le \delta 1_{kz}(t)$.

An increase in the affine relationship value between $H_k(t)$ and $H_z(t)$ is performed by implementing selective pairwise rearrangements of genes between loci in the $H_k(t)$ position.

A set of $L(t) = < l_i(t) \mid i = 1, 2, \ldots, n_l >$ is generated, where genes are located not coinciding with the genes located in the corresponding loci of the $H_z(t)$ chromosome.

With the $\pi_1 = \varphi / n_l$ probability, the $l_i(t) \in L(t)$ locus is selected in the $H_k(t)$ chromosome, and the $h_{ki}(t) \in H_k(t)$ gene located in this $l_i(t)$ locus is determined, φ is the coefficient.

Sequentially starting from the first, the set of $L(t) = < l_i(t) \mid i = 1, 2, \ldots, n_l >$ loci in the $H_z(t)$ chromosome is considered, and the $l_j(t) \in L(t)$ locus is identified, where the $h_{zj}(t) \in H_k(t)$ gene is located, such that $h_{zj}(t) = h_{ki}(t)$.

Genes located in the $l_i(t)$ and $l_j(t)$ loci of the $H_k(t)$ chromosome are reversing. From now on, genes with the same value $\delta 1_{kz}(t + 1) \le \delta 1_{kz}(t)$ are located in the $l_i(t)$ locus of the $H_k(t + 1)$ and $H_z(t)$ chromosomes. The μ number of such paired permutations is a control parameter, and the $\mu < n_l$ condition should be satisfied.

An example of modifying the $H_k(t)$ position, when performing displacement.

Let the $H_k(t)$ and $H_z(t)$ positions have the following form:

$$H_k(t) = \{1, 3, 2, 4, 5, 6\},\ H_z(t) = \{1, 4, 2, 3, 5, 6\}.$$

A set of $L(t) = < 2, 4 >$ loci is formed in the $H_k(t)$ chromosome, where genes are located not coinciding with the genes located in the corresponding loci of the $H_z(t)$ chromosome, $\delta 1_{kz}(t) = 2$. After rearranging genes in $H_k(t)$ between the second and fourth loci, the following is obtained:

$$H_k(t + 1) = \{1, 4, 2, 3, 5, 6\} \text{ and } \delta 1_{kz}(t + 1) = 0.$$

Distance between $H_k(t)$ and $H_z(t)$ decreased, connection affinity between $H_k(t)$ and $H_z(t)$ increased.

The number of axes of the ξ_w search subspace of the α_{wk} particle described by the $W_k = \{g_{ki} \mid i = 1, 2, \ldots, n_i\}$ chromosome is the same, as in the ξ_h search subspace. The $g_{ki}(t) \in W_k$ gene value is an attribute value variant. If $g_{ki}(t) = 1$, then the first value of the corresponding attribute is selected, if $g_{ki}(t) = 2$, then the second value. Each $g_{ki}(t) \in W_k$ corresponds to its own axis, which scale includes two reference points, i.e., x_{i1}, x_{i2}. If $g_{ki}(t) = 1$, then $x_{i1} = 1$. If $g_{ki} = 2$, then $x_{i2} = 2$.

For example, position in the search space has the following form: $W = \{2, 2, 1, 1, 2, 2\}$.

The $\delta 2_{kz}(t)$ difference degree between two chromosomes of the same length is the number of mismatched gene values in the same loci. Let $\delta 2_{kz}(t)$ be the calculated difference degree between the $W_k(t)$ and $W_z(t)$ positions. By modifying the $W_k(t)$, the α_{wk} particle is displaced to a new $W_k(t+1)$ position with a lower value of the difference degree: $\delta_{kz}(t+1) \le \delta_{kz}(t)$.

Gene values in each i locus of the new $W_k(t+1)$ position are determined as follows:

$$\text{if } g_{ki}(t) = g_{zi}(t), \text{ then } g_{ki}(t+1) = g_{ki}(t);$$

$$\text{if } g_{ki}(t) \ne g_{zi}(t), \text{ then } g_{ki}(t+1) = g_{zi} \text{ with the } \pi \text{ probability;}$$

$$\pi = \varepsilon \delta 2_{\mathrm{kz}}(t)(t) / n_i.$$

Where ε is the coefficient; n_i is the number of genes in chromosomes. The higher is the $\delta 2_{kz}(t)$ difference degree between $W_k(t)$ and $W_z(t)$, the higher is probability that the $g_{zi}(t)$ value would become the $g_{ki}(t+1)$ value.

Example. Let $W_z(t) = <1, 2, 2, 1, 1, 2, 2, 2, 1, 1, 2>$;

$$W_k(t) = <2, 1, 2, 2, 1, 2, 1, 2, 2, 1, 2>.$$

Here $|W_z(t)| = |W_k(t)| = 11$. Gene values do not coincide in loci 1, 2, 4, 7, 9, $\delta_{kzk}(t) = 5$. Let genes 2, 4, and 7 mutate in the $W_k(t)$ with a certain probability $\pi = \varepsilon \cdot 5/11$. The $W_k(t+1)$ modified position has the following form: $W_k(t+1) = <2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2>$. The difference degree is $\delta 2_{kz}(t+1) = 2$. Connection affinity between $W_k(t)$ and $W_z(t)$ increased by the $AS_w = k_2(\delta 2_{kz}(t+1) - \delta 2_{kz}(t))$ value.

Affine connection total value increased by the following value:

$$AS = AS_h + AS_w =$$

$$= k_1(\delta 1_{kz}(t)(t+1) - \delta 1_{kz}(t)(t)) + k_2(2\delta_{kz}(t)(t+1) - \delta 2_{kz}(t)(t)).$$

Experimental research. The developed algorithm for constructing a qualification model was implemented in the form of a GA-RCh DR program for constructing the decision tree.

GA-RCh DR program testing was carried out on test cases with the known K_{opt} optimum [13, 16, 17]. Obtained solutions quality level was assessed by the

$P = K_{\text{opt}}/K$ indicator, where K is the optimization criterion value used in the GA-RCh DR program. The number of iterations, where the algorithm reached the maximum quality level, was not exceeding 135 (Fig. 3).

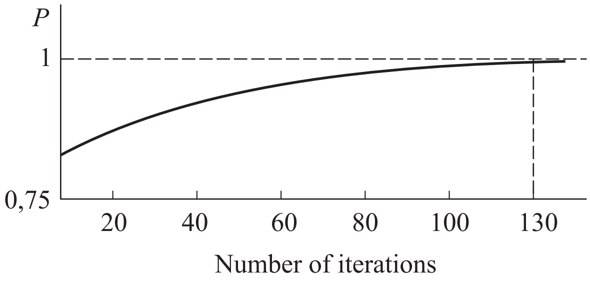


Fig. 3. Dependence of the GA-RCh DR algorithm quality level (P) on the number of iterations

Comparison of the GA-RCh DR algorithm in terms of quality level with the genetic algorithm and the particle swarm algorithm demonstrated that with comparable time expenditure, the P indicator in GA-RCh DR algorithm was higher in average by 9–11 %. The average quality level achieved by the GA-RCh DR algorithm at 130 iterations differs from the maximum value by 0.15 %. Overall estimate of the time complexity lies in the $O(n^2)$ — $O(n^3)$ range, where n is the number of features.

Conclusion. Solving the problem of constructing a classification model is presented in the form of a sequence of considered attributes and values thereof included in the M_k route from the root vertex to the dangling vertex. Developed interpretation of the decision tree is presented as a pair of chromosomes (S_k, W_k). List of genes of the S_k chromosome corresponds to the list of all attributes included in the M_k route in the decision tree. Gene values of the W_k chromosome correspond to the attribute values included in the M_k route.

For hybridization, unification of data structures, search space and integrable algorithms modernization were performed. Hybrid algorithm operators use integer-valued parameters and synthesize new integer parameter values.

Modified hybrid metaheuristic of the search algorithm is proposed for constructing a classification model through recombination of swarm and genetic search algorithms. The first approach initially uses the genetic algorithm and then the particle swarm algorithm.

The second approach implies the high-level nesting hybridization method based on combining genetic algorithm and particle swarm algorithm [10, 11]. Position alteration of a particle represented as a genotype leads to both parametric and structural changes. Thus, the proposed modified system is capable of adapting based on parametric and structural changes. A method was elaborated to account for the α_i particle simultaneous attraction to several attractors, when displaced to a new position.

The proposed approach to constructing a modified paradigm uses chromosomes with integer values of parameters in the indicated hybrid algorithm and operators allowing chromosomes to evolve according to the rules of particle swarm and genetic search.

Translated by K. Zykova

REFERENCES

[1] Witten I.H., Frank E., Hall M.A. Data mining. San Francisco, Morgan Kaufmann, 2011.

[2] Zhuravlev Yu.I., Ryazanov V.V., Sen'ko O.V. Raspoznavanie. Matematicheskie metody. Programmnaya sistema. Prakticheskie primeneniya [Recognition. Mathematical methods. Software system. Practical applications]. Moscow, Fazis Publ., 2006.

[3] Berikov V.S., Lbov G.S. [Modern trends in cluster analysis]. *Vserossiyskiy konkursnyy otbor obzorno-analiticheskikh statey po prioritetnomu napravleniyu "Informatsionno-telekommunikatsionnye sistemy"* [All-Russian competitive selection of review and analytical articles in the priority area of "Information and Telecommunication Systems"]. Moscow, Informika Publ., 2008, art. 126 (in Russ.).

[4] Barsegyan A.A., Kupriyanov M.S., Stepanenko V.V., et al. Metody i modeli analiza dannykh: OLAP i Data Mining [Methods and models of data analysis: OLAP and Data Mining]. St. Petersburg, BKhV-Peterburg Publ., 2004.

[5] Karpenko A.P. Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennye prirodoy [Modern search optimization algorithms. Algorithms inspired by nature]. Moscow, Bauman MSTU Publ., 2014.

[6] Wang X. Hybrid nature-inspired computation method for optimization. Doc. Diss. Helsinki University of Technology, 2009.

[7] Lebedev B.K., Lebedev O.B., Lebedev V.B. Mechanisms of the roving algorithm for finding the solution of the problem of distribution of connections. *Programmnye produkty, sistemy i algoritmy* [Software Journal: Theory and Applications], 2017, no. 4 (in Russ.). Available at: http://swsys-web.ru/en/mechanisms-of-the-roving-algorithm-for-finding-the-solution-of-the-problem-of-distribution-of-connections.html

[8] Lebedev B.K., Lebedev O.B. Hybrid bioinspired algorithm for solving symbolic regression problem. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, no. 6 (167), pp. 28–41 (in Russ.).

[9] Kureychik V.M., Lebedev B.K., Lebedev O.B. Poiskovaya adaptatsiya: teoriya i praktika [Search adaptation: theory and practice]. Moscow, FIZMATLIT Publ., 2006.

[10] Lebedev B.K., Lebedev O.B., Lebedeva E.M. Distribution of resources based on hybrid models of swarm intelligence. *Nauchno-tekhnicheskiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics], 2017, vol. 17, no. 6, pp. 1063–1073 (in Russ.).
DOI: https://doi.org/10.17586/2226-1494-2017-17-6-1063-1073

[11] Kureychik V.V., Kureychik Vl.Vl. The architecture of hybrid search for design. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, no. 7 (132), pp. 22–27 (in Russ.).

[12] Clerc M. Particle swarm optimization. London, ISTE, 2006.

[13] Lebedev B.K., Lebedev V.B. The evolutionary learning procedure for pattern recognition. *Izvestiya TSREU* [Izvestiya TRTU], 2004, no. 8 (43), pp. 83–84 (in Russ.).

[14] Lebedev B.K., Lebedev V.B., Lebedev O.B. The solution of the symbolic regression problem by genetic search methods. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, no. 2 (163), pp. 212–225 (in Russ.).

[15] Kennedy J., Eberhart R.C. Particle swarm optimization. *Proc. ISNN*, 1995, pp. 1942–1948. DOI: https://doi.org/10.1109/ICNN.1995.488968

[16] Lebedev B.K., Lebedev O.B., Lebedeva E.M. Partition a class method alternative collective adaptation. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2016, no. 7 (180), pp. 89–101 (in Russ.).

[17] Cong J., Romesis M., Xie M. Optimality, scalability and stability study of partitioning and placement algorithms. *Proc. ISPD*, 2003, pp. 88–94.

Lebedev B.K. — Dr. Sc. (Eng.), Professor, Department of Computer Aided Design Systems, Academy for Engineering and Technologies, Southern Federal University (Nekrasovsky pereulok 44, Taganrog, 347900 Russian Federation).

Lebedev O.B. — Cand. Sc. (Eng.), Assoc. Professor, Department of Computer Aided Design Systems, Academy for Engineering and Technologies, Southern Federal University (Nekrasovsky pereulok 44, Taganrog, 347900 Russian Federation).

Zhiglaty A.A. — Assistant of the Department of Mathematical Software and Computer Applications, Academy for Engineering and Technologies, Southern Federal University (Nekrasovsky pereulok 44, Taganrog, 347900 Russian Federation).

Please cite this article as:

Lebedev B.K., Lebedev O.B., Zhiglaty A.A. Binary decision tree construction using the hybrid swarm intelligence. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2021, no. 2 (135), pp. 52–65.
DOI: https://doi.org/10.18698/0236-3933-2021-2-52-65