Kureichik Viktor Mikhaylovich – Southern Federal University, e-mail: vmkureychik@sfedu.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634311487; the department of computer aided design; dr. of eng. sc.; professor.

Safronenkova Irina Borisovna – Federal Research Centre The Southern Scientific Centre of the Russian Academy of Sciences; e-mail: safronenkova050788@yandex.ru; 41, Chekhov street, Rostov-on-Don, 344006, Russia; phone: +79604678753; junior researcher.

УДК 004.896 DOI 10.18522/2311-3103-2020-4-82-93

Б.К. Лебедев, В.Б. Лебедев, О.Б. Лебедев

ПОПУЛЯЦИОННЫЙ АЛГОРИТМ ПОСТРОЕНИЯ ДЕРЕВА РЕШЕНИЙ МЕТОДОМ КРИСТАЛЛИЗАЦИИ РОССЫПИ АЛЬТЕРНАТИВ*

В ряде случаев возникает необходимость установления соответствия между заявленным и фактическим значением категориальной переменной на основе совокупности признаков объекта. В этом случае возникает потребность в классификаторе с оптимальной последовательностью рассматриваемых атрибутов с заданным значением целевой функции. Значением целевой переменной может быть: да, нет, номер сорта, номер класса и т.д. В работе решается задача построения классификационной модели в виде оптимальной последовательность рассматриваемых атрибутов и их значений, входящих в состав маршрута от корневой вершины к концевой вершине с заданным значением целевой переменной. Если требуется классификатор, включающий возможность альтернативных ответов, то вначале строятся независимо друг от друга оптимальные маршруты для каждого значения целевой переменной, а затем эти маршруты объединяются («склеиваются») в единое бинарное дерево решений. В алгоритме построения классификатора на основе метода кристаллизации россыпи альтернатив, каждое решение Q_k интерпретируется в виде в ориентированного маршрута M_k на бинарном дереве решений. Назовем порядковый номер элемента в ориентированном маршруте M_k позицией $s_i \in S=\{s_i|i=1,2,...,n_A\}$. Элементом маршрута M_k является пара (x_i,u_{i-}), где x_i соответствует A_i. u_{i-} в маршруте M_k является ребром, выходящим из x_i и соответствует выбранному вместе с A_i значению A_i. Второй индекс элемента u_{i-} определится после выбора A_i, помещенного в соседнюю с s_j позицию s_{j+1}. Работа алгоритма построения дерева решений базируется на использовании коллективной эволюционной памяти, под которой подразумевается информация, отражающая историю поиска решения. Алгоритм учитывает тенденции к использованию альтернатив из наилучших найденных решений. Особенностями являются наличие непрямого обмена информацией – стигмержи. Совокупность данных об альтернативах и их оценках составляет россыпь альтернатив. Рассмотрены ключевые моменты анализа альтернатив в процессе эволюционной коллективной адаптации. Экспериментальные исследования показали, что разработанный алгоритм находит решения, не уступающие по качеству, а иногда и превосходящие своих аналогов в среднем на 3–4 %. Временная сложность алгоритма, полученная экспериментальным путем, лежит в пределах $O(n^2)$-$O(n^3)$.

Классификация; дерево решений; оптимизация; популяционный алгоритм; адаптивное поведение; метод кристаллизации россыпи альтернатив.

B.K. Lebedev, V.B. Lebedev, O.B. Lebedev

POPULATION ALGORITHM FOR CONSTRUCTING A TREE OF SOLUTIONS BY METHOD OF CRYSTALLIZATION OF ALTERNATIVES FIELD

In some cases, it becomes necessary to establish a correspondence between the declared and actual value of a categorical variable on the basis of a set of object characteristics. In this case, there is a need for a classifier with an optimal sequence of the considered attributes with a

* Работа выполнена при финансовой поддержке гранта РФФИ № 20–07–00260 А.

given value of the objective function. The target variable can be: yes, no, variety number, class number, etc. This paper solves the problem of constructing a classification model in the form of an optimal sequence of the considered attributes and their values included in the route from the root vertex to the terminal vertex with a given value of the target variable. If a classifier is required that includes the possibility of alternative answers, then first, independently from each other, optimal routes are built for each value of the target variable, and then these routes are combined ("glued") into a single binary decision tree. In the algorithm for constructing a classifier based on the method of crystallization of a placer of alternatives, each solution Q_k is interpreted as an oriented route M_k on a binary decision tree. Let us call the ordinal number of an element in the directed route M_k the position $s_i \in S=\{s_i|i=1,2,\ldots,n_A\}$. An element of the route M_k is the pair $(x_i, u_{i\cdot})$, where x_i corresponds to A_i. $u_{i\cdot}$ in the route M_k is an edge outgoing from x_i and corresponds to the value A_i chosen together with A_i. The second index of the element $u_{i\cdot}$ is determined after the choice of A_i, placed in the position s_j+1 adjacent to s_j. The work of the decision tree construction algorithm is based on the use of collective evolutionary memory, which is understood as information reflecting the history of the search for a solution. The algorithm takes into account the tendency to use alternatives from the best solutions found. The peculiarities are the presence of an indirect exchange of information – stigmerges. The totality of data on alternatives and their assessments constitutes a scattering of alternatives. The key points of the analysis of alternatives in the process of evolutionary collective adaptation are considered. Experimental studies have shown that the developed algorithm finds solutions that are not inferior in quality, and sometimes surpass their counterparts by an average of 3–4 %. The time complexity of the algorithm, obtained experimentally, lies within $O(n^2)$-$O(n^3)$.

Classification; decision tree; optimization; population algorithm; adaptive behavior; method of crystallization of alternatives placer.

Введение. Наиболее распространенные методы решения задач *классификации* используют в качестве квалификационной модели дерево решений $D=(X,U)$, где $X=\{x_i|i=1,2,\ldots,n\}$ – множество вершин, $U=\{u_i|i=1,2,\ldots,m\}$ – множество ребер [1–3]. Множество X включает множество X_1 внутренних вершин и множество X_2 ««концевых»» вершин. Внутренние вершины дерева решений соответствуют признакам, характеризующим объект и подвергающимся разбиению, в зависимости от значений признаков. «Концевые» вершины соответствуют значениям категориальных переменных (конкретный класс, сорт и т.д.) [4,5]. Все ребра ориентированные. При этом ребро, выходящее из вершины x_i, соответствует значению признака x_i [4], другими словами все ребра дерева решений D помечены метками, соответствующими значениям признаков. Решение задачи классификации нового объекта, заключается в построении на дереве решений ориентированного маршрута M, начиная от корня до одной из концевых вершин. Порядок вершин в ориентированном маршруте определяет порядок учета признаков. Если целевая переменная принимает дискретные значения, то решается задача классификации. Самым распространенным, и наиболее простым случаем являются бинарные деревья решений (БДР) [3–4]. Эффективность БДР во многом зависит от правильного выбора последовательности и критерия ветвления внутренних вершин дерева решений.

Большинство из известных алгоритмов (CART, C4.5, NewId, ITrule, CHAID, CN2 и т.д. [1–4]) являются «жадными алгоритмами» последовательного типа. При использовании такого подхода построение БДР происходит сверху вниз. На каждом шаге жадного алгоритма разбиение множества объектов производится по признаку, обеспечивающему возникновение максимального различия между подмножествами. Последовательные алгоритмы отличаются наименьшей трудоемкостью, но дают наименьшее качество.

В работе решается задача построения классификационной модели в виде оптимальной последовательности рассматриваемых атрибутов и их значений, входящих в состав маршрута от корневой вершины к концевой вершине с заданным значением целевой переменной.

Если требуется классификатор, включающий возможность альтернативных ответов, то вначале строятся независимо друг от друга оптимальные маршруты для каждого значения целевой переменной, а затем эти маршруты объединяются («склеиваются») в единое БДР.

В ряде случаев возникает необходимость установления соответствия между заявленным и фактическим значением категориальной переменной на основе совокупности признаков объекта. В этом случае возникает потребность в классификаторе с оптимальной последовательностью рассматриваемых атрибутов с заданным значением целевой функции. Значением целевой переменной может быть: да, нет, номер сорта, номер класса и т.д.

Эффективным направлением повышения качества решений стало использование стохастических **популяционных** алгоритмов [5], которые, как правило, итерационные и работают в пространстве полных решений. Широкое распространение получили роевые и генетические алгоритмы. Исследования эффективности популяционных алгоритмов показали, что мощным средством повышения эффективности новых алгоритмов является их гибридизация [6, 7]. Рекомбинация метаэвристик популяционных алгоритмов обеспечивает более равномерный и обоснованный просмотр пространства поиска и более высокую эффективность интегрированных алгоритмов [8].

Разработка новых поисковых алгоритмов заключается в использовании и модификации метаэвристик, заложенных в природных механизмах принятия решений. В работе по аналогии с меаэвристиками, на которых построены роевые алгоритмы, используется метаэвристика [10], учитывающая тенденцию к использованию альтернатив (вариантов компонентов) из наилучших найденных решений. Особенностями являются наличие непрямого обмена информацией – стигмержи [11]. Совокупность данных об альтернативах и их оценках составляет **россыпь альтернатив.** Рассмотрены ключевые моменты анализа альтернатив в процессе эволюционной коллективной адаптации [12, 13]. Алгоритм на основе кристаллизации россыпи альтернатив был успешно применен для решения задачи построения дерева решений. В процессе эволюционной коллективной адаптации методами дискриминантного анализа формируются оценки приспособленности альтернатив. Приспособленность альтернатив рассматривается как вероятность ее использования в формируемом решении. Совокупность данных об альтернативах и их оценках составляет **россыпь альтернатив.** Дискриминантный анализ и поиск эффективных альтернатив, в процессе эволюционной коллективной адаптации назван по аналогии с процессами вычленения объектов (формирования кристаллов) кристаллизацией. Другими словами, в процессе эволюционной коллективной адаптации производится вычленение из множества вариантов наиболее приспособленных альтернатив. Отсюда название метода оптимизации − метод кристаллизации россыпи альтернатив (КРА), (Crystallization of alternatives field (CAF)).

1. Постановка задачи построения бинарного дерева решений методами кристаллизации россыпи альтернатив. Имеется множество объектов $W=\{w_i|i=1,2,\ldots,n_o\}$, каждый из которых характеризуется n_A признаками $A=\{A_i|i=1,2,\ldots,n_A\}$. Задано некоторое обучающее множество примеров $P=\{P_i|i=1,2,\ldots,n_p\}$ для объектов с описанием значений признаков, и указаний класса объекта. Каждый признак A_i имеет два отличных друг от друга значения Z_{i1}, Z_{i2}.

Необходимо разработать алгоритм построения решения бинарной классификационной модели в виде дерева решений, который позволил бы классифицировать новые поступающие извне данные. Целью построения дерева решения является определение значения категориальной зависимой переменной.

В общем случае маршрут M_k на дереве решений включает n_A вершин и n_A ребер. Вершины x_i и x_{i+1} маршрута M_k соответствуют признакам A_i, $A_{i+1.}$ Каждое ребро u_{ij} соответствует одному из двух значений $Z_{it}(t=1,2)$ признака A_i, выходит из x_i и входит в x_{i+1}. Последнее ребро в маршруте выходит из последней вершины списка вершин S_k и входит в вершину L_c меткой *list*, значение которой соответствует номеру распознаваемого класса. Значения ребер, входящих в маршрут $M_{k,}$ задается списком W_k.

Пример. Для построения классификатора сорта пшена задана обучающая выборка $P=\{P_i|i=1,2,\ldots,n_p\}$, представленная в таблице 1. Каждый признак A_i имеет два значения Z_{i1}, Z_{i2}.

Таблица 1

Обучающая выборка

	Признаки				
№	***A₁.Влажность в%. Не более***	***A₂.Сорная примесь в%. Не более***	***A₃. Почерневшие Ядра в%. Не более***	***A₄.Неочищенные ядра в%. Не более***	***Сорт***
p_1	$Z_{11}\leq 10\%$	$Z_{21}\leq 0.2\%$	$Z_{31}\leq 2\%$	$Z_{41}=0$	2
p_2	$Z_{11}\leq 10\%$	$Z_{21}\leq 0.2\%$	$Z_{31}\leq 2\%$	$Z_{42}\leq 2\%$	1
p_3	$Z_{11}\leq 10\%$	$Z_{21}\leq 0.2\%$	$Z_{32}=0$	$Z_{42}\leq 2\%$	1
p_4	$Z_{12}\leq 18\%$	$Z_{21}\leq 0.2\%$	$Z_{32}=0$	$Z_{42}\leq 2\%$	1
p_5	$Z_{12}\leq 18\%$	$Z_{22}\leq 0.3\%$	$Z_{32}=0$	$Z_{42}\leq 2\%$	2
p_6	$Z_{12}\leq 18\%$	$Z_{21}\leq 30.2\%$	$Z_{32}=0$	$Z_{41}=0$	1
p_7	$Z_{11}\leq 10\%$	$Z_{22}\leq 0.3\%$	$Z_{31}\leq 2\%$	$Z_{41}=0$	2

Рассмотрим построенный на базе обучающей выборки в дереве решений маршрут M_k концевая вершина которого соответствует заданному классу (1 сорту).

(M_k=<x_2, u_{24}; x_4, u_{41}; x_1, u_{13}; $x_{3,}$ u_{3L}; L>. $u_{24}= Z_{21}$, $u_{41}= Z_{42}$, $u_{13}= Z_{11}$,$u_{3L}= Z_{32}$. L=1copm).

Маршруту M_k соответствует фрагмент классификатора, представленный на рис. 1. Все вершины БДР обладают памятью. В памяти внутренней вершины БДР хранится информации о признаке и его значениях, в соответствии с которыми вершина подвергалась разбиению. В памяти концевой вершины (листа) хранится информация о номере класса (сорта), в состав которого входят примеры данной вершины, и о последовательности признаков, входящих в состав маршрута M_k. В результате прохождения от корня дерева до концевой вершины решается задача о принадлежности объекта к номеру класса, хранящемся в концевой вершине. Введем обозначения.

n_i – число примеров с признаком A_i.

n_{ij} – число примеров j-го сорта с признаком A_i.

Параметр $П=(\pi_1:\pi_2)$фиксирует отношение числа π_1 примеров первого сорта к числу π_2 примеров второго сорта.

В качестве оценки качества классификации выбрана величина $F_o=(n_o-n^*_o)/n_o$, где n_o – общее число объектов, n_{*o} – количество правильно классифицированных объектов.

В работе на этапе построения модели формируется упорядоченная последовательность признаков, входящих в состав маршрута на дереве решений от корневой вершины к висячей вершине. Построение маршрута заканчивается, если достигнуто минимальное значение F_o (нулевое значение) или глубина поиска C (число признаков в последовательности) достигла предельного значения – C_{max}. При этом висячая вершина объявляется листом. Оценкой маршрута в первом случае является параметр C. Во втором случае оценкой маршрута является параметр F:

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

где α, β – коэффициенты пропорциональности.

Цель оптимизации – минимизация критерия F.

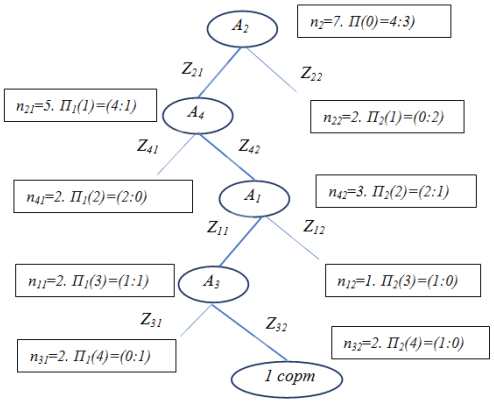

Рис. 1. Фрагмент классификатора сорта пшена, соответствующий маршруту M_k

2. Механизмы построения классификатора на основе метода кристаллизации россыпи альтернатив. В алгоритме построения классификатора на основе методе КРА [10-14] каждое решение Q_k интерпретируется в виде ориентированного маршрута M_k на БДР. Назовем порядковый номер элемента в ориентированном маршруте M_k позицией $s_i \in S=\{s_i|i=1,2,\ldots,n_A\}$. Элементом маршрута M_k является пара $(x_i,u_{i\text{-}})$, где x_i соответствует признаку A_i. Элемент $u_{i\text{-}}$ в маршруте M_k является ребром, выходящим из x_i и соответствует выбранному вместе с A_i значению A_i. Второй индекс элемента $u_{i\text{-}}$ определится после выбора A_i, помещенного в соседнюю с s_j позицию s_{j+1}.

Каждый агент A_i с выбранным значением может располагаться в одной из альтернативных позиций $s_i \in S=\{s_i|i=1,2,\ldots,n_A\}$, в соответствии с маршрутом M_k.

В каждой из позиций может располагаться только один агент. Каждый агент может иметь одно из двух значений. Решение Q_k определяется совокупностью альтернативных значений и позиций множества агентов A.

В работе под *россыпью альтернатив* (РА) решения Q_k называется структура данных, в виде матрицы $R_k=||r_{kjl}||_{m\times 2n}$, несущая информацию об занимаемых агентами альтернативных позициях (порядковых номерах), значениях агентов в ориентированном маршруте M_k и значениях их пригодностей, составляющих коллективную эволюционную память (КЭП) (рис.2).

Столбцы матрицы попарно соответствуют агентам, строки – альтернативным позициям, в которых может располагаться агент.

Номер строки (вектора $R_{kj}=\{r_{jl}|l=(1,2),(3,4),\ldots,2n_A\}$) матрицы R_k соответствует порядковому номеру j позиции в маршруте. Число строк равно числу позиций. Размер строки равен удвоенному числу агентов (признаков).

Каждому агенту A_i соответствует два столбца – два вектора $R1_{k\alpha}=\{r_{kj\alpha}|\alpha=(2i\text{-}1);\ i=1,2,\ldots,n_A\}$ и $R2_{k\beta}=\{r_{kj\beta}|\beta=2i;\ i=1,2,\ldots,n_A\}$). Элементы вектора $R1_{k\alpha}$ соответствуют паре $(x_i,u_{i\text{-}})$ с первым значением признака. Элементы вектора $R2_{k\beta}$ соответствуют паре $(x_i,u_{i\text{-}})$ со вторым значением признака. Размер векторов $R1_{k\alpha}$ и $R2_{k\beta}$ равен числу позиций агентов (признаков). Число столбцов матрицы R_k равно удвоенному числу агентов (признаков).

Для любого A_i в векторах $R1_{k\alpha}$ и $R2_{k\beta}$ только один из двух элементов $r_{kj\alpha}$ или $r_{kj\beta}$ соответствующих состоянию c_{ij} в котором находится агент имеет значение, отличное от нуля. Остальные элементы вектора R_{ki} имеют нулевые значения.

В каждой строке (векторе R_{kj}) только один элемент r_{kji} имеет значение, отличное от нуля. Для фиксации значения каждого агента A_i в векторе R_{kj} используются два соседних элемента $r_{kj\alpha}$ и $r_{kj\beta}$, где $\alpha=(2i-1)$, $\beta=2i$: $r_{kj\alpha}$ – для первого значения, $r_{kj\beta}$ – для второго значения.

Каждый, отличный от нуля элемент r_{kji} матрицы $R_k=||r_{kji}||_{m\times 2n}$, имеет значение, равное полезности δ_k решения Q_k, при котором агент a_i назначен в позициюs_j. $\delta_k=f(\xi_k)$, где ξ_k – оценка решения, а δ_k – оценка полезности этого решения.

Пусть для формирования квалификационной модели построено множество решений $Q=\{Q_k|k=1,2,\ldots,n_k\}$.

Рассмотрим процедуру отображения решения Q_k в матрице R_k. В качестве примера используем рассмотренный выше маршрут M_k концевая вершина которого соответствует заданному классу (1 сорту).

($M_k=<x_2, u_{24}; x_4, u_{41}; x_1, u_{13}; x_{3,} u_{3L};L>$. $u_{24}= Z_{21}$, $u_{41}= Z_{42}$, $u_{13}= Z_{11}$,$u_{3L}= Z_{32}$. $L=1$ сорт).

Решение Q_k формируется четырьмя агентами. Для решения Q_k рассчитаны оценки ξ_k и δ_k.

Для отображения решения Q_k в матрице R_k формируется набор ячеек Θ_k, соответствующих состояниям агентов в маршруте M_k.

Первый элемент в маршруте M_k – агент A_2, следовательно, номер агента i=2, номер позиции агента $j=1$. Выбрано первое значение признака A_2. Данной ситуации в матрице R_k соответствует элемент $r_{kj\alpha}$ с индексами: $j=1$, $\alpha=2i-1=3$. $r_{k13.}$

Второй элемент в маршруте M_k – агент A_4, следовательно, номер агента i =4, номер позиции агента $j=2$. Выбрано второе значение признака A_4. Данной ситуации в матрице R_k соответствует элемент $r_{kj\beta}$ с индексами: $j=2$, $\beta=2i=8$. r_{k28}

Третий элемент в маршруте M_k – агент A_1, следовательно, номер агента i =1, номер позиции агента $j=3$. Выбрано первое значение признака A_1. Данной ситуации в матрице R_k соответствует элемент $r_{kj\alpha}$ с индексами: $j=3$, $a=2i-1=1$. r_{k31}

Четвертый элемент в маршрутеM_k – агентA_3, следовательно номер агента i =3, номер позиции агента $j=4$. Выбрано второе значение признака. Данной ситуации в матрице R_k соответствует элемент $r_{kj\beta}$ с индексами: $j=4$, $\beta=2i=6$. r_{k46}

Все определенные ячейки включаются в набор $\Theta_k=\{r_{k13,}r_{k28,}r_{k31,}r_{k46}\}$.

После формирования набора Θ_k, и расчета оценки полезности δ_k этого решения все элементы набора Θ_k в матрице R_k увеличиваются на величину δ_k.

На рис. 2. всем элементам набора Θ_k в матрице R_k присвоено значение δ_k.

Позиция	Признаки							
	A_1		A_2		A_3		A_4	
	Z_{11}	Z_{12}	Z_{21}	Z_{22}	Z_{31}	Z_{32}	Z_{41}	Z_{42}
1			δ_k					
2								δ_k
3	δ_k							
4						δ_k		

Рис. 2. Россыпь альтернатив R_k

3. Построение классификатора (маршрута) в БДР с заданным значением концевой вершины (листа). Работа алгоритма построения классификационной модели базируется на использовании коллективной эволюционной памяти (КЭП), под которой подразумевается любой вид информации, которая отражает прошлую историю развития и хранится независимо от индивидуумов. Алгоритм, связанный с эволюционной памятью, стремится к запоминанию и многократному использованию способов достижения лучших результатов. КЭП алгоритма построение

классификатора представляет собой набор статистических показателей, отражающих для каждого фрагмента решения число μ вхождений фрагмента в лучшие решения на предыдущих итерациях алгоритма и число δ показывающее полезность фрагмента при построении решений на предыдущих итерациях алгоритма. Рассматриваемый алгоритм относится к классу популяционных. Процесс поиска решений итерационный. В процессе поиска эволюционирует популяция решений. Дано: обучающая выборка; множество признаков; бинарные значения признаков; метка класса для концевой вершины. Каждая итерация l включает три этапа.

На первом этапе каждой итерации конструктивным алгоритмом формируется множество решений $Q=\{Q_k|k=1,2,\ldots,n_k\}$. Формирование каждого решения Q_k выполняется конструктивным алгоритмом ***Маршрут***, путем последовательного выбора на каждом шаге признака (агента A_i), одного из двух значений признака $Z_{it}(t=1,2)$ A_i, и позиции s_j, в которую назначается агент A_i. Работа конструктивного алгоритма базируется на базе показателей основной интегральной россыпи альтернатив – матрицы $R=||r_{jl}||_{m\times 2n}$, структура которой описана выше, и в которой хранятся интегральные показатели решений, полученных на предыдущих итерациях. Формирование основной интегральной россыпи альтернатив осуществляется в два этапа с использованием основной R и промежуточной R^*матриц россыпей альтернатив.

После построения очередного решения Q_k его показатели заносятся в промежуточную россыпь альтернатив − матрицу R^*. Для этого рассчитывается оценка ξ_k и оценка полезности δ_k решения Q_k (маршрута M_k). Далее формируется набор Θ_k ячеек для отображения решения Q_k в матрице R^*. После расчета оценки полезности δ_k и формирования Θ_k все элементы набора Θ_k в матрице R^* увеличиваются на величину δ_k. В работе используется циклический метод формирования решений. В этом случае наращивание оценок интегральной полезности δ_k в основной интегральной россыпи альтернатив R выполняется после полного формирования множества решений Q, в результате которого будет сформирована промежуточная матрица россыпи альтернатив R^*.

На втором этапе итерации производится наращивание оценок интегральной полезности δ_k в основной интегральной россыпи альтернатив – матрице R, путем добавления матрицы R^* к матрице R. На третьем этапе итерации осуществляется снижение интегральных оценок полезности δ_k в матрице R на величину δ^*.

Формирование конструктивным алгоритмом решения Q_k (маршрута M_k) производится последовательно, пошагово. Процесс выбора на каждом шаге признака (агента A_i), одного из двух значений признака $Z_{it}(t=1,2)$, и позиции s_j, в которую назначается агент A_i, включает две стадии. На первой стадии выбирается агент и его значение (в маршруте M_k это выбор пары $(x_i,u_{i\text{-}})$), а на второй стадии – позиция s_j. При этом должно выполняться ограничение: каждому агенту множества A соответствует один единственный элемент множества S и наоборот.

На шаге t для каждого, еще не назначенного в позицию, агента $A_i \in A(t)$, путем просмотра двух соответствующих ему столбцов матрицы R (вектора $R1_{k\alpha}=\{r_{kj\alpha}|\alpha=(2i-1);i=1,2,\ldots,n_A\}$) и $R2_{k\beta}=\{r_{kj\beta}|\beta=2i;\ i=1,2,\ldots,n_A\}$) среди свободных позиций $S(t)$ отыскивается позиция $s_j \in S(t)$ с максимальным значением полезности назначения в нее A_i, которое назовем стоимостью $\varepsilon_i(t)$ агента A_i на шаге t. Среди $A_i \in A(t)$ с вероятностью $P_{ik}(t)=\varepsilon_i(t)/\sum_i(\varepsilon_i(t))$, пропорциональной стоимости $\varepsilon_i(t)$ агента A_i,выбирается агент A^*_i. В маршрут M_k в позицию s_j (порядковый номер) заносится пара элементов $(x_i,u_{i\text{-}})$. Элемент $u_{i\text{-}}$ в маршруте M_k является ребром, выходящим из x_i , и соответствует выбранному вместе с A_i значению A_i. Второй индекс элемента $u_{i\text{-}}$ определится после выбора A_i, помещенного в соседнюю с s_j позицию s_{j+1}.

Предварительно формируется структура основной R и промежуточной R^* матриц россыпей альтернатив. Введем обозначения:

$A(t)$ – множество агентов не размещенных в узлах элементов на шаге t;

$S(t)$ – множество свободных позиций шаге t;

k – номер решения;

l – номер итерации.

Задаются параметры:

N_k – объем популяции решений;

N_l – число итераций;

δ – начальное значение оценки полезности.

Алгоритм.

1. Выбор значений параметров N_k, N_l, δ.

2. Всем элементам основной матрицы россыпи альтернатив R присваивается начальное значение оценки полезности δ.

3. $l=1$.

4. Элементы промежуточной матрицы россыпи альтернатив R^* обнуляются.

5. $k=1$.

6. Формирование конструктивным алгоритмом решения Q_k (маршрута M_k) на базе матрицы R.

7. Расчет оценки полезности δ_k решения Q_k.

8.Формирование набора Θ_k ячеек для отображения решения Q_k в матрице $R^*(l)$.

9. Все элементы набора Θ_k в матрице $R^*(l)$ увеличиваются на величину δ_k.

10. Если $(k<n_k)$, то $k=k+1$ и переход к 6, иначе переход к 11.

11. Сложение матриц R и R^*. $R=R+R^*$.

12. Снижение всех интегральных оценок полезности r_{ij} интегральной россыпи альтернатив $R(l)$ на величину δ^*.

13. Если $(l<N_l)$, то $l=l+1$ и переход к 4, иначе переход к 14.

14. Конец работы алгоритма.

4. Экспериментальные исследования. Разработанный алгоритм построения квалификационной модели реализован в виде программы построения дерева решений **БДР-КРА.**

Тестирование программы **БДР-КРА** производилось на контрольных примерах с известным оптимумом $K_{опт}$ [14–21]. Уровень качества полученных решений оценивался по показателю $P=K_{опт}/K$, где K – значение критерия оптимизации, используемого в программе **БДР-КРА.** Число итераций, при котором алгоритм достигал максимального уровня качества не превышает 135.

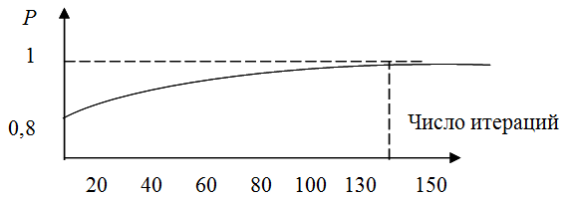

Рис. 3. Зависимость качества решений алгоритма БДР-КРА от числа итераций

Сравнительный анализ с разработанного алгоритма производился с генетическим алгоритмом и алгоритмом роя частиц. Результаты сравнения представлены в табл. 2.

Сравнение алгоритма БДР-КРА по показателю уровень качества с генетическим алгоритмом и алгоритмом роя частиц показало, что при сравнимых временных затратах у алгоритма БДР-КРА показатель P выше в среднем на 9–11 %. В среднем уровень качества, достигнутый алгоритмом БДР-КРА на 135-ой итерации, отличается от максимального значения на 0.12 процента. Общая оценка временной сложности лежит в пределах $O(n^2)$-$O(n^3)$, где n – число признаков.

Таблица 2

Сравнительная оценка работы алгоритмов

Алгоритм / Параметр		**Тест 1**	**Тест 2**	**Тест 3**	**Тест 4**	**Тест 5**
ГА	**P**	332	534	221	189	626
	t	5	12	5	72	27
РЧ	**P**	281	619	268	223	683
	t	4	12	7	78	34
БДР-КРА	**P**	268	502	209	169	611
	t	3	6	5	36	29

Заключение. В ряде случаев возникает необходимость установления соответствия между заявленным и фактическим значением категориальной переменной на основе совокупности признаков объекта. В этом случае возникает потребность в классификаторе с оптимальной последовательностью рассматриваемых атрибутов с заданным значением целевой функции. Значением целевой переменной может быть: да, нет, номер сорта, номер класса и т.д.

В работе решается задача построения классификационной модели в виде оптимальной последовательность рассматриваемых атрибутов и их значений, входящих в состав маршрута от корневой вершины к концевой вершине с заданным значением целевой переменной.

Если требуется классификатор, включающий возможность альтернативных ответов, то вначале строятся независимо друг от друга оптимальные маршруты для каждого значения целевой переменной, а затем эти маршруты объединяются («склеиваются») в единое бинарное дерево решений.

Эффективным направлением повышения качества решений стало использование стохастических **популяционных** алгоритмов, которые, как правило, итерационные и работают в пространстве полных решений. В работе по аналогии с меа-эвристиками, на которых построены роевые алгоритмы, используется метаэвристика, учитывающая тенденцию к использованию альтернатив (вариантов компонентов) из наилучших найденных решений. Особенностями являются наличие непрямого обмена информацией – стигмержи. Совокупность данных об альтернативах и их оценках составляет **россыпь альтернатив.**

Разработан конструктивный алгоритм формирования решения (маршрута M_k). Процесс выбора на каждом шаге признака (агента A_i), одного из двух значений признака, и позиции, в которую назначается агент A_i, включает две стадии. На первой стадии выбирается агент и его значение, а на второй стадии – позиция. При этом должно выполняться ограничение: каждому агенту соответствует один единственный порядковый номер и наоборот.

Предложена эффективная методика расчета оценки полезности решения. Построение маршрута заканчивается, если достигнуто минимальное значение или глубина поиска достигла предельного значения.

В алгоритме построения классификатора на основе метода кристаллизации россыпи альтернатив, каждое решение Q_k интерпретируется в виде в ориентированного маршрута M_k на бинарном дереве решений. Работа алгоритма построения дерева решений базируется на использовании коллективной эволюционной памяти, под которой подразумевается информация, отражающая историю поиска решения. Рассмотрены ключевые моменты анализа альтернатив в процессе эволюционной коллективной адаптации. Экспериментальные исследования показали, что разработанный алгоритм находит решения, не уступающие по качеству, а иногда и превосходящие своих аналогов в среднем на 3–4 %.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Witten Ian H., Frank Eibe, Hall Mark A.* Data Mining: Practical Machine Learning Tools and Techniques. – 3rd Edition. – Morgan Kaufmann, 2011. – 178 p.
2. *Журавлев Ю.И., Рязанов В.В., Сенько О.В.* «Распознавание». Математические методы. Программная система. Практические применения. – М.: Фазис, 2006. – 122 с.
3. *Бериков В.С., Лбов Г.С.* Современные тенденции в кластерном анализе // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы». – 2008. – 126 с.
4. *Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И.* Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004. – 93 с.
5. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 448 с.
6. *Wang X.* Hybrid nature-inspired computation method for optimization: Doctoral Dissertation. Helsinki University of Technology, TKK Dissertations, Espoo 2009. – 161 p.
7. *Лебедев Б.К., Лебедев О.Б., Лебедев В.Б.* Гибридизация роевого интеллекта и генетической эволюции на примере размещения // Программные продукты, системы и алгоритмы. – 2017. – №4.
8. *Лебедев Б.К., Лебедев О.Б.* Гибридный биоинспирированный алгоритм решения задачи символьной регрессии // Известия ЮФУ. Технические науки. – 2015. – № 6 (167). – С. 28-41.
9. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Поисковая адаптация: теория и практика. – М.: Физматлит, 2006. – 288 с.
10. *Лебедев Б.К., Лебедев В.Б.* Оптимизация методом кристаллизации россыпи альтернатив // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 11-17.
11. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.М.* Распределение ресурсов на основе гибридных моделей роевого интеллекта // Научно-технический вестник информационных технологий, механики и оптики. – 2017. – Т. 17, № 6. – С. 1063-1073.
12. *Курейчик В.В., Курейчик Вл.Вл.* Архитектура гибридного поиска при проектировании // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 22-27.
13. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.О.* Решение задачи символьной регрессии методами генетического поиска // Известия ЮФУ. Технические науки. – 2015. – № 2 (163). – С. 212-225.
14. *Kennedy J., Eberhart R.C.* Particle swarm optimization // In Proceedings of IEEE International Conference on Neural Networks. – 1995. – P. 1942-1948.
15. *Clerc M.* Particle Swarm Optimization. – ISTE, London, UK, 2006. – 133 p.
16. *Лебедев Б.К., Лебедев В.Б.* Эволюционная процедура обучения при распознавании образов // Известия ТРТУ. – 2004. – № 8 (43). – С. 83-88.
17. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.О.* Разбиение на классы методом альтернативной коллективной адаптации // Известия ЮФУ. Технические науки. – 2016. – № 7 (180). – С. 89-101.
18. *Cong J., Romesis M., Xie M.* Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms // Proc. of the International Symposium on Physical Design. – Monterey, CA, 2003. – P. 88-94.
19. *Захаров Д.О, Карпенко А.П.* Исследование эффективности популяционного алгоритма лиги чемпионов для задачи глобальной оптимизации // Математика и математическое моделирование. – 2020. – № 2. – С. 25-45.
20. *Карпенко А.П.* Методы повышения эффективности популяционных алгоритмов глобальной оптимизации // Матер. V межрегиональной научно-практической конференции. – Севастополь: Изд-во: Севастопольский государственный университет, 2019. – С. 87-88.
21. *Кучуганов В.Н., Кучуганов А.В., Каксимов Д.Р.* Алгоритм кластеризации множества деталей по чертежам // Программирование. – М.: Изд-во: Российская академия наук, 2020. – С. 29-38.

REFERENCES

1. *Witten Ian H., Frank Eibe, Hall Mark A.* Data Mining: Practical Machine Learning Tools and Techniques. 3rd Edition. Morgan Kaufmann, 2011, 178 p.
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, 2006, 122 p.

3. *Berikov V.S., Lbov G.S.* Sovremennye tendentsii v klasternom analize [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 direction "Information and telecommunication systems], 2008, 126 p.
4. *Barsegyan A.A., Kupriyanov M.S., Stepanenko V.V., Kholod I.I.* Metody i modeli analiza dannykh: OLAP i Data Mining [Methods and models for data analysis: OLAP and Data Mining]. Saint Petersburg: BKhV-Peterburg, 2004, 93 p.
5. *Karpenko A.P.* Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature: textbook]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2014, 448 p.
6. *Wang X.* Hybrid nature-inspired computation method for optimization: Doctoral Dissertation. Helsinki University of Technology, TKK Dissertations, Espoo 2009, 161 p.
7. *Lebedev B.K., Lebedev O.B., Lebedev V.B.* Gibridizatsiya roevogo intellekta i geneticheskoy evolyutsii na primere razmeshcheniya [Hybridization of swarm intelligence and genetic evolution on the example of placement], *Programmnye produkty, sistemy i algoritmy* [Software products, systems and algorithms], 2017, No.4.
8. *Lebedev B.K., Lebedev O.B.* Gibridnyy bioinspirirovannyy algoritm resheniya zadachi simvol'noy regressii [Hybrid bioinspired algorithm for solving the problem of symbolic regression], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 6 (167), pp. 28-41.
9. *Kureychik V.M., Lebedev B.K., Lebedev O.B.* Poiskovaya adaptatsiya: teoriya i praktika [Search engine adaptation: theory and practice]. Moscow: Fizmatlit, 2006, 288 p.
10. *Lebedev B.K., Lebedev V.B.* Optimizatsiya metodom kristallizatsii rossypi al'ternativ [Optimization by the method of crystallization of alternatives field], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 11-17.
11. *Lebedev B.K., Lebedev O.B., Lebedeva E.M.* Raspredelenie resursov na osnove gibridnykh modeley roevogo intellekta [Resource allocation based on hybrid models of swarm intelligence], *Nauchno-tekhnicheskiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki* [Scientific and technical bulletin of information technologies, mechanics and optics], 2017, Vol. 17, No. 6, pp. 1063-1073.
12. *Kureychik V.V., Kureychik Vl.Vl.* Arkhitektura gibridnogo poiska pri proektirovanii [Architecture of hybrid search in design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 22-27.
13. *Lebedev B.K., Lebedev O.B., Lebedeva E.O.* Reshenie zadachi simvol'noy regressii metodami geneticheskogo poiska [The solution of the problem of symbolic regression by methods of genetic search], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 2 (163), pp. 212-225.
14. *Kennedy J., Eberhart R.C.* Particle swarm optimization, *In Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.
15. *Clerc M.* Particle Swarm Optimization. ISTE, London, UK, 2006, 133 p.
16. *Lebedev B.K., Lebedev V.B.* Evolyutsionnaya protsedura obucheniya pri raspoznavanii obrazov [Evolutionary training procedure for pattern recognition], *Izvestiya TRTU* [Izvestiya TSURE], 2004, No. 8 (43), pp. 83-88.
17. *Lebedev B.K., Lebedev O.B., Lebedeva E.O.* Razbienie na klassy metodom al'ternativnoy kollektivnoy adaptatsii [Division into classes by the method of alternative collective adaptation], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2016, No. 7 (180), pp. 89-101.
18. *Cong J., Romesis M., Xie M.* Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms, *Proc. of the International Symposium on Physical Design*. Monterey, CA, 2003, pp. 88-94.
19. *Zakharov D.O, Karpenko A.P.* Issledovanie effektivnosti populyatsionnogo algoritma ligi chempionov dlya zadachi global'noy optimizatsii [Research of the efficiency of the population algorithm of the Champions League for the global optimization problem], *Matematika i matematicheskoe modelirovanie* [Mathematics and Mathematical Modeling], 2020, No. 2, pp. 25-45.

20. *Karpenko A.P.* Metody povysheniya effektivnosti populyatsionnykh algoritmov global'noy optimizatsii [Methods for increasing the efficiency of population algorithms for global optimization], *Mater. V mezhregional'noy nauchno-prakticheskoy konferentsii* [Proceedings of the V interregional scientific and practical conference]. Sevastopol': Izd-vo: Sevastopol'skiy gosudarstvennyy universitet, 2019, pp. 87-88.
21. *Kuchuganov V.N., Kuchuganov A.V., Kaksimov D.R.* Algoritm klasterizatsii mnozhestva detaley po chertezham [Algorithm for clustering a set of parts according to drawings], *Programmirovanie* [Programming]. Moscow: Izd-vo: Rossiyskaya akademiya nauk, 2020, pp. 29-38.

Статью рекомендовал к опубликованию д.т.н., профессор А.Г. Коробейников.

Лебедев Борис Константинович – Южный федеральный университет; e-mail: lebedev.b.k@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 89282897933; кафедра систем автоматизированного проектирования; профессор.

Лебедев Олег Борисович – e-mail: lebedev.ob@mail.ru; тел.: 89085135512; кафедра систем автоматизированного проектирования; доцент.

Лебедев Владимир Борисович – Московский государственный технический университет имени Н.Э. Баумана; e-mail: lebedev.vlad.bor@mail.ru; 105005, г. Москва, ул. Бауманская 2-я, д. 5, стр. 1; тел.: 89287775005; Научно-производственное объединение «Новые технологии»; с.н.с.

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.b.k@gmail.com; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: +79282897933; the department of computer aided design; professor.

Lebedev Oleg Borisovich – e-mail: lebedev.ob@mail.ru; phone: +79085135512; the department of computer aided design; associate professor.

Lebedev Vladimir Borisovich – Moscow State Technical University named after N.E. Bauman; e-mail: lebedev.vlad.bor@mail.ru; 105005, Moscow, st. Baumanskaya 2-nd, 5, build. 1; phone: +79287775005; Research and Production Association "New Technologies"; senior researcher.

УДК 004.041 DOI 10.18522/2311-3103-2020-4-93-107

А.А. Сорокин, И.М. Бородянский, А.В. Дагаев

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ВОССТАНОВЛЕНИЯ ПРОПУЩЕННЫХ ДАННЫХ

В последние десятилетия качественно развиваются методы системного анализа, что связано с увеличением скорости технического развития, уплотнением временных процессов, быстрым ростом накапливаемой информации и новыми возможностями вычислительной техники. К этим методам относятся методы анализа большого объема данных, методы добычи данных, методы аналитического моделирования, методы параллельной обработки данных, нейросетевые методы, методы прогнозирования и другие. Представленные методы позволяют быстро и качественно обрабатывать разнородные кластеры информации, аккумулировать и синтезировать данные, обобщать и классифицировать информацию. К последним из представленных методов относятся методы интерполяции и экстраполяции потерянной, поврежденной или неполученной информации. Данные методы позволяют структурировать, восстанавливать и моделировать информацию на основе статистических данных, математических и алгоритмических методов. Таким образом в статье рассматривается проблема восстановления пропущенных данных в графических и сложных объектах. Приводятся литературные источники по рассматриваемым задачам. В них приводится обширная информация по рассматриваемой тематике: представлены генетические алгоритмы используемые для пространственной интерполяции; рассмотрено решение задач неоднородности интерполяции сейсмических данных; описано использо-