УДК 681.325 DOI 10.18522/2311-3103-2016-7-7488

**Б.К. Лебедев, О.Б. Лебедев, Е.М. Лебедева**

# МУРАВЬИНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ БИНАРНОГО ДЕРЕВА РЕШЕНИЙ*

*Рассматривается задача классификации, заключающаяся в нахождении моделей или функций, которые описывают и различают классы для того, чтобы иметь возможность предсказывать класс произвольного заданного объекта с известными атрибутами, но неизвестной меткой класса. Полученная модель основывается на анализе обучающей выборки, то есть множества объектов, чья метка класса известна. Для решения задач классификации используется метод, основанный на построении дерева решений. Дерево решений (ДР) – это дерево, в котором каждой внутренней вершине поставлен в соответствие некоторый атрибут, каждая ветвь, выходящая из данной вершины, соответствует одному из возможных значений атрибута, а каждому листу дерева сопоставлен конкретный класс или набор вероятностей классов. Для того чтобы классифицировать новый объект, необходимо двигаться по дереву сверху вниз, начиная с корня. При этом на каждом внутреннем узле дерева выбирается та ветвь, которая соответствует фактическому значению соответствующего атрибута. Добравшись до листа дерева, получаем тот класс, которому принадлежит объект согласно классифицирующему правилу. В работе исследуется дихотомические классификационные модели формируемые алгоритмом построения бинарного дерева решений. Каждый узел дерева при разбиении имеет только двух потомков. В работе рассматривается муравьиный алгоритм построения дерева решений, основанный на использовании эффективной оценочной функции для выбора атрибута. Общее правило для выбора атрибута можно сформулировать следующим образом: выбранный атрибут разбивает множество так, чтобы получаемые в итоге подмножества состояли из объектов, принадлежащих к одному классу, или были максимально приближены к этому. В общем случае поиск решения задачи построения ДР осуществляется коллективом муравьев $Z=\{ z_k |k=1,2,...,l\}$. На каждой итерации муравьиного алгоритма каждый муравей $z_k$ строит свое конкретное решение задачи построения ДР, представляемое маршрутом $M_k$ в графе $G=(X,U)$. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае ферромон откладывается агентами на ребрах и вершинах после полного формирования решений. На первом этапе каждой итерации каждый k-й муравей формирует свой собственный маршрут $M_k$. Процесс построения маршрута $M_k$ агентом $z_k$ пошаговый двухтактный. На каждом шаге t агент $z_k$ применяет вероятностные правила выбора следующей вершины (атрибута) и ребра (значения атрибута) для их включения в формируемый маршрут $M_k(t)$. В памяти агента хранится информация: число выполненных шагов - t; маршрут $M_k(t)$ , построенный за t шагов; вершина $x_i$ (атрибут $A_i$) выбранная и включенная в маршрут $M_k(t)$ на шаге t; значение $A^l_i$ атрибута $A_i$ (вершины $x_i$), выбранное на шаге t. На втором этапе все муравьи откладывают феромон. На третьем этапе осуществляется испарение феромона. Поиск решения задачи выполняется на полном ориентированном графе решений $G(X,U)$, где $X=\{x_i \mid i=1,2,...,n\}$ множество вершин соответствует множеству атрибутов A, U – множество бинарных ребер полного графа, соответствующих значениям атрибутов. Каждая пара вершин $(x_i, x_k)$ связана двумя ориентированными бинарными ребрами. Одно бинарное ребро выходит из $x_i$ и входит в $x_k$, другое наоборот выходит из $x_k$ и входит в $x_i$. Для всех k бинарное ребро $u_{ik}$, выходящее из данной вершины $x_i$, используется для моделирования двух значений атрибута $A_i$. Каждое бинарное ребро может находиться в одном из двух состояний, соответствующих двум значениям атрибута. Состояние ребра $u_{ik}$, связывающего пару вершин $(x_i, x_k)$, фиксируется с помощью параметра $V_{ik}$. Если $V_{ik}$ =1, то ребро $u_{ik}$ находится в состоянии соответствующем первому значению $A^1_i$, атрибута $A_i$. Если $V_{ik}$ =2, то ребро $u_{ik}$ находится в состоянии соответствующем второму значению атрибута $A_i$. Если $V_{ik}$ =0, то ребро не входит в состав маршрута. Для учета оценки состояния ребра $u_{ik}$ (количества отложенного феромона), вво-*

* Работа выполнена при финансовой поддержке гранта РФФИ №15-01-05297.

*дится два счетчика $H^1_{ik}$ и $H^2_{ik}$. Решение задачи построения ДР представляется в виде кода – некоторого маршрута $M_k$, включающего вершины и «бинарные» ребра с выбранными состояниями на графе решений G(X,U). Для того чтобы получить ДР нужна процедура декодирования. ДР формируется последовательно в соответствии с построенным маршрутом M, начиная с первоначального ввода безымянной вершины. На каждом шаге t декодирования в маршруте M выбирается очередная пара – вершина $x_i$ и выходящее из нее ребро $u_{ik}$, для которого фиксируется значение параметра $V_{ik}$. Вершине $x_i$ соответствуют атрибут $A_i$ . Если $V_{ik}$ =1, то ребру $u_{ik}$ соответствует первое значение $A^1_i$ атрибута $A_i$. Если $V_{ik}$ =2, то ребру $u_{ik}$ соответствует второе значение $A^2_i$ атрибута $A_i$. Временная сложность этого алгоритма зависит от времени жизни колонии l (число итераций), количества вершин графа n и числа муравьев m, и определяется как $O(l*n^2*m)$.*

*Распознавание образов; классификация; дерево решений; муравьиная колония; граф поиска решений; маршрут.*

**B.K. Lebedev, O.B. Lebedev, E.M. Lebedeva**

## ANT ALGORITHMS BUILD A BINARY DECISION TREE

*The paper deals with the classification task is to find models or features that describe and distinguish classes in order to be able to predict the class of any given object with known attributes, but an unknown class label. The resulting model is based on a training sample analysis, ie a set of objects whose class label is known for. To solve the problems of classification method based on the construction of a decision tree. The decision tree (DR) – the tree in which each internal vertex there corresponds an attribute, each branch coming out of this summit, corresponds to one of the possible values of the attribute, and each tree leaf is associated a specific class or set of classes of probability. To classify a new object, it is necessary to move the tree top down starting at the root. At each internal node of the tree is selected the branch, which corresponds to the actual value of the corresponding attribute. Reaching the tree leaf, we get the class to which the object belongs according to the classifying rule. We study the dichotomous classification model generated by the algorithm of constructing a binary decision tree. Each node of the tree in the division has only two children. This paper considers the ant algorithm for constructing decision tree based on the use of an effective evaluation function to select an attribute. A general rule for selecting an attribute can be summarized as follows: The selected attribute splits the set so that the result obtained in the subset composed of objects belonging to the same class, or were as close as possible to it. In general, the search for solutions to the problem of constructing the DR team made ants $Z = \{z_k \mid k = 1,2, ..., l\}$. At each iteration of the algorithm, each ant $z_k$ ant builds his particular solution to the problem of constructing DT $M_k$ route represented in the graph G = (X, U). We use the cyclic (ant-cycle) method of ant systems. In this case, the agents pheromones deposited on the edges and vertices after the complete formation of solutions. In the first phase of each iteration of each k-th ant creates its own route $M_k$. The process of building the route $M_k$ by agent $z_k$ are incremental push-pull. At each step t the agent $z_k$ uses probabilistic rules for selecting the next vertex (attribute) and edges (the attribute value) to be included in the generated route $M_k$ (t). The memory stores the information agent: number of executed steps – t; $M_k$(t)route, built in t steps; vertex $x_i$ (attribute $A_i$) selected and included in the route $M_k$ (t) at step t; $A^f_i$ value of attribute $A_i$ (vertex $x_i$), selected at step t. At the second stage, all the ants lay pheromone. In the third step, the pheromone evaporation. Search for solution of the problem is carried out on a complete solutions-oriented graph G (X, U), where $X = \{x_i \mid i = 1,2, ..., n\}$ are the set of vertices corresponds to a set of attributes A, U - set of binary edges of a complete graph, the corresponding attribute values. Each pair of vertices ($x_i$, $x_k$) is associated with two binary oriented edges. One binary edge exits from xi and enters to $x_k$, another opposite exits from $x_k$ and enters to $x_i$. For all k the binary edge $u_{ik}$, emanating from this node $x_i$, used to simulate two attribute values $A_i$. Each binary edge may be in one of two states, corresponding to two values of the attribute. Status of edge $u_{ik}$, connecting a pair of vertices ($x_i$, $x_k$), is secured by $V_{ik}$ parameter. If $V_{ik}$ = 1, then $u_{ik}$ edge is at the respective first value $A^1_i$ of attribute $A_i$. If $V_{ik}$ = 2, then $u_{ik}$ edge is at a second value corresponding to attribute $A_i$. If $V_{ik}$ = 0, then the edge is not included in the route. To account for the assessment of the state of the edges $u_{ik}$ (the amount of deferred pheromone), introduced two counters $H^1_{ik}$ and $H^2_{ik}$. The decision task of building the DT is represented as a code - a route $M_k$, including the vertices and "bi-*

*nary" edges with selected states in the decision graph G (X, U). To obtain the required DT decoding procedure. DT is formed sequentially in accordance with the built route M, from the initial input unnamed peaks. At each step t of route M decoding chosen another couple - vertice $x_i$ and edge $u_{ik}$ which coming out of her, for which the fixed value of the parameter $V_{ik}$. If Vik = 1, the edge $u_{ik}$ corresponds to the first value $A^1_i$ of attribute $A_i$. If $V_{ik}$ = 2, the edge $u_{ik}$ corresponds to the second value $A^2_i$ of attribute $A_i$.The time complexity of this algorithm depends on the lifetime of colonies l (number of iterations) and the number n of the graph and the vertices of ants m, and is defined as O ($l * n^2 * m$).*

*Pattern recognition; classification; decision tree; ant colony; making the search graph route.*

**Введение.** Технологии Интеллектуального Анализа Данных (Data Mining) – одна из активно развивающихся областей информационных технологий, предназначенной для выявления полезных знаний из баз данных различной природы [1]. Классификация – это процесс нахождения моделей или функций, которые описывают и различают классы для того, чтобы иметь возможность предсказывать класс произвольного заданного объекта с известными атрибутами, но неизвестной меткой класса. Полученная модель основывается на анализе обучающей выборки, то есть множества объектов, чья метка класса известна [1–4]. Деревья решений – один из методов автоматического анализа данных, успешно применяемый для решения задач классификации и регрессии [5].

Дерево решений – это дерево, в котором каждой внутренней вершине поставлен в соответствие некоторый атрибут, каждая ветвь, выходящая из данной вершины, соответствует одному из возможных значений атрибута, а каждому листу дерева сопоставлен конкретный класс или набор вероятностей классов. Для того, что бы классифицировать новый объект, необходимо двигаться по дереву сверху вниз, начиная с корня. При этом на каждом внутреннем узле дерева выбирается та ветвь, которая соответствует фактическому значению соответствующего атрибута. Добравшись до листа дерева, получаем тот класс, которому принадлежит объект согласно классифицирующему правилу [6–11].

Наибольшее распространение получили дихотомические классификационные модели формируемые алгоритмом построения бинарного дерева решений [8–14]. Каждый узел дерева при разбиении имеет только двух потомков. Рассмотрим пример построения деревьев решений.

Пусть задана обучающая выборка $P=\{p_k \mid k=1,2,\ldots,n_k\}$ – (табл. 1). Каждый пример описывается следующим набором $A=\{A_i|i=1,2,\ldots,n_i\}$ (атрибутов): $A_1$ – Тип автомобиля; $A_2$ – Тип топлива; $A_3$ – КПП; $A_4$- Мощность. Число классов – 2 (1 и 2). Каждый атрибут $A_i$ имеет два значения $A^1_i$, $A^2_i$ (см. табл. 1).

В табл. 1 приведена обучающая выборка, включающая 7 примеров.

Таблица 1

| № | $A_1$. Тип автомобиля | $A_2$. Тип топлива | $A_3$. Коробка передач | $A_4$. Мощность | Класс |
|---|---|---|---|---|---|
| 1 | $A^1_1$. седан | $A^1_2$.дизель | $A^1_3$. механика | $A^1_4$. 200 | 2 |
| 2 | $A^1_1$. седан. | $A^1_2$. дизель | $A^1_3$. механика | $A^2_4$. 150 | 1 |
| 3 | $A^1_1$. седан | $A^1_2$. дизель | $A^2_3$. автомат | $A^2_4$. 150 | 1 |
| 4 | $A^2_1$. кроссовер | $A^1_2$. дизель | $A^2_3$. автомат | $A^2_4$. 150 | 1 |
| 5 | $A^2_1$. кроссовер | $A^2_2$. бензин | $A^2_3$. автомат | $A^2_4$. 150 | 2 |
| 6 | $A^2_1$. кроссовер | $A^1_2$.дизель | $A^2_3$. автомат | $A^1_4$. 200 | 1 |
| 7 | $A^1_1$. седан | $A^2_2$. бензин | $A^1_3$. механика | $A^1_4$.200 | 2 |
| 8 | $A^2_1$. кроссовер | $A^2_2$. бензин | $A^1_3$. механика | $A^2_4$. 150 | ? |

На рис. 1 приведен возможный классификатор, позволяющий осуществлять отнесение объекта к классу 1. В узлах дерева, не являющихся листьями, находятся атрибуты, которые различаются значениями. В листьях находятся значения целевой функции. На рис. 1 имена атрибутов записаны прописными буквами, а альтернативные значения атрибутов записаны курсивом. В листьях находятся значения целевой функции. По ребрам осуществляется спуск, чтобы классифицировать имеющиеся случаи.

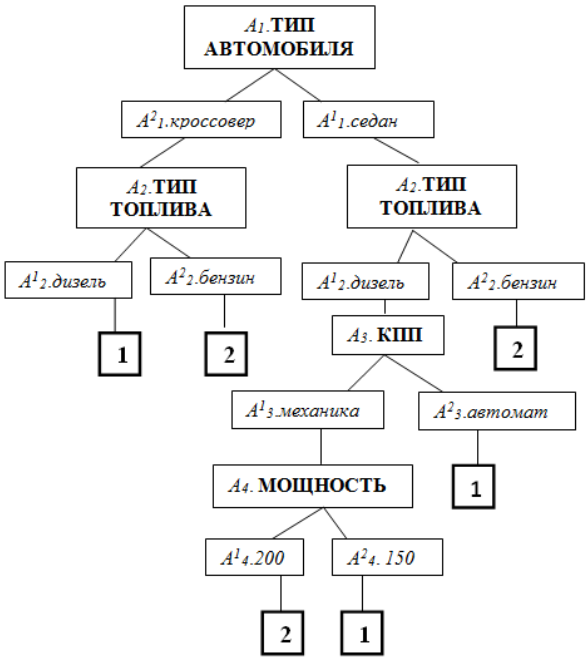


*Рис. 1. Классификатор класса автомобиля*

Использование дерева принятия решений для поиска ответа на интересующий нас вопрос сводится к тому, чтобы пройтись по дереву сверху вниз и определить, в какой из листьев попадает интересующая нас ситуация. Для получения ответа на вопрос – относится предъявленный объект к 1-му классу? – необходимо задать 4 вопроса в заданной последовательности.

Алгоритмом построения бинарного дерева решений решаются задачи классификации и регрессии.

Обучение дерева решений относится к классу обучения с учителем, то есть обучающая и тестовая выборки содержат *классифицированный* набор примеров. Для построения дерева на каждом внутреннем узле необходимо найти такое условие (проверку), которое бы разбивало множество, ассоциированное с этим узлом на подмножества. В качестве такой проверки должен быть выбран один из атрибутов. Общее правило для выбора атрибута можно сформулировать следующим образом: выбранный атрибут должен разбить множество так, чтобы получаемые в итоге подмножества состояли из объектов, принадлежащих к одному классу, или были максимально приближены к этому, т.е. количество объектов из других классов ("примесей") в каждом из этих множеств было как можно меньше [8–15].

Эта процедура будет рекурсивно продолжаться до тех пор, пока конечное множество не будет состоять из примеров, относящихся к одному и тому же классу. Вышеописанная процедура лежит в основе многих современных алгоритмов

построения деревьев решений, этот метод известен еще под названием разделения и захвата (divide and conquer). Очевидно, что при использовании данной методики, построение дерева решений происходит сверху вниз.

Очень часто алгоритмы построения деревьев решений дают сложные деревья, которые "переполнены данными", имеют много узлов и ветвей. Такие "ветвистые" деревья очень трудно понять. К тому же ветвистое дерево, имеющее много узлов, разбивает обучающее множество на все большее количество подмножеств, состоящих из все меньшего количества объектов. Гораздо предпочтительнее иметь дерево, состоящее из малого количества узлов, которым бы соответствовало большое количество объектов из обучающей выборки. И тут возникает вопрос: а не построить ли все возможные варианты деревьев, соответствующие обучающему множеству, и из них выбрать дерево с наименьшей глубиной? К сожалению, это задача является NP-полной, это было показано Л. Хайфилем (L. Hyafill) и Р. Ривестом (R. Rivest), и, как известно, этот класс задач не имеет эффективных методов решения [14].

В течение последних лет были предложены различные подходы к решению проблемы классификации. Большинство из известных алгоритмов являются "жадными алгоритмами". На каждом шаге жадный алгоритм должен выбирать тот атрибут, для которого прирост информации максимален. На сегодняшний день существует значительное число алгоритмов, реализующих деревья решений CART, C4.5, NewId, ITrule, CHAID, CN2 и т.д. [14, 15]. Но наибольшее распространение и популярность получили следующие два: CART, C4.5 [14]. Оценочная функция основана на идее уменьшения неопределенности в узле. Если один раз был выбран атрибут, и по нему было произведено разбиение на подмножества, то алгоритм не может вернуться назад и выбрать другой атрибут, который дал бы лучшее разбиение. И поэтому на этапе построения нельзя сказать даст ли выбранный атрибут, в конечном итоге, оптимальное разбиение. Главным образом это алгоритмы, основанные на эвристиках, обеспечивающих получение приемлемого результата за полиномиальное время. Тем не менее, возросшие сложность решаемых задач и требования к качеству решения делают актуальной разработку новых более эффективных методов.

В работе предлагаются новые технологии решения задачи классификации, использующие математические методы, в которых заложены принципы природных механизмов принятия решений [15–20]. Задача классификации, представляется в виде адаптивной системы на основе интеграции жадной стратегии, использующей иерархический подход с перекрестным анализом и муравьиного подходов к поиску решения.

**Алгоритм построения дерева решений в соответствии со стратегией поиска в глубину, основанный на использовании эффективной оценочной функции для выбора атрибута.** Общее правило для выбора атрибута можно сформулировать следующим образом: выбранный атрибут должен разбить множество так, чтобы получаемые в итоге подмножества состояли из объектов, принадлежащих к одному классу, или были максимально приближены к этому.

Под точностью (распознавания) дерева решений понимается отношение правильно классифицированных объектов при обучении к общему количеству объектов из обучающего множества, а под ошибкой – количество неправильно классифицированных [3, 4].

Множество примеров $T_i$, соответствующих атрибуту $A_i$ разбивается в соответствии с двумя значениями атрибута $A_i$ на два подмножества $T^1_i$ и $T^2_i$ таким образом, что в подмножество $T^1_i \in T_i$ входят примеры с 1-м значением $A^1_i$ атрибута, а в подмножество $T^2_i \in T_i$ входят примеры со 2-м значением $A^2_i$ атрибута, $T^1_i \cup T^2_i = T_i$.

Введем обозначения:

$S_i$ – число примеров в обучающей выборке (ОВ) с признаком (атрибутом) $A_i$;

$S^j_i$ – число примеров в обучающей выборке (ОВ) со значением атрибута $A^j_i$, $j \in \{1,2\}$;

$$S^1_i + S^2_i = S_i. \quad (1)$$

Допустим, есть узел на дереве, соответствующий атрибуту $A_i$. Для нашего примера (см. табл. 1) значения параметров $S_i$ и $S^j_i$ представлены в табл. 2.

Таблица 2

| $S_1$ | $S^1_1$ | $S^2_1$ | $S_2$ | $S^1_2$ | $S^2_2$ | $S_3$ | $S^1_3$ | $S^2_3$ | $S_4$ | $S^1_4$ | $S^2_4$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 4 | 3 | 7 | 5 | 2 | 7 | 3 | 4 | 7 | 3 | 4 |

$S1^j_i$ – число примеров в обучающей выборке со значением $A^j_i$ атрибута, относящихся к классу *1*.

$S2^j_i$ – число примеров в обучающей выборке со значением $A^j_i$ атрибута, относящихся к классу *2*.

$$S1^j_i + S2^j_i = S^j_i. \quad (2)$$

Для нашего примера (см. табл. 1) значения параметров $S1^j_i$ и $S2^j_i$ представлены в табл. 3.

Таблица 3

| $S^1_1$ | | $S^2_1$ | | $S^1_2$ | | $S^2_2$ | | $S^1_3$ | | $S^2_3$ | | $S^1_4$ | | $S^2_4$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S1^1_1$ | $S2^1_1$ | $S1^2_1$ | $S2^2_1$ | $S1^1_2$ | $S2^1_2$ | $S1^2_2$ | $S2^2_2$ | $S1^1_3$ | $S2^1_3$ | $S1^2_3$ | $S2^2_3$ | $S1^1_4$ | $S2^1_4$ | $S1^2_4$ | $S2^2_4$ |
| 2 | 2 | 2 | 1 | 4 | 1 | 0 | 2 | 1 | 2 | 3 | 1 | 1 | 2 | 3 | 1 |

Введем параметры $F^1_i$ и $F^2_i$ – характеризующие степень количественного соотношения объектов разных классов в подмножествах $T^1_i$ и $T^2_i$.

$F^1_i$ – разница между числом $S1^1_i$ примеров подмножества $T^1_i$, относящихся к классу 1, и числом $S2^1_i$ примеров, относящихся к классу 2.

$F^2_i$– разница между числом $S1^2_i$ примеров подмножества $T^2_i$, относящихся к классу 1, и числом $S2^2_i$ примеров относящихся к классу 2.

$$F^1_i = | S1^1_i - S2^1_i |, \quad F^2_i = | S1^2_i - S2^2_i |. \quad (3)$$

Назовем показатели $F^1_i$, $F^2_i$ показателями контрастности значений $A^j_i$ атрибута $A_i$. Введем оценочную функцию

$$F_i = F^1_i + F^2_i . \quad (4)$$

Чем больше значение функции $F_i$, тем в большей степени атрибут $A_i$ контрастен и обладает свойством разбиения на классы. Если для разбиения множества примеров $T$ выбран атрибут $A_i$, то для дальнейшего разбиения (ветвления) выбирается то подмножество $T^1_i$ или $T^2_i$, у которого больше $F^1_i$ или $F^2_i$.

Для нашего примера значения $F^1_i$, $F^2_i$, $F_i$ имеют вид:

Таблица 4

| $F^1_1$ | $F^2_1$ | $\boldsymbol{F_1}$ | $F^1_2$ | $F^2_2$ | $\boldsymbol{F_2}$ | $F^1_3$ | $F^2_3$ | $\boldsymbol{F_3}$ | $F^1_4$ | $F^2_4$ | $\boldsymbol{F_4}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | ***1*** | 3 | 2 | ***5*** | 1 | 2 | ***3*** | 1 | 2 | ***3*** |

В процессе построения дерева путем последовательного разбиения исходного множества примеров в его состав входят “висячие ” соответствующие значениям атрибута вершины, которые могут подвергаться дальнейшему ветвлению и вершины – листы, которые дальнейшему ветвлению не подлежат. Все вершины обладают памятью. В памяти листа хранится информация о целевой функции, т.е. о номере класса, в состав которого входят примеры данной вершины, и о последовательности атрибутов, входящих в состав маршрута от корневой вершины к вершине – листу. В памяти “висячей” вершины хранится информация о последовательности атрибутов, входящих в состав маршрута от корневой вершины к “висячей” и значения параметров.

Выбор атрибута и его значения осуществляется за два такта. На первом такте выбирается атрибут $A_i$, у которого оценочная функция $F_i$ имеет лучшее значение. На втором такте выбирается то значение атрибута,у которого характеристика значения атрибута ($F^1{}_i$ или $F^2{}_i$) имеет большее значение. На первом шаге, на первом такте выбирается и объявляется "висячей" вершина – корень дерева, у которой оценочная функция, имеет лучшее значение. Этой вершине соответствует исходное множество примеров (в нашем случае – 7). Для нашего примера в качестве корневой вершины выбирается атрибут "Тип топлива" для которого оценочная функция максимальна $F_2$=5 (см. табл. 4). Выбранный атрибут "Тип топлива " имеет два значения: "дизель" и "бензин", которым соответствуют два ребра и вводятся две вершины, смежные этим ребрам.

Вершины с однозначным разбиением на классы (в нашем случае это вершина смежная ребру со значением атрибута – "бензин", (рис. 2)) объявляется листом и дальнейшему ветвлению не подлежит. Остальные вершины остаются в ранге "висячих" и являются претендентами для дальнейшего разбиения с помощью еще не использованных атрибутов.

На втором такте для дальнейшего ветвления выбирается вершина, смежная ребру со значением атрибута – "дизель", так как $F^1{}_2$ =3, $F^2{}_2$=2. а $F^1{}_2 > F^2{}_2$.

В табл. 5 приведено множество примеров $T^1{}_2$, соответствующих значению "дизель" атрибута $A_2$.

Таблица 5

| № | $A_1$. Тип автомобиля | $A_2$. Тип топлива | $A_3$. Коробка передач | $A_4$. Мощность | Класс |
|---|---|---|---|---|---|
| 1 | $A^1{}_1$. седан | $A^1{}_2$.дизель | $A^1{}_3$. механика | $A^1{}_4$. 200 | 2 |
| 2 | $A^1{}_1$. седан. | $A^1{}_2$. дизель | $A^1{}_3$. механика | $A^2{}_4$. 150 | 1 |
| 3 | $A^1{}_1$. седан | $A^1{}_2$. дизель | $A^2{}_3$. автомат | $A^2{}_4$. 150 | 1 |
| 4 | $A^2{}_1$. кроссовер | $A^1{}_2$. дизель | $A^2{}_3$. автомат | $A^2{}_4$. 150 | 1 |
| 6 | $A^2{}_1$. кроссовер | $A^1{}_2$.дизель | $A^2{}_3$. автомат | $A^1{}_4$. 200 | 1 |

Просматриваются характеристики (параметры – $F_i$ , $F^1{}_i$, $F^2{}_i$) всех не подвергавшихся ветвлению вершин (атрибутов) (см. табл. 5).

Рассчитаем значения параметров $S_i$, $S^j{}_i$, $S1^j{}_i$, $S2^j{}_i$, $F1_i$, $F^2{}_i$ , $F_i$ – для множества примеров приведенных в табл. 5. $S_1$= $S_2$= $S_3$= $S_4$=5. (см. табл. 6-8).

Таблица 6

| $S_1$ | $S^1{}_1$ | $S^2{}_1$ | $S_3$ | $S^1{}_3$ | $S^2{}_3$ | $S_4$ | $S^1{}_4$ | $S^2{}_4$ |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 2 | 5 | 2 | 3 | 5 | 2 | 3 |

Таблица 7

| $S^1{}_1$ | | $S^2{}_1$ | | $S^1{}_3$ | | $S^2{}_3$ | | $S^1{}_4$ | | $S^2{}_4$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $S1^1{}_1$ | $S2^1{}_1$ | $S1^2{}_1$ | $S2^2{}_1$ | $S1^1{}_3$ | $S2^1{}_3$ | $S1^2{}_3$ | $S2^2{}_3$ | $S1^1{}_4$ | $S2^1{}_4$ | $S1^2{}_4$ | $S2^2{}_2$ |
| 2 | 1 | 1 | 1 | 1 | 1 | 3 | 0 | 1 | 1 | 3 | 0 |

Таблица 8

| $F^1{}_1$ | $F^2{}_1$ | $\boldsymbol{F_1}$ | $F^1{}_3$ | $F^2{}_3$ | $\boldsymbol{F_3}$ | $F^1{}_4$ | $F^2{}_4$ | $\boldsymbol{F_4}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | ***1*** | 0 | 3 | ***3*** | 0 | 3 | ***3*** |

На первом такте второго шага для дальнейшего ветвления среди атрибутов "Тип автомобиля", "КПП" и "Мощность" выбирается атрибут "Мощность", у которого оценочная функция в случае его применения к рассматриваемой "висячей" вершине, имеет лучшее значение, $F_4$=3. Выбранный атрибут "Мощность" имеет два значения: "200" и "150", которым соответствуют два ребра и вводятся две вершины, смежные этим ребрам. Вершины с однозначным разбиением на классы (в нашем

случае это вершина смежная ребру со значением атрибута – “150”, (рис. 2)) объявляется листом и дальнейшему ветвлению не подлежит.

На втором такте второго шага для дальнейшего ветвления выбирается вершина, смежная ребру “200”, которая остается в ранге “висячих” и является претендентом для дальнейшего разбиения. В табл. 9 приведено множество примеров $T^1_4$, соответствующих значению “200” атрибута $A_4$..

Таблица 9

| № | $A_1$. Тип автомобиля | $A_2$. Тип топлива | $A_3$. Коробка передач | $A_4$. Мощность | Класс |
|---|---|---|---|---|---|
| 1 | $A^1_1$. седан | $A^1_2$.дизель | $A^1_3$. механика | $A^1_4$. 200 | 2 |
| 6 | $A^2_1$. кроссовер | $A^1_2$.дизель | $A^2_3$. автомат | $A^1_4$. 200 | 1 |

Рассчитаем значения параметров $S_i$, $S^j_i$, $S1^j_i$, $S2^j_i$, $F_{1i}$, $F_{2i}$ , $F_i$ – для множества примеров приведенных в табл. 8. $S_1$= $S_2$= $S_3$= $S_4$=2. (см. табл. 10–12).

Таблица 10

| $S_1$ | $S^1_1$ | $S^2_1$ | $S_3$ | $S^1_3$ | $S^2_3$ |
|---|---|---|---|---|---|
| 2 | 1 | 1 | 2 | 1 | 1 |

Таблица 11

| $S^1_1$ | | $S^2_1$ | | $S^1_3$ | | $S^2_3$ | |
|---|---|---|---|---|---|---|---|
| $S1^1_1$ | $S2^1_1$ | $S1^2_1$ | $S2^2_1$ | $S1^1_3$ | $S2^1_3$ | $S1^2_3$ | $S2^2_3$ |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

Таблица 12

| $F_{11}$ | $F_{21}$ | $\boldsymbol{F_1}$ | $F_{13}$ | $F_{23}$ | $\boldsymbol{F_3}$ |
|---|---|---|---|---|---|
| 1 | 1 | ***2*** | 1 | 1 | ***2*** |

На третьем шаге в качестве выбранной для дальнейшего ветвления висячей вершины среди атрибутов “Тип автомобиля” и “КПП” выбирается атрибут “Тип автомобиля”, у которого оценочная функция в случае его применения к рассматриваемой “висячей” вершине, имеет лучшее значение, $F_1$=2.

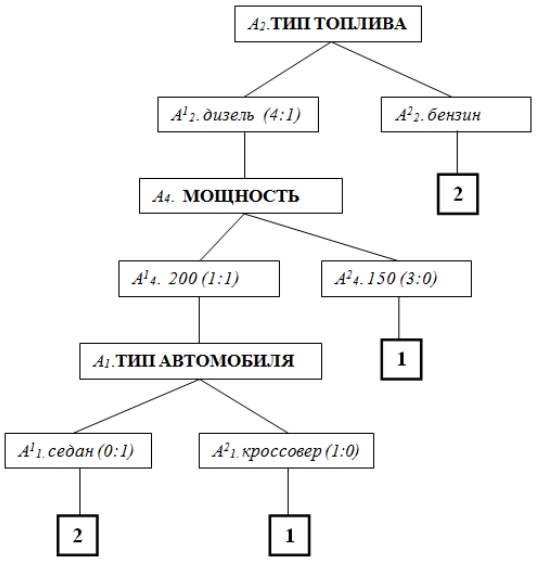


*Рис. 2. Оптимальный классификатор класса автомобиля*

Выбранный атрибут “Тип автомобиля” имеет два значения: “седан” и “ кроссовер”, которым соответствуют два ребра и вводятся две вершины, смежные этим ребрам. Поскольку эти вершины имеют однозначное разбиение на классы, каждая из них объявляется листом и дальнейшему ветвлению не подлежит. Окончательно построенное дерево решений представлено на рис. 2.

**Муравьиный алгоритм построения дерева решений.** Для того чтобы построить муравьиный алгоритм для решения какой-либо задачи, нужно представить задачу в виде набора компонент: в первую очередь сформировать граф поиска решений и определить эвристику поведения муравья [16–20].

Поиск решения задачи выполняется на полном ориентированном графе решений $G(X,U)$, где $X=\{x_i \mid i=1,2,\ldots,n\}$ множество вершин соответствует множеству атрибутов $A$, $U$ – множество бинарных ребер полного графа, соответствующих значениям атрибутов (рис. 3). Каждая пара вершин $(x_i, x_k)$ связана двумя ориентированными бинарными ребрами. Одно бинарное ребро выходит из $x_i$ и входит в $x_k$, другое наоборот выходит из $x_k$ и входит в $x_i$. Для всех $k$ бинарное ребро $u_{ik}$, выходящее из данной вершины $x_i$, используется для моделирования двух значений атрибута $A_i$. Каждое бинарное ребро может находиться в одном из двух состояний, соответствующих двум значениям атрибута. Состояние ребра $u_{ik}$, связывающего пару вершин $(x_i, x_k)$, фиксируется с помощью параметра $V_{ik}$. Если $V_{ik}$ =1, то ребро $u_{ik}$ находится в состоянии соответствующем первому значению $A^1_i$, атрибута $A_i$. Если $V_{ik}$ =2, то ребро $u_{ik}$ находится в состоянии соответствующем второму значению атрибута $A_i$. Если $V_{ik}$ =0, то ребро не входит в состав маршрута. Для учета оценки состояния ребра $u_{ik}$ (количества отложенного феромона), вводится два счетчика $H^1_{ik}$ и $H^2_{ik}$. Решение задачи построения ДР представляется в виде кода –некоторого маршрута $M_k$, включающего вершины и «бинарные» ребра с выбранными состояниями на графе решений $G(X,U)$.

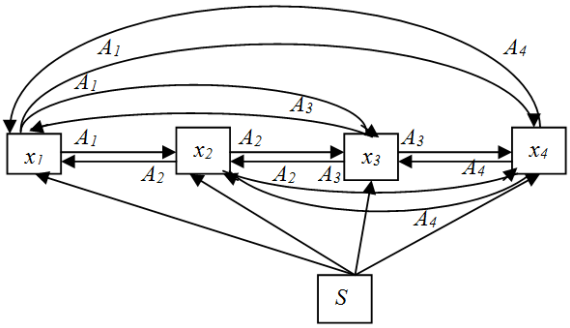


*Рис. 3. Граф поиска решений*

Для того чтобы получить дерево принятия решений нужна процедура декодирования. Дерево принятия решений формируется последовательно в соответствии с построенным маршрутом $M$, начиная с первоначального ввода безымянной вершины. На каждом шаге $t$ декодирования в маршруте $M$ выбирается очередная пара – вершина $x_i$ и выходящее из нее ребро $u_{ik}$, для которого фиксируется значение параметра $V_{ik}$. Вершине $x_i$ соответствуют атрибут $A_i$ . Если $V_{ik}$ =1, то ребру $u_{ik}$ соответствует первое значение $A^1_i$ атрибута $A_i$. Если $V_{ik}$ =2, то ребру $u_{ik}$ соответствует второе значение $A^2_i$ атрибута $A_i$.

**Пример.** Пусть на графе $G(X,U)$ (рис. 3) в соответствии с предъявленным набором атрибутов объекта построен для классификации объекта маршрут $M= x_2, u_{24}; x_4, u_{41}; x_1, u_{12}; L$. Маршрут завершается листом – $L$. При этом были определены параметры: $V_{24}$=1, $V_{41}$=1, $V_{12}$=2.На первом шаге выбирается пара $x_2, u_{24}$; и определяется, что $V_{24}$=1. Вершине $x_2$ соответствует атрибут $A_2$, имеющий два значения $A^1_2$ и

$A^2_2$. Ребру $u_{24}$ со значением параметра $V_{24}$=1, соответствует значение атрибута $A^1_2$. Дальнейшему ветвлению подвергается ребро, соответствующее значению атрибута $A^1_2$. Участок ДР после учета первой пары имеет вид, представленный на рис. 4.

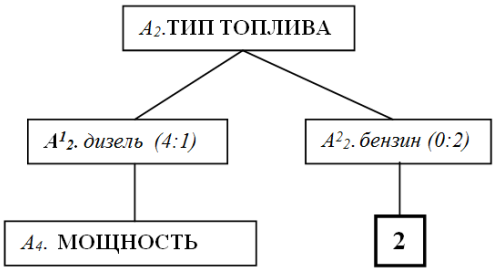


*Рис. 4. Участок ДР после учета первой пары*

На втором шаге выбирается пара $x_4$, $u_{41}$; и определяется, что $V_{41}$=1. Вершине $x_4$ соответствует атрибут $A_4$, имеющий два значения $A^1_4$ и $A^2_4$. Ребру $u_{41}$ со значением параметра $V_{41}$=1, соответствует значение атрибута $A^1_4$. Дальнейшему ветвлению подвергается ребро, соответствующее значению атрибута $A^1_4$. Участок ДР после учета второй пары имеет вид, представленный на рис. 5.

На третьем шаге выбирается пара $x_1$, $u_{12}$; и определяется, что $V_{12}$=2. Вершине $x_1$ соответствует атрибут $A_1$, имеющий два значения $A^1_1$ и $A^2_1$. Ребру $u_{12}$ со значением параметра $V_{12}$=1, соответствует значение атрибута $A^2_1$. Дальнейшему ветвлению подвергается ребро, соответствующее значению атрибута $A^1_4$. ДР после учета третьей пары имеет вид, представленный на рис. 2. Поскольку "висячии" вершины отсутствуют, процесс ветвления завершается.

**Процесс построения дерева.** В общем случае поиск решения задачи построения ДР осуществляется коллективом муравьев $Z=\{ z_k | k=1,2,...,l\}$. На каждой итерации муравьиного алгоритма каждый муравей $z_k$ строит свое конкретное решение задачи построения ДР, представляемое маршрутом $M_k$ в графе $G=(X,U)$ [16]. В работе используется циклический (ant-cycle) метод муравьиных систем [17]. В этом случае ферромон откладывается агентами на ребрах и вершинах после полного формирования решений. На первом этапе каждой итерации каждый $k$-й муравей формирует свой собственный маршрут $M_k$. На втором этапе все муравьи откладывают феромон. На третьем этапе осуществляется испарение феромона.

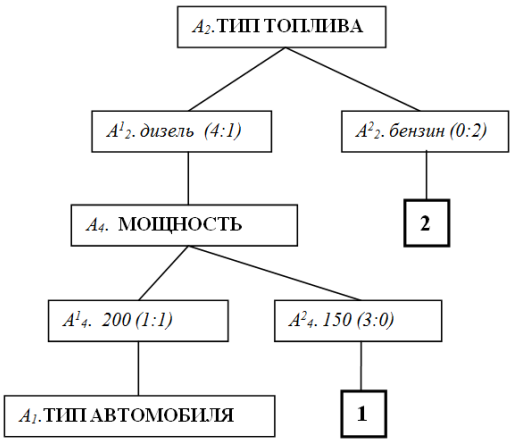


*Рис. 5. Участок ДР после учета второй пары*

Для равномерного распределения муравьев и создания равных стартовых условий при выборе начальных вершин у формируемых муравьями маршрутов, используется стартовая вершина *S,* связанная ориентированными ребрами со всеми вершинами множества *X* (см. рис. 3). Каждый муравей начинает построение маршрута с вершины *S.*

В качестве исходных данных служит обучающая выборка (см. табл. 1).

Процесс построения маршрута Mk агентом zk пошаговый двухтактный. На каждом шаге t агент zk применяет вероятностные правила выбора следующей вершины (атрибута) и ребра (значения атрибута) для их включения в формируемый маршрут Mk(t). В памяти агента хранится информация: число выполненных шагов – t; маршрут Mk(t) , построенный за t шагов; вершина xi (атрибут Ai) выбранная и включенная в маршрут Mk(t) на шаге t; значение Afi атрибута Ai (вершины xi), выбранное на шаге t. На первом такте шага (t+1) формируется множество Xk(t+1)⊂ X таких вершин xj∈ Xk(t+1), которые еще не вошли в состав строящегося маршрута и каждая из них может быть добавлена в формируемый маршрут. Другими словами множество Xk(t+1) соответствует множеству A(t+1) атрибутов еще не входивших в состав строящегося дерева решений. Агент просматривает все вершины xj∈Xk(t+1). Для каждой вершины xj∈Xk(t+1) рассчитывается параметр cjk – суммарный уровень размещенного на ней феромона. В соответствии с выражениями (1-4) для вершины $x_j \in X_k(t+1)$ рассчитываются показатели качества $F_{1jk}(t+1)$, $F_{2jk}(t+1)$, $F_{jk}(t+1)$,

Определяется стоимость $h_{jk}$ вершины $x_j \in X_k(t+1)$.

$$h_{jk}= \alpha c_{jk}+ \beta F_{jk}(t+1),$$

где $\alpha$, $\beta$ – управляющие параметры, которые подбираются экспериментально.

Вероятность $P_{jk}$ включения вершины $x_j \in X_k(t+1)$ в формируемый маршрут $M_k(t+1)$ определяется следующим соотношением

$$P_{jk}=h_{jk} / \sum_j (h_{jk}). \quad (5)$$

Агент с вероятностью $P_{jk}$ выбирает одну из вершин $x_j$, которая включается в маршрут $M_k(t+1)$.

После выбора (на шаге $t+1$) вершины (атрибута) $x_j$ в маршрут $M_k(t+1)$ включается бинарное ребро $u_{ij}$, связывающее выбранную на шаге $t$ вершину $x_i$ с вершиной $x_j$. Параметру $V_{ij}$ бинарного ребра $u_{ij}$ присваивается номер варианта выбранного значения атрибута $A^f_i$.

На втором такте шага $t+1$ осуществляется выбор значения $A^f_j$ атрибута (вершины $x_j$). Для каждого значения $A^f_i$ ($f$=1,2)выбранного атрибута $x_j$ определяется его стоимость

$$g_{1jk} =\mu d_{1jk}+\gamma F_{1jk}(t+1),$$
$$g_{2jk} =\mu d_{2jk}+\gamma F_{2jk}(t+1), \quad (6)$$

где μ, γ – управляющие параметры, которые подбираются экспериментально,

$d_{1jk}$ – количество феромона, накопленного в счетчике $H_{ik1}$,

$d_{2jk}$ – количество феромона, накопленного в счетчике $H_{ik2}$.

Вероятность $P_{1jk}$ включения в маршрут 1-го значения выбранного атрибута

$$P_{1jk}= g_{1jk} / g_{1jk} + g_{2jk}.$$

Вероятность $P_{2ik}$ включения в маршрут 2-го значения выбранного атрибута

$$P_{2jk}= g_{2jk} / g_{1ik} + g_{2jk}.$$
$$P_{1jk} + P_{2jk}=1.$$

Агент в соответствии с заданным распределением с вероятностей выбирает значение $A^f_j$ атрибута $A_j$.

Процесс построения агентом $z_k$ маршрута $M_k$ завершается, если после разбиения с помощью последнего в маршруте выбранного атрибута получаемые в итоге подмножества состоят из объектов, принадлежащих к одному классу, или максимально приближены к этому.

Целевая функция имеет вид

$$G_k(l)=k_1L_S + k_2D_S, \quad (7)$$

где $k_1$, $k_2$ – коэффициенты важности; $L_M$ –число вершин, входящих в маршрут $M_k$; $D_M$ – число ребер в дереве решений.

На втором этапе итерации, каждый муравей $z_k$ откладывает феромон на соответствующих вершинах и рёбрах построенного маршрута.

Количество феромона $\tau_k(l)$, откладываемое муравьем $z_k$ на каждой вершине и ребре построенного маршрута $M_k$, определяется следующим образом:

$$\delta\ \tau_k(l)= \delta\ Q / G_k(l), \quad (8)$$

где $l$ – номер итерации, $\delta$ – коэффициент отложения, $Q_i$ – базовое количество феромона, откладываемое муравьем на ребрах маршрута $M_k$, $G_k(l)$ – целевая функция для решения, полученного муравьем $z_k$ на $l$-й итерации. Чем меньше $G_k(l)$, тем больше феромона откладывается на ребрах построенного маршрута и, следовательно, тем больше вероятность выбора этих ребер при построении маршрутов на следующей итерации.

Сначала феромон откладывается на всех вершинах построенного маршрута $M_k$. Затем последовательно просматриваются, начиная с первой, вершины маршрута. Пусть рассматривается вершина $x_i$ (атрибут $A_i$). Определяется значение $A^f_i$ атрибута $A_i$. Количество феромона в счетчике $H_{fik}$, соответствующем значению $A^f_i$ атрибута $A_i$ каждого бинарного ребра, смежного вершине $x_i$ увеличивается на величину $\tau_k(l)$.

После того, как каждый агент сформировал решение и отложил феромон на графе поиска решений, на третьем этапе происходит общее испарение феромона на ребрах и вершинах полного графа $G$ в соответствии с нижеприведенной формулой.

$$\delta_{ik} = \delta_{ik}(1 - \rho), \quad (9)$$

где $\rho$ – коэффициент обновления.

После выполнения всех действий на итерации находится агент с лучшим решением, которое запоминается. Далее осуществляется переход на следующую итерацию.

Временная сложность этого алгоритма зависит от времени жизни колонии $l$ (число итераций), количества вершин графа $n$ и числа муравьев $m$, и определяется как $O(l*n^2*m)$.

Алгоритм построения ДР на основе метода муравьиной колонии формулируется следующим образом [16–20].

1. В соответствии с исходными данными формируется граф поиска решений $G$.
2. Определяются число агентов и вершины, в которые они помещаются.
3. Задается значение параметра $Q_i$и число итераций – $N_l$.
4. На всех вершинах и бинарных ребрах графа $G$ откладывается начальное количество феромона. $l=1$.
5. На первом этапе $l$-й итерации на ГПР $G$ каждым агентом $z_k$ находится маршрут $M_k(l)$.
6. Для каждого маршрута $M_k(l)$, строится соответствующее дерево решений.
7. Для каждого решения задачи построения дерева решений находится значение целевой функции $F_k(l)$.
8. На вершинах и ребрах каждого найденного маршрута $M_k(l)$ в графе $G$ откладывается феромон. Количество феромона, откладываемого каждым агентом, пропорционально $F_k(l)$.
9. Выполняется процедура испарения феромона на ребрах графа $G$.

10. Выбор лучшего решения, полученного на протяжении всех выполненных итераций.
11. Если все итерации выполнены, то конец работы алгоритма, в противном случае, переход к пункту 5 для выполнения очередной итерации.

**Заключение**. На основе сравнительного анализа существующих подходов и методов для решения задачи классификации использованы мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании адаптивного поведения муравьиной колонии. Разработана модель пространства поиска решений в виде полного ориентированного графа поиска решений. Отличительная особенность разработанной модели заключается в том, что в графе поиска решений использованы бинарные ребра и введена функция состояния ребра. Это позволяет осуществлять комбинирование набором, направлением, выбранным состоянием ребра при построении маршрута на графе поиска решений. Интеграция эвристик с механизмами муравьиной колонии позволило синтезировать эффективную оценочную функцию, основанную на идее уменьшения неопределенности в узле. В работе предлагается муравьиный алгоритм построения дерева решений в соответствии со стратегией поиска в глубину, основанный на использовании эффективной оценочной функции для выбора атрибута.

Для анализа точности получаемых решений был синтезирован ряд примеров с априори известным оптимальным значением целевой функции. Исследованию подвергались примеры, у которых обучающая выборка содержала до 1000 примеров. Сравнение с известными алгоритмам [14] показало, что при меньшем времени работы у полученных с помощью разработанного алгоритма решений значения целевой функции лучше (меньше) в среднем на 6 %. Вероятность получения оптимального решения составила 0.9.

Временная сложность алгоритма (ВСА), полученная экспериментальным путем, лежит в пределах $O(n^2)$-$O(n^3)$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК


1. *Han J., Kamber M.* Data mining: Concepts and Techniques. – Morgan Kaufmann Publishers, 2001.
2. *Ian H. Witten, Eibe Frank and Mark A.* Hall Data Mining: Practical Machine Learning Tools and Techniques. – 3rd Edition. – Morgan Kaufmann, 2011.
3. *Журавлев Ю.И., Рязанов В.В., Сенько О.В.* Распознавание. Математические методы. Программная система. Практические применения. – М.: Фазис, 2006. – 159 с.
4. *Шлезингер М., Главач В.* Десять лекций по статистическому и структурному распознаванию. – Киев: Наукова думка, 2004. – 545 с.
5. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning. – Springer, 2001. COMPACT - Comparative Package for Clustering Assessment. A free Matlab package, 2006.
6. *Berkhin P.* Survey of Clustering Data Mining Techniques, Accrue Software, 2002.
7. *Радченко С.Г.* Методология регрессионного анализа: монография. – К.: Корнийчук, 2011. – 376 с.
8. *Бериков В.С., Лбов Г.С.* Современные тенденции в кластерном анализе // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы», 2008. – 26 с.
9. *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. – 336 с.
10. *Лебедев Б.К., Лебедев В.Б.* Эволюционная процедура обучения при распознавании образов // Известия ТРТУ. – 2004. – № 8 (43). – С. 83-88.
11. *Konar A.* Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain. – CRC Press LLC. – Boca Raton,Florida, 2000.
12. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Поисковая адаптация: теория и практика. – М.: Физматлит, 2006. – 272 с.

13. *Лебедев Б.К., Лебедев О.Б.* Моделирование адаптивного поведения муравьиной колонии при поиске решений, интерпретируемых деревьями // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 27-35.
14. *Курейчик В.В., Курейчик В.М., Гладков Л.А., Сороколетов П.В.* Бионспирированные методы в оптимизации. – М.: Физмалит, 2009. – 384 с.
15. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Разбиение на основе моделирования адаптивного поведения биологических систем // Нейрокомпьютеры: разработка, применение. – 2010. – № 2. – С. 28-34.
16. *Лебедев В.Б., Лебедев О.Б.* Роевой интеллект на основе интеграции моделей адаптивного поведения муравьиной и пчелиной колоний // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 41-47.
17. *Лебедев О.Б.* Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог: Изд-во ЮФУ, 2013.
18. *Dorigo M. and Stützle T.* Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
19. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.М.* Решение однородной распределительной задачи на основе моделей адаптивного поведения муравьиной колонии // Вестник РГУПС. – 2016. – № 2 (62). – С. 71-77.
20. *Engelbrecht A.P.* Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.

REFERENCES

1. *Han J., Kamber M.* Data mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.
2. *Ian H. Witten, Eibe Frank and Mark A.* Hall Data Mining: Practical Machine Learning Tools and Techniques. 3rd Edition. Morgan Kaufmann, 2011.
3. *Zhuravlev Yu.I., Ryazanov V.V., Sen'ko O.V.* Raspoznavanie. Matematicheskie metody. Programmnaya sistema. Prakticheskie primeneniya [Recognition. Mathematical methods. Software system. Practical application]. Moscow: Fazis, 2006, 159 p.
4. *Shlezinger M., Glavach V.* Desyat' lektsiy po statisticheskomu i strukturnomu raspoznavaniyu [Ten lectures on statistical and structural recognition]. Kiev: Naukova dumka, 2004, 545 p.
5. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning. Springer, 2001. COMPACT - Comparative Package for Clustering Assessment. A free Matlab package, 2006.
6. *Berkhin P.* Survey of Clustering Data Mining Techniques, Accrue Software, 2002.
7. *Radchenko S.G.* Metodologiya regressionnogo analiza: monografiya [Methodology regression analysis: monograph]. Kiev: Korniychuk, 2011, 376 p.
8. *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», 2008* [all-Russian competitive selection of survey and analytical articles on priority direction "Information-telecommunication systems", 2008], 26 p.
9. *Barsegyan A.A., Kupriyanov M.S., Stepanenko V.V., Kholod I.I.* Metody i modeli analiza dannykh: OLAP i Data Mining [Methods and models of data analysis: OLAP and Data Mining]. St. Petersburg: BKhV-Peterburg, 2004, 336 p.
10. *Lebedev B.K., Lebedev V.B.* Evolyutsionnaya protsedura obucheniya pri raspoznavanii obrazov [The evolutionary procedure learning for image recognition], *Izvestiya TRTU* [Izvestiya TSURE], 2004, No. 8 (43), pp. 83-88.
11. *Konar A.* Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain. CRC Press LLC. Boca Raton, Florida, 2000.
12. *Kureychik V.M., Lebedev B.K., Lebedev O.B.* Poiskovaya adaptatsiya: teoriya i praktika [Search adaptation: theory and practice]. Moscow: Fizmatlit, 2006, 272 p.
13. *Lebedev B.K., Lebedev O.B.* Modelirovanie adaptivnogo povedeniya murav'inoy kolonii pri poiske resheniy, interpretiruemykh derev'yami [Modelling of an ant colony adaptive behaviour by search of the decisions interpreted by trees], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 27-35.
14. *Kureychik V.V., Kureychik V.M., Gladkov L.A., Sorokoletov P.V.* Bionspirirovannye metody v optimizatsii [Inspirowanie methods in optimization]. Moscow: Fizmalit, 2009, 384 p.

15. *Kureychik V.M., Lebedev B.K., Lebedev O.B.* Razbienie na osnove modelirovaniya adaptivnogo povedeniya biologicheskikh sistem [Partitioning based on simulation of adaptive behavior of biological systems], *Neyrokomp'yutery: razrabotka, primenenie* [Neurocomputers: development, application], 2010, No. 2, pp. 28-34.
16. *Lebedev V.B., Lebedev O.B.* Roevoy intellekt na osnove integratsii modeley adaptivnogo povedeniya murav'inoy i pchelinoy koloniy [Swarm intelligence on the basis of the adaptive behaviour models integration of the ant and beer colonies], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 41-47.
17. *Lebedev O.B.* Modeli adaptivnogo povedeniya murav'inoy kolonii v zadachakh proektirovaniya [Models of adaptive behaviour of ant colony in task design]. Taganrog: Izd-vo YuFU, 2013.
18. *Dorigo M. and Stützle T.* Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
19. *Lebedev B.K., Lebedev O.B., Lebedeva E.M.* Reshenie odnorodnoy raspredelitel'noy zadachi na osnove modeley adaptivnogo povedeniya murav'inoy kolonii [The solution of the homogeneous distribution of tasks based on models of adaptive behavior ant colony], *Vestnik RGUPS* [Bulletin of the Rostov state transport University], 2016, No. 2 (62), pp. 71-77.
20. *Engelbrecht A.P.* Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.

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.

**Лебедева Елена Михайловна** – e-mail: lebedev.ob@mail.ru; тел.: 89081702418; кафедра систем автоматизированного проектирования; аспирант.

**Lebedev Boris Konstantinovich** – Southern Federal University; e-mail: lebedev.b.k@gmail.com; 44, Nekrasovskiy, 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.

**Lebedeva Elena Mikhaylovna** – e-mail: lebedev.ob@mail.ru; phone: +79081702418; the department of computer aided design; postgraduate student.