УДК 621.3.049.771.14

Б.К. Лебедев, О.Б. Лебедев

ГИБРИДНЫЙ БИОИНСПИРИРОВАННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ СИМВОЛЬНОЙ РЕГРЕССИИ*

Рассматривается задача символьной регрессии, заключающаяся в нахождении математического выражения в символьной форме, аппроксимирующего зависимость между конечным набором значений независимых переменных и соответствующими значениями зависимых переменных. Критерием качества приближения (целевой функции) является среднеквадратичная ошибка: сумма квадратов разности значений модели и зависимой переменной для всех значений независимой переменной в качестве аргумента. Символьная регрессия – метод построения регрессионных моделей путем перебора различных произвольных суперпозиций функций из некоторого заданного набора. Предлагается гибридный алгоритм для решения задачи символьной регрессии. Используется традиционное представление алгебраической формулы в виде синтаксического дерева. Листовые узлы соответствуют переменным или числовым константам, а не листовые узлы содержат операцию, которая выполняется над дочерними узлами. В процессе синтеза алгебраической формулы решаются две задачи. Первая задача заключается в построении структуры дерева с безымянными вершинами. Вторая задача заключается в конкретизации значений вершин дерева. Листовые узлы сопоставляются с терминальным множеством, а нелистовые узлы сопоставляются с функциональным множеством. Первая задача решается методами муравьиной колонии. Для решения второй задачи используется генетический алгоритм. Оценка формулы вычисляется после решения обеих задач – построения муравьем дерева с безымянными вершинами и последующей идентификацией вершин с помощью генетического алгоритма Разработана структура графа поиска решений $G=(X,U)$. При больших размерностях временные показатели разработанного алгоритма превосходят показатели сравниваемых алгоритмов при лучших значениях целевой функции. Экспериментальная временная сложность алгоритма на одной итерации при фиксированных значениях управляющих параметров составляет $O(n\lg n)$, где n – мощность терминального множества.

Символьная регрессия; синтаксическое дерево; терминальное множество; функциональное множество; муравьиная колония; генетический поиск; гибридный алгоритм.

B.K. Lebedev, O.B. Lebedev

HYBRID BIOINSPIRED ALGORITHM FOR SOLVING SYMBOLIC REGRESSION PROBLEM

The problem of symbolic regression is to find mathematical expressions in symbolic form, approximating the relationship between the finite set of values of the independent variables and the corresponding values of the dependent variables. The criterion of quality approach (objective function) is the mean square error: the sum of the squares of the difference between the model and the values of the dependent variable for all values of the independent variable as an argument. Symbolic regression – method of constructing regression models by trying different superpositions of arbitrary functions from a given set. The paper is offered hybrid algorithm for solving symbolic regression. Use the traditional idea of an algebraic formula in the form of syntax tree. Leaf nodes correspond to variables or numeric constants rather than leaf nodes contain the operation that is performed on the child nodes. A distinctive feature of the process tree representation as a linear recording is preclude loss plurality of terminal elements, but the model can be an arbitrary function of the superposition of a set. In the process of synthesis of algebraic formula two problems are solved. The first task is to build a tree structure with anonymous tips. The second task is to specify

* Исследование выполнено за счет гранта Российского научного фонда (проект № 14-11-00242) в Южном федеральном университете.

the values of the tree tops. Leaf nodes are compared with the terminal set, a non-leaf nodes are matched with a functional set. The first problem is solved by ant colony. To solve the second problem is used a genetic algorithm. The rating formula is calculated after solving both problems – the construction of an ant tree unnamed peak and subsequent identification of vertices using ant colony The structure of the graph to find solutions G = (X, U). It is possible to create a solution space in which organized the search process based on the simulation of adaptive behavior of an ant colony. Formulated the necessary conditions under which the built in G route is represented as a legitimate expression D, is a solution of a symbolic regression. Developed heuristic of ant behavior when ant moving in graph to find solutions. For large dimension of time parameters of the algorithm outperformed compared algorithms for the best value of the objective function. Experimental time complexity of the algorithm on a single iteration for fixed values of the control parameters is O (nlgn), where n – the power terminal set.

Symbolic regression; syntax tree; terminal set; functional set; ant colony; genetic search; hybrid algorithm.

Введение. Одной из центральных проблем машинного обучения и интеллектуального анализа данных является проблема построения адекватных моделей регрессии и классификации при решении задач прогнозирования [1, 2]. В большинстве численных методов идентификации для аппроксимации экспериментальных (статистических) данных используются регрессионные модели [3, 4]. Задача заключается в построении математического выражения W, задаваемого примерами пар (x_1, y_1), (x_2, y_2),…, (x_n, y_n), где x_i, и y_i – входные и выходные записи. Регрессия – это оценка функциональной зависимости условного среднего значения результативного признака Y от факторных признаков $X= (x_1 ,x_2 ,...,x_n)$, т.е. регрессия – это некоторая усредненная количественная зависимость между выходными и входными переменными $Y=W(X)$. Критерием качества приближения (целевой функции) обычно является *среднеквадратичная ошибка:* сумма квадратов разности значений модели и зависимой переменной для всех значений независимой переменной в качестве аргумента.

В регрессионном анализе задача регрессии решается путем выбора функциональной формы и последующим нахождением ее численных коэффициентов (любым подходящим методом). Например, линейная – $(y = a_0 + a_1 x)$, квадратичная – $y = a_0 + a_1 x + a_2 x^2$, полиномиальная регрессия и др. [5]. На первом этапе регрессионного анализа данные наблюдений или эксперимента представляют графически. Зависимость между переменными X и Y изображают точками на координатной плоскости (x, y) и соединяют их ломаной линией. Этот ломаный график называется *эмпирической линией регрессии Y по X.* По виду эмпирической линии регрессии делают предположение о виде (форме) зависимости переменной Y от X. В простейшем случае предполагают линейную зависимость. Очевидно, что качество аппроксимации при данном подходе напрямую зависит от выбора конкретной параметрической модели [6, 7] .

Задача символьной регрессии заключается в нахождении математического выражения в символьной форме, аппроксимирующего зависимость между конечным набором значений независимых переменных и соответствующими значениями зависимых переменных. Таким образом, символьная регрессия дает нам не только вычислительную процедуру, но и формулу (символьное математическое выражение). Символьная регрессия – метод построения регрессионных моделей путем перебора различных произвольных суперпозиций функций из некоторого заданного набора. Суперпозиция функций при этом называется «программой», а построение таких суперпозиций осуществляется эволюционными стохастическими оптимизационными алгоритмами [8–10]. Подобные алгоритмы являются переборными и требуют значительных вычислительных ресурсов [6–11]. Формулы составляются из переменных, констант и функций, которые связаны некоторыми синтак-

сическими правилами. Поэтому необходимо определить *терминальное множество*, содержащее *константы* и переменные, и *функциональное множество*, которое состоит, прежде всего, из операторов и необходимых элементарных функций. Терминальное множество включает в себя: 1) внешние входы в программу; 2) используемые в программе константы; 3) функции, которые не имеют аргументов. Существуют два подхода к выбору констант. При первом подходе множество числовых констант выбирается для всей популяции и не меняется при поиске решения. При втором – случайно генерированные (обычно из заданного диапазона) константы могут мутировать.

В настоящее время наиболее распространенными структурами для представления математических выражений являются: древовидное представление; линейная структура; графоподобная структура [4, 5].

Достаточно удобным вариантом представления алгебраических формул является синтаксическое дерево [11]. Листовые узлы соответствуют переменным или числовым константам, а нелистовые узлы содержат операцию, которая выполняется над дочерними узлами. Стоит заметить, что для каждого синтаксического дерева существует бесконечное количество семантически эквивалентных деревьев. Все дело в коэффициентах. Коэффициенты каждого дерева оптимизируются методами генетического поиска.

Большинство разработанных алгоритмов символьной регрессии базируется на методе генетического программирования [12] и использовании для представления математических выражений древовидных структур. Данному подходу присущи существенные недостатки [8–14]. Во-первых, достаточно сложными являются принципы кодирования хромосом. Во-вторых, используются хромосомы разной длины, что значительно усложняет операцию кроссинговера. В-третьих, и это наиболее существенный недостаток – проблема избыточного разрастания деревьев. Для решения этой проблемы используют два подхода. Первый связан с ограничением максимальной глубины дерева [10, 14]. Использование для этого различных отсечек может привести к потери оптимальных решений. Второй подход связан с использованием правил трансформации деревьев [1]. С помощью этих правил производятся эквивалентные преобразования структуры деревьев, их упрощение, но при этом функциональные свойства остаются без изменений. Использование таких правил приводит к значительному росту трудоемкости алгоритма. В связи с этим создание нового математического подхода для решения задачи последовательного выбора регрессионных моделей является актуальным

Исходя из вышеизложенного, представляет интерес разработка и применение новых поисковых алгоритмов для решения задачи символьной регрессии на базе эффективных метаэвристик. Результатом непрекращающегося поиска наиболее эффективных методов стало использование бионических методов и алгоритмов интеллектуальной оптимизации, базирующихся на моделировании коллективного интеллекта [15–21]. К таким методам можно отнести и муравьиные алгоритмы (Ant Colony Optimization – ACO) [17–21]. Основу поведения муравьиной колонии составляет самоорганизация, обеспечивающая достижения общих целей колонии на основе низкоуровневого взаимодействия

В работе рассматривается гибридный алгоритм для решения задачи символьной регрессии. Используется традиционное представление алгебраической формулы в виде синтаксического дерева. Листовые узлы соответствуют переменным или числовым константам, а не листовые узлы содержат операцию, которая выполняется над дочерними узлами.

В процессе синтеза алгебраической формулы решаются две задачи. Первая задача заключается в построении структуры дерева с безымянными вершинами. Вторая задача заключается в конкретизации значений вершин дерева. Листовые узлы сопоставляются с терминальным множеством, а нелистовые узлы сопостав-

ляются с функциональным множеством. Первая задача решается методами муравьиной колонии. Для решения второй задачи используется генетический алгоритм. Оценка формулы вычисляется после решения обеих задач – построения муравьем дерева с безымянными вершинами и последующей идентификацией вершин с помощью генетического алгоритма

1. Постановка задачи. Обозначим как y_i^* значение выходной записи, получаемой с помощью выражения *W*. Для оценки математического выражения *W* введем критерий

$$F=\Sigma|y_i - y_i^*|^2. \quad (1)$$

Для решения задачи символьной регрессии необходимо выполнить следующие подготовительные шаги, а именно определить:

- терминальное множество;
- функциональное множество;
- функцию пригодности;
- параметры, контролирующие работу алгоритма;
- критерий остановки.

На первом этапе определяется множество термов, из которых будет строиться решение. В задаче символьной регрессии терминальное множество *T* содержит набор переменных x_i, $i = \overline{1,N}$, где *N* – размерность поставленной задач), и набор констант c_j, $j = \overline{1,\ K}$.

На втором этапе пользователь метода должен определить множество *Ф*-функций, которые будут использованы для построения решений. Пользователь должен априори предполагать некоторую комбинацию функций, которые могли бы содержаться в решении задачи. Функциональное множество может содержать:

- арифметические операции (+, −, ×, ÷);
- математические функции (*Sin, Cos, Exp, Log* и т.д.);
- булевы операции ($\wedge$, $\vee$, $\rightarrow$, $\neg$);
- специальные предопределенные функции (Automatically Defined Functions).

Необходимо учитывать то обстоятельство, что заданное универсальное множество $C=F\cup T$ должно удовлетворять условию замкнутости. Например, для исключения деления на нуль вводят «защищенное деление», которое возвращает единицу, когда знаменатель равен нулю.

На третьем этапе для оценки уравнения символьной регрессии задается целевая функция, вычисляемая по заданной обучающей выборке.

С одной стороны, *терминальное* и *функциональное множества* должны быть достаточно большими для представления потенциального решения. С другой стороны не следует сильно без необходимости расширять *функциональное множество*, поскольку при этом резко возрастает пространство поиска решений.

2. Описание дерева. Рассмотрим структуру выражения для описания бинарного дерева с безымянными вершинами. Введём алфавит $A=\{\mathbf{o}, \bullet\}$. Структуру дерева можно задать, используя на базе алфавита *A* польское выражение для бинарного дерева, где знак **o** соответствует листьям дерева (термам), а знак • соответствует внутренним вершинам дерева (функциям) [11]. Польское выражения для дерева, представленного на рис. 1, имеет следующий вид:

o o • o o • • o • .

Процесс восстановления дерева по польскому выражению достаточно прост. Последовательно слева направо просматривается польское выражение и отыскиваются буквы типа •, соответствующие внутренним (нелистовым) вершинам дере-

ва. Каждая такая вершина объединяет два ближайших образованных на предыдущих шагах подграфа, расположенных в польской записи слева от знака •. Проиллюстрируем процесс свертки с помощью скобок: [{(**o o** •)(**o o** •)• }**o** •].

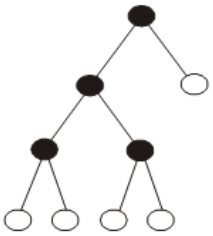

Рис. 1. Пример дерева

Отметим основные свойства польского выражения, для выполнения которых необходимо, чтобы записи соответствовало дерево [11]. Обозначим через *n* число элементов польского выражения типа **o**, а через *m* – число элементов типа •. Пронумеруем позиции между знаками **o**, как показано ниже.

o o *1* **o** *2* **o** *3* **o** *4* … **o** *m* **.**

Условия (свойства) легитимности польского выражения.

1. Для дерева всегда выполняется равенство n = m + 1.

2. Если в польском выражении провести справа от знака • сечение, то слева от сечения число знаков X больше числа знаков • по крайней мере на единицу.

3. Первый знак • в польском выражении (при просмотре слева направо) может появиться только после двух знаков **oo.**

4. Максимальное число знаков •, которое может появиться в позиции, равно номеру позиции.

Если польское выражение соответствует вышеперечисленным свойствам, то ему соответствует дерево. Назовем польское выражение *R*, построенное на базе алфавита *A={***o, •***}*, легитимным, если оно удовлетворяет вышеперечисленным условиям. Таким образом, легитимное выражение *R* является символьным представлением дерева. Различные решения получаются путём комбинирования взаимным расположением элементов алфавита *A={***o, •***}*, удовлетворяющим условиям легитимности. В работе пространство решений представляется множеством легитимных выражений *R*. Поиск решения сводится к поиску такого легитимного выражения *R*, которое оптимизирует показатель качества (критерий).

3. Построение муравьем дерева с безымянными вершинами. 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. Задача формулируется как задача поиска минимального по стоимости маршрута на графе поиска решений (ГПР) [17].

В работе задача синтеза дерева сводится к задаче формирования соответствующего польского выражения. Предварительно формируется заготовка в виде вектора *R*=**o o** *1* **o** *2* **o** *3* **o** *4* ... **o** *m* с пронумерованными позициями. Задача формирования соответствующего польского выражения заключается в назначении *m* элементов типа • в позиции заготовки *R* с соблюдением рассмотренных выше свойств польского выражения (1-4). Поиск решения сводится к поиску такого легитимного польского выражения *E* и соответствующего ему дерева D^0, которое после идентификации вершин дерева D^0 с помощью ГА оптимизирует показатель качества (критерий).

Поиск решений осуществляется на графе поиска решений $G=(X,U)$. Базовая структура ГПР формируется следующим образом. Вершины множества X размещены в узлах решетки размером $m \times m$. Множество вершин графа G разбито на m стадий X_l. Каждая стадия X_l представляет собой столбец из m-l+1 вершин. Вершины x_{il} стадии X_l пронумерованы снизу вверх. i – номер вершины в стадии l (множестве X_l). Ребра графа $G=(X,U)$ ориентированные и связывают вершины соседних стадий X_l и X_{l+1} по следующему правилу. Вершина $x_{il} \in X_l$ связана со всеми такими вершинами $x_{j,l+1} \in X_{l+1}$, что $j \le i$. В общем случае ГПР представляется совокупностью из m стадий (по числу элементов типа ●) и начальной вершины O (рис. 2).

Задача каждого муравья a_k найти в графе G маршрут M_k из вершины O до вершины x_{1m} стадии X_m. В маршрут входят по одной вершине из каждой стадии. Если вершина x_{il} вошла в состав маршрута M_k, то это значит, что в l-ю позицию заготовки R включается элемент типа ●. В стадиях, начиная со второй, содержатся свободные вершины. Поскольку свободные вершины не могут входить в маршрут, то они исключаются из графа поиска решений.

Пример. Необходимо синтезировать дерево с размерами n=6, m=5. Формируется заготовка в виде вектора R = **o o** $\underline{1}$ **o** $\underline{2}$ **o** $\underline{3}$ **o** $\underline{4}$ **o** $\underline{5}$. На рис. 3 представлен граф поиска решений и найденный на нем муравьем a_k маршрут. В соответствии с найденным маршрутом польское выражение имеет вид E = **o o ● o o ●● o ● o ●.** Дерево, построенное по полученному польскому выражению, представлено на рис. 4.

В общем случае поиск решения задачи синтеза дерева осуществляется коллективом муравьев $A=\{a_k \mid k=1,2,\ldots,NR\}$. На каждой итерации муравьиного алгоритма каждый муравей a_k строит свое конкретное решение задачи синтеза дерева. Решением является маршрут M_k в графе $G=(X,U)$, включающий m вершин, принадлежащих множествам $X_1 - X_m$, построенный в соответствии с условиями 1–4. В этом случае построенный маршрут M_k представляется в виде легитимного вектора E_k. По вектору E_k строится дерево деревьев $D_k^{\,0}$ с безымянными вершинами, на базе которого строится математическое выражение Q.

Моделирование поведения колонии муравьёв в задаче построения дерева связано с распределением феромона на ребрах графа G. На начальном этапе на всех ребрах графа G откладывается одинаковое (небольшое) количество феромона ξ/v, где $v=|U|$. Параметр ξ задается априори. Процесс поиска решений итерационный. Каждая итерация l включает три этапа.

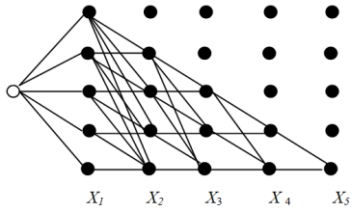

Рис. 2. Граф поиска решений

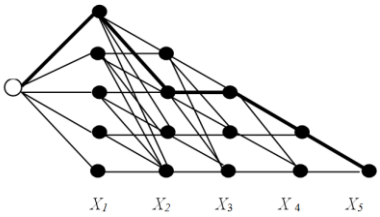

Рис. 3. Маршрут, построенный муравьем на графе поиска решений

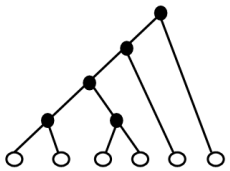

Рис. 4. Дерево со структурой, соответствующей записи D = ***o o • o o •• o • o •***

На первом этапе каждый муравей находит решение (строит маршрут, который интерпретируется как дерево с последующей идентификацией вершин методами генетического поиска), рассчитывается оценка решения. На втором этапе каждый муравей на ребрах построенного им маршрута откладывает феромон, в количестве, соответствующем оценке решения. На третьем этапе осуществляется испарения феромона на ребрах ГПР. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае ферромон откладывается агентом на ребрах после полного формирования решения. На первом этапе каждой итерации каждый *k*-й муравей формирует на ГПР свой собственный маршрут M_k. Процесс построения маршрута M_k пошаговый. На каждом шаге l выбирается вершина из множества X_l. Пусть выполнено l-1шагов, а $x_{e,l-1}$ – последняя вершина частично построенного за $(l-1)$ шагов маршрута M_k. $x_{e,l-1} \in X_{l-1}$. На шаге l агент применяет вероятностное правило выбора следующей вершины из стадии X_l для включения ее в формируемый маршрут M_k. Для этого формируется множество вершин $Y_k(l) \subset X_l$, таких, что каждая из вершин $x_{il} \in Y_k(l)$ может быть добавлена в формируемый маршрут M_k с соблюдением условий 1–4.

Агент просматривает все вершины $x_{il} \in Y_k(l)$. Для каждой вершины $x_{il} \in Y_k(l)$ рассчитывается параметр h_{il} – суммарный уровень феромона на ребре графа G, связывающего последнюю вершину маршрута $x_{e,l-1} \in X_{l-1}$ с вершиной $x_{il} \in Y_k(l) \subset X_l$,

Вероятность P_{il} включения вершины $x_{il} \in Y_k(l)$в формируемый маршрут M_k определяется следующим соотношением:

$$P_{il}=h_{il} / \sum_i h_{il} , \quad (i \mid x_{il} \in Y_k(l)). \qquad (2)$$

Агент с вероятностью P_{il} выбирает одну из вершин, которая включается в маршрут M_k.

После построения агентом маршрута на базе заготовки R формируется польское выражение E_k., в соответствии с которым строится дерево с безымянными вершинами D^0_k.

Математическое выражение Q_k, для которого рассчитывается оценка F_k, формируется после идентификации вершин построенного муравьем a_k дерева D^0_k методами генетического поиска. Алгоритм идентификации дерева D^0_k рассматривается ниже.

На втором этапе итерации каждый муравей откладывает феромон на рёбрах построенного маршрута. Количество феромона Δ_k, откладываемое муравьем a_k на каждом ребре построенного маршрута M_k, определяется следующим образом:

$$\Delta_k= \lambda / F_k . \qquad (3)$$

Параметр λ задается априори. F_k – целевая функция для решения Q_k,, полученного муравьем a_k на t-й итерации. Чем меньше F_k, тем больше феромона откладывается на ребрах построенного маршрута и, следовательно, тем больше вероятность выбора этих ребер при построении маршрутов на следующей итерации.

Обозначим как $\varphi_{ij}(t)$ суммарное количество феромона, отложенного на дуге (i,j) всеми муравьями на t-й итерации. После того, как каждый агент сформировал решение и отложил феромон, на третьем этапе происходит общее испарение феромона на ребрах графа G в соответствии с формулой (4).

$$\delta_{ij}(t) = (\delta_{ij}(t-1) + \varphi_{ij}(t)) \cdot (1 - \rho), \quad (4)$$

где $\delta_{ij}(t)$ – уровень феромона на ребре (i,j), ρ – 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 t (число итераций), количества вершин графа n и числа муравьев m и определяется как $O(t*n^2*m)$.

4. Принцип гибридизации. Метаэвристика муравьиного алгоритма основывается на комбинации двух техник: общая схема строится на базовом методе, в которую включается та или иная встроенная процедура. Важным аспектом есть то, что встроенная процедура – это в большинстве случаев самостоятельный алгоритм решения той же задачи, что и метаэвристический метод в целом. Базовый метод заключается в реализации итерационной процедуры поиска лучшего решения, на основе механизмов адаптивного поведения муравьиной колонии. Основу встроенной процедуры составляет конструктивный алгоритм построения муравьем некоторой конкретной интерпретации решения. В оптимизации муравьиными колониями [8] конструктивный блок – деятельность искусственных муравьев играет ключевую роль. Классификация гибридных метаэвристик детально рассмотрена в работе [14].

Методами генетического поиска производится идентификация безымянных вершин каждого построенного муравьем a_k дерева D^0_k. Для этого формируется структура хромосомы, состоящая из трех частей. Значения генов первой части соответствуют элементам функционального множества (функциям и правилам). Значения генов второй и третьей частей части соответствуют элементам терминального множества (переменным и константам). Отметим, что функциональное и терминальное множества формируются на подготовительном этапе, причем для констант задаются границы возможных значений. Алгоритмы символьной регрессии на основе гибридизации методов муравьиной колонии и генетического поиска формулируются следующим образом.

Алгоритм поведения муравьиной колонии

1. Производится предварительный анализ задачи символьной регрессии. Формируются функциональное и терминальное множества. Для констант задаются границы возможных значений.

2. В соответствии с исходными данными формируются заготовка R для польского выражения и граф поиска решений G, на ребрах которого отложено начальное количество феромона.

3. Задается: число итераций – NT; число муравьев, формирующих независимо друг от друга решения на одной итерации – NR.

4. $t=1$. (t– номер итерации).

5. $k=1$. (k– номер агента).

6. *(Алгоритм муравья).* Муравей a_k строит на графе поиска решений G маршрут M_k из вершины O до вершины x_{1m} стадии X_m.

7. В соответствии с построенным маршрутом M_k и заготовкой R строится польское выражение E_k , по которому строится дерево D^0_k с безымянными вершинами.

8. *(Алгоритм идентификации вершин).* Методом генетического поиска производится идентификация вершин дерева D^0_k (т.е. построение дерева D^i_k).

9. По дереву с идентифицированными вершинами D^i_k строится математическое выражение Q_k, для которого находится значение целевой функции F_k.

10. Если $k< NR$, то $k=k+1$ и переход к пункту 6, иначе переход к пункту 11.

11. $k=1$.

12. Муравей a_k откладывает на каждом ребре построенного им маршрута M_k в графе поиска решений G ферромон в количестве

$$\Delta_k = \lambda / F_k.$$

13. Если $k< NR$, то $k= k+1$ и переход к пункту 14, иначе переход к пункту 15.

14. На третьем этапе итерации t происходит общее испарение феромона на всех ребрах графа G в соответствии с формулой

$$\delta_{ij}(t) =(\delta_{ij}(t-1) + \varphi_{ij}(t)) (1-\rho),$$

где ρ – коэффициент обновления.

15. Находится агент a_k с лучшей оценкой $F_{опт}$ решения, полученного после выполнения t итераций, которое запоминается.

16. Если $t< NT$, то $t= t+1$ и переход к пункту 6, иначе переход к пункту 17.

17. Конец работы алгоритма.

Рассмотрим теперь конструктивный алгоритм построения муравьем маршрута $M_k(t)$ в графе поиска решений G из вершины O до вершины x_{1m} стадии X_m

Алгоритм муравья

1. Муравей помещается в вершину О графа поиска решений.

2. *END*=О. (*END* – последняя вершина, вошедшая в формируемый маршрут M_k).

3. $l=1$. (l – номер шага).

4. Формируется множество вершин ГПР – $Y_k(l) \subset X_l$, таких, что каждая из вершин $x_{il} \in Y_k(l)$ может быть добавлена в формируемый маршрут M_k с соблюдением условий 1–4.

5. Для каждой вершины $x_{il} \in Y_k(l)$рассчитывается параметр h_{il} – суммарный уровень феромона на ребре графа G, связывающего последнюю вершину *END* маршрута M_k с вершиной $x_{il} \in Y_k(l) \subset X_l$,

6. Вероятность P_{il} включения вершины $x_{il} \in Y_k(l)$в формируемый маршрут M_k определяется следующим соотношением

$$P_{il} = h_{il} / \sum_i h_{il} , \ (i \mid x_{il} \in Y_k(l)).$$

7. Случайным образом, в соответствии с распределением вероятностей, рассчитанным в пункте 5, выбирается вершина x_{il}, которая включается в конец маршрута M_k. *END*= x_{il}.

8. Если $l< m$, то $l= l+1$ и переход к пункту 3, иначе переход к пункту 8.

9. В соответствии с построенным маршрутом M_k на заготовке R строится польское выражение E_k.

10. В соответствии с польским выражением E_k строится дерево D_k с безымянными вершинами.

11. Конец работы алгоритма.

Как уже указывалось выше, идентификация вершин дерева осуществляется методами генетического поиска. Структура хромосомы имеет вид: $H=\{g_i \mid i=1,2,\ldots, n_1, (n_1+1),\ldots, n_2, (n_2+1), \ldots, n_3\}$. Гены в локусах с 1 по n_1, предназначены для записи элементов функционального множества, гены в локусах $(n_1+1),\ldots, n_2$, предназначены для записи переменных, гены в локусах $(n_2+1), \ldots, n_3$, предназначены для записи констант.

На структуру хромосомы в зависимости от постановки задачи могут налагаться ограничения. Первое связано с тем, что все гены должны иметь отличное друг от друга значение. Это порождает проблему легитимности хромосом, образующихся после выполнения генетических операторов кроссинговера и мутации. Для выполнения этого ограничения должны использоваться специальные операторы кроссинговера и мутации либо специальные методы кодирования.

Второе связано с расширением пространства поиска за счет увеличения числа комбинаций генов. С этой целью увеличиваются метрические параметры хромосомы n_1, n_2, n_3 и допускается использование повторяющихся генов в пределах каждой из трех областей хромосомы. Пусть у синтезированного муравьем дерева v_1, v_2, v_3 – количество вершин, соответствующих функциональному множеству, множеству переменных и множеству констант. Идентификация дерева по выбранной хромосоме производится следующим способом. Для идентификации вершин функционального множества используются гены хромосомы, расположенные в локусах с 1 по v_1. Для идентификации вершин множества переменных используются гены, расположенные в локусах с (n_1+1) по (n_1+1+v_2). Для идентификации вершин множества констант используются гены, расположенные в локусах с (n_2+1) по $(n_2+1+ v_3)$. Отметим, что $n_1 \geq v_1$, $n_2 \geq v_1$, $n_3 \geq v_3$.

Для выполнения идентификации вершин требуется выполнить следующие подготовительные шаги.

1. В соответствии с заданными составами функционального и терминального множеств формируется структура хромосомы $H=\{g_i|i=1,2,\ldots, n_1, (n_1+1),\ldots , n_2, (n_2+1), \ldots, n_3\}$. В соответствии с постановкой определяются метрические параметры хромосомы n_1, n_2, n_3, v_1, v_2, v_3 и задаются ограничения.

2. В соответствии с постановкой задачи идентификации осуществляется выбор операторов кроссинговера и мутации.

3. Задается размер начальной популяции.

Алгоритм идентификации вершин дерева

1. Генерируется начальная популяция $П$ хромосом, структура которых сформирована на подготовительном этапе.

2. На базе построенного муравьем a_k дерева D^0_k и популяции хромосом $П$ формируется множество формул Θ_k. Для каждой формулы $\theta_{ik} \in \Theta_k$ подсчитывается оценка F_{ik}.

3. Задается: число итераций генетического алгоритма – NG.

4. $t=1$. (t – номер итерации).

5. На популяции хромосом $П$ выполняются операции кроссинговера и мутации и формируется множество новых хромосом $П^*$.

6. На базе построенного муравьем a_k дерева D^0_k и популяции хромосом $П^*$ формируется множество формул Θ^*_k. Для каждой формулы $\theta^*_{ik} \in \Theta^*_k$. подсчитывается оценка F^*_{ik}.

7. Популяции $П$ и $П^*$ объединяются, $П_o = П \cup П^*$.

8. Производится редукция популяции $П_o$ до размеров популяции $П$.

9. Если $t< NG$, то $t= t +1$ и переход к пункту 5, иначе переход к пункту 10.

10. Выбирается решение D^*_k с лучшей оценкой.

11. Конец работы алгоритма.

5. Экспериментальные исследования. Для оценки эффективности предлагаемого гибридного биоинспирированного алгоритма (ГБА) был проведен ряд численных экспериментов. Результаты экспериментов ГБА сравнивались с экспериментальными результатами работы метода стандартного генетического про-

граммирования (МСГП) и метода гибридного генетического программирования (МГГП) [10]. В качестве общего для всех задач критерия останова использовалось условие достижения уровня относительной ошибки моделирования либо выполнение заданного максимального числа вычислений функции пригодности. Эффективность методов оценивалась по критерию надежности, который определялся как отношение числа запусков, в которых была достигнута заданная точность аппроксимации исходных данных, к общему числу запусков [10]. При этом максимальное число вычислений функции пригодности не должно превышать заданное в критерии останова. Статистика для получения оценок надежности набиралась по 50 запускам каждого из рассмотренных методов. Значимость в различиях результатов алгоритмов проверялась методами ANOVA. Проверка выполнялась при уровне значимости α=0,05.Описание тестовых задач приведено в табл 1. В табл. 2 представлены результаты сравнительного исследования эффективности, полученные для рассматриваемых алгоритмов на тестовых задачах. Жирным шрифтом выделен метод, победивший на тестовой задаче, т.е. статистически значимо превосходящий по надежности конкурирующий метод. Разработанный гибридный биоинспирированный алгоритм оказался эффективнее методов стандартного и гибридного генетического программирования.

Таблица 1

№ задачи	Моделируемая функция	Интервал варьирования переменных	Функциональное множество	Объем выборки
1	$y = \sin(x)$	$x\epsilon[-3;4]$	$\{+,-,\times,/\}$	100
2	$y = x^2 + 2x + 3$	$x\epsilon[-3;4]$	$\{+,-,\times,/\}$	100
3	$y = x_1^2 + x_2^2$	$x_1, x_2\epsilon[-4;4]$	$\{+,-,\times,/\}$	200
4	Функция Растригина $y = 0{,}1x_1^2 + 0{,}1x_2^2 - 4\cos(0{,}8x_1) - 4\cos(0{,}8x_1) + 8$	$x_1, x_2\epsilon[-3;3]$	$\{+,-,\times,/,cos, sin,\sqrt{x},exp\}$	200
5	$y = x_1^2\sin(x_1) + x_2^2\sin(x_2)$	$x_1, x_2\epsilon[-4;4]$	$\{+,-,\times,/,cos, sin,\sqrt{x},exp\}$	200
6	Функция Розенброка $y = 100(x_2 - x_1^2)^2 - (1 - x_1)^2$	$x_1, x_2\epsilon[-2;2]$	$\{+,-,\times,/\}$	200
7	$y = x_1^2 + x_1x_2 + x_2^2$	$x_i\epsilon[-4;4]$, $i = \overline{1,3}$	$\{+,-,\times,/\}$	300

Таблица 2

Тестовая задача	1	2	3	4	5	6	7
МСГП	0.6	0.9	0.5	0.45	0.95	0.6	0.45
МГГП	0.9	1	0.9	O.95	1	0.95	0.85
ГБА	**O.95**	**1**	**O.95**	**O.95**	**1**	**0.95**	**0.9**

Заключение. В работе рассматриваются новые принципы решения задачи множественной нелинейной символьной регрессии на основе моделей адаптивного поведения биологических систем. Рассмотрены бионические алгоритмы порождения допустимых существенно нелинейных суперпозиций. Предложен муравьиный алгоритм, порождающий все возможные суперпозиции заданной сложности за конечное число шагов. Сформулированный гибридный алгоритм решает некоторые типичные проблемы предложенных ранее методов генетического программирования. Для компактного представления решения задачи символьной регрессии

используется модифицированная польская запись. Архитектура задачи символьной регрессии представлена в виде набора компонент муравьиного алгоритма и генетического поиска. Разработана структура графа поиска решений $G=(X,U)$. Это позволило создать пространство решений, в рамках которого организован поисковый процесс, базирующийся на моделировании адаптивного поведения муравьиной колонии. Сформулированы необходимые условия, при выполнении которых построенный в графе G маршрут представляется в виде легитимного выражения D, являющегося решением задачи символьной регрессии. Разработаны эвристики поведения муравья при перемещениях в графе поиска решений. Отличительной особенностью способов представления деревьев в виде линейной записи является то, что они исключают возможность потери элементов терминального множества, но при этом модель может быть произвольной суперпозицией функций из некоторого набора. Эксперименты показали, что представленный биоинспирированный алгоритм, базирующийся на гибридизации методов муравьиной колонии и генетического поиска, строит более простые модели и более точные в сравнении с алгоритмами на основе принципов генетического программирования.

При больших размерностях временные показатели разработанного алгоритма превосходят показатели сравниваемых алгоритмов при лучших значениях целевой функции.

Экспериментальная временная сложность алгоритма на одной итерации при фиксированных значениях управляющих параметров составляет $O(nlgn)$, а временная сложность существующих алгоритмов [3–15] – $O(n^2)$, где n – мощность терминального множества.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Ian H. Witten, Eibe Frank and Mark A.* Hall Data Mining: Practical Machine Learning Tools and Techniques. – 3rd Edition. – Morgan Kaufmann, 2011.
2. *Sammut C., Webb G.I.* Symbolic regression // Encyclopedia of Machine Learning. – Berlin: Springer, 2010.
3. *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.
4. *Радченко С.Г.* Методология регрессионного анализа: Монография. – К.: Корнийчук, 2011. – 376 с.
5. *Дрейпер Н., Смит Г.* Прикладной регрессионный анализ. – М.: Вильямс, 2007. – 912 с.
6. *Стрижов В.В., Крымова Е.А.* Методы выбора регрессионных моделей. – М.: ВЦ РАН, 2010. – 60 с.
7. *Zelinka I., Oplatkova Z., Nolle L.* Analytic programming and symbolic regression by means of arbitrary evolutionary algorithms // Int. J. Simulation Syst. Sci. Technol. – 2005. – Vol. 6, No. 9. – P. 44-56.
8. *Лебедев Б.К., Лебедев В.Б.* Эволюционная процедура обучения при распознавании образов // Известия ТРТУ. – 2004. – № 8 (43). – С. 83-88.
9. *Рудой Г.И., Стрижов В.В.* Алгоритмы индуктивного порождения суперпозиций для аппроксимации измеряемых данных // Информатика и её применения. – 2013. – Т. 7. – Вып. 1. – С. 44-53.
10. *Бухтояров В.В., Семенкин Е.С.* Разработка и исследование гибридного метода генетического программирования // Программные продукты и системы. – 2010. – № 3. – С. 34-38.
11. *Kureichik V.M., Lebedev B.K. and Lebedev V.B.* VLSI Floorplanning Based on the Integration of Adaptive Search Models // Journal of Computer and Systems Sciences International. – 2013. – Vol. 52, No. 1. – P. 80-96. ISSN 1064_2307.
12. *Koza, J.R.* Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Springer. 2005.
13. *Lebedev B.K. and. Lebedev V.B.* Synthesis of Mathematical Expressions by Methods of Genetic Search // Proceedings of the International Scientific Conferences “Intelligent Systems (IEEE AIS’06)”and “Intelligent CAD’s (CAD- 2006)”. Scientific publication in 3 vol. Vol. 3. – M.: Physmathlit, 2006. – P. 29-34.

14. *Barmpalexis P.,Kachrimanis K., Tsakonas A., Georgarakis E.* Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation // Chemometrics and Intelligent Laboratory Systems. – 2011. – Vol. 107, No. 1. – P. 75-82.
15. *Colin G. Johnson*. Artificial Immune Systems Programming for Symbolic Regression // Genetic Programming: 6th European Conference. – 2003. – P. 345-353. – ISBN=3-540- 00971-X.
16. *Ушаков С.А.* Использование распределенных искусственных иммунных систем для решения задачи символьной регрессии // Инноватика. Научный электронный журнал. – 2014. – № 1. Свидетельство о регистрации ЭЛ № ФС 77-5722.
17. *Лебедев О.Б.* Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог: Изд-во ЮФУ, 2013. – 199 с.
18. *Лебедев Б.К., Лебедев В.Б.* Оптимизация методом кристаллизации россыпи альтернатив (КРА) // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 11-17.
19. *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.
20. *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.
21. *Kureichik V.M., Lebedev B.K., Lebedev O.B.* A hybrid partitioning algorithm based on natural mechansms of decision making // Scientific and Technical Information Processing. – 2012. – № 39 (6). – P. 317-327.

REFERENCES

1. *Ian H. Witten, Eibe Frank and Mark A.* Hall Data Mining: Practical Machine Learning Tools and Techniques. 3rd Edition. Morgan Kaufmann, 2011.
2. *Sammut C., Webb G.I.* Symbolic regression, *Encyclopedia of Machine Learning*. Berlin: Springer, 2010.
3. *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. Peterburg: BKhV-Peterburg, 2004, 336 p.
4. *Radchenko S.G.* Metodologiya regressionnogo analiza: Monografiya [Methodology regression analysis: Monograph]. K.: Korniychuk, 2011, 376 p.
5. *Dreyper N., Smit G.* Prikladnoy regressionnyy analiz [Applied regression analysis]. Moscow: Vil'yams, 2007, 912 p.
6. *Strizhov V.V., Krymova E.A.* Metody vybora regressionnykh modeley [Methods selection of regression models]. Moscow: VTs RAN, 2010, 60 p.
7. *Zelinka I., Oplatkova Z., Nolle L.* Analytic programming and symbolic regression by means of arbitrary evolutionary algorithms, *Int. J. Simulation Syst. Sci. Technol*., 2005, Vol. 6, No. 9, pp. 44-56.
8. *Lebedev B.K., Lebedev V.B.* Evolyutsionnaya protsedura obucheniya pri raspoznavanii obrazov [Evolutionary procedure learning in pattern recognition], *Izvestiya TRTU* [Izvestiya TSURe], 2004, No. 8 (43), pp. 83-88.
9. *Rudoy G.I., Strizhov V.V.* Algoritmy induktivnogo porozhdeniya superpozitsiy dlya approksimatsii izmeryaemykh dannykh [Algorithms for inductive generation of superpositions for approximation of the measured data], *Informatika i ee primeneniya* [Informatics and Applications], 2013, Vol. 7, Issue 1, pp. 44-53.
10. *Bukhtoyarov V.V., Semenkin E.S.* Razrabotka i issledovanie gibridnogo metoda genetiche-skogo programmirovaniya [Research and development of hybrid method of genetic programming], *Programmnye produkty i sistemy* [Software products and systems], 2010, No. 3, pp. 34-38.
11. *Kureichik V.M., Lebedev B.K. and Lebedev V.B.* VLSI Floorplanning Based on the Integration of Adaptive Search Models, *Journal of Computer and Systems Sciences International*, 2013, Vol. 52, No. 1, pp. 80-96. ISSN 1064_2307.
12. *Koza, J.R.* Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Springer. 2005.
13. *Lebedev B.K. and. Lebedev V.B.* Synthesis of Mathematical Expressions by Methods of Genetic Search, *Proceedings of the International Scientific Conferences "Intelligent Systems (IEEE AIS'06)"and "Intelligent CAD's (CAD- 2006)". Scientific publication in 3 vol*. Vol. 3. Moscow: Physmathlit, 2006, pp. 29-34.

14. *Barmpalexis P.,Kachrimanis K., Tsakonas A., Georgarakis E.* Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation, *Chemometrics and Intelligent Laboratory Systems*, 2011, Vol. 107, No. 1, pp. 75-82.
15. *Colin G. Johnson*. Artificial Immune Systems Programming for Symbolic Regression, *Genetic Programming: 6th European Conference*, 2003, pp. 345-353. ISBN=3-540- 00971-X.
16. *Ushakov S.A.* Ispol'zovanie raspredelennykh iskusstvennykh immunnykh sistem dlya resheniya zadachi simvol'noy regressii [The use of distributed artificial immune system for solving symbolic regression], *Innovatika. Nauchnyy elektronnyy zhurnal* [Innovation. Scientific electronic journal], 2014, No. 1. Certificate of registration EL № FS 77-5722.
17. *Lebedev O.B.* Modeli adaptivnogo povedeniya murav'inoy kolonii v zadachakh proektirovaniya [Models of adaptive behavior, ant colony in the task of designing]. Taganrog: Izd-vo YuFU, 2013, 199 p.
18. *Lebedev B.K., Lebedev V.B.* Optimizatsiya metodom kristallizatsii rossypi al'ternativ (KRA) [Optimization by the crystallization of alternatives field (CAF) method], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 11-17.
19. *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.
20. *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.
21. *Kureichik V.M., Lebedev B.K., Lebedev O.B.* A hybrid partitioning algorithm based on natural mechansms of decision making, *Scientific and Technical Information Processing*, 2012, No. 39 (6), pp. 317-327.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.А. Гатчин.

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: oblebedev@sfedu.ru; тел.: 89085135512; кафедра систем автоматизированного проектирования; доцент.

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: oblebedev@sfedu.ru; phone: +79085135512; the department of computer aided design; associate professor.

УДК 004.822

В.В. Бова, Д.В. Заруба, В.В. Курейчик

ЭВОЛЮЦИОННЫЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧИ ИНТЕГРАЦИИ ОНТОЛОГИЙ*

В настоящее время интеграция данных и знаний является одной из наиболее важных задач обеспечения интероперабельности информационных систем на структурном и семантическом уровне. Рассматривается технология интеграции онтологий, предполагающая проведение автоматического сопоставления понятий интегрируемых онтологий с помощью составной семантической метрики (названий и значений понятий или их контекстов), множеств атрибутов и их

* Работа выполнена при поддержке Министерства образования и науки РФ. Проект № 8.823.2014.