

УДК 004.85

## ИДЕНТИФИКАЦИЯ ПАРАМЕТРОВ НЕЧЁТКИХ АППРОКСИМАТОРОВ И КЛАССИФИКАТОРОВ НА ОСНОВЕ АЛГОРИТМА «КУКУШКИН ПОИСК»

**И. А. Ходашинский, Д. Ю. Минина, К. С. Сарин**

*Томский государственный университет систем управления и радиоэлектроники,  
634050, г. Томск, просп. Ленина, 40  
E-mail: hodashn@rambler.ru*

Представлен алгоритм «кукушкин поиск» для идентификации параметров нечётких аппроксиматоров и классификаторов. Описаны результаты экспериментов на восьми наборах данных из репозитория KEEL. Даны сравнения с аналогами.

*Ключевые слова:* нечёткий аппроксиматор, нечёткий классификатор, идентификация параметров, алгоритм «кукушкин поиск».

**Введение.** Нечёткие системы, основанные на правилах, широко применяются для решения проблем аппроксимации и классификации. Существует два подхода к построению таких систем: на базе знаний и опыта эксперта и на наблюдаемых данных. Идентификация рассматривается в предлагаемой работе как процесс построения системы, связанный с установлением закономерностей между входными и выходными переменными наблюдаемого объекта. В нечётких системах такие закономерности представлены в виде правил «ЕСЛИ–ТО». Идентификация параметров — это поиск оптимальных значений всех параметров нечёткой системы, т. е. определение значений консеквентной части (ТО-части) правила и параметров функций принадлежности в антецедентной части (ЕСЛИ-части) каждого правила на основе заданных критериев качества и метода оптимизации выбранного критерия. Начиная с ранних работ [1, 2], проблема идентификации параметров нечётких моделей решается нелинейными методами оптимизации. Однако трудности применения классических методов оптимизации нелинейных функций, такие как проблема локального экстремума и «проклятие размерности», заставили специалистов обратиться к метаэвристическим методам оптимизации, например эволюционным вычислениям или методам роевого интеллекта [3–7]. Проведённые в [8–10] исследования показали высокую эффективность в решении задач оптимизации нового популяционного алгоритма «кукушкин поиск» (АКП).

Цель данной работы — исследование применения алгоритма «кукушкин поиск» для идентификации параметров нечётких систем.

**Постановка задачи.** Нечёткий аппроксиматор задаётся правилами вида

$$\text{ЕСЛИ } x_1 = A_{1i} \text{ AND } x_2 = A_{2i} \text{ AND } \dots \text{ AND } x_n = A_{ni} \text{ ТО } y = r_i,$$

где  $A_{ji}$  — лингвистический терм, которым оценивается входная переменная  $x_j$ ;  $r_i$  — действительное число, которым оценивается выход  $y$ .

Выход нечёткого аппроксиматора определяет следующее отображение:

$$f(\mathbf{x}; \boldsymbol{\theta}) = \frac{\sum_{i=1}^R \mu_{A_{1i}}(x_1) \mu_{A_{2i}}(x_2) \dots \mu_{A_{ni}}(x_n) r_i}{\sum_{i=1}^R \mu_{A_{1i}}(x_1) \mu_{A_{2i}}(x_2) \dots \mu_{A_{ni}}(x_n)},$$

где  $\mathbf{x}$  — входной вектор;  $R$  — число правил;  $n$  — количество входных переменных;  $\mu_{A_{ij}}$  — функция принадлежности  $j$ -й входной переменной;  $\boldsymbol{\theta} = \|\theta_1, \dots, \theta_N\|$  — вектор параметров нечёткого аппроксиматора.

Пусть имеется таблица наблюдений  $\{(\mathbf{x}_p; t_p), p = 1, \dots, m\}$ , тогда критерий качества аппроксимации может быть выражен среднеквадратической функцией ошибки

$$\text{MSE}(\boldsymbol{\theta}) = \sum_{p=1}^m (t_p - f(\mathbf{x}_p, \boldsymbol{\theta}))^2 / m.$$

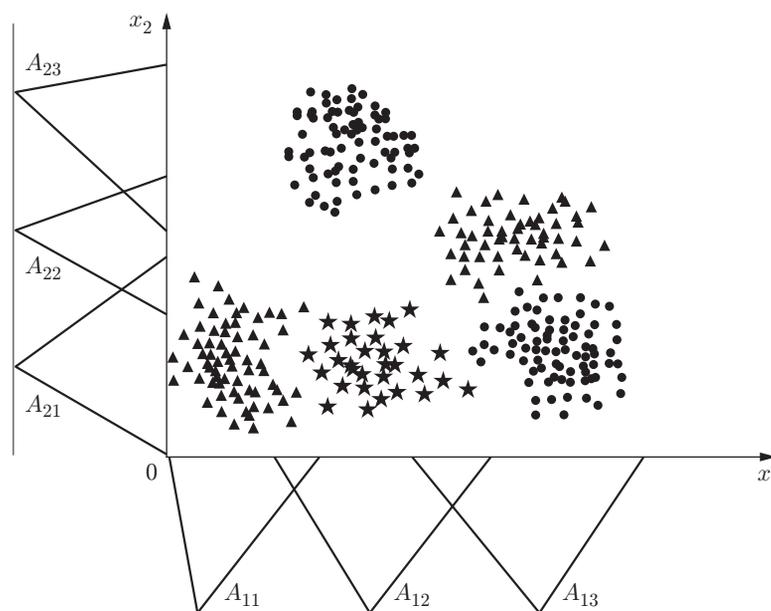
Для оптимизации параметров  $\boldsymbol{\theta}$  предлагается применить алгоритм «кукушкин поиск». Основная идея нечёткого классификатора состоит в описании предполагаемого кластера нечётким прототипом, размерность которого определена размерностью пространства исследуемых данных. Таким образом,  $i$ -й кластер задаётся нечётким правилом следующего вида:

$$R_{ij}: \text{ЕСЛИ } x_1 = A_{1i} \text{ И } x_2 = A_{2i} \text{ И } x_3 = A_{3i} \text{ И } \dots \text{ И } x_n = A_{ni} \text{ ТО } \text{class} = c_j,$$

где  $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$  — вектор признаков классифицируемого объекта;  $A_{ki}$  — нечёткий терм, характеризующий  $k$ -й признак в  $i$ -м правиле ( $i \in [1, R]$ );  $c_j$  — идентификатор  $j$ -го класса ( $j \in [1, m]$ ).

Пример представления в двумерном пространстве трёх классов пятью кластерами приведён на рисунке. База правил для данного примера имеет следующий вид:

$$\begin{aligned} &\text{ЕСЛИ } x_1 = A_{11} \text{ И } x_2 = A_{21} \text{ ТО } \text{class} = 3, \\ &\text{ЕСЛИ } x_1 = A_{12} \text{ И } x_2 = A_{21} \text{ ТО } \text{class} = 2, \\ &\text{ЕСЛИ } x_1 = A_{13} \text{ И } x_2 = A_{21} \text{ ТО } \text{class} = 1, \\ &\text{ЕСЛИ } x_1 = A_{12} \text{ И } x_2 = A_{23} \text{ ТО } \text{class} = 1, \\ &\text{ЕСЛИ } x_1 = A_{13} \text{ И } x_2 = A_{22} \text{ ТО } \text{class} = 3. \end{aligned}$$



Пример разделения трёх классов в двумерном пространстве на пять кластеров:  
класс 1 (●), класс 2 (★), класс 3 (▲)

В процессе нечёткой классификации объект относится к каждому классу с определённой степенью принадлежности, вычисленной следующим образом:

$$\beta_j(\mathbf{x}) = \sum_{R_{ij}} \prod_{k=1}^n A_{ki}(x_k), \quad j = 1, 2, \dots, m.$$

Выходом классификатора является метка класса, заданная выражением

$$\text{class} = c_{j^*}, \quad j^* = \arg \max_{1 \leq j \leq m} \beta_j.$$

Нечёткий классификатор может быть представлен функцией  $c = f(\mathbf{x}, \boldsymbol{\theta})$ , где  $\boldsymbol{\theta}$  — вектор, описывающий базу правил.

На множестве обучающих данных (таблице наблюдений)  $\{(\mathbf{x}_p; c_p), p = 1, \dots, z\}$  найдём единичную функцию

$$\text{delta}(p, \boldsymbol{\theta}) = \begin{cases} 1, & \text{если } c_p = f(\mathbf{x}_p, \boldsymbol{\theta}), \\ 0 & \text{иначе,} \end{cases} \quad p = 1, 2, \dots, z,$$

тогда численный критерий качества классификации выражается как

$$E(\boldsymbol{\theta}) = \sum_{p=1}^z \text{delta}(p, \boldsymbol{\theta}) / z.$$

Проблема идентификации нечёткого классификатора сводится к проблеме поиска максимума указанной функции в многомерном пространстве, координаты которого соответствуют параметрам нечёткого классификатора. Для оптимизации параметров  $\boldsymbol{\theta}$  предлагается использовать АКП.

**Алгоритм «кукушкин поиск».** Оптимизационный алгоритм «кукушкин поиск» — метаэвристика, построенная на основе имитации способа размножения кукушки, когда она находит недавно построенные гнёзда и подкладывает в них свои яйца (заменяет своими), которые в итоге могут быть выкинуты хозяином гнезда [8]. В основе алгоритма лежат три правила: 1) кукушка откладывает по одному яйцу в случайно выбранное гнездо, которое представляет собой решение; 2) часть лучших решений будет перенесена в следующее поколение; 3) количество гнёзд фиксировано, и есть вероятность того, что хозяин может обнаружить чужое яйцо; в этом случае хозяин может выбросить яйцо из гнезда или вовсе отказаться от гнезда и построить новое на новом месте.

Важной составляющей АКП является использование «полётов Леви» для локального и глобального поисков [11]. Процесс полёта Леви — это случайное блуждание, характеризующееся серией скачков, обусловленных функцией плотности вероятности с «толстыми» хвостами, за счёт которых вероятность значительных отклонений от среднего больше, чем у нормального распределения.

Пошаговая реализация алгоритма для классификатора и аппроксиматора представлена далее.

**Шаг 1. Инициализация исходной популяции.** Задаётся  $S$  — размер популяции  $\Theta = (\boldsymbol{\theta}^s, s \in [1, S])$ . Случайным образом определяется «начальное положение кукушки», являющееся текущим решением  $\boldsymbol{\theta}^{\text{cur}}$ . При этом область определения каждой входной переменной полностью покрыта функциями принадлежности, по крайней мере, одна функция принадлежности получает значение не равное нулю в любой точке области определения. Если

в процессе инициализации или оптимизации указанное требование нарушается, то специальная процедура алгоритма производит перекрытие двух соседних термов [12]. Устанавливается  $p$  — вероятность, с которой гнездо может быть «покинуто» хозяином, т. е. вероятность удаления векторов из множества  $\Theta$ . Задаётся количество итераций  $N$  в качестве критерия останова и  $m$  — число худших решений. Количество генерируемых векторов  $l$  полагаем равным  $S$ .

Шаг 2. Случайным образом генерируется  $l$  векторов решений  $\theta$ .

Шаг 3. Генерация нового решения на основе полётов Леви. Выполняется «случайное перемещение кукушки», которое выражено изменением  $\theta^{\text{cur}}$  по закону Леви  $\theta^{\text{cur}} = \theta^{\text{cur}} + \text{Levi}$ , где  $\text{Levi}$  — случайный прыжок полёта Леви, вычисляемый по правилу  $\text{Levi} = \gamma u/|v|^{1/\beta}$ , где  $\gamma$  — коэффициент прыжка полёта Леви; параметр  $\beta$  принимает значения из интервала  $[1, 2]$ ;  $u, v$  — нормально распределённые величины:  $v \sim N(0; \sigma_v^2)$ ,  $u \sim N(0; \sigma_u^2)$ ,  $\sigma_v^2 = 1$ ,  $\sigma_u^2 = \left\{ \frac{\Gamma(1+\beta) \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2] \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta}$ ,  $\Gamma(x)$  — гамма-функция.

Случайным образом выбирается другое решение  $\theta^s$  из популяции  $\Theta$ .

Шаг 4. Оценка качества решения. Для нечёткого классификатора сравниваются значения функций  $E(\theta)$ :

$$\text{Если } E(\theta^{\text{cur}}) < E(\theta^s), \text{ то } \theta^{\text{cur}} = \theta^s, \text{ иначе } \theta^s = \theta^{\text{cur}}.$$

Для нечёткого аппроксиматора сравниваются значения функций  $\text{MSE}(\theta)$

$$\text{Если } \text{MSE}(\theta^{\text{cur}}) > \text{MSE}(\theta^s), \text{ то } \theta^{\text{cur}} = \theta^s, \text{ иначе } \theta^s = \theta^{\text{cur}}.$$

Шаг 5. Удаление плохих решений («неудачных гнёзд»). Если не выполнено заданное количество итераций  $N$ , то выбирается заранее определённое количество  $m$  худших решений; для каждого из них генерируется случайное число  $h$  в диапазоне  $[0, 1]$  и, если значение  $h$  для решения оказывается больше значения заданной вероятности  $p$ , гнездо-решение удаляется; вместо удалённого решения формируется новое;  $l$  полагается равным количеству уничтоженных векторов-решений, переход на шаг 2; иначе выбирается лучшее решение (с максимальным значением  $E(\theta)$  для классификатора, с минимальным значением  $\text{MSE}(\theta)$  для аппроксиматора).

**ВЫХОД.**

**Эксперимент.** Для оценки эффективности нечёткого классификатора, настроенного приведённым алгоритмом, были проведены по схеме кроссвалидации тесты на четырёх наборах данных из репозитория KEEL [13]: balance, rima, vira, iris. В этой схеме набор разбивался на пять файлов, 80 % набора помещалось в обучающую выборку, а остальные 20 % — в тестовую. Проведённые многочисленные опыты показали, что точность результата существенно зависит от числа итераций, в меньшей степени — от количества особей в популяции и уничтожаемых худших решений. Не выявлена зависимость точности результата от коэффициента прыжка полёта Леви  $\gamma$  и параметра  $\beta$ . Эмпирические знания позволили выбрать следующие параметры алгоритма и нечёткого классификатора: число итераций 5000; количество особей в популяции 10; количество уничтожаемых худших решений 3; коэффициент прыжка полёта Леви  $\gamma = 0,1$ ; параметр  $\beta = 1,1$ ; функции принадлежности треугольные. Число правил в классификаторах на наборах данных: balance 3, rima 2, vira 2, iris 81.

Результаты проведённых экспериментов были сопоставлены с результатами работы пяти алгоритмов из [14]. Краткое описание указанных алгоритмов приведено далее.

Алгоритм Ant-Miner (Ant Colony-based Data Miner [15]) основан на системе муравьиной колонии [16]. Каждый муравей в системе, начиная с «пустого» правила, добавляет в

него термины, выбор которых зависит от значения заданной эвристической функции и количества феромона на дуге, соответствующей выбираемому терму. Сформированному таким образом antecedенту ставится в соответствие подходящая метка класса. На каждой итерации выбирается лучшее правило и заносится в базу правил. Процесс заканчивается, когда правилами покрыты все образцы из таблицы наблюдений. В методике коэволюционной классификации CORE (CO-evolutionary Rule Extractor) применяются эволюционные алгоритмы для формирования правил классификации в двух взаимодействующих популяциях [17]. Алгоритм HIDER (Hierarchical DEcision Rules) также основан на применении эволюционных алгоритмов для формирования иерархических (вложенных) решающих правил классификации [18]. Генетический алгоритм извлечения нечётких правил классификации из данных SGERD (Steady-State Genetic Algorithm for Extracting Fuzzy Classification Rules From Data) имеет элитарную селекцию и специфические операторы скрещивания и мутации. Несомненным достоинством алгоритма является его способность генерировать короткие и хорошо интерпретируемые правила [19]. В методе TARGET (Tree Analysis with Randomly Generated and Evolved Trees) генетический алгоритм применяется для поиска в пространстве деревьев решений, оценка каждого дерева основана на байесовском информационном критерии. Полученные деревья обладают большей интерпретируемостью по сравнению с ансамблевыми методами при сопоставимой точности [20].

В табл. 1 для каждого из алгоритмов указаны усреднённые значения процента правильной классификации на четырёх наборах данных для обучающей и тестовой выборок.

Классификаторы, оптимизированные АКП, имеют хорошие способности к обучению (высокий процент правильной классификации на обучающей выборке) и не менее хорошие прогностические способности (высокий процент правильной классификации на тестовой выборке).

Исследование алгоритма при решении задач аппроксимации проводилось по схеме кроссвалидации на четырёх наборах данных: Diabetes, DEE, ELE-2, Quake — из репозитория KEEL. Эффективность алгоритма «кукушкин поиск» сравнивалась с алгоритмами, решающими ту же задачу аппроксимации; далее приведены их краткие описания [21].

COR-BWAS — алгоритм оптимизации на основе муравьиной колонии, ориентированный на достижение компромисса между точностью и интерпретируемостью полученных правил.

Thrift — это метод обучения нечёткого аппроксиматора типа Мамдани, основанный на питтсбургском представлении нечёткой системы.

Таблица 1

Набор данных		Алгоритм					Наш алгоритм
		Ant Miner	CORE	HIDER	SGERD	TARGET	
balance	обучающая	73,65	68,64	75,86	76,96	77,29	<b>89,88</b>
	тестовая	70,24	70,08	69,60	75,19	75,62	<b>87,74</b>
pima	обучающая	71,86	72,66	<b>77,82</b>	73,71	73,42	76,25
	тестовая	66,28	73,06	<b>73,18</b>	73,71	73,02	70,87
bupa	обучающая	<b>80,38</b>	61,93	73,37	59,13	68,86	71,80
	тестовая	57,25	61,97	65,83	57,89	65,97	<b>68,48</b>
iris	обучающая	97,26	95,48	97,48	97,33	93,50	<b>98,63</b>
	тестовая	96,00	92,67	<b>96,67</b>	<b>96,67</b>	92,93	<b>96,67</b>

Таблица 2

Набор данных		Алгоритм					
		COR-BWAS	Thrift	Fuzzy-GAP	Pitts-DNF med	Pitts-DNF max	Наш алгоритм
Diabetes	обучающая	0,17496	0,07448	0,14292	0,12958	0,10656	<b>0,06358</b>
	тестовая	1,45869	0,87825	0,50141	<b>0,32134</b>	0,63396	0,38932
DEE	обучающая	0,12463	0,38778	0,17751	0,13821	<b>0,11267</b>	0,19444
	тестовая	<b>0,20513</b>	0,45830	0,20633	0,27465	0,21692	0,20628

Таблица 3

Алгоритм	Число правил	Обучающая выборка	Тестовая выборка
ELE-2			
TS-NSGA-II [22]	41	14488	18419
TS-SPEA2 [22]	33	13272	17533
Pulkkinen [22]	24,9	9366	10429
TS[23]	41,3	13387	17784
TS-SP2-St [23]	21,7	17619	22099
Наш алгоритм	41,3	11404	12023
Наш алгоритм	98,6	8972	9611
Quake			
TS [23]	33,5	0,0173	0,0428
TS-SP2-St [23]	27,2	0,0173	0,0182
Наш алгоритм	27,0	0,0174	0,0188

Метод Fuzzy-GAP генерирует сложные нечёткие правила, используя гибридную модель генетических программирования и алгоритма.

Алгоритмы Pitts-DNF med и Pitts-DNF max предназначены для поиска компромисса между точностью и интерпретируемостью в нечётких системах. Антецеденты нечётких правил этих систем содержат помимо операции конъюнкции ещё и операцию дизъюнкции. Префикс Pitts свидетельствует о питтсбургском представлении нечёткой системы. Суффиксы med и max задают ограничения на количество правил — среднее или максимальное.

В табл. 2, 3 представлены усреднённые значения MSE и MSE/2. На наборе данных ELE-2 точность нашего алгоритма при сравнимом количестве правил уступает алгоритму Pulkkinen и сопоставима с точностью алгоритмов-аналогов. Для достижения более высокой точности приходится увеличивать число правил.

Результаты экспериментов показывают, что точность аппроксимации нечётких аппроксиматоров, идентифицированных алгоритмом «кукушкин поиск», сопоставима с точностью аппроксиматоров, построенных известными лучшими алгоритмами.

**Заключение.** В предлагаемой работе рассмотрена процедура построения нечётких систем на основе алгоритма «кукушкин поиск». Работоспособность нечётких классификаторов и аппроксиматоров, настроенных приведённым алгоритмом, проверена на восьми наборах данных из репозитория KEEL. Сравнения с аналогами показали достаточно высокую точность АКП при решении задачи оптимизации параметров нечётких систем. Таким образом, алгоритм «кукушкин поиск» может быть рекомендован для практическо-

го применения при решении задач оптимизации параметров нечётких классификаторов и аппроксиматоров.

### СПИСОК ЛИТЕРАТУРЫ

1. **Takagi T., Sugeno M.** Fuzzy identification of systems and its application to modeling and control // *IEEE Trans. Syst., Man, and Cybern.* 1985. **15**, N 1. P. 116–132.
2. **Sugeno M., Yasukawa T.** A fuzzy-logic-based approach to qualitative modeling // *IEEE Trans. Fuzzy Syst.* 1993. **1**, N 1. P. 7–31.
3. **Herrera F.** Genetic fuzzy systems: taxonomy, current research trends and prospects // *Evolutionary Intelligence.* 2008. **1**, N 1. P. 27–46.
4. **Ходашинский И. А., Горбунов И. В.** Алгоритмы поиска компромисса между точностью и сложностью при построении нечётких аппроксиматоров // *Автометрия.* 2013. **49**, № 6. С. 51–61.
5. **Ходашинский И. А., Синьков Д. С.** Применение гибридного квантового алгоритма роящихся частиц для идентификации параметров нечётких аппроксиматоров // *Информатика и системы управления.* 2013. **36**, № 2. С. 56–63.
6. **Juang C.-F., Chang P.-H.** Designing fuzzy-rule-based systems using continuous ant-colony optimization // *IEEE Trans. Fuzzy Syst.* 2010. **18**, N 1. P. 138–149.
7. **Ходашинский И. А., Дудин П. А., Лавыгина А. В.** Биоинспирированные методы параметрической идентификации нечетких моделей // *Докл. ТУСУР.* 2007. **16**, № 2. С. 81–92.
8. **Yang X.-S., Deb S.** Engineering optimisation by cuckoo search // *Intern. Journ. Math. Modelling and Numerical Optimisation.* 2010. **1**, N 4. P. 330–343.
9. **Yang X.-S., Deb S.** Cuckoo search: recent advances and applications // *Neural Comput. and Appl.* 2014. **24**, N 1. P. 169–174.
10. **Civicioglu P., Besdok E.** A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms // *Artificial Intell. Rev.* 2013. **39**, N 4. P. 315–346.
11. **Mantegna R. N.** Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes // *Phys. Rev. E.* 1994. **49**, N 5. P. 4677–4683.
12. **Ходашинский И. А., Горбунов И. В.** Построение нечётких классификаторов на основе алгоритма пчелиной колонии // *Матер. Всерос. конф. с междунар. участием «Знания-Онтологии-Теории» (ЗОНТ-2011).* Новосибирск: ИМ СО РАН, 2011. Т. 2. С. 117–125.
13. **KEEL.** URL: <http://www.keel.es> (дата обращения: 09.06.2014).
14. **Alcala-Fdez J., Fernandez A., Luengo J. et al.** KEEL data-mining software tool: data, set repository, integration of algorithms and experimental analysis framework // *Journ. Multiple-Valued Logic and Soft Comput.* 2011. **17**, N 2–3. P. 255–287.
15. **Parpinelli R. S., Lopes H. S., Freitas A. A.** Data mining with an ant colony optimization algorithm // *IEEE Trans. Evolutionary Computation.* 2002. **6**, N 4. P. 321–332.
16. **Dorigo M., Gambardella L. M.** Ant colony system: A cooperative learning approach to the traveling salesman problem // *IEEE Trans. Evolutionary Computation.* 1997. **1**, N 1. P. 53–66.
17. **Tan K. C., Yu Q., Ang J. H.** A coevolutionary algorithm for rules discovery in data mining // *Intern. Journ. Syst. Sci.* 2006. **37**, N 12. P. 835–864.
18. **Aguilar-Ruiz J. S., Riquelme J. C., Toro M.** Evolutionary learning of hierarchical decision rules // *IEEE Trans. Syst., Man, and Cybern. Pt. B: Cybernetics.* 2003. **33**, N 2. P. 324–331.
19. **Mansoori E. G., Zolghadri M. J., Katebi S. D.** SGERD: A steady-state genetic algorithm for extracting fuzzy classification rules from data // *IEEE Trans. Fuzzy Syst.* 2008. **16**, N 4. P. 1061–1071.

20. **Gray J. B., Fan G.** Classification tree analysis using TARGET // Computat. Stat. & Data Anal. 2008. **52**, N 3. P. 1362–1372.
21. **Casillas J., Martinez P., Benitez A. D.** Learning consistent, complete and compact sets of fuzzy rules in conjunctive normal form for regression problems // Soft Comput. 2009. **13**, N 5. P. 451–465.
22. **Pulkkinen P., Koivisto H.** A dynamically constrained multiobjective genetic fuzzy system for regression problems // IEEE Trans. Fuzzy Syst. 2010. **18**, N 1. P. 161–177.
23. **Gacto M. J., Alcalá R., Herrera F.** Integration of an index to preserve the semantic interpretability in the multiobjective evolutionary rule selection and tuning of linguistic fuzzy systems // IEEE Trans. Fuzzy Syst. 2010. **18**, N 3. P. 515–531.

*Поступила в редакцию 9 июня 2014 г.*

---