

УДК 004.82

И. А. Ходашинский, д-р техн. наук, проф.,

hodashn@rambler.ru,

И. В. Горбунов, аспирант,

noby.Ardor@gmail.com,

Томский государственный университет
систем управления и радиоэлектроники

Оптимизация параметров нечетких систем на основе модифицированного алгоритма пчелиной колонии¹

Представлен анализ использования алгоритма пчелиной колонии для оптимизации параметров нечетких систем. Выполнена оценка влияния параметров на результаты работы алгоритма. Выявлена зависимость между временем работы алгоритма и его параметрами. Работоспособность и эффективность алгоритма успешно подтверждены на серии имитационных экспериментов и рядом тестов.

Ключевые слова: нечеткие системы, оптимизация параметров, алгоритм пчелиной колонии

Введение. Нечеткие системы находят широкое применение при решении задач автоматического управления, распознавания образов, прогнозирования, принятия решений и др. Общепризнанной проблемой при построении нечетких систем является формирование нечеткой базы правил [1]. Часто только экспертного знания бывает недостаточно для автоматизации проектирования нечетких систем. В последнее время для этих целей применяются методы роевого интеллекта, в частности метод рояющихся частиц [2–4], методы муравьиной колонии [5–7] и перемещения бактерий [8].

В данной работе для оптимизации параметров нечеткого аппроксиматора применяется алгоритм пчелиной колонии. Алгоритм пчелиной колонии (АПК) — сравнительно новый, но хорошо зарекомендовавший себя инструмент решения сложных, нелинейных проблем оптимизации, основанный на моделировании специфического поведения роя медоносных пчел [9]. Для оптимизации параметров нечетких систем этот алгоритм используется не часто. В работе [10] АПК используется для обучения рекуррентной нейронной сети на основе нечеткой системы вывода. Авторы работы [11] при-

менили АПК для настройки параметров нечеткого регулятора. В работе [12] АПК используется для настройки нечетких классификаторов на основе таблиц наблюдений.

Нечеткая система. Базовая концепция нечеткого моделирования заключается в использовании степени принадлежности, которая является эффективным средством описания поведения плохо формализованных объектов, систем и процессов. Построение нечетких моделей возможно на основе наблюдаемых данных, а также с использованием априорного знания и опыта. Основная задача нечеткого моделирования заключается в нахождении конечного множества локальных отношений вход-выход, которые описывают систему или процесс в виде нечетких "IF — THEN" правил.

Предметная область может быть описана посредством лингвистических переменных и правил естественного языка, содержащих качественную оценку ситуации. Основой для описания ситуации является нечеткое высказывание следующего вида:

$$x_i \text{ есть } A_i \text{ или } x_i = A_i,$$

где x_i — некоторая величина; A_i — элемент терм-множества лингвистической переменной из исследуемой предметной области [13].

Кратко опишем работу обобщенной нечеткой системы. На вход поступает вектор четких значений входных переменных, которые преобразуются в нечеткие значения (этап фазификации). С помощью нечеткой базы правил и машины логического вывода из фазифицированных значений формируется функция принадлежности выходной переменной, которая операцией дефазификации преобразуется в действительное число.

Нечеткие правила имеют следующую структуру:

IF <нечеткое описание входных переменных>
THEN <описание выходной переменной>.

В зависимости от того, как представлено описание выходной переменной, различают три основных типа нечетких систем "много входов — один выход".

Нечеткая система типа *синглтон* задается правилами следующего вида:

$$\begin{aligned} \text{IF } x_1 = A_{1i} \text{ AND } x_2 = A_{2i} \text{ AND } \dots \\ \dots \text{ AND } x_n = A_{ni} \text{ THEN } y = r_i, \end{aligned} \quad (1)$$

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

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 12-07-00055).

Модель типа синглтон осуществляет отображение $f: \mathbb{R}^n \rightarrow \mathbb{R}$, заменяя оператор нечеткой конъюнкции произведением, а оператор агрегации нечетких правил — сложением. Отображение f определяется следующей формулой:

$$f(\mathbf{x}) = \sum_{i=1}^R \mu_{A1i}(x_1) \mu_{A2i}(x_2) \cdot \dots \cdot \mu_{Ani}(x_n) r_i / \sum_{i=1}^R \mu_{A1i}(x_1) \mu_{A2i}(x_2) \cdot \dots \cdot \mu_{Ani}(x_n),$$

где \mathbf{x} — входной вектор, R — число правил; n — число входных переменных; μ_{Aij} — функция принадлежности i -й входной переменной.

В нечеткой системе типа Такаги—Сугено правая часть правила задается как линейная функция, определенная на входных переменных. Нечеткая система типа Мамдани в правой части правила содержит нечеткое описание выходной переменной.

Постановка задачи. Рассматривается нечеткая система типа синглтон, которая может быть представлена как

$$y = f(\mathbf{x}, \theta),$$

где $\theta = \|\theta_1, \dots, \theta_M\|$ — вектор параметров нечеткой системы, y — скалярный выход системы.

Пусть дано множество обучающих данных (таблица наблюдений) $\{(\mathbf{x}_p, t_p), p = 1, \dots, m\}$, тогда среднеквадратические функции ошибки, являющиеся численным критерием адекватности модели, вычисляются по формулам

$$MSE(\theta) = \frac{\sum_{p=1}^m (t_p - f(\mathbf{x}_p, \theta))^2}{m}, \quad (2)$$

$$RMSE(\theta) = \sqrt{\frac{\sum_{p=1}^m (t_p - f(\mathbf{x}_p, \theta))^2}{m}}. \quad (3)$$

Проблема оптимизации параметров нечеткой системы сводится к поиску минимума заданной функции в многомерном пространстве, координаты которого соответствуют параметрам нечеткой системы. В силу того, что поверхность поиска в указанном пространстве может иметь сложный рельеф, методы поиска, основанные на производных, здесь не всегда эффективны. Для решения проблемы минимизации предлагается использовать метаэвристический АПК.

Алгоритм пчелиной колонии относится к классу методов, получивших название "роевой интеллект" и сочетающих в себе биологические и социальные эвристики. Основной задачей данного направления является создание децентрализованных самоорганизующихся систем с приложениями в робототехнике, кибернетике, информатике. Роевой интеллект основан на обратной связи, адаптации к изменяющейся окружающей среде и децентрализованном

взаимодействии между особями. С вычислительной точки зрения роевые алгоритмы — это стохастические методы поиска, в которых эффективно сочетаются нахождение новых решений и улучшение существующих решений. Указанные свойства позволяют избежать ранней сходимости к локальному решению и найти глобальный оптимум. Идея этих метаэвристик основана на том, что каждая особь находится в пространстве поиска возможных решений задачи. Особь меняет свое положение, основываясь на знании трех факторов: текущего положения или приспособленности, предыдущих состояний данной особи, предыдущих состояний соседей данной особи. Для поведения роя характерны следующие признаки: автономность, распределение функций и самоорганизация [14, 15].

Разработка формальных средств оптимизации основана на таких аспектах поведения медоносных пчел в природе, как коммуникативный танец, поиск пищи, коллективное принятие решений, спаривание и выбор местоположения нового улья.

При создании алгоритмов оптимизации наиболее часто имитируют коммуникативный танец и поиск пищи, в частности, эти аспекты поведения пчел лежат в основе таких алгоритмов, как Bees Algorithm [11], Bee colony optimization [16], Artificial Bee Colony (ABC) [17], Virtual Bee Algorithm [18], Bee Hive algorithm [19].

Для описания поведения пчел в природе используются следующие понятия: источник нектара, пчелы-фуражиры, пчелы-разведчики, пчелы-наблюдатели. Источник нектара характеризуется значимостью, определяемой различными факторами, такими как удаленность от улья, концентрация нектара, удобство добычи нектара. Пчелы-фуражиры закреплены за отдельным источником, на котором они добывают нектар. Фуражиры владеют такой информацией о данном источнике нектара, как расстояние и направление от улья, полезность источника. Разведчики ищут новые источники нектара. Пчелы-наблюдатели могут полететь к источнику нектара, следя за пчелой-разведчиком. Это достигается за счет того, что каждый улей имеет так называемую закрытую площадку для танца, на которой пчелы, обнаружившие источники нектара, выполняют специфический танец, пытаясь привлечь других, незанятых пчел последовать за ними. Механизмы, в соответствии с которыми пчела решает следовать за другой пчелой, исследованы недостаточно, но предполагается, что вербовка пчел с математической точки зрения является функцией качества источника нектара [15].

Представленный далее в статье алгоритм основан на поведении пчел во время добычи пищи и соединяет в себе поисковый подход алгоритма ВСО [16], особенности поиска каждого типа пчел алгоритма ABC [17] и элементы их лучших модификаций, таких как вербовка в соответствии с методами селекции генетического алгоритма и выбор решения методом имитации отжига. Алгоритм применяется для

настройки нечеткого аппроксиматора. В качестве источника нектара выступают параметры Θ нечеткого правила (1), пригодность которых оценивается среднеквадратической ошибкой.

Алгоритм пчелиной колонии для настройки нечеткого аппроксиматора (АПКНА) настраивает антецеденты (IF-части) нечетких правил, оптимизация консеквентов (THEN-части) выполняется методом наименьших квадратов (МНК) [20].

Ниже приведен собственно алгоритм оптимизации параметров нечеткой системы.

Вход: число итераций *IterMax*, требуемая точность *E*, тип алгоритма формирования популяции *Alg*, тип алгоритма инициализации нечеткой базы правил *Init*.

Выход: вектор Θ оптимальных параметров нечеткой системы.

BS — случайные векторы-решения пчел разведчиков.

W — архив решений.

best — лучший вектор-решение.

NW — векторы-решения, формируемые пчелами-фуражирами на основе архива решений.

NB — векторы-решения, формируемые пчелами-наблюдателями на основе **best**-решения.

Шаг 1. Инициализация базы правил нечеткой системы алгоритмом *Init*. Сохранение вектора-решения в **BS**.

Шаг 2. Поиск нектара пчелами-разведчиками — случайный поиск параметров антецедентов на всей области решений задачи. Найденные решения сохраняются в **BS**.

Шаг 3. Оптимизация консеквентов правил из **BS** алгоритмом МНК.

Шаг 4. Танец пчел-разведчиков — формирование алгоритмом отжига из **BS** решений **W**.

Шаг 5. Добыча нектара пчелами-фуражирами — локальный поиск параметров антецедентов на основе архива **W**. Сохранение решений в **NW**.

Шаг 6. Добыча нектара пчелами-наблюдателями — локальный поиск параметров антецедентов на основе лучших решений из архива **W** и **best**-решения. Сохранение решений в **NB**.

Шаг 7. Формирование нового архива решений **W** алгоритмом *Alg* из векторов **NW**, **NB**, **W**.

Шаг 8. ЕСЛИ точность *E* достигнута или превышено число итераций *IterMax*

ТО поиск лучшего решения в **W**, сохранение его как Θ , **ВЫХОД**

ИНАЧЕ переход на шаг 2.

В качестве алгоритма *Init* можно использовать инициализацию антецедентов равномерным покрытием, методом экстремумов, методом диффузии и соответствующей инициализацией консеквентов методом ближайшего соседа или МНК [21].

В качестве алгоритма *Alg* можно использовать любой оператор селекции генетического алгоритма. В данной работе используются следующие опера-

торы: бинарные турниры, рулетка, элитарный отбор и случайный выбор.

Исследование алгоритма. Исследование алгоритма оптимизации параметров нечетких систем проводили при решении задач аппроксимации. В качестве тестовых функций были выбраны следующие:

$$1) f(x_1, x_2) = x_1 \sin(x_2), -\pi/2 < x_1, x_2 < \pi/2;$$

$$2) f(x_1, x_2) = \sin(2x_1/\pi) \sin(2x_2/\pi), -5 < x_1, x_2 < 5;$$

$$3) f(x) = (1 + 10 \exp(-100(x - 0,7)^2)) \left(\frac{\sin\left(\frac{125}{x+1,5}\right)}{x+0,1} \right), \\ 0 < x < 1;$$

$$4) f(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1,5})^2, x_1, x_2 \in [1, 5];$$

$$5) f(x_1, x_2, x_3) = 1 + x_1^{0,5} + x_2^{-1} + x_3^{-1,5}, \\ x_1, x_2, x_3 \in [1, 5].$$

На основе каждой тестовой функции формировали таблицу наблюдений, по которой строили нечеткую систему, аппроксимирующую данную функцию. Критерием качества аппроксимации была среднеквадратическая ошибка вывода.

По умолчанию нечеткая система при исследовании алгоритма пчелиной колонии имеет следующие параметры: равномерное покрытие пятью треугольными термами каждой входной переменной, кроме пятой тестовой функции, переменные которой покрыты тремя термами; на этапе инициализации функции принадлежности двух соседних треугольных термов пересекаются на уровне 0,5. Параметры алгоритма по умолчанию следующие: начальная температура в алгоритме отжига равна 70, коэффициент охлаждения 0,95, алгоритм формирования популяции — бинарные турниры, размер улья — 100, процент пчел разведчиков — 50, число итераций — 100. Для каждого набора параметров тест выполнен 10 раз.

В процессе исследования АПКНА изменяли следующие параметры алгоритма: число итераций, размер улья, процент пчел-разведчиков, начальная температура в алгоритме отжига, коэффициент охлаждения, алгоритм формирования популяции.

На рис. 1 показано динамическое распределение среднеквадратической ошибки (3) лучшего решения в зависимости от процента пчел-разведчиков в улье. При малом проценте пчел-разведчиков алгоритм работает быстрее, однако и ошибка решения в этом случае большая. Большой процент пчел-разведчиков обеспечивает быстрое нахождение перспективных решений, однако в силу того, что пчел-фуражиры недостаточно, улучшить эти решения не всегда удается. Когда пчел-разведчиков в улье приблизительно столько же, сколько и пчел-фуражиров, алгоритм находит лучшие решения.

Были исследованы четыре алгоритма формирования новой популяции: tournament — бинарные турниры, roulette — рулетка, elite — элитарный и

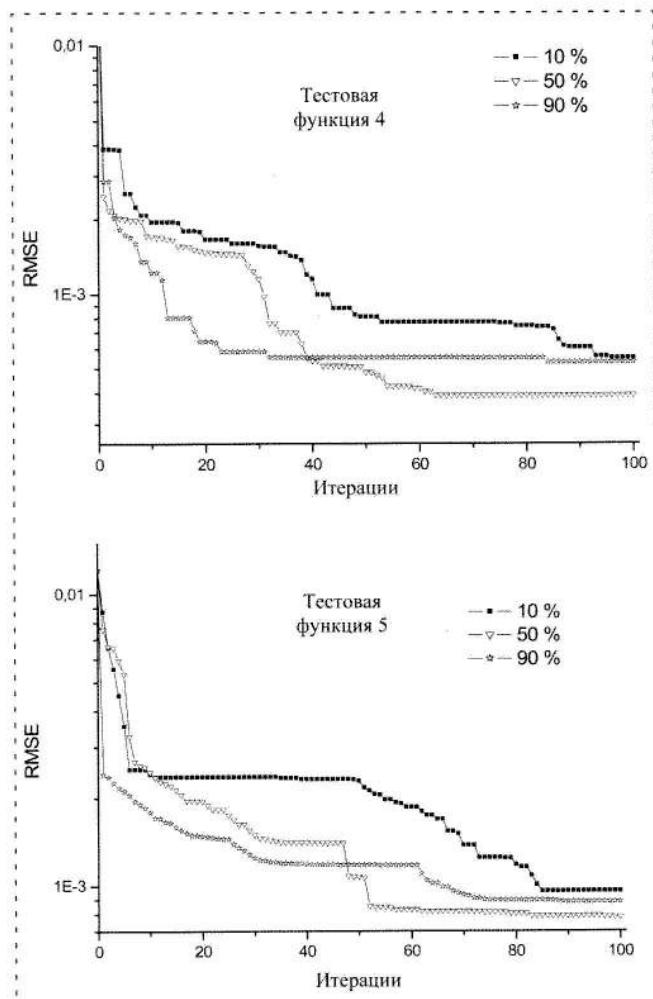


Рис. 1. Динамика изменения ошибки в зависимости от числа разведчиков в популяции

random — случайный. На рис. 2 показано динамическое распределение среднеквадратической ошибки лучшего решения в зависимости от алгоритма формирования популяции. Выделить лучший алгоритм формирования популяции на данных тестовых функциях не представляется возможным. Худшие позиции занимает алгоритм формирования популяции методом случайной выборки. Для настройки нечеткого аппроксиматора немультимодальных тестовых функций (типичный представитель — тестовая функция 1) требуется не более 50 итераций. А для аппроксимации мультимодальных тестовых функций (функция 3) необходимо более 90 итераций.

Размер популяции играет важную роль в роевых алгоритмах, оказывая влияние на точность и время вычисления. Малая популяция приводит к сходению в локальный оптимум, большая популяция увеличивает время вычисления и ухудшает сходимость. Эксперимент показал, что при аппроксимации второй и третьей тестовых функций этот параметр оказывает влияние только на значениях до 100, для четвертой и пятой тестовых функций влияние параметра незначительно. Разумный компромисс между точностью и временем выполнения для тес-

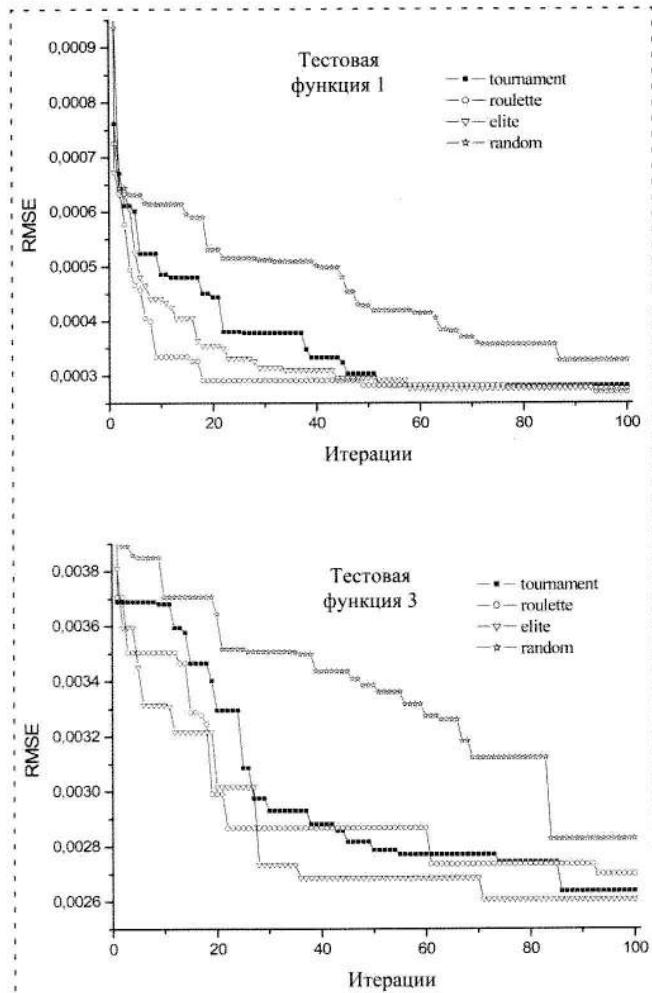


Рис. 2. Динамика изменения ошибки в зависимости от алгоритма формирования популяции

тируемых функций достигается в среднем при 100 единицах.

Эксперимент показал, что такие параметры, как начальная температура и коэффициент охлаждения, существенного влияния на качество решения не оказывают.

Таблица 1
Зависимость времени оптимизации нечеткой системы
от параметров алгоритма

Функция	Время от числа итераций i	Время от размера улья s	Время от процента p числа разведчиков в улье
1	$-0,425486 + 0,02661i$ ($R^2 = 0,9986$)	$0,05s$ ($R^2 = 1$)	$0,05p$ ($R^2 = 1$)
2	$-0,1406 + 0,02622i$ ($R^2 = 0,9981$)	$0,05s$ ($R^2 = 1$)	$0,05p$ ($R^2 = 1$)
3	$-0,4778 + 0,007833i$ ($R^2 = 0,9775$)	$-0,21428 + 0,013575s$ ($R^2 = 0,9377$)	
4	$-0,32678 + 0,02623i$ ($R^2 = 0,9988$)	$0,05s$ ($R^2 = 1$)	$0,05p$ ($R^2 = 1$)
5	$-0,535 + 0,05385i$ ($R^2 = 0,9997$)	$-0,47024 + 0,1068s$ ($R^2 = 0,9991$)	$0,1p$ ($R^2 = 1$)

Компьютерный эксперимент позволил установить, что зависимость времени настройки нечеткой системы от параметров алгоритма носит линейный характер. В табл. 1 приведены регрессионные модели зависимости времени оптимизации нечеткой системы, в скобках указаны квадраты коэффициентов корреляции R^2 как мера точности полученных зависимостей.

Сравнение разработанного алгоритма с аналогами. Для сравнения разработанного алгоритма с существующими подходами построения нечетких баз правил было проведено исследование результатов аппроксимации нескольких нелинейных функций. Сравнение проводили со следующими алгоритмами: прямой алгоритм муравьиной колонии [5], непрерывный алгоритм муравьиной колонии [7], алгоритм Mitaim-Kosko [22], алгоритм Lisin-Gennert [22], алгоритм Teng-Wang-Chiu [23], алгоритм Aliyari-Teshnehlab-Sedigh [24]. Критерием сравнения выступала ошибка $RMSE$ (3). Результаты аппроксимации показали превосходство АПКНА над всеми другими алгоритмами для функций 2, 3, 4 со значениями $RMSE$ 3,86E-04; 2,17E-03; 4,33E-04, соответственно. АПКНА уступил по результатам аппроксимации функции 5 прямому алгоритму муравьиной колонии, показав результат 5,72E-04 против 4,93E-06.

Исследование было проведено также на реальных данных, представленных в репозитории KEEL (Knowledge Extraction Evolutionary Learning) [25]. Характеристики аппроксимируемых данных представлены в табл. 2. Все входные и выходные переменные — вещественные числа. Каждая выборка, кроме *Kinematics*, разделена на пять наборов, из которых строятся обучающая и тестовая выборки, содержащие 80 % и 20 % данных, соответственно. Для *Kinematics* разделение было на десять наборов, в каждом по 90 % и 10 % для обучающей и тестовой выборок, соответственно. Разделение проводилось

таким образом, чтобы каждый набор попал во все тестовые выборки ровно один раз.

Результаты работы АПКНА сравнивали с результатами других алгоритмов глобальной оптимизации [26], такими как Wang-Mendel, COR-BWAS, Thrift, Pittsburgh, Fuzzy-GAP, Pitts-DNF min, Pitts-DNF med, Pitts-DNF max. Критерием сравнения выступала ошибка MSE (2). На выборке *Diabetes* по результатам аппроксимации обучающей выборки лучшим был Thrift с результатом 0,0745 против результата АПКНА 0,1235, но среднее число правил для Thrift 46,2, что говорит о том, что на каждый образец обучающей выборки приходится более одного правила. Именно это привело к переобучению. На тестовой выборке ошибка аппроксимации алгоритма Thrift составила 0,8783, что более чем на порядок больше показателя обучающейся выборки. АПКНА показал лучший результат со значением 0,1018. При обработке данных *ELE 2* АПКНА показал лучшие результаты со значениями 41 778 и 10 571 для обучающей и тестовой выборок, соответственно. На выборке *DEE* АПКНА показал худший результат среди конкурентов с результатом 0,48662 против лучшего результата 0,11267, показанного Pitts-DNF max. Однако на тестовой выборке АПКНА показал результат 0,13767 против результата COR-BWAS, равного 0,20513.

Результаты исследований, проведенных на наборах данных *Delta ailerons* и *Kinematics*, сравнивали с результатами алгоритмов для задачи глобальной оптимизации, основанных на подходах, отличных от нечетких систем. По большей части здесь представлены нейронные сети и статистические методы и SVR подход. Описание каждого из алгоритмов можно найти в соответствующих статьях: SMOTI, MS, RETIS [27]; SLFN sigmoid, SLFN RBF, SLFN sin, RAN, MRAN [28]; SVR [29]; FF-SVESMSs, SVR-SMO [30].

В работах [28–30] для расчета $RMSE$ использовали формулу

$$RMSE = \sqrt{\frac{\sum_{p=1}^m (t_p - f(x_p, \theta))^2}{m}},$$

поэтому для проведения сравнения значения $RMSE$, приведенные в указанных работах, были умножены на значение $\frac{1}{\sqrt{m}}$, где m — число образцов в наборе.

По обучающей выборке *Delta ailerons* АПКНА уступил лишь MS, показав результат $RMSE$ 2,5304E-06 против 2,4886E-06. На тестовой выборке АПКНА показал лучший результат, равный 6,9734E-06. На самой большой и по числу данных и входных параметров выборке *Kinematics* АПКНА на обучающей выборке показал средний из всех сравниваемых алгоритмов результат 2,3131E-03 против лучшего результата SMOTI 1,8411E-03. Сравнение по тестовой выборке не проводили,

Таблица 2
Описание данных

Название данных	Число образцов	Число входных переменных	Краткое описание
<i>DEE</i>	365	6	Прогнозирование среднесуточной цены на электроэнергию
<i>Delta Ailerons</i>	7129	5	Формирование управляющего воздействия на элероны воздушного судна
<i>Diabetes</i>	43	2	Прогнозирование развития сахарного диабета у инсулиновых детей
<i>ELE2</i>	1066	4	Прогнозирование оценки стоимости обслуживания городских электрических сетей
<i>Kinematics</i>	8192	8	Прогнозирование расстояния от датчика робота до цели

в силу отсутствия этих значений у сравниваемых алгоритмов.

Результаты экспериментов показали, что модифицированный АПК имеет высокую прогностическую способность. По результатам видно, что АПКНА в некоторых тестах показывает лучшие результаты, по другим более скромные, но в подавляющем большинстве случаев входит в тройку лидеров. Особого внимания заслуживает факт устойчивости к переобучению, во многом полученный благодаря сбалансированному числу правил по сравнению с аналогами, позволяющий победить на тестовой выборке и показать неплохие результаты на обучающей выборке.

Заключение

В работе описан эффективный алгоритм оптимизации параметров нечетких систем, основанный на моделировании поведения медоносных пчел. Эксперименты, проведенные на классических тестовых задачах, а также сравнительный анализ показали, что предложенный алгоритм имеет достаточно хорошие характеристики по сравнению с другими методами формирования нечетких баз правил. Особо отметим его высокие прогностические возможности.

Список литературы

1. Espinosa J., Vandewalle J., Wertz V. Fuzzy logic, identification and predictive control. London: Springer-Verlag, 2005. 263 p.
2. Zhao L., Qian F., Yang Y., Zeng Y., Su H. Automatically extracting T-S fuzzy models using cooperative random learning particle swarm optimization // Applied Soft Computing. 2010. V. 10. N 3. P. 938–944.
3. Ходашинский И. А. Идентификация параметров нечетких моделей типа синглтон на основе алгоритма роящихся частиц // Информационные технологии. 2009. № 6. С. 8–11.
4. Ходашинский И. А., Сильков Д. С. Идентификация параметров нечетких систем на основе адаптивного алгоритма роящихся частиц // Информационные технологии. 2011. № 8. С. 2–5.
5. Ходашинский И. А., Дудин П. А. Идентификация нечетких систем на основе прямого алгоритма муравьиной колонии // Искусственный интеллект и принятие решений. 2011. № 3. С. 26–33.
6. Juang C.-F., Chang P.-H. Designing Fuzzy-Rule-Based Systems Using Continuous Ant-Colony Optimization // IEEE Transactions Fuzzy Systems. 2010. V. 18. N 1. P. 138–149.
7. Ходашинский И. А., Дудин П. А. Идентификация нечетких систем на основе непрерывного алгоритма муравьиной колонии // Автометрия. 2012. Т. 48. № 1. С. 63–71.
8. Ходашинский И. А., Земцов Н. Н., Мещеряков Р. В. Построение нечетких аппроксиматоров на основе метода перемещения бактерий // Известия вузов. Физика. 2012. Т. 55. № 3. С. 57–61.
9. Karaboga D., Akay B. A survey: algorithms simulating bee swarm intelligence // Artificial Intelligence Review. 2009. V. 31. P. 61–85.
10. Khanmirzaei Z., Teshnehab M. Prediction Using Recurrent Neural Network Based Fuzzy Inference System by the Modified Bees Algorithm // International Journal of Advancements in Computing Technology. 2010. V. 2. P. 42–55.
11. Pham D. T., Darwish A. H., Eldukhri E. E. Optimisation of a Fuzzy Logic Controller using the Bees Algorithm // International Journal of Computer Aided Engineering and Technology. 2009. V. 1. P. 250–264.
12. Ходашинский И. А., Горбунов И. В. Построения нечетких классификаторов на основе алгоритма пчелиной колонии // Матер. Всеросс. Конф. с междунар. участием ЗОНТ-2011. М.: Институт математики им. С. Л. Соболева, 2011. Т. 2. С. 117–125.
13. Ходашинский И. А. Оценивание величин: подход на основе мягких вычислений // Информационные технологии. 2006. № 4. С. 13–21.
14. Karaboga D., Akay B. A survey: algorithms simulating bee swarm intelligence // Artificial Intelligence Review. 2009. V. 31. P. 61–85.
15. Alatas B. Chaotic bee colony algorithms for global numerical optimization // Expert Systems with Applications. 2010. V. 37. P. 5682–5687.
16. Teodorović D. Bee Colony Optimization (BCO) // Innovations in Swarm Intelligence. Heidelberg: Springer-Verlag, 2009. P. 39–60.
17. Karaboga D., Basturk B. On the performance of artificial bee colony (ABC) algorithm // Applied Soft Computing. 2008. V. 8. P. 687–697.
18. Yang X.-S. Engineering optimizations via nature-inspired virtual bee algorithms // IWINAC 2005, LNCS 3562. Berlin: Springer-Verlag, 2005. P. 317–323.
19. Wedde H. F., Farooq M., Zhang Y. BeeHive: an efficient fault-tolerant routing algorithm inspired by honey bee behavior // Ant colony optimization and swarm intelligence, LNCS 3172. Berlin: Springer-Verlag, 2004. P. 83–94.
20. Катаев М. Ю., Лавыгина А. В., Ходашинский И. А., Эштейн Д. А. Нечеткий аппроксиматор атмосферных температурных полей // Автометрия. 2010. Т. 46. № 2. С. 39–48.
21. Ходашинский И. А. Идентификация нечетких систем: методы и алгоритмы // Проблемы управления. 2009. № 4. С. 15–23.
22. Lisin D., Gennert M. A. Optimal Function Approximation Using Fuzzy Rules // Proc. 18th Int. Conf. North American Fuzzy Information Processing Society. 1999. P. 184–188.
23. Teng Y.-W., Wang W.-J., Chiu C.-H. Function approximation via particular input space partition and region-based exponential membership functions // Fuzzy Sets and Systems. 2004. V. 142. P. 267–291.
24. Aliyari M., Teshnehab M. Sh., Sedigh A. K. Novel Hybrid Learning Algorithms for Tuning ANFIS Parameters Using Adaptive Weighted PSO // IEEE International Conference on Fuzzy Systems. 2007. P. 111–116.
25. Data-mining software Tool: data, set repository. URL: <http://www.keel.es>.
26. Alcala-Fdez J., Fernandez A., Luengo J., Derrac J., Garcia S., Sanchez L., Herrera F. KEEL Data-mining software Tool: data, set repository, integration of algorithms and experimental analysis framework // Journal of Multiple-Valued Logic and Soft Computing. 2011. V. 17. P. 255–287.
27. Malerba D., Esposito F., Ceci M., Appice A. Top-Down Induction of Model Trees with Regression and Splitting Nodes // IEEE Transactions on pattern analysis and machine intelligence. 2004. V. 26. N 6. P. 1–13.
28. Guang-Bin H., Lei C., Chee-Kheong S. Universal Approximation Using Incremental Constructive Feedforward Networks With Random Hidden Nodes // IEEE Transactions on Neural Networks. 2006. V. 17. N 4. P. 879–892.
29. Yongping Z., Jianguo S. Improved scheme to accelerate support vector regression // Journal of Systems Engineering and Electronics. 2009. V. 20. N 5. P. 1086–1090.
30. Zhiwei S., Min H. γ -C plane and robustness in static reservoir for nonlinear regression estimation // Neurocomputing Advances in Machine Learning and Computational Intelligence. 2009. V. 72. P. 1732–1743.