

УДК 004.8.023 : 681.51.015

АЛГОРИТМЫ ПОИСКА КОМПРОМИССА МЕЖДУ ТОЧНОСТЬЮ И СЛОЖНОСТЬЮ ПРИ ПОСТРОЕНИИ НЕЧЁТКИХ АППРОКСИМАТОРОВ*

И. А. Ходашинский, И. В. Горбунов

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

Рассмотрены два важных этапа в построении нечётких аппроксиматоров: генерация структуры и оптимизация параметров. Предложены два критерия оптимизации: точность, измеренная среднеквадратической ошибкой, и сложность, выраженная числом нечётких правил. Приведены результаты исследований полученных аппроксиматоров на реальных данных из репозитория KEEL, проведено сравнение результатов с аналогами.

Ключевые слова: нечёткий аппроксиматор, генерация структуры, оптимизация параметров, метаэвристики.

Введение. Проблема оценки неизвестной функции на основе таблицы наблюдений является одним из ключевых вопросов в области моделирования нечётких систем (НС) и теории аппроксимации функций. Популярность и практичность НС объясняются следующими причинами:

- 1) НС могут быть идентифицированы путём объединения наблюдаемых данных и знаний эксперта;
- 2) природа нечётких правил позволяет описать поведение моделируемой системы в терминах причинно-следственных отношений;
- 3) НС являются универсальными аппроксиматорами, способными представить любую непрерывную нелинейную функцию с любой степенью точности [1].

При этом к НС, построенной на базе реальных данных, выдвигаются два основных требования:

- 1) система должна точно воспроизводить информацию из анализируемой таблицы наблюдений и обладать высокими обобщающими способностями;
- 2) система должна быть представлена в формате, понятном пользователю, и помогать выявить наиболее существенные зависимости и соотношения между входными и выходными переменными, т. е. нечёткие правила могут быть интерпретированы пользователем в контексте данного приложения.

Интерпретируемость можно трактовать как возможность описать поведение моделируемой системы в понятной форме. Это желательное свойство для всех типов приложений, но оно является особенно важным требованием для систем, основанных на знаниях, с человеко-машинным взаимодействием, например систем поддержки принятия решений. Базы знаний таких систем должны быть понятны пользователям для повышения доверия к системам, к их советам и предложениям. Специалисты в области нечёткого моделирования оценивают интерпретируемость либо через сложность, либо через семантику [2–5].

*Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 12-07-00055) и Российского гуманитарного научного фонда (проект № 12-06-12008).

Семантика подразумевает придание смысла функциям принадлежности, когда каждому нечёткому терму можно предписать осмысленное лингвистическое значение, например «очень малое», «малое», «среднее», «большое», «очень большое». Сложность выражается через число правил, переменных, нечётких термов. Отметим, что несложная НС более проста в настройке, требует меньше памяти и времени вывода, чем более сложная НС. Оба названных критерия — точность и интерпретируемость — являются противоречивыми и должны быть учтены при проектировании НС.

В основе построения НС лежит многокритериальная оптимизация, направленная на отыскание решения, одновременно оптимизирующего более чем одну целевую функцию. В этом случае ищется компромиссное решение, недоминируемое никаким другим решением, или оптимальное в смысле Парето. После построения области (фронта) парето-лицу, принимающему решение, предоставляется возможность выбора единственного из парето-оптимальных решений.

Существует достаточно много публикаций по построению двухкритериальных (точность—сложность) нечётких классификаторов, в которых решается задача поиска компромисса между обоими критериями, сошлёмся здесь лишь на [2, 3]. Работ по построению двухкритериальных нечётких аппроксиматоров (НА) значительно меньше, и в их основе лежит подход к многокритериальной оптимизации с помощью эволюционных алгоритмов [1, 4–6].

В данной работе компромисс между точностью и сложностью предлагается искать в два этапа: сначала генерируется структура аппроксиматора заданной сложности, затем проводится оптимизация его параметров роевыми алгоритмами. Так как сложность в нашей работе принимает целочисленные значения, ограничив это значение сверху, можно сложность указывать в качестве входного параметра оптимизируемой системы, что позволяет решать поставленную проблему как задачу одноцелевой оптимизации.

Цель предлагаемой работы — описание методов и средств, учитывающих компромисс между точностью и сложностью при разработке нечётких аппроксиматоров.

Постановка задачи. Точность — это способность адекватно представить реальную систему. Общепринятой мерой точности является среднеквадратическая ошибка. Сложность — субъективное свойство, связанное с такими факторами, как структура модели, число входных переменных, число нечётких правил, число и форма нечётких термов лингвистических переменных и т. д. Не существует универсального способа измерения сложности моделей [2, 3, 7].

В данной работе сложность определена как число нечётких правил. Кроме того, на НС наложены следующие ограничения:

- 1) число термов для каждой входной переменной находится в разумных пределах (от 2 до 9);
- 2) функции принадлежности (ФП) нечётких термов выпуклы и нормализованы;
- 3) область определения полностью покрыта;
- 4) функции принадлежности различимы, т. е. две ФП не принимают близких значений на области определения;
- 5) используются глобально определённые ФП;
- 6) в базе нет правил с одинаковыми антецедентами, но различными консеквентами.

Поскольку точность и сложность являются противоречивыми критериями, генерируется не одна НС, а набор таких систем. Из сгенерированного набора выбирается множество недоминируемых решений, обозначенное как парето-множество, характерное свойство которого — оптимальное соотношение между критериями точности и сложности.

Нечёткий аппроксиматор задаётся правилами следующего вида:

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

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

Аппроксиматор осуществляет отображение $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$ следующим образом:

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

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 среднеквадратические функции ошибки

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

Проблема построения парето-оптимального множества НА с разными соотношениями между значениями их точности и сложности основана на оптимизации заданной функции в многомерном пространстве, координаты которого соответствуют параметрам $\boldsymbol{\theta}$. Целевая функция $\text{MSE}(\boldsymbol{\theta})$ определена на пространстве поиска $\Omega \subseteq \mathfrak{R}^N$, причём $\boldsymbol{\theta}_{\min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_{\max}$, где $\boldsymbol{\theta}_{\min}$, $\boldsymbol{\theta}_{\max}$ — нижняя и верхняя границы параметров решения.

Оптимизация проводится итерационно: сначала генерируется заданная структура аппроксиматора, затем проводится оптимизация его параметров, если выполнено условие окончания построения НА, то работа заканчивается, в противном случае генерируется очередная структура и работа продолжается.

Главными задачами предлагаемого исследования являются разработка алгоритмов генерации структуры и оптимизации параметров НА, позволяющих достичь компромисса между сложностью и точностью, а также экспериментальные исследования и сравнительный анализ полученных результатов с аналогами.

Генерация структуры. Существует три основных способа задания структуры на базе разделения входного пространства наблюдаемых данных [8].

1. Разделение на основе сетки (grid partition). Все входные переменные разделены на заданное число областей, каждая из которых описывается своим нечётким термом. База правил формируется с использованием декартова произведения множеств входных нечётких термов. Достоинство — простота идентификации и интерпретации нечёткой системы; главный недостаток — экспоненциальный рост числа правил с увеличением числа входных переменных.

2. Разделение на основе дерева поиска (tree partition). Пространство входных переменных итерационно разбивается на непересекающиеся гиперпрямоугольники, ортогональные осям координат. В центре каждого прямоугольника помещается центр ФП нечёткого термина, размах функции принадлежности (альфа-срез) пропорционален размеру гиперпрямоугольника. Достоинство алгоритма заключается в хорошем разделении ФП. Недостаток состоит в том, что разбиение входного пространства ортогонально осям и может требовать весьма большого количества правил для аппроксимации функций со сложным рельефом.

3. Разделение с перекрытием (scatter partition). Этот способ отличается от предыдущего тем, что области разделения могут иметь формы, отличные от гиперпрямоугольника, и они могут пересекаться. В зависимости от формы выделяют три типа разделения: гиперпараллелепипед, эллипсоид, многогранник. Достоинством данного подхода является меньшее число генерируемых правил, недостатком — сложность процедур реализации.

В представленной работе на этапе генерации структуры задаётся сложность системы, определяемая количеством нечётких правил. Для генерации структур предлагается использовать следующие методы и алгоритмы:

- 1) метод нечёткой кластеризации;
- 2) алгоритм генерации базы правил НС равномерным разбиением и перебором;
- 3) алгоритм генерации базы правил НС исключением неэффективных правил;
- 4) алгоритм генерации базы правил с заданной структурой.

Хорошо известный итерационный алгоритм нечёткой кластеризации, относящийся к группе алгоритмов разделения с перекрытием, основан на минимизации функционала

$$J_2 = \sum_{i=1}^m \sum_{j=1}^c u_{ij}^2 d^2(\mathbf{x}_i, \mathbf{c}_j),$$

где u_{ij} — степень принадлежности вектора наблюдений \mathbf{x}_i j -му кластеру; $d(\mathbf{x}_i, \mathbf{c}_j)$ — расстояние Махаланобиса между вектором \mathbf{x}_i и центром j -го кластера \mathbf{c}_j (необходимая для вычисления расстояния матрица ковариации определяется по данным таблицы наблюдений, попавшим в j -й кластер); c — число кластеров; m — объём таблицы наблюдений. На каждой итерации u_{ij} и \mathbf{c}_j изменяются следующим образом:

$$u_{ij} = 1 / \sum_{h=1}^c (d_{ij}/d_{ih})^2, \quad \mathbf{c}_j = \sum_{i=1}^m u_{ij}^2 \mathbf{x}_i / \sum_{i=1}^m u_{ij}^2.$$

Алгоритм прекращает работу на k -й итерации при выполнении условия

$$\max_{ij} |u_{ij}^k - u_{ij}^{k-1}| < \varepsilon,$$

где ε — критерий останова.

Полученные кластеры преобразуются в нечёткие термы, количество таких термов, относящихся к каждой переменной, фиксировано и равно количеству кластеров [9].

Алгоритм генерации равномерным разбиением и перебором основан на сеточном разделии. Здесь генерируется заданное количество термов на каждую входную переменную, покрывающих всю область определения переменной. Из полученных термов путём полного перебора сочетаний термов по разным входным переменным формируются antecedentes правил, консеквенты формируются методом ближайшего соседа.

Алгоритм генерации базы правил исключением неэффективных правил убирает из базы те правила, которые минимально влияют на точность, формирует нечёткие системы с малой ошибкой при невысокой сложности, не превышающей 50, но при больших сложностях системы он практически бесполезен.

В разработанном нами алгоритме генерации базы правил с заданной структурой указывается общая сложность системы без точного указания разбиения входных переменных на нечёткие термы. Алгоритм перебирает варианты разбиения и рассчитывает ошибку данной версии системы. Выходом алгоритма является база правил, соответствующая структуре и имеющая минимальную ошибку из сгенерированных вариантов. Алгоритм

генерирует аппроксиматор с ошибкой, сравнимой с ошибкой алгоритма генерации базы правил перебором при меньшем количестве нечётких термов и правил в базе. В меньшей степени он подвержен влиянию эффекта «проклятия размерности» по сравнению с алгоритмом генерации базы правил НС перебором.

Оптимизация параметров. Для настройки antecedентов правил в работе используются метаэвристики, описанные далее, а для оптимизации консеквентов правил применяется метод наименьших квадратов (МНК) [10].

Классический и модифицированный алгоритмы роящихся частиц (КАРЧ и МАРЧ) [11, 12]. Классический алгоритм роящихся частиц — это стохастический метод поиска, основанный на итеративном взаимодействии частиц, образующих рой [13]. Перемещение частицы в пространстве поиска определяют три фактора: инерция, память (лучшая позиция частицы), сотрудничество (известна лучшая позиция частиц в рое).

В нашем случае пространство поиска $\theta \subset \mathbb{R}^N$, рой состоит из M частиц. Позиция i -й частицы задаётся вектором $\mathbf{x}_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{iN}) \in \theta$. Лучшая позиция, которую занимала i -я частица, определяется вектором $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{iN}) \in \theta$, а скорость частицы — N -местным вектором $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iN}) \in \theta$. Скорость и положение i -й частицы на $(k + 1)$ -м шаге вычисляются из следующих уравнений:

$$\mathbf{v}_i(k + 1) = w\mathbf{v}_i(k) + c_1 \text{rand}(\mathbf{p}_i(k) - \mathbf{x}_i(k)) + c_2 \text{rand}(\mathbf{p}_g(k) - \mathbf{x}_i(k)),$$

$$\mathbf{x}_i(k + 1) = \mathbf{x}_i(k) + \mathbf{v}_i(k + 1),$$

где $i = 1, 2, \dots, M$; $\mathbf{v}_i(k)$ — вектор скорости частицы i на итерации k ; $\mathbf{x}_i(k)$ — координаты частицы i на итерации k ; c_1, c_2 — положительные коэффициенты ускорения; $\mathbf{p}_i(k)$ — лучшая позиция частицы i на первых k итерациях; $\mathbf{p}_g(k)$ — глобально лучшая позиция частицы в рое на первых k итерациях (задаётся индексом g); w — эмпирический коэффициент инерции; rand — случайное число из интервала $[0, 1]$.

Непрерывный алгоритм муравьиной колонии (НАМК) [14]. Алгоритм предложен в [15], где для выбора пути используется функция плотности вероятности с гауссовым ядром $G^i(x)$, которое основано на взвешенной сумме нескольких одномерных гауссовых функций g_l^i :

$$G^i(x) = \sum_{l=1}^k \omega_l g_l^i(x) = \sum_{l=1}^k \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-(x - \theta_l^i)^2 / (2\sigma_l^{i2})}.$$

Каждому параметру НС соответствует своё гауссово ядро, $i = 1, \dots, N$ (N — число идентифицируемых параметров). Каждая функция $G^i(x)$ описывается тремя векторами: $\theta^i = \{\theta_1^i, \theta_2^i, \dots, \theta_k^i\}$, ω — вектор весов, связанных с индивидуальными гауссовыми функциями, и σ^i — вектор среднеквадратичных отклонений, который вычисляется следующим образом:

$$\sigma_l^i = \xi \sum_{j=1}^k \frac{|\theta_j^i - \theta_l^i|}{k - 1}.$$

Параметр $\xi > 0$ вносит эффект, подобный норме испарения феромона в дискретном алгоритме муравьиной колонии. Чем больше значение ξ , тем ниже скорость сходимости алгоритма [15]. В непрерывном алгоритме вводится архив решений, представленный таблицей из k строк. Каждая строка состоит из трёх частей:

- 1) найденное муравьём решение $\theta_l = \{\theta_l^1, \theta_l^2, \dots, \theta_l^N\}$;
- 2) ошибка, вычисленная нечёткой системой;
- 3) вес решения ω_l .

Решения упорядочены в архиве по возрастанию ошибки. Вес ω_l решения θ_l вычисляется согласно формуле

$$\omega_l = \frac{1}{qk\sqrt{2\pi}} e^{-(l-1)^2/(2q^2k^2)},$$

где q — задаваемый параметр алгоритма.

При добавлении нового решения в архив худшее из них удаляется. Этот процесс аналогичен процессу испарения феромона в классическом алгоритме.

Прямой алгоритм муравьиной колонии (ПАМК) [16]. Алгоритм предложен в [17], муравей в данном алгоритме отвечает за вычисление значений закреплённого за ним параметра. Каждый i -й муравей создаёт своё решение, генерируя нормально распределённое действительное число $N(\mu_i, \sigma_i)$. В алгоритме используются два вида феромонов: первый связан с центрами нормальных распределений $\mu = \|\mu_1, \dots, \mu_N\|$, второй — с разбросом $\sigma = \|\sigma_1, \dots, \sigma_N\|$. Количество феромона определяет значения параметров μ и σ . Для каждого параметра θ_j задан интервал изменения $[a_j, b_j]$, где b_j и a_j — верхняя и нижняя граница параметра θ_j .

В качестве начальных значений для параметров μ используются заданные случайным или иным способом значения параметров θ . Начальные значения параметров σ вычисляются по формуле

$$\sigma_i = \frac{b_i - a_i}{2}.$$

После того как муравьи нашли решения, определяется испарение феромона. Для текущей t -й итерации это выражения

$$\mu(t) = (1 - \rho)\mu(t - 1), \quad \sigma(t) = (1 - \rho)\sigma(t - 1),$$

где ρ — эмпирический коэффициент испарения феромона, заданный на интервале $[0, 1]$.

Далее происходит нанесение феромона:

$$\mu(t) = \mu(t) + \rho\theta(t), \quad \sigma(t) = \sigma(t) + \rho|\theta(t) - \mu(t)|,$$

где $\theta(t)$ — решение, найденное муравьиной колонией на текущей итерации, оно совпадает с глобальным лучшим решением.

Для преодоления локальных минимумов в алгоритме используется обновление параметров σ . В этих целях введён параметр конвергенции, вычисляемый по следующей формуле:

$$cf = \sum_{j=1}^N \frac{2\sigma_j}{b_j - a_j} / N.$$

Когда алгоритм приближается к локальному минимуму, коэффициент конвергенции cf приближается к 0. Как только коэффициент конвергенции становится меньше критического значения cf , вектор σ возвращается в начальное состояние.

Алгоритм пчелиной колонии (АПК) [18]. Алгоритм имеет множество модификаций и основан на процессе добычи нектара пчёлами, в нашей модификации количество нектара

обратно пропорционально $MSE(\theta)$, а вербовка пчёл производится в соответствии с методами селекции генетического алгоритма. Алгоритм настраивает antecedенты нечётких правил, оптимизация консеквентов выполняется МНК [9].

Приведём собственно алгоритм оптимизации параметров нечёткой системы.

Вход: количество итераций $IterMax$, требуемая точность E , вид алгоритма формирования популяции Alg , вид алгоритма инициализации нечёткой базы правил $Init$.

Выход: вектор θ оптимальных параметров нечёткой системы.

Примем обозначения переменных: **BS** — случайные векторы-решения (нечёткие базы правил) пчёл-разведчиков; **W** — архив решений; **best** — лучший вектор-решение (нечёткая база правил); **NW** — векторы-решения (нечёткие базы правил), формируемые на основе архива решений пчёлами-фуражирами; **NB** — векторы-решения (нечёткие базы правил), формируемые на основе **best**-решения пчёлами-наблюдателями.

Шаг 1. Поиск нектара пчёлами-разведчиками — случайный поиск параметров antecedентов на всей области решений задачи. Найденные решения сохраняются в **BS**.

Шаг 2. Оптимизация консеквентов правил из **BS** по алгоритму МНК.

Шаг 3. Танец пчёл-разведчиков — отсеивание решений в **BS** алгоритмом отжига. Оставшиеся правила добавляются в **W**.

Шаг 4. Добыча нектара пчёлами-фуражирами — локальный поиск параметров antecedентов на основе **W**. Сохранение решений в **NW**.

Шаг 5. Добыча нектара пчёлами-наблюдателями — локальный поиск параметров antecedентов на основе лучших из **W** и **best**-решения. Сохранение решений в **NB**.

Шаг 6. Формирование нового архива решений **W** алгоритмом Alg из векторов **NW**, **NB**, **W**.

Шаг 7. ЕСЛИ точность E достигнута или превышено количество итераций $IterMax$, ТО поиск лучшего решения в **W**, сохранение его как θ , ВЫХОД, ИНАЧЕ переход на шаг 1.

Эксперимент. Для решения задачи поиска компромисса между точностью и сложностью нами были построены гибридные алгоритмы, включающие изложенные выше алгоритмы генерации структуры и оптимизации параметров нечётких аппроксиматоров. Исследование алгоритмов проводилось при решении задач аппроксимации данных из репозитория KEEL (Knowledge Extraction based on Evolutionary Learning) [19]. Характеристики данных представлены в табл. 1. Все входные и выходные переменные — вещественные числа. Каждая выборка разделена на пять наборов, из которых строятся обучающая и тестовая выборки, содержащие 80 и 20 % данных соответственно. Разделение проводилось таким образом, чтобы каждое наблюдение попало во все тестовые выборки ровно 1 раз. На одной и той же выборке настройка нечёткого аппроксиматора проводилась 5 раз.

Эффективность разработанных нами гибридных алгоритмов сравнивалась с алгоритмами, решающими те же задачи аппроксимации. Для этого из полученных парето-множеств недоминируемых решений были отобраны аппроксиматоры с лучшими сочетаниями точности и сложности. Цифры, характеризующие другие алгоритмы, были взяты из [4–6] и сведены в табл. 2 и 3, в которых первые шесть строк — разработанные нами гибридные алгоритмы. Здесь MSE — среднее значение ошибки, вычисленное по формуле (2) и усреднённое по пяти выборкам; CKO — среднеквадратическое отклонение от среднего значения количества правил, от среднего значения ошибки MSE обучающей и тестовой выборок. В работах [4, 6] ошибка представлена как $MSE/2$, поэтому в табл. 2 и 3 оригинальное значение умножено на 2. Из таблиц видно, что аппроксиматоры, построенные с помощью наших алгоритмов, дают лучшие результаты по точности на всех выборках при сопоставимых сложностях.

Таблица 1

Название данных	Количество наблюдений	Количество входных переменных	Краткое описание
Diabetes	43	2	Прогнозирование развития сахарного диабета у инсулинозависимых детей
ELE2	1066	4	Проблема оценки стоимости обслуживания городских электрических сетей

Таблица 2

Алгоритм	Правила		Обучающая выборка		Тестовая выборка	
	Число	СКО	MSE	СКО	MSE	СКО
Diabetes						
Генерация равномерным разбиением и НАМК	25	0	0,02502	0,0116	0,5963	0,1918
Генерация равномерным разбиением и ПАМК	9	0	0,1156	0,0058	0,2315	0,0151
Нечёткая кластеризация и МНК	3	0	0,3367	0,033	0,3911	0,1206
Генерация базы правил исключением неэффективных правил и МАРЧ	12	0	0,01472	0,0123	0,1533	0,1216
Генерации базы правил с заданной структурой и КАРЧ	12	0	0,0442	0,0149	0,152	0,077
Генерация базы правил исключением неэффективных правил и АПК	20	0	0,0183	0,01036	0,043	0,0234
Wang — Mendel [5]	18,6	1,4	0,2284	0,0425	1,4024	0,6890
COR — BWAS [5]	18,6	1,4	0,1750	0,0250	1,4587	0,7091
Thrift [5]	46,2	0,7	0,0745	0,0098	0,8783	0,3575
Pittsburgh [5]	15	2,9	0,1040	0,0182	0,9509	0,7881
Fuzzy — GAP [5]	10	0	0,1429	0,0376	0,5014	0,3014
Pitts — DNF min [5]	1,6	0,5	0,4162	0,1231	0,4539	0,1288
Pitts — DNF med [5]	5,4	0,5	0,1296	0,0136	0,3213	0,1922
Pitts — DNF max [5]	9,6	1,2	0,1066	0,0150	0,6340	0,5276

Таблица 3

Алгоритм	Правила		Обучающая выборка		Тестовая выборка	
	Число	СКО	MSE	СКО	MSE	СКО
ELE2						
Генерация равномерным разбиением и НАМК	37,3	12,3	13510	2611	17235	2420
Генерация равномерным разбиением и ПАМК	30,8	10,7	15656	1871	18458	1951
Нечёткая кластеризация и МНК	3	0	339902	9848	349418	45178
Генерация базы правил исключением неэффективных правил и АПК	4	0	29427	6464	24385	13567
Генерации базы правил с заданной структурой и КАРЧ	24	0	26668	3359	24141	5907
Генерация базы правил исключением неэффективных правил и МАРЧ	20,2	8,0	33236	6973	34852	8743
Wang — Mendel [5]	65	0	112270	1498	112718	4685
COR — BWAS [5]	65	0	102664	1080	102740	4321
Thrift [5]	524,6	6,4	146305	12991	168472	20135
Pittsburgh [5]	240	21,1	210717	32027	265130	30161
Fuzzy — GAP [5]	33	0	279166	90017	290062	89155
Pitts — DNF min [5]	12,2	0,7	202943	43684	212018	44616
Pitts — DNF med [5]	18,6	1,4	86930	3955	99310	12996
Pitts — DNF max [5]	32,4	6,6	70207	1658	88017	8968
DynMO GFS [4]	25	—	18732	—	20858	—
NSGA — ILRB [6]	29	—	34230	—	39668	—
NSGA — ILKB [6]	29	—	26272	—	31174	—
PAES_RB [6]	30	—	15453	—	30906	—
PAES_KB [6]	30	—	22086	—	25212	—

Заключение. В данной работе предлагается подход к поиску компромисса между точностью и сложностью. Реализуется подход в два этапа: сначала генерируется структура аппроксиматора заданной сложности, затем проводится оптимизация его параметров ровными алгоритмами. Так как сложность в нашей работе принимает целочисленное значение, ограничив его сверху, можно сложность указывать в качестве входного параметра оптимизируемой системы, что позволяет решать поставленную проблему как задачу одноцелевой оптимизации.

Сильные стороны предлагаемого подхода:

1) пользователь заранее может определить желаемый уровень сложности, все получаемые нечёткие системы будут отвечать данному требованию;

2) поскольку сложность не является оптимизируемым параметром, это позволяет использовать алгоритмы однокритериальной оптимизации.

Недостаток данного подхода заключается в его ориентации только на дискретное описание критерия сложности нечёткой системы.

СПИСОК ЛИТЕРАТУРЫ

1. **Gonzalez J., Rojas I., Pomares H. et al.** Improving the accuracy while preserving the interpretability of fuzzy function approximators by means of multi-objective evolutionary algorithms // Intern. Journ. Approximate Reasoning. 2007. **44**, N 1. P. 32–44.
2. **Fazzolari M., Alcalá R., Nojima Y. et al.** A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions // IEEE Trans. Fuzzy Systems. 2013. **21**, N 1. P. 45–65.
3. **Gacto M. J., Alcalá R., Herrera F.** Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures // Inform. Sci. 2011. **181**, N 20. P. 4340–4360.
4. **Pulkkinen P., Koivisto H.** A dynamically constrained multiobjective genetic fuzzy system for regression problems // IEEE Trans. Fuzzy Systems. 2010. **18**, N 1. P. 161–177.
5. **Casillas J., Martínez P., Benitez A. D.** Learning consistent, complete and compact sets of fuzzy rules in conjunctive normal form for regression problems // Soft Computing. 2009. **13**, N 5. P. 451–465.
6. **Alcalá R., Ducange P., Herrera F. et al.** A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems // IEEE Trans. Fuzzy Systems. 2009. **17**, N 5. P. 1106–1122.
7. **Mencar C., Castiello C., Cannone R., Fanelli A. M.** Interpretability assessment of fuzzy knowledge bases: A cointension based approach // Intern. Journ. Approximate Reasoning. 2011. **52**, N 4. P. 501–518.
8. **Ho S.-Y., Chen H.-M., Chen T.-K.** Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space // IEEE Trans. Syst., Man, Cybern. B. 2004. **34**, N 2. P. 1031–1044.
9. **Ходашинский И. А.** Идентификация нечетких систем: методы и алгоритмы // Проблемы управления. 2009. № 4. С. 15–23.
10. **Ходашинский И. А.** Идентификация нечётких систем на базе алгоритма имитации отжига и методов, основанных на производных // Информационные технологии. 2012. № 3. С. 14–20.
11. **Ходашинский И. А.** Идентификация параметров нечетких моделей типа синглтон на основе алгоритма роящихся частиц // Информационные технологии. 2009. № 6. С. 8–11.
12. **Ходашинский И. А., Синьков Д. С.** Идентификация параметров нечётких систем на основе адаптивного алгоритма роящихся частиц // Информационные технологии. 2011. № 8. С. 2–5.
13. **Kennedy J., Eberhart R.** Particle swarm optimization // Proc. of the IEEE Intern. Conf. Neural Networks. Perth: IEEE Service Center, 1995. P. 1942–1948.
14. **Ходашинский И. А., Дудин П. А.** Идентификация нечётких систем на основе непрерывного алгоритма муравьиной колонии // Автометрия. 2012. **48**, № 1. С. 63–71.
15. **Socha K., Dorigo M.** Ant colony optimization for continuous domains // Europ. Journ. Operational Res. 2008. **185**, N 3. P. 1155–1173.
16. **Ходашинский И. А., Дудин П. А.** Идентификация нечётких систем на основе прямого алгоритма муравьиной колонии // Искусственный интеллект и принятие решений. 2011. № 3. С. 26–33.

-
17. **Kong M., Tian P.** Application of ACO in continuous domain // Proc. of the 2nd Intern. Conf. on Advances in Natural Computation (ICNC 2006). Pt. II: Lecture Notes in Computer Science. Vol. 4222. Berlin: Springer-Verlag, 2006. P. 126–135.
 18. **Ходашинский И. А., Горбунов И. В.** Оптимизация параметров нечётких систем на основе модифицированного алгоритма пчелиной колонии // Мехатроника, автоматизация, управление. 2012. № 10. С. 15–20.
 19. **Knowledge** Extraction based on Evolutionary Learning. URL: <http://www.keel.es> (дата обращения: 6.03.2013).

Поступила в редакцию 6 марта 2013 г.
