



ИДЕНТИФИКАЦИЯ НЕЧЕТКИХ СИСТЕМ: МЕТОДЫ И АЛГОРИТМЫ¹

И.А. Ходашинский

Рассмотрены основные этапы построения нечетких моделей: экспертное оценивание, идентификация структуры, идентификация параметров. Отмечено, что экспертное оценивание включает в себя определение типа нечеткой модели, выбор t -нормальных функций для задания нечетких логических операций, выбор способов нечеткого вывода. Для идентификации структуры предложено применять два метода: нечеткий кластерный анализ и алгоритм перебора. Для идентификации параметров выбраны следующие алгоритмы: генетический, муравьиной колонии, роящихся частиц, имитации отжига. Приведены результаты имитационного эксперимента.

Ключевые слова: идентификация нечетких моделей, метаэвристики, имитация отжига, генетический алгоритм, алгоритм муравьиной колонии, алгоритм роящихся частиц.

ВВЕДЕНИЕ. ПОСТАНОВКА ЗАДАЧИ

Идентификация систем, в том числе и нечетких, играет принципиальную роль в теории управления. Нечеткая система может быть построена либо на основе знаний эксперта, либо на основе наблюдаемых данных, либо на совместном использовании знаний и данных. Идентификация рассматривается в настоящей работе как процесс построения модели, связанный с установлением закономерностей между входными и выходными переменными объекта.

Нечеткая модель определена как система с n входными переменными $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$, определенными на входной области рассуждений $DX = DX_1 \times DX_2 \times \dots \times DX_n$, и одной выходной переменной Y , определенной на выходной области рассуждений DY . Четкое значение, которое принимает входная переменная X_i , обозначается как x_i и как u для выходной переменной Y [1, 2].

Нечеткая область определения i -й входной переменной X_i обозначена как $FX_i = \{LX_{i,1}, \dots, LX_{i,p_i}\}$, где p_i — число лингвистических термов (нечетких значений), на которых определена входная переменная, $LX_{i,k}$ задает функцию принадлежности и имя k -го лингвистического терма. Аналогично, $FY = \{LY_1, \dots, LY_q\}$ — нечеткая область определе-

ния выходной переменной, q — число нечетких значений, LY_j — функция принадлежности и имя выходного лингвистического терма.

База правил в нечеткой системе типа Мамдани представляет собой множество нечетких правил вида

$$R_j: LX_{1,j_1} \text{ AND } \dots \text{ AND } LX_{n,j_n} \rightarrow LY_j. \quad (1)$$

Нечеткое j -е правило в системе типа синглтон имеет вид

$$R_j: LX_{1,j_1} \text{ AND } \dots \text{ AND } LX_{n,j_n} \rightarrow r_j, \quad (2)$$

где r_j — действительное число, которым оценивается выход u .

Нечеткое j -е правило в модели типа Такаги—Сугено имеет вид

$$R_j: LX_{1,j_1} \text{ AND } \dots \text{ AND } LX_{n,j_n} \rightarrow \\ \rightarrow r_{0j} + r_{1j}x_1 + r_{2j}x_2 + \dots + r_{nj}x_n,$$

здесь выход u оценивается линейной функцией.

Таким образом, нечеткая система осуществляет отображение $F: \mathfrak{R}^n \rightarrow \mathfrak{R}$. Важную роль в этом отображении играют функции принадлежности $\mu_{LX_{ij}}(x_i)$, указывающие степень принадлежности четкой переменной x_i нечеткому понятию LX_{ij} . Каждая функция принадлежности задается своим набором параметров, например, треугольная — тройкой параметров, трапециевидная — четверкой, гауссова и параболическая — двойкой.

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 06-08-00248).

Обучение системы ведется на основе таблиц наблюдений или тестовых функций $f(\mathbf{x})$. Необходимо так построить базу правил, чтобы ошибка вывода, определяемая как

$$\frac{1}{M} \sum_{i=1}^M |f(\mathbf{x}_i) - F(\mathbf{x}_i)|, \text{ или } \frac{1}{M} \sqrt{\sum_{i=1}^M (f(\mathbf{x}_i) - F(\mathbf{x}_i))^2},$$

$$\text{или } \max_{i \in M} |f(\mathbf{x}_i) - F(\mathbf{x}_i)|, \quad (3)$$

была минимальной, где M — число наблюдений.

В общем случае построение нечеткой системы состоит из следующих основных этапов [3]:

— *экспертное оценивание* — задание типа нечеткой системы и сопутствующих ему параметров;

— *идентификация структуры* — выбор переменных (\mathbf{X} , $D\mathbf{X}$, Y , DY , p_i , q) и нечетких правил R_j ;

— *идентификация параметров* — поиск оптимальных значений всех параметров, задействованных в нечеткой системе, т. е. нахождение значений консеквента (ТО-части правила) и параметров функций принадлежности в антецеденте (ЕСЛИ-части) каждого правила ($F\mathbf{X}_j$, $F\mathbf{Y}$) на основе заданных критериев качества и метода оптимизации выбранного критерия;

— *проверка правильности модели*.

Цель работы состоит в описании и сравнении методов, алгоритмов и эвристик реализации перечисленных этапов построения нечетких систем.

1. ЭКСПЕРТНОЕ ОЦЕНИВАНИЕ

Идентификация нечетких систем базируется на данных наблюдений, однако пока не представляется возможным исключить в этом процессе участие эксперта, который решает следующие задачи: определение типа нечеткой системы (синглтон, Мамдани или Такаги—Сугено); выбор t -нормальных функций для задания нечетких логических операций; выбор способов нечеткого вывода (для системы типа Мамдани: аппроксимация Мамдани либо формальнологический вывод). В связи с изложенным, в процесс построения нечеткой системы нами включен рассматриваемый этап.

Выбор типа системы определяется решаемой задачей. Если решается задача интерполяции или аппроксимации, причем точность является определяющим фактором, то выбор должен быть сделан в пользу нечеткой системы типа Такаги—Сугено.

Если решается задача извлечения знаний из данных (в виде лингвистических правил) или поиск ассоциативных связей во множестве данных, то для этих целей необходимо использовать нечеткую систему типа Мамдани. Несомненное достоинство таких моделей заключается в их понятности и интерпретируемости. Недостаток — в высо-

ких вычислительных затратах ведения нечеткого вывода.

Система типа синглтон может быть использована как для решения задач аппроксимации, так и извлечения знаний.

Нечеткая модель типа синглтон, задаваемая правилами (2), осуществляет отображение $F: \mathfrak{R}^n \rightarrow \mathfrak{R}$, заменяя оператор нечеткой конъюнкции произведением, а оператор агрегации нечетких правил — сложением. Отображение F определяется следующей формулой:

$$F(\mathbf{x}) = \frac{\sum_{i=1}^m \mu_{LX1i}(x_1) \mu_{LX2i}(x_2) \dots \mu_{LXni}(x_n) r_i}{\sum_{i=1}^m \mu_{LX1i}(x_1) \mu_{LX2i}(x_2) \dots \mu_{LXni}(x_n)},$$

где $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathfrak{R}^n$, m — число правил нечеткой модели, n — число входных переменных в модели, μ_{LXij} — функции принадлежности.

Отображение F для модели типа Такаги—Сугено определяется формулой:

$$F(\mathbf{x}) = \frac{\sum_{i=1}^m \mu_{LX1i}(x_1) \mu_{LX2i}(x_2) \dots \mu_{LXni}(x_n) (r_{0i} + r_{1i}x_1 + r_{2i}x_2 + \dots + t_{ni}x_n)}{\sum_{i=1}^m \mu_{LX1i}(x_1) \mu_{LX2i}(x_2) \dots \mu_{LXni}(x_n)}.$$

Для описания отображения входного вектора \mathbf{X} в выходное значение y в нечетких системах типа Мамдани применяются методы нечеткой логики, например, аппроксимация Мамдани или метод, основанный на формальном логическом доказательстве [4, 5], при этом эксперт должен задать тип оператора импликации.

2. ИДЕНТИФИКАЦИЯ СТРУКТУРЫ

Идентификация структуры — определение структурных характеристик нечеткой системы, таких как число нечетких правил, число лингвистических термов, на которое разбиты входные переменные. Такая идентификация выполняется с помощью нечеткого кластерного анализа либо метода перебора.

Применение алгоритмов *кластерного анализа* направлено на разделение общей генеральной совокупности на кластеры так, чтобы в каждом из них находились наиболее близкие по свойствам объекты. Нечеткая кластеризация является одним из наиболее интересных методов выявления возможных групп и проверки гипотез о структуре данных. Методы нечеткой кластеризации позволяют одному и тому же объекту принадлежать одновременно нескольким кластерам, но с различной степенью принадлежности. Обычно каждый кластер



характеризуется некоторым прообразом (прототипом), который описывается центром кластера и некоторой дополнительной информацией, например, размером и формой кластера. Существует множество методик кластеризации. В данной работе рассмотрен алгоритм нечетких c -средних [6].

Исходной информацией для кластеризации служит таблица наблюдений $\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_M\}$, где k -й элемент $\mathbf{t}_k = (x_k, y_k)$, $\mathbf{x}_k = [x_{k1}, x_{k2}, \dots, x_{kn}]$ — вектор входных значений, y_k — выходное значение, M — число наблюдений, n — число входных переменных, $x_k \in DX, y_k \in DY$. Пусть $P(\mathbf{T})$ — мощность множества \mathbf{T} , т. е. множество всех подмножеств множества \mathbf{T} . Нечеткое c -разбиение \mathbf{T} есть множество $\{A_i | A_i \in P(\mathbf{T}), 1 \leq i \leq c, 2 \leq c \leq M - 1\}$, c — число кластеров. Кластеры характеризуются матрицей нечеткого разбиения $\mathbf{F} = [\mu_{ki}]$, которая обладает следующими свойствами:

$$\forall_{k=1, \dots, M} \left(\sum_{i=1}^c \mu_{ki} = 1 \right), \quad \forall_{i=1, \dots, c} \left(0 < \sum_{k=1}^M \mu_{ki} < M \right). \quad (4)$$

Алгоритм нечетких c -средних основан на минимизации расстояния от наблюдаемых данных до центров кластеров. Евклидово расстояние вычисляется по формуле [6]:

$$\sum_{i=1}^c \sum_{k=1}^M (\mu_{ki})^m \|\mathbf{v}_i - \mathbf{t}_k\|^2,$$

где $\mathbf{v}_i = \frac{\sum_{k=1}^M (\mu_{ki})^m \mathbf{t}_k}{\sum_{k=1}^M (\mu_{ki})^m}$ — вектор центров нечетких кластеров, $m \in [1, \infty)$ — эмпирическое число, обычно равное двум.

Собственно алгоритм нечетких c -средних состоит в следующем [6].

Вход. Таблица наблюдений \mathbf{T} , число кластеров c , число m , параметр останова ϵ .

Выход. Матрица нечеткого разбиения \mathbf{F} и матрица центров кластеров \mathbf{V} .

Шаг 1. Случайным образом инициализировать матрицу нечеткого разбиения \mathbf{F} , удовлетворяющую условиям (4).

Шаг 2. Вычислить вектор центров кластеров \mathbf{v}_i .

Шаг 3. Рассчитать расстояние d_{ki} между k -м наблюдением из матрицы \mathbf{T} и i -м центром кластера $d_{ki} = \sqrt{\mathbf{t}_k - \mathbf{v}_i}$.

Шаг 4. Рассчитать очередное приближение матрицы \mathbf{F} ;

$$\text{если } d_{ki} > 0, \text{ то } \mu_{ki} = \left(d_{ki}^2 \sum_{j=1}^c \frac{1}{d_{kj}^2} \right)^{\frac{-1}{(m-1)}},$$

$$\text{если } d_{ki} = 0, \text{ то } \mu_{ki} = \begin{cases} 1, & j \neq i, \\ 0, & j = i, \end{cases} \quad j = \overline{1, c}.$$

Шаг 5. Если $\|F - F^*\|^2 < \epsilon$, то выход из алгоритма, иначе переход на шаг 2. Здесь F^* — матрица нечеткого разбиения, полученная на предыдущей итерации алгоритма.

В результате работы алгоритма вычисляется матрица центров кластеров $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c\}$, где каждое $\mathbf{v}_k = \{v_{k1}, v_{k2}, \dots, v_{kn}, v_{kn+1}\}$, v_{ki} — значение координаты i -й переменной в k -м кластере, v_{kn+1} — значение координаты выходной переменной в k -м кластере. Каждый выделенный кластер соответствует одному лингвистическому правилу базы правил нечеткой системы.

Каждый лингвистический терм, находящийся в antecedенте нечетких правил, задается своей функцией принадлежности. В качестве функции принадлежности выбрана функция гауссова типа, которая описывается формулой:

$$\mu(x) = \exp\left(-\frac{(x-a)^2}{2\sigma^2}\right).$$

Опираясь на результаты кластерного анализа, можно определить параметры функции принадлежности следующим образом:

$$a = v_{ij},$$

$$\sigma_{ij}^2 = \frac{\sum_{k=1}^M (\mu_{ki})^m (x_{kj} - v_{ij})^2}{\sum_{k=1}^M (\mu_{ki})^m}.$$

Далее, после определения необходимых параметров и построения функций принадлежности, идет формирование правил нечеткой базы. Нечеткое понятие $LX_{i,k}$ в правилах (1) и (2) соответствует суждению « X_i приблизительно равно v_{ij} ». Общий вид правила в модели типа синглтон будет иметь следующий вид:

$$R_j: X_1 \approx v_{j1} \text{ AND } \dots \text{ AND } X_n \approx v_{jn} \rightarrow v_{jn+1},$$

а в модели типа Мамдани —

$$R_j: X_1 \approx v_{j1} \text{ AND } \dots \text{ AND } X_n \approx v_{jn} \rightarrow y \approx v_{jn+1},$$

где v_{ji} — значение координаты i -й входной переменной в j -м кластере, v_{jn+1} — значение координаты выходной переменной в j -м кластере, $1 \leq j \leq c$. Таким образом, решена задача определения лингвистических термов и формирования базы нечетких правил.

Использование гауссовых функций принадлежности в нечетких правилах не единственно возможное. Методы и алгоритмы перехода от гауссовых функций к треугольным, трапециевидным и параболическим описаны в работе [7].

Кластерный анализ — это в конечном итоге метод выбора небольшого числа нечетких «репрезентативных» правил, описывающих пространство входных и выходных данных. Достоинство данного метода состоит в алгоритмическом разбиении входного пространства на фиксированное число кластеров, а значит, и на фиксированное число термов и правил. К недостаткам можно отнести отсутствие гарантии, что входное пространство должным образом представлено посредством центров кластеров. Кроме того, ряд авторов отмечают чувствительность алгоритма к шумам и выбросам в данных [8, 9].

Алгоритм перебора находит число термов, на которое необходимо разбить область определения входных переменных. Сначала генерируется один терм, охватывающий всю область определения входных переменных, определяется ошибка аппроксимации. Если ошибка превышает заранее заданную, то начинается последовательное разбиение области определения на большее число термов, пока не будет найдено такое число термов, при котором заданная ошибка будет достигнута.

Вход: число входных переменных, таблица наблюдений, заданная ошибка.

Выход: база правил.

Шаг 1. Задать начальное число термов для каждой переменной равным 1.

Шаг 2. Сформировать базу правил, при заданном числе термов.

Шаг 3. Найти выходное значение нечеткой системы.

Шаг 4. Найти ошибку вычислений.

Шаг 5. Если ошибка больше заданной, то увеличить число термов каждой переменной на 1 и перейти на шаг 2, иначе *Выход*.

У алгоритма перебора есть важное достоинство: функции принадлежности накрывают всю область определения входных переменных. Недостаток такой схемы — экспоненциальный рост числа правил с ростом числа входных переменных и лингвистических термов. Нечеткие системы с большим числом правил имеют более высокую точность, но более слабую вычислительную эффективность и интерпретируемость.

3. ИДЕНТИФИКАЦИЯ ПАРАМЕТРОВ НЕЧЕТКИХ МОДЕЛЕЙ

Начиная с ранних работ [1, 2], проблема идентификации параметров нечетких моделей решается методами оптимизации нелинейных функций. На этапе идентификации параметров происходит настройка параметров antecedentes и консеквентов нечетких правил путем оптимизации работы нечеткой системы по заданному критерию.

Трудности применения классических методов оптимизации нелинейных функций, такие как проблема локального экстремума и «проклятие

размерности», заставили специалистов обратиться к таким метаэвристическим подходам, как алгоритмы имитации отжига, генетические алгоритмы, алгоритмы муравьиной колонии.

Алгоритм имитации отжига основан на аналогии с процессом кристаллизации с минимальной энергией при охлаждении [10], в нем используется упорядоченный случайный поиск. Особенность алгоритма — допустимость принятия решений, приводящих к увеличению ошибки.

На начальном этапе работы алгоритма создается текущее решение, которое, при его изменении случайным образом, переходит в рабочее. Рабочее решение может опять вернуться в текущее либо перейти в лучшее при условии уменьшения энергии δE . Но рабочее решение может быть принято в качестве текущего, даже если его энергия превышает энергию текущего, в том случае, когда выполняется критерий допуска.

Критерий допуска основывается на следующем уравнении:

$$P(\delta E) = \exp(-\delta E/T).$$

При высокой температуре T плохие решения принимаются чаще, чем отбрасываются. При снижении температуры вероятность принятия худшего решения уменьшается. Таким образом, при высоких температурах алгоритм отжига выполняет поиск оптимального решения во всем диапазоне решений. При понижении температуры диапазон поиска уменьшается. Существует множество вариантов снижения температуры. В данной работе используется функция $T_{i+1} = \alpha T_i$, где α — коэффициент снижения температуры, $0 < \alpha < 1$.

Собственно алгоритм имитации отжига для параметрической идентификации нечетких систем состоит в следующем.

Вход: начальная температура, конечная температура, число итераций при одном значении температуры, функция изменения температуры.

Выход: оптимальные параметры функций принадлежности нечеткой системы.

Шаг 1. Генерируются значения параметров функций принадлежности нечеткой системы. Эти значения записываются в текущее решение и в лучшее решение.

Шаг 2. Значения параметров передаются нечеткой системе, которая на их основе находит значение выходной переменной. Это значение сравнивается с эталонным и вычисляется ошибка (ошибка текущего решения).

Шаг 3. Текущее решение копируется в рабочее, и рабочее решение изменяется случайным образом.

Шаг 4. Из рабочего решения значения параметров передаются нечеткой системе, которая на их основе находит значение выходной переменной. Это значение сравнивается с эталонным и вычисляется ошибка (ошибка рабочего решения).

Шаг 5. Сравниваются ошибки текущего и рабочего решения. Если ошибка рабочего решения меньше, чем



ошибка текущего решения, то переход к шагу 6. В противном случае, определяется критерий допуска. Если критерий допуска выполняется, то переход к шагу 6. В противном случае текущее решение копируется в рабочее, и осуществляется переход к шагу 7.

Шаг 6. Рабочее решение копируется в текущее, и ошибка текущего решения сравнивается с ошибкой лучшего решения. Если ошибка текущего решения меньше, то текущее решение копируется в лучшее решение.

Шаг 7. Уменьшается температура. Если температура становится равной заранее заданному конечному значению, т. е. выполняется условие окончания, то переход к шагу 8. В противном случае переход к шагу 4.

Шаг 8. Процесс поиска прекращается. Оптимальные параметры системы сохраняются как лучшее решение.

Достоинство алгоритма отжига заключается в успешном прохождении локальных минимумов и простоте реализации, недостаток — в больших временных затратах при не всегда высокой точности решений.

Решение задачи параметрической идентификации нечеткой системы возможно с помощью непрерывного либо дискретного *алгоритма муравьиной колонии* (НАМК и ДАМК, соответственно). Основу ДАМК составляет полный ориентированный граф с взвешенными дугами [11]. Веса дуг являются нормированными значениями параметров функций принадлежности [12]. Задача каждого муравья пройти столько дуг, сколько параметров необходимо оптимизировать. Например, для одной треугольной функции принадлежности — три параметра. Значение фермента на дугах определяет выбор муравьем очередной дуги. Значения весов пройденных муравьем дуг передаются в нечеткую систему, которая находит ошибку, и на пройденные дуги наносится количество фермента, обратно пропорциональное найденной ошибке. Нечеткая система описывается несколькими лингвистическими переменными. Каждая переменная описана несколькими функциями принадлежности. Муравьи в алгоритме делятся на колонии. Каждая колония муравьев отвечает за нахождение параметров своей функции.

Количество фермента, наносимого на дуги, пропорционально качеству решения, чем меньше ошибка вывода нечеткой системы, выполненного на выбранных параметрах, тем больше фермента наносится на дуги:

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L^k(t), & \text{если муравей проходит дугу } (i, j), \\ 0, & \text{иначе,} \end{cases}$$

где Q — константа, определяющая количество фермента у отдельного муравья, $L^k(t)$ — значение ошибки на t -й итерации для параметров, выбранных k -м муравьем.

Вероятность выбора муравьем дуги (i, j) на t -й итерации, если дуга не является последним выбранным параметром функции принадлежности, вычисляется по формуле

$$p_{ij}(t) = \tau_{ij}(t)^\alpha / \sum_{l=1}^N \tau_{il}(t)^\alpha; \quad (5)$$

вероятность выбора k -м муравьем дуги (i, j) на t -й итерации, если дуга является последним выбранным параметром функции принадлежности, вычисляется по формуле

$$p_{ij}^k(t) = \frac{\tau_{ij}(t)^\alpha (1/L_j^k(t))^\beta}{\sum_{l=1}^N (\tau_{il}(t)^\alpha (1/L_l^k(t))^\beta)}, \quad (6)$$

где α — параметр, учитывающий важность (приоритет, значимость) фермента на пути; β — параметр, учитывающий важность (приоритет, значимость) ошибки; $\tau_{ij}(t)$ — интенсивность фермента

на дуге между узлами i и j на t -й итерации; $L_o^k(t)$ — значение ошибки на t -й итерации для параметров, выбранных k -м муравьем, если дуга (i, o) является последней, т. е. определяет третий параметр; N — множество узлов, смежных вершине i , причем узел j может быть выбран многократно; в отличие от классического применения алгоритма в нашем случае не формируется запрещенный список узлов; $l \neq i$.

Увеличение количества фермента определяется следующим образом:

$$\tau_{ij}(t+1) = \sum_{k=1}^M (\Delta\tau_{ij}^k(t) + \tau_{ij}(t)) \cdot \rho, \quad (7)$$

а его испарение — по формуле:

$$\tau_{ij}(t+1) = \tau_{ij}(t)(1 - \rho), \quad (8)$$

где $\rho \in [0; 1]$ — коэффициент снижения интенсивности фермента, M — число муравьев, прошедших по дуге (i, j) .

Далее приведен *алгоритм оптимизации трехпараметрической функции принадлежности*.

Шаг 1. Задать начальные параметры алгоритма и нечеткой системы.

Шаг 2. Задать популяции муравьев в колониях.

Шаг 3. Для всех муравьев текущей колонии определить три дуги, используя два раза формулу (5), затем формулу (6).

Шаг 4. Передать в нечеткую систему значения параметров функций принадлежности, определенных муравьями текущей колонии, и вычислить ошибки. Если параметры, переданные муравьем в нечеткую систему, лучше текущих, то сохранить новые значения параметров.

Шаг 5. Если имеется следующая колония, то сделать ее текущей и перейти на шаг 3, иначе перейти на шаг 6.

Шаг 6. Вычислить количество фермента на каждой дуге по формуле (7).

Шаг 7. Вычислить количество испаренного фермента по формуле (8).

Шаг 8. Если условие окончания работы алгоритма выполнено, то останов, иначе перейти к шагу 2.

Условием окончания работы алгоритма служит достижение заданного числа итераций либо получение ошибки, меньше заданной.

В ДАМК выбор очередной дуги ведется муравьем на основе дискретного распределения вероятности. В НАМК дискретное распределение заменяется на непрерывное с использованием функции плотности вероятности [13]. Наиболее часто применяется функция Гаусса. Ее преимущество в простом способе генерации случайных чисел, но очевидный недостаток в том, что она имеет только один максимум. Поэтому в НАМК используется функция плотности вероятности с гауссовым ядром, под которым понимается функция, основанная на взвешенной сумме нескольких одномерных гауссовых функций:

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

Каждому идентифицируемому параметру соответствует свое гауссово ядро, $i = 1, \dots, n$, ω — вектор весов. В ДАМК информация о ферменте хранится в виде таблицы. В НАМК вводится понятие архива решений T . Архив решений представлен таблицей, в которой k строк. Каждая строка состоит из трех частей: найденное муравьем решение $s_l = \{s_l^1, s_l^2, \dots, s_l^n\}$, ошибка, вычисленная нечеткой системой, и вес решения ω_l . Решения в архиве расположены по возрастанию ошибки. Вес ω_l решения s_l вычисляется по формуле

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

где q — эмпирический параметр алгоритма.

Параметры гауссовой функции $g_l^i(x)$, определяются следующим образом:

$$\mu_l^i = s_l^i, \sigma_l^i = \xi \sum_{e=1}^k \frac{|s_e^i - s_l^i|}{k-1}.$$

Параметр $\xi > 0$ одинаков для всех размерностей и имеет эффект, подобный норме испарения фер-

мента в АМК. Чем выше значение ξ , тем ниже скорость сходимости алгоритма [13].

При добавлении нового решения в архив решений T худшее из них удаляется. Этот процесс аналогичен процессу испарения фермента в АМК.

Далее приведен собственно НАМК для параметрической идентификации нечетких систем.

Шаг 1. Задать начальные параметры.

Шаг 2. Сгенерировать популяцию муравьев в колониях.

Шаг 3. Сгенерировать несколько случайных решений для всех архивов решений с последующим оцениванием и ранжированием.

Шаг 4. Найти значения вектора весов. Сделать текущим первого муравья первой колонии.

Шаг 5. Для текущего муравья текущей колонии вычислить номер используемой функции Гаусса. Определить параметры функций Гаусса. Сгенерировать n случайных величин $\{x_1, x_2, \dots, x_n\}$ на основе полученных функций.

Шаг 6. Найти ошибку вывода нечеткой системы при параметрах $\{x_1, x_2, \dots, x_n\}$, если ошибка меньше текущей, то сохранить новые параметры.

Шаг 7. Добавить в архив новое решение, ранжировать архив, удалить из архива худшее решение.

Шаг 8. Если в текущей колонии имеется следующий муравей, то сделать его текущим и перейти на шаг 5, иначе перейти на шаг 9.

Шаг 9. Если имеется следующая колония, то сделать текущим первого муравья в этой колонии и перейти на шаг 5, иначе перейти на шаг 10.

Шаг 10. Если условие окончания работы алгоритма выполнено, то останов, иначе сделать текущим первого муравья первой колонии и перейти на шаг 5.

Условием окончания работы алгоритма служит достижение заданного числа итераций либо получение ошибки, меньше заданной.

Отметим, что для идентификации нечетких моделей дискретный и непрерывный алгоритмы муравьиной колонии нами применены впервые.

Работа генетического алгоритма (ГА) основана на принципах естественного отбора и наследования. Основное преимущество ГА перед другими методами оптимизации заключается в параллельной обработке множества альтернативных решений. При работе алгоритма большее внимание уделяется наиболее перспективным решениям, однако и худшие решения также участвуют в поиске.

В случае параметрической идентификации ген хромосомы кодирует один параметр нечеткой системы. Популяция — это множество хромосом, созданных на текущем шаге работы алгоритма. В качестве целевой функции выступает ошибка вывода нечеткой системы.

Схема генетического алгоритма сводится к следующей последовательности шагов.

Шаг 1. Генерируется популяция.

Шаг 2. Оценивается популяция. Если достигнуто условие окончания работы, то переход на шаг 4.



Шаг 3. Выбираются пары хромосом для скрещивания в соответствии с заданной стратегией и применяются генетические операторы скрещивания и мутации. Переход на шаг 2.

Шаг 4. Окончание работы. Вывод «наилучшего» решения.

В ГА для идентификации используются классические стратегии селекции хромосом (рулеточный отбор, турнирный отбор, элитаризм и случайный отбор), типы операторов скрещивания (одноточечный, многоточечный) и мутации (случайная одноточечная и многоточечная).

Алгоритм роящихся частиц (АРЧ) позволяет осуществлять поиск глобального оптимума в пространстве непрерывно меняющихся переменных [14]. Как и генетический алгоритм, это основанный на популяции метод поиска, в котором, вместо хромосомы — частица, вместо генов — координаты частицы в многомерном пространстве. Каждая частица представляет потенциальное решение, имеет память и хранит свою лучшую позицию в пространстве поиска. Известна частице и лучшая позиция роя. Зная эти две позиции, частица динамически изменяет скорость согласно ее собственному опыту и опыту полета других частиц. Таким образом, движение каждой частицы задается ее лучшей позицией и ее текущей скоростью, ускорением, заданным предыдущей позицией, и ускорением, заданным лучшей частицей в рое.

Пусть имеется n -мерное пространство поиска $S \subset \mathcal{R}^n$, рой состоит из N частиц. Позиция i -й частицы определяется вектором $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in}) \in S$. Лучшая позиция, которую занимала i -я частица, определяется вектором $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{in}) \in S$. Скорость частицы определяется также n -местным вектором $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{in})$. Движение частиц определяют простые уравнения:

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

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

где $i = 1, 2, \dots, N$; $\mathbf{v}_i(k)$ — вектор скорости частицы i на итерации k ; $\mathbf{x}_i(k)$ — координаты частицы i на итерации k ; c_1 и c_2 — эмпирические константы ускорения из интервала; $\mathbf{p}_i(k)$ — лучшая позиция частицы i на первых k итерациях; $\mathbf{p}_g(k)$ — лучшая позиция частицы в рое (задается индексом g) на первых k итерациях; w — эмпирический коэффициент инерции из интервала; rand и Rand — случайные числа из интервала $[0, 1]$.

Обобщенный алгоритм метода состоит в следующем.

Шаг 1. Случайным образом генерируется рой.

Шаг 2. Оценка каждой частицы роя путем вычисления ошибки по формулам (3). Если достигнуто условие окончания работы (число итераций, значение ошибки), то переход на шаг 5.

Шаг 3. Определение для каждой частицы ее новых координат и вектора скорости.

Шаг 4. Расчет новых лучших позиций каждой частицы. Определение частицы с лучшей позицией в рое. Переход на шаг 2.

Шаг 5. Окончание работы.

По сравнению с генетическим алгоритмом алгоритм роящихся частиц прост в реализации и не требует больших вычислительных затрат, в нем отсутствуют операции скрещивания и мутации, частицы меняют позицию в пространстве поиска, не генерируя новых частиц.

4. ЭКСПЕРИМЕНТ

Две таблицы наблюдений были сформированы на основе следующих тестовых функций: 1) $f(x_1, x_2) = x_1 \sin(x_2)$, 2) $f(x_1, x_2) = \sin(2x_1/\pi) \sin(2x_2/\pi)$. Суть эксперимента заключалась в аппроксимации нечеткой системой заданных тестовых функций. Критерием аппроксимации служила ошибка вывода (3). Некоторые выводы по каждому из алгоритмов приведены ниже.

Алгоритм имитации отжига. Среднее значение ошибки не зависит от числа итераций для одного значения температуры, однако время работы системы заметно увеличивается при увеличении числа итераций. Средняя абсолютная ошибка (формула (3), первый функционал) уменьшается при увеличении коэффициента снижения температуры. При этом время работы меняется незначительно. Средняя абсолютная ошибка уменьшается при уменьшении начальной температуры от 1 до 10^{-4} , дальнейшее уменьшение начальной температуры приводит к увеличению средней абсолютной ошибки.

На основе проведенных экспериментов были получены следующие параметры алгоритма имитации отжига: начальная температура — 0,01, конечная температура — 0,00001, число итераций для одного значения температуры — 20, коэффициент снижения температуры — 0,8.

Алгоритм муравьиной колонии. В результате проведенных исследований было установлено, что увеличение числа итераций работы ДАМК, числа муравьев и значения параметра β уменьшают ошибку, увеличивая при этом время работы алгоритма. Такие параметры ДАМК, как α и Q и ρ не оказывают существенного влияния на время работы алгоритма и незначительно меняют ошибку вывода.

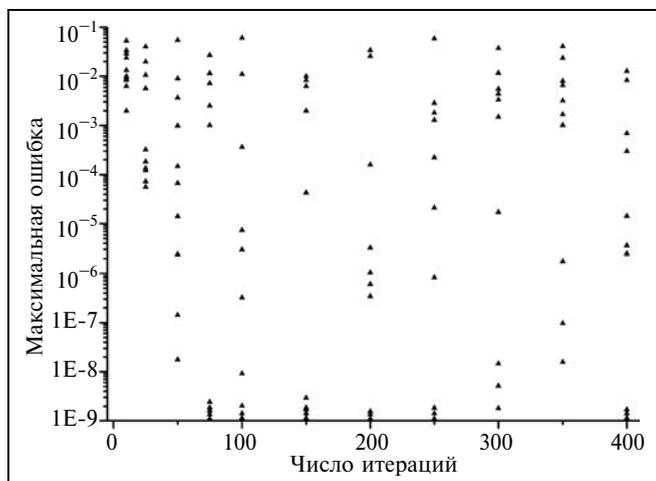


Рис. 1. Зависимость максимальной ошибки от числа итераций в непрерывном алгоритме муравьиной колонии

Непрерывный АМК имеет свойство застревать в локальных минимумах, по этой причине необходимо говорить о лучших и худших решениях при одних и тех же параметрах алгоритма. Разные решения получаются в силу случайности, присутствующей в самом алгоритме, однако не всегда этот случайный выбор способен вывести алгоритм из локального минимума. На рис. 1 (шкала логарифмическая) приведено распределение максимальной ошибки (формула (3), третий функционал) от числа итераций при аппроксимации первой тестовой функции (число муравьев — 24, размер архива — 16, число испытаний — 10, $q = 0,6$).

Эксперимент показывает, что худшее решение (большие ошибки) не зависит от числа итераций, лучшие решения (малые ошибки) появляются уже при числе итераций равном 75. Похожая зависимость наблюдается и при изменении числа муравьев. Время работы НАМК линейно зависит от числа итераций и от числа муравьев.

На рис. 2 (шкала логарифмическая) приведено распределение максимальной ошибки при изменении размера архива при аппроксимации первой тестовой функции. Здесь худшее решение не зависит от размера архива, наилучшие решения получены при размере архива 12...16. Размер архива не оказывает существенного влияния на время вывода.

Генетический алгоритм. Увеличение числа итераций алгоритма ведет к уменьшению значений ошибки, при этом линейно увеличивается и время работы алгоритма. Было установлено, что увеличение числа термов уменьшает значения ошибки, но ведет к возникновению следующих проблем: увеличивается время вывода, так как база правил растет экспоненциально; увеличивается размер хромосомы, и, следовательно, генетическому алго-

риту требуется большее число шагов для настройки параметров функций принадлежности. Для указанных тестовых функций лучшие результаты показал способ формирования новой популяции по принципу, частичной замены, когда формируется «расширенная» популяция, в которую помещаются особи, как нового поколения, так и старого.

Алгоритм роящихся частиц. Большое влияние на сходимость и точность работы алгоритма имеют эмпирические константы ускорения и эмпирический коэффициент инерции. При небольших значениях коэффициента инерции ($w = 0,2...0,4$) имеется тенденция к застреванию в локальных минимумах, большие значения коэффициента инерции ($w = 1,4...1,6$) расширяют диапазон поиска на начальных стадиях и не способствуют точному определению минимума на заключительных стадиях. Эксперименты показали, что лучшие значения для

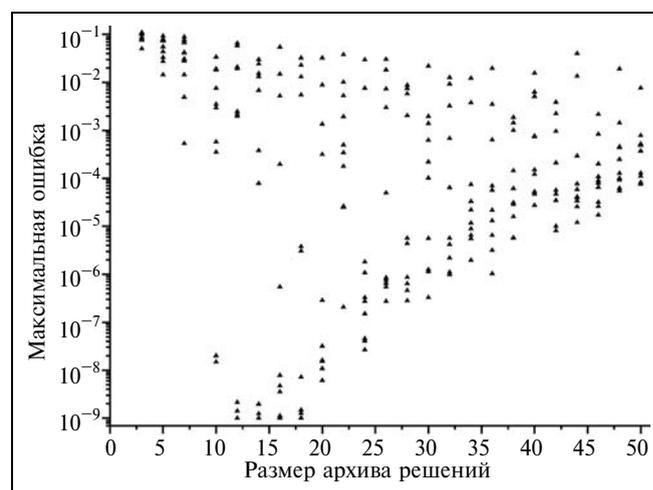


Рис. 2. Зависимость максимальной ошибки от размера архива в непрерывном алгоритме муравьиной колонии

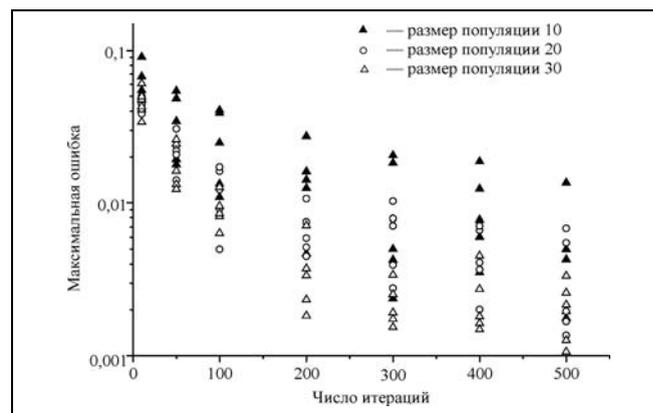


Рис. 3. Распределение максимальной ошибки решения при $w = 0,7$; $c_1 = 1,4$; $c_2 = 1,4$



Сравнение метаэвристик

Алгоритм	Параметры	Точность аппроксимации	Время поиска
Алгоритм отжига	Температура, закон охлаждения, критерий допуска	Очень низкая	Очень большое
Дискретный алгоритм муравьиной колонии	Фермент, число муравьев, эмпирические коэффициенты	Невысокая	Небольшое
Непрерывный алгоритм муравьиной колонии	Число муравьев, размер архива решений, эмпирические коэффициенты	Высокая	Небольшое
Генетический алгоритм	Размер популяции, селекция, мутация, скрещивание	Невысокая	Большое
Алгоритм роящихся частиц	Размер роя, коэффициент инерции, константы ускорения	Высокая	Большое

$w = 0,6...1,2$; константы ускорения должны быть определены в интервале $1,2...1,8$.

Распределение максимальной ошибки (формула (3), третий функционал) для пяти опытов в зависимости от числа итераций и размера популяции приведены на рис. 3 (шкала логарифмическая). Отметим, что здесь представлены не все проведенные нами опыты, а только наиболее типичные. Так, например, в одном из не представленных опытов максимальная ошибка аппроксимации на пятистах итерациях составила $4,092857719066 \cdot 10^{-10}$.

Качественный анализ рассмотренных алгоритмов приведен в таблице.

ЗАКЛЮЧЕНИЕ

Рассмотрены методы идентификации нечетких систем, работающие последовательно и независимо. Дальнейшее развитие работы предполагает совместное использование различных метаэвристик для решения задачи идентификации нечетких моделей. С помощью таких метаэвристик на начальных этапах вычисляется начальное приближение, локализованное в области экстремума, на заключительном этапе уточняется положение экстремума градиентным методом или методом калмановской фильтрации.

Последовательная и независимая схема идентификации себя исчерпала, выбрать какой-либо один эффективный алгоритм решения проблемы

оптимизации не представляется возможным ввиду сложности и многоплановости самой проблемы, поэтому необходимо применять несколько алгоритмов и эвристик одновременно. При параллельном выполнении имеется возможность сотрудничества между эвристиками, а решение, произведенное одной процедурой, может быть впоследствии улучшено другой. Таким образом, сокращается время выполнения отдельных процедур, улучшается качество решения и повышается ошибкоустойчивость.

ЛИТЕРАТУРА

1. Sugeno M., Yasukawa T. A fuzzy-logic-based approach to qualitative modeling // IEEE Trans. on Fuzzy Systems. — 1993. — Vol. 1. — P. 7—31.
2. Takagi T., Sugeno M. Fuzzy identification of systems and its application to modeling and control // IEEE Trans. Systems, Man and Cybernetics. — 1985. — Vol. 15. — P. 116—132.
3. Ходашинский И.А. Технология идентификации нечетких моделей типа сингтон и Мамдани // Тр. VII междунар. конф. «Идентификация систем и задачи управления» SICPRO'08 / Ин-т пробл. упр. — М., 2008. — С. 137—163.
4. Emami M.R., Turksen I.B., Goldenberg A.A. A unified parameterized formulation of reasoning in fuzzy modeling and control // Fuzzy Sets and Systems. — 1999. — Vol. 108. — P. 59—81.
5. Ходашинский И.А. Формальнологический метод и аппроксимация Мамдани в нечетком оценивании величин // Автометрия. — 2006. — № 1. — С. 55—67.
6. Espinosa J., Vandewalle J., Wertz V. Fuzzy logic, identification and predictive control. — London: Springer-Verlag, 2005. — 263 p.
7. Ходашинский И.А. Оценивание величин средствами нечеткой арифметики // Автометрия. — 2004. — № 3. — С. 21—31.
8. Chuang C.C., Su S.F., Chen S.S. Robust TSK fuzzy modeling for function approximation with outliers // IEEE Trans. on Fuzzy Systems. — 2001. — Vol. 9. — P. 810—821.
9. Menard M. Fuzzy clustering and switching regression models using ambiguity and distance rejects // Fuzzy Sets and Systems. — 2001. — Vol. 122. — P. 363—399.
10. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by Simulated Annealing // Science. — 1983. — Vol. 220. — P. 671—680.
11. Dorigo M., Maniezzo V., Colomi A. Ant System: Optimization by Colony of Cooperating Agents // IEEE Trans. Systems, Man and Cybernetics. Part B. — 1996. — Vol. 26. — P. 29—41.
12. Ходашинский И.А., Дудин П.А. Оценивание параметров функций принадлежности на основе алгоритма муравьиной колонии / Тр. междунар. науч.-техн. конф. «Интеллектуальные системы» (IEEE AIS'07). — М.: Физматлит, 2007. — Т. 1. — С. 88—94.
13. Socha K., Dorigo M. Ant Colony Optimization for Continuous Domains / Technical Report TR/IRIDIA/2005-037, Universite Libre de Bruxelles. — Bruxelles, 2005. — 34 p.
14. Kennedy J., Eberhart R. Particle Swarm Optimization / Proc. of the 1995 IEEE Intern. Conf. on Neural Networks. — Perth: IEEE Service Center, 1995. — P. 1942—1948.

Статья представлена к публикации членом редколлегии Ф.Ф. Пащенко.

Ходашинский Илья Александрович — д-р техн. наук, профессор, Томский государственный университет систем управления и радиоэлектроники, ☎(3822) 70-15-93, ✉hodashn@rambler.ru.