

УДК 004.42:004.8

**И.С. Корниенко, О.А. Серебренникова, И.А. Ходашинский**

## **Программно-инструментальный комплекс идентификации нечетких систем**

Предложен подход к созданию программно-инструментального комплекса идентификации нечетких систем. Настройка параметров нечетких систем ведется на основе таблиц наблюдений. Определены структура комплекса и входящие в него подсистемы, описаны их функции. Приведен пример построения нечеткой системы.

**Ключевые слова:** нечеткие системы, идентификация, метаэвристики, программное обеспечение.

**Постановка задачи.** Моделирование стало основной методологией познания действительности. Традиционные подходы к математическому моделированию требуют от разработчиков достаточно глубоких математических знаний и навыков. Поэтому большинство математических моделей – это результат работы математика, но не исследователя конкретной проблемы или предметной области. Однако даже если этот исследователь не в состоянии разработать математическую модель, он может описать изучаемую систему и ее поведение лингвистически. Остается решить вопрос, как преобразовать лингвистическое описание в математическое представление.

Нечеткое моделирование – эффективный подход к преобразованию лингвистических данных в математические формулы и алгоритмы. Нечеткая модель представляет реальную систему в форме, легко понимаемой и интерпретируемой непрофессиональным математиком, такая модель может быть легко модифицирована. В системе нечеткого моделирования собственно знания или модель изучаемой системы отделены от средств манипулирования этими знаниями, что позволяет исследователям ускорить процесс разработки модели, сосредоточившись на вопросе «что делать» вместо вопроса «как делать» [1].

Известно, что моделирование систем включает два основных этапа: идентификацию структуры и идентификацию параметров. Идентификация структуры – определение таких характеристик нечеткой системы, как число нечетких правил, количество лингвистических термов, на которое разбиты входные и выходные переменные. Параметрическая идентификация нечеткой системы заключается в том, что необходимо оптимально подобрать параметры функций принадлежности лингвистических переменных и параметры консеквента в правиле таким образом, что бы минимизировать ошибку системы на обучающих данных. Для оптимизации параметров нечетких систем используются две группы методов. Первая группа – классические методы оптимизации, основанные на производных: метод наименьших квадратов, градиентный метод, фильтр Калмана, метод Левенберга–Марквардта. Эти методы дают точные результаты, но они имеют тенденцию сходиться к локальным оптимумам. Трудности применения классических методов оптимизации, в частности проблема локального экстремума и «проклятие размерности», заставляют обратиться ко второй группе методов – метаэвристических, таких как алгоритмы муравьиной колонии [2–6], роящихся частиц [7, 8], имитации отжига [9, 10], дифференциальной эволюции [11]. Метаэвристика – это метод оптимизации, многократно использующий простые правила или эвристики для достижения оптимального или субоптимального решения [9]. Достоинство метаэвристических методов заключается в большей устойчивости. Но это методы грубой настройки, требующие больших временных ресурсов. Кроме того, применение метаэвристик не гарантирует нахождения оптимального решения и, как правило, связано с эмпирической настройкой параметров используемых алгоритмов. Использование гибридных алгоритмов позволит объединить преимущества метаэвристических методов с преимуществами классических методов, основанных на производных. Такое объединение повысит качество решений при умеренном количестве ресурсов и за приемлемое время.

Из развитых программных пакетов, использующих нечеткую логику, можно отметить Fuzzy Logic Toolbox в составе MATLAB, fuzzyTECH, CubiCalc. Но для того что бы работать с данными программными средствами, необходимо быть экспертом в нечетком моделировании, и данные продукты не обеспечивают полную автоматическую идентификацию нечетких моделей на основе таблиц наблюдений.

Целью нашей работы является разработка алгоритмов и программного комплекса проектирования и идентификации нечетких систем (НС), позволяющего производить настройку на основе наблюдаемых данных.

**Нечеткая система.** Обработка лингвистической информации в нечеткой системе происходит при помощи базы правил. Каждое правило состоит из двух частей: условной и заключительной. Антецедент, или условная часть (ЕСЛИ-часть), содержит утверждение относительно значений входных переменных, в консеквенте, или заключительной части (ТО-части), указывается значение, которое принимает выходная переменная.

В нечеткой системе типа синглтон  $i$ -е правило имеет следующий вид:

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

где  $A_{ij}$  – лингвистический терм, которым оценивается переменная  $x_i$ ;  $r_i$  – действительное число, которым оценивается выход  $y$ .

Нечеткая система осуществляет отображение  $f : \Re^n \rightarrow \Re$ :

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

где  $\mathbf{x}$  – входной вектор;  $R$  – число правил;  $n$  – число входных переменных;  $\mu_{Aij}$  – функция принадлежности, определяемая набором своих параметров.

Нечеткая система может быть представлена как

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

где  $\boldsymbol{\theta} = \|\theta_1, \dots, \theta_N\|$  – вектор параметров;  $N$  – общее число идентифицируемых параметров НС.

Пусть дано множество обучающих данных (таблица наблюдений)  $\{(\mathbf{x}_p, t_p), p = 1, \dots, m\}$ , тогда среднеквадратическая функция ошибки, являющаяся численным критерием адекватности модели, вычисляется по следующей формуле:

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

Проблема идентификации сводится к проблеме поиска минимума заданной функции в многомерном пространстве, координаты которого соответствуют параметрам нечеткой системы.

**Структура и функции комплекса.** Программно-инструментальный комплекс включает в себя следующие компоненты: визуальную среду проектирования; сервер оптимизации НС; клиентов, реализующих параллельную оптимизацию и обработку данных; а также программу для использования конечным пользователем спроектированной и оптимизированной НС.

Под сервером оптимизации понимается приложение принимающее входящие подключения по сети и обрабатывающее их. К серверу предусмотрено подключение из визуальной среды с указанием метода оптимизации НС. Сервер дает возможность наблюдать за процессом оптимизации, который выполняют клиенты, подключенные к серверу. На сервере существует очередь оптимизируемых НС. Все НС, загруженные на сервер, хранятся в жестком диске определенное время, установленное администратором сервера.

Программная реализация методов оптимизации построена следующим образом. Существует абстрактный класс `Method`, от которого наследуются все остальные методы. Данный класс содержит такие поля, как имя метода, разработчик метода, описание, способ запуска оптимизации и ссылку на объект класса `ParamsMethod` и т.п. Класс `ParamsMethod` отвечает за параметры метода, данный класс позволяет добавлять параметры методу, загружать их значения, значения по умолчанию, указывать тип параметра. Каждый параметр можно описать с помощью объекта класса `ParameterMethod`. Существует класс `MethodWorker`, который управляет работой метода оптимизации. С помощью данного класса методы запускаются в отдельных потоках, что обеспечивает параллельную оптимизацию НС несколькими методами одновременно. Объекту класса `MethodWorker` перед запуском необходимо передать оригинальную систему и ее копию.

На рис. 1 приведена основная форма программы, состоящая из четырех областей: верхнее меню управления проектом; панель инструментов, расположенная в левой части формы; рабочая область, расположенная справа от панели инструментов; строка состояния, которая отображает имя проекта и директорию расположения файла проекта.

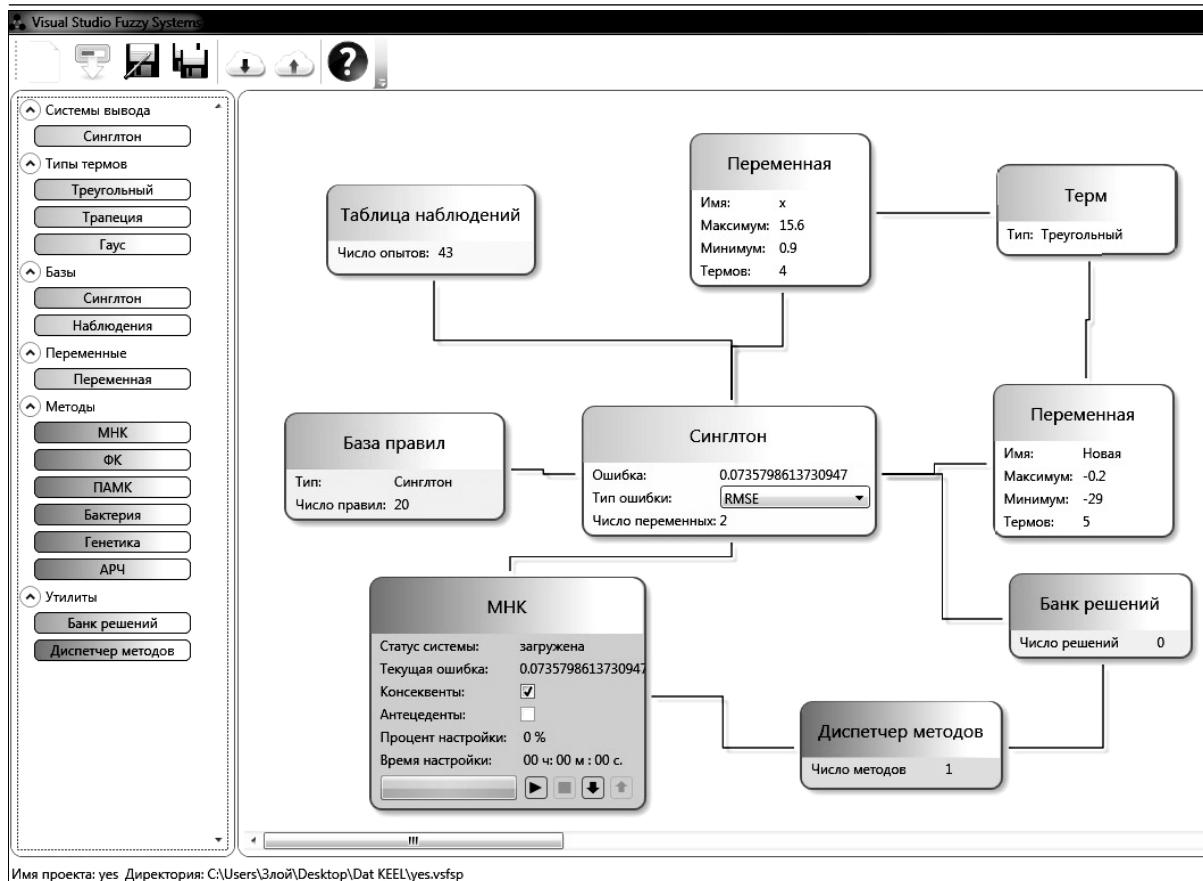


Рис. 1. Главная форма визуальной среды проектирования

Для создания элемента рабочей области необходимо, зажав левую кнопку мыши над одним из элементов панели инструментов, переместить его на рабочую область. Для соединения двух элементов необходимо зажать правую кнопку над шапкой одного элемента и отпустить ее над шапкой другого элемента. Не все элементы можно связывать между собой, а только те, которые логически связаны, например систему вывода и переменную, базу правил и систему вывода, переменную и тип терма и т.п.

В системе предусмотрена работа с данными в формате Excel. Для этого необходимо корректно сформировать файл. Рассмотрим пример формирования файла diabetes из репозитория KELL [10]. В файле необходимо первой строкой обозначить имена столбцов таблицы, а ниже расположить сами данные. Лист необходимо переименовать в Observation. Для загрузки данных необходимо перейти на вкладку обучающей выборки и указать имя файла. Далее выбираются столбцы загружаемых из файла данных. Указывается столбец, используемый в качестве выходного значения системы, остальные выбранные столбцы будут автоматически расценены как входные значения для НС. В нашем случае входными являются столбцы Age и Deficit, а выходными – C\_peptide.

После того как были загружены данные diabetes, необходимо выбрать тип системы вывода, он выбирается на панели инструментов и переносится на рабочую область (на рисунке «singleton»), после чего соединяется с ранее созданной таблицей наблюдений. После этого описываются переменные, делается это с помощью элемента «переменная». Далее данный элемент соединяется с элементом «singleton». После выбирается тип функции принадлежности и выбранный элемент соединяется с элементом «переменная». Далее необходимо задать описывающие переменную нечеткие термы. Это можно сделать либо создавая по одному терму, нажимая кнопку, добавив терм и редактируя его, либо же сгенерировать равномерное покрытие переменной заданным количеством термов.

Элемент «база правил» ответствен за создание, редактирование и хранение нечетких правил. Правила можно создавать как вручную, так и автоматически. После того как база правил сформирована, будет создан проект, который может быть сохранен в файл или использован для оптимизации заданной НС. Для оптимизации спроектированной системы необходимо на рабочей области расположить один из выбранных методов и соединить его с системой вывода. В нашем примере в качест-

всего такого метода выбран метод наименьших квадратов (МНК). Методы оптимизации имеют параметры, для их изменения и отображения предусмотрены свои средства. После установки параметров метод можно запускать. В ходе работы отображаются число выполненных и оставшихся итераций, а также отработанное и оставшееся время работы.

Программно-инструментальный комплекс имеет возможности графического отображения функций принадлежности как до, так и после оптимизации. На рис. 2 приведены функции принадлежности переменной Deficit до и после оптимизации.

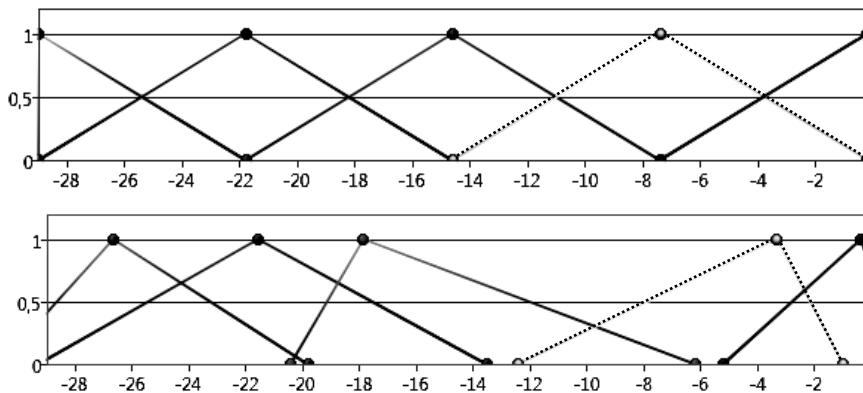


Рис. 2. Функции принадлежности переменной Deficit до и после оптимизации

В систему включен элемент «диспетчер методов», позволяющий управлять методами оптимизации, строить последовательности выполнения методов и исследовать результаты их работы. Для этого устанавливают такие параметры, как количество прогонов метода оптимизации, изменение какого-либо параметра метода после каждого запуска, имя следующего метода. Последовательность выполнения методов можно циклически замкнуть, тогда все методы будут работать до вмешательства эксперта.

Элемент «банк решений» предназначен для сбора и хранения выполненных решений. Для любого решения из банка можно посмотреть, при каких параметрах метода решение было получено, за какое время и с какими ошибками. Среда разработки позволяет сохранять НС. Для этого необходимо выбрать элемент вывода и нажать кнопку импорта данных, после чего выбранная НС будет сохранена в файл с выбранным расширением.

Визуальная среда проектирования позволяет интегрировать в нее методы оптимизации, разработанные третьей стороной. То есть существует возможность разработать свой метод и интегрировать его в систему.

Визуальная среда и сервер оптимизации предназначены для эксперта, который проектирует НС и ее оптимизирует. Оптимизированные нечеткие системы использует конечный пользователь в отдельном приложении.

**Заключение.** Разработанный программный комплекс позволяет автоматизировать процесс проектирования нечетких систем и их оптимизацию. Серверная часть комплекса значительно сокращает время оптимизации. Существует возможность разработки и интеграции в визуальную среду проектирования методов оптимизации, разработанных третьими лицами. Предложенный подход к проектированию облегчает процесс создания нечеткой системы, так как сама система представлена в графическом виде. Наглядное отображение системы в виде отдельных узлов способствует ее лучшему пониманию.

Программно-инструментальный комплекс прошел успешное тестирование при выполнении лабораторных работ по курсу «Базы знаний».

Дальнейшее расширение программного инструментария предполагает добавление новых систем нечеткого вывода (Такаги-Сугено, Мамдани) и создание новых утилит, а также создание сервера оптимизации нечеткой системы и методов параллельной и распределенной оптимизации, утилит управления и хранения копий нечетких систем на сервере.

#### Литература

1. Tron E. Mathematical modeling of observed natural behavior: a fuzzy logic approach / E. Tron, M. Margaliot // Fuzzy Sets and Systems. – 2004. – Vol. 146. – P. 437–450.

2. Ходашинский И.А. Идентификация нечетких систем на основе непрерывного алгоритма муравьиной колонии / И.А. Ходашинский, П.А. Дудин // Автометрия. – 2012. – Т. 48, № 1. – С. 63–71.
  3. Khodashinskii I.A. Identification of fuzzy systems using a continuous ant colony algorithm / I.A. Khodashinskii, P.A. Dudin // Optoelectronics, Instrumentation and Data Processing. – 2012. – Vol. 48, №. 1. – P. 54–61.
  4. Ходашинский И.А. Идентификация нечетких систем на основе прямого алгоритма муравьиной колонии / И.А. Ходашинский, П.А. Дудин // Искусственный интеллект и принятие решений. – 2011. – №3. – С. 26–33.
  5. Ходашинский И.А. Параметрическая идентификация нечетких моделей на основе гибридного алгоритма муравьиной колонии / И.А. Ходашинский, П.А. Дудин // Автометрия. – 2008. – Т. 44, № 5. – С. 24–35.
  6. Khodashinsky I.A. Parametric Fuzzy Model Identification Based on a Hybrid Ant Colony Algorithm / I.A. Khodashinskii, P.A. Dudin // Optoelectronics, Instrumentation and Data Processing. – 2008. – Vol. 44, № 5. – P. 402–411.
  7. Ходашинский И.А. Идентификация параметров нечетких систем на основе адаптивного алгоритма роящихся частиц / И.А. Ходашинский, Д.С. Синьков // Информационные технологии. – 2011. – №8. – С. 2–5.
  8. Ходашинский И.А. Идентификация параметров нечетких моделей типа синглтон на основе алгоритма роящихся частиц // Информационные технологии. – 2009. – №6. – С. 8–11.
  9. Ходашинский И.А. Идентификация нечетких систем: методы и алгоритмы // Проблемы управления. – 2009. – № 4. – С. 15–23.
  10. Ходашинский И.А. Идентификация нечетких систем на базе алгоритма имитации отжига и методов, основанных на производных // Информационные технологии. – 2012. – №3. – С. 14–20.
  11. Ходашинский И.А. Идентификация нечетких систем на основе метода дифференциальной эволюции / И.А. Ходашинский, П.А. Дудин // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2011. – Т. 23, № 1, ч. 1. – С. 178–183.
  12. KEEL-dataset repository [Электронный ресурс]. – Режим доступа: <http://sci2s.ugr.es/keel/>, свободный (дата обращения: 10.04.2012).
- 

**Корниенко Игорь Сергеевич**

Студент каф. автоматизации обработки информации ТУСУРа

Тел.: 8 (383-2) 41-34-26

Эл. почта: zloi.linux@gmail.com

**Серебренникова Ольга Александровна**

Студентка каф. автоматизации обработки информации ТУСУРа

Тел.: 8 (383-2) 41-34-26

Эл. почта: avayrin@gmail.com

**Ходашинский Илья Александрович**Д-р техн. наук, профессор каф. комплексной информационной безопасности  
электронно-вычислительных систем ТУСУРа

Тел.: 8 (383-2) 41-34-26

Эл. почта: hodashn@rambler.ru

Kornienko I.S., Serebrennikova O.A., Hodashinsky I.A.

**Software for fuzzy systems identification**

An approach to building software-tool complex of fuzzy systems identification proposed. The fuzzy systems parameters identification based on the observation table. The complex structure, and their functions are described. An example of the fuzzy systems design is considered.

**Keywords:** fuzzy system, identification, metaheuristics, software.