

УДК 519.71



РАЗВИТИЕ МЕТОДОВ НА ОСНОВЕ МНОГОЗНАЧНОЙ ИНТЕРВАЛЬНОЙ ЛОГИКИ В ЗАДАЧАХ НАСТРОЙКИ ФУНКЦИЙ ПРИНАДЛЕЖНОСТИ

Е.И. Кучеренко¹, А.В. Корниловский¹, И.С. Глушенкова²

¹ Харьковский национальный университет радиоэлектроники

² Харьковская национальная академия городского хозяйства

Предложено и реализовано развитие метода оперативной настройки параметров функций принадлежности нечетких баз знаний. Исследованы точностные параметры при настройке функций принадлежности на множестве лингвистических термов. Предложена программная реализация подхода, представленная в работе множеством классов UML-диаграмм, оценена вычислительная сложность на множестве показателей точности.

НЕЧЕТКАЯ БАЗА ЗНАНИЙ, ФУНКЦИИ ПРИНАДЛЕЖНОСТИ, НАСТРОЙКА, ОПТИМИЗАЦИЯ, ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ

Введение

Важным аспектом реализации технологий принятия решений на множестве альтернатив $\{Alt_i\}, i \in I$ сетевой модели, отображающей процессы предметной области $\{Pr_k\}, k \in K$, является использование знание ориентированных технологий [1], отображающих процессы моделей правил продукционных баз знаний

$$\{ \text{if } x \text{ is } \mu(x) \text{ then } y \text{ is } \mu(y) \}. \quad (1)$$

В связи с этим представляется актуальным в задачах повышения достоверности принимаемых решений повышение адекватности применения функций принадлежности нечетких процессов [1] в задачах нечетких баз знаний. В работах [2, 3, 4] были исследованы некоторые аспекты описания и настройки компонент $\mu(x) = f(a, b, x, y, z)$ функций принадлежности процессов предметной области, ориентированных на обучение функций принадлежности нечетких правил продукций. Решения задач нечеткого логического вывода может быть представлено в виде нахождения решения уравнения вида:

$$y' = x' \wedge \mu(x, y), \quad (2)$$

где $\mu(x, y)$ — нечеткое отношение правила (1).

Существующие решения (2) основаны на модификации и развитии подходов методов дихотомии и многозначной логики в задачах настройки параметров функций принадлежности. Критический анализ результатов показал, что полученные решения обладают сложностью, близкой к экспоненциальной, а решения характеризуются наличием существенной неустойчивости.

В связи с этим существует актуальность и перспективность исследований, отображающих процедуры нечеткого логического вывода на множествах возможных решений.

1. Постановка задачи

Ранее был предложен подход для реализации процедуры настройки функций принадлежности

нечеткого логического вывода посредством развития метода дихотомии [5]. Половинное деление позволяет для множества функций принадлежности $\{\mu(x)\}$ производить настройку параметра крутизны k в $\mu(x) = f(k, a, b, x, y, z)$ в зависимости от заданного значения ожидаемой величины дефазсифицированного значения функции принадлежности $y_{ож}$. Установлено, что такая реализация обладает вычислительной сложностью, характеризующейся [3]:

$$O(n) \approx e^{1/\varepsilon}, \quad (3)$$

где ε — параметр метода настройки параметров, определяющий норму точности аппроксимации.

Вычислительная сложность ранее предложенного алгоритма не является оптимальной. При решении задачи с использованием метода многозначной логики при некоторых произвольных входных данных [6] возникала проблема скачка вычислительной сложности, что приводило к недопустимо большому увеличению времени работы программы.

В связи с изложенным выше, необходимо модифицировать существующие решения, что позволит существенно снизить вычислительную сложность с одновременным повышением стабильности работы.

2. Разработка модификации метода настройки параметров функций принадлежности

Известно, что множество функций принадлежности $\{\mu_i\}, i \in I$ является конечным в рамках некоторого класса решаемых задач: $\{\mu_i\} = \text{const}, \{\mu_i\} < \infty$. Для каждой операции a_i настройки параметра k функция принадлежности $\{\mu_i\}, i \in I$ четко определена и может быть представлена в виде:

$$\mu_{\bar{a}_i} = f(x, k, a, b),$$

где x, k, a, b — некоторые переменные.

В изложенном [1] методе развития дихотомии операция соответствует итеративному процессу:

$$k_i = k_{i-1} \pm \varepsilon / 2, \quad (4)$$

где k_i — i шаг операции; k_{i-1} — $i-1$ шаг операции; ε — норма ошибки,

Процесс продолжается до тех пор, пока не реализован критерий точности аппроксимации Δy :

$$\Delta y = |y_{\text{ф}} - y_{\text{ож}}| \leq \varepsilon, \quad (5)$$

где $y_{\text{ф}}$ — фактическое значение дефаззифицированной величины.

В работе [2] предлагается использовать текущее значение делителя как минимальное и ввести ряд дополнительных целочисленных значений делителя [3]:

$$d \in \{d_1, d_2, \dots, d_{n-1}, 2\}, \quad (6)$$

где $d_1, d_2, \dots, d_n, d_l > 2, l \in L$.

Такой вариант выбора делителя позволит в ряде случаев осуществить более быстрое достижение заданной точности аппроксимации ε .

Для каждого типа функции принадлежности и каждого делителя d_i можно измерить время работы предлагаемого алгоритма τ_i , где τ_i — модельное время. Критерием решения задачи (останова) будет из $\{\tau_k\}, i \in K$ как нахождение

$$\forall \tau_k, \tau_k \in \{\tau_k\} \xrightarrow{F^*} \text{extr},$$

где F^* — ограничения предметной области. На множестве $\{\mu_i\}$ осуществляется ранжирование τ_k по критерию минимизации модельного времени. Для соответствующих значений из $\{\tau_{k_{\min}}\}$ определяется их делитель $d_{\min_i}$ для μ_i :

$$k_i = k_{i-1} \pm \varepsilon / d_{\min}. \quad (7)$$

Таким образом, можно выделить для нахождения (7) ряд этапов модифицированного метода при настройке параметров функций принадлежности:

- фаззификация правила продукции, выбор типов функции принадлежности;
- используя экспертные оценки, задание параметров функций принадлежности;
- решение задачи нечеткого логического вывода;
- дефаззификация нечеткого логического вывода;
- определение оценки ожидаемого дефаззифицированного значения — $y_{\text{ож}}$;
- задание делителя дихотомии d и требуемой точности вычисления ε ;
- запуск процесса итеративной аппроксимации;
- остановка при достижении необходимой точности ε ;
- повторный прогон изложенных выше процедур, модификация параметров функции принадлежности;
- останов.

Оценка времени проведения операции настройки τ_k определена в мс — 10^{-3} с.

Решение данной задачи посредством использования методов многозначной интервальной логики [3] позволяет определять минимальное значение модельного времени τ [2]. Однако в ряде случаев оно не является алгоритмически оптимальным. Это связано с тем, что не всегда возможно достичь минимальных значений модельного времени по причине использования в этом подходе значения Δy (5) в качестве числителя шага аппроксимации [3], что не всегда адекватно.

Знаменатель шага определяется как $A \in (0,1), A \in \mathbb{R}$. В предлагаемом подходе в роли делимого выступает значение априорно заданной точности вычислений ε (5), что позволяет выполнить условие для модельного времени τ_k :

$$\tau_{\Delta y, A} < \tau_{\varepsilon, d}, \quad (8)$$

где $\tau_{\Delta y, A}$ — модельное время при использовании числителя шага Δy и знаменателя A ; $\tau_{\varepsilon, d}$ — ε и d — соответственно.

3. Программная реализация предлагаемого подхода

Программная реализация предложенного решения является модификацией версии приложения, реализующего n -разрядную интервальную настройку функций [3], которая выполнена в программной среде C++ [7]. В приложении в качестве множества функций принадлежности представлено множество гауссианов функций принадлежности:

$$\text{— большого } \mu^b = 1 - e^{-k_1(x-a)^2},$$

$$\text{— малого } \mu^m = e^{-k_2 x^2},$$

$$\text{— среднего } \mu^{cp} = e^{-k_3(x-b)^2}$$

и их производные, где k_1, k_2, k_3 — элементы настройки крутизны, a, b — некоторые константы. Соответственно подбор делителя дихотомии $d_{\min}$ осуществляется для каждого из трех видов функций.

UML-диаграмма классов приложения представлена на рис. 1.

Для реализации модифицированного подхода в программный код введены классы FVData и Divider. Класс FVData представляет собой структуру, содержащую данные о времени выполнения операции настройки для конкретного типа нечеткой величины и значения делителя d . Таким образом, класс FVData агрегируется классом FuzzyValue в отношении «один ко многим». Поля агрегированных экземпляров класса заполняются динамически и подлежат дальнейшему считыванию, поэтому определены модификатором доступа public [7].

Класс Divider описывает сущность динамически изменяющегося делителя d . Объект класса Divider хранит в себе текущее значение делителя, а

также функционал для его изменения и измерения текущего числа операций при настройке функций принадлежности. Класс *Divider* не предполагает наследования, поэтому определён модификатором *final*. Классовая структура приложения выполнена таким образом, что предполагает повторное использование кода и дальнейшую модификацию. Выполнены требования инкапсуляции и модульности. Важным нововведением в программной оболочке алгоритма стала поддержка вычислений и вызова приложений посредством командной строки. Входные данные для работы алгоритма при этом передаются в виде строкового массива.

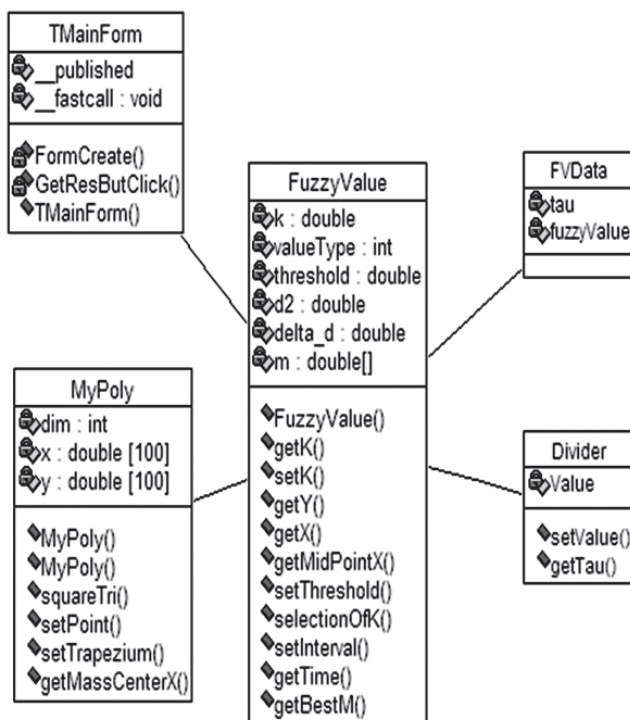


Рис. 1. UML-диаграмма классов приложения

Важным нововведением в программной оболочке алгоритма стала поддержка вычислений и вызова приложений посредством командной строки. Входные данные для работы алгоритма при этом передаются в виде строкового массива. Таким образом, вызов приложения имеет следующий вид:

FuzzyP.exe arg1 arg2 arg3 ... argN,

где arg_i ($i = 1..N$) – входные значения типов функций принадлежности, входное значение аргумента, ожидаемое значение дефазифицированной величины,

Вывод информации при вызове приложения через командную строку осуществляется в файл, который впоследствии может быть использован и обработан другими системами.

Функционал консольного вызова приложения позволяет интегрировать его с другими программными вычислительными средствами при решении комплексных задач.

Пользовательский интерфейс приложения представлен на рис. 2 для случая настройки функции $\mu^m = e^{-k_2 x^2}$:

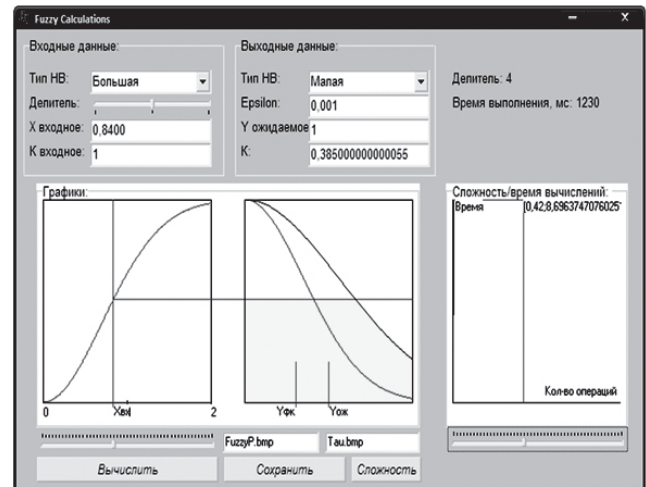


Рис. 2. Интерфейс приложения при выполнении настройки функции принадлежности «малое»

4. Экспериментальные исследования

В процессе работы приложения были получены следующие фактические значения времени работы алгоритма τ для трёх типов функций принадлежности и множества делителей дихотомии $d = \{2, 4, 8\}$. Определена точность аппроксимации $\varepsilon = 10^{-3}$ в зависимости от коэффициента деления (табл. 1).

Таблица 1

Время работы приложения, мс

Тип НВ \ Делитель d_i	2	4	8
Малое	103	80	97
Среднее	123	69	112
Большое	45	75	69

Экспериментальные данные (табл. 1) позволяют в задачах настройки нечетких баз знаний использовать конкретный коэффициент деления d для соответствующей функции. Точные значения времени выполнения τ (мс) изменяются в зависимости от аппаратного обеспечения компьютерных средств. Результаты эксперимента получены для компьютера с центральным процессором Intel Core2Duo 2GHz и 1 GB RAM.

Зависимость вычислительной сложности от различных значений делителя d представлена на рис. 3.

Как следует из рис. 3, на участке $0 < d < d_3$ существует принципиальная возможность вычислений с приемлемой точностью ε без влияния экспоненциальной сложности вычислений, что важно в приложениях. В работе также выполнены замеры модельного времени для различных значений за-

данной точности вычислений ε , что в ряде случаев позволяет варьировать значениями параметров настройки.

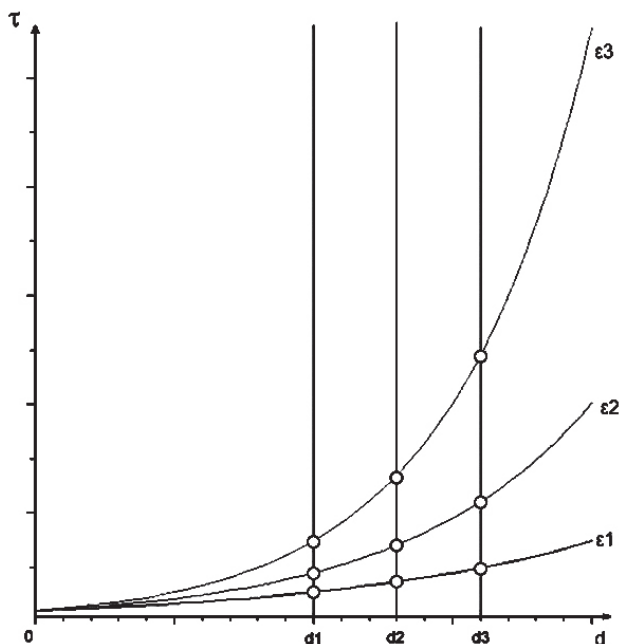


Рис. 3. Зависимость сложности вычислений от различных значений делителя d при $d_3 > d_2 > d_1$; $\varepsilon_1 > \varepsilon_2$; $\varepsilon_2 > \varepsilon_3$

Выводы

1. Реализовано развитие метода настройки параметров нечетких функций принадлежности с использованием кратных значений делителя многозначной логики в знаниеориентированных технологиях. Предложен подход для нахождения оптимальных значений параметров операции настройки по критерию точности аппроксимации, что позволяет существенно снизить время работы метода.

2. Модифицировано программное решение путем ввода дополнительных функций, внесены соответствующие изменения в классовую структуру UML-диаграммы, реализован более удобный пользовательский интерфейс.

3. Проведены экспериментальные исследования, подтверждающие эффективность данного

подхода для трёх базовых видов функций принадлежности на множестве термов лингвистических переменных: «малое», «большое» и «среднее», которые реализованы в виде гауссианов в знаниеориентированных технологиях.

Список литературы: 1. Бодянский, Е.В. Нейро-фаззи модели и мультиагентные технологии в сложных системах [Текст] / Е. В. Бодянский, В. Е. Кучеренко, Е. И. Кучеренко, А. И. Михалев, В. А. Филатов — Дніпропетровськ: Системні технології, 2008. — 412 с. 2. Кучеренко, Е. И. О методах настройки функций принадлежности [Текст] / Е. И. Кучеренко, А. В. Корниловский, И. С. Творошенко — Системы управления навигации и связи. Вып. 1, 2010. — С. 95-99. 3. Кучеренко, Е. И. О модификации методов оперативной настройки функций принадлежности в знание ориентированных моделях [Текст] / Е. И. Кучеренко, А. В. Корниловский, И. С. Глушенкова // Системы обработки информации. — 2010. — № 5 (86). — С. 54 — 57. 4. Способы формирования нечетких множеств. Функции принадлежности [Электронный ресурс]. Режим доступа: http://www.lifeprog.ru/view_zam.php?id=87&cat=5&page=5. 5. Корн, Г. Справочник по математике для научных работников и инженеров [Текст] / Г. Корн, Т. Корн — М.: Наука, 1970. — 720 с. 6. Васильев, Ф. П. Методы оптимизации [Текст] / Ф. П. Васильев — М.: «Факториал Пресс», 2002. — 824 с. 7. Шилдт, Г. Полный справочник по C++ / Г. Шилдт. — М.: Издательский дом «Вильямс», 2004. — 800 с.

Поступила в редакцию 25.02.2011.

Розвиток методів на основі багатозначної інтервальної логіки в задачах налаштування функцій належності / Є. І. Кучеренко, А. В. Корніловський, І. С. Глушенкова // Біоніка інтелекту: наук.-техн. журнал. — 2011. — № 1 (75). — С. 75–78.

Запропоновано розвиток методу налаштування функцій належності в знанняорієнтованих моделях. Досліджено питання обчислювальної складності та точності реалізації.

Табл. 1. Іл. 3. Бібліогр.: 7 найм.

The development of methods on based of multi-valued interval logic in tasks of setting membership functions / Ye. I. Kucherenko, A. V. Kornilovski, I. S. Gluchenkova // Bionics of Intelligence: Sci. Mag. — 2011. — № 1 (75). — P. 75–78.

Proposed development of methods for setting membership functions in a knowledge-based models. The issues of computational complexity and accuracy of implementation.

Tab. 1. Fig. 3. Ref.: 7 items.