

А.А. ОЛЕЙНИК, ЗНТУ (г. Запорожье),

С.А. СУББОТИН, канд. техн. наук, доц. ЗНТУ (г. Запорожье)

МУЛЬТИАГЕНТНЫЙ МЕТОД ПАРАМЕТРИЧЕСКОГО СИНТЕЗА НЕЙРО-ФАЗЗИ СЕТЕЙ С ПРЯМОЙ СВЯЗЬЮ МЕЖДУ АГЕНТАМИ

Рассмотрена задача параметрической идентификации нейро-фаззи сетей, для решения которой разработан мультиагентный метод с прямой связью между агентами и ускоренной сходимостью к оптимальному решению, и более подробным изучением пространства поиска. Предложенный метод программно реализован и позволяет выполнять параметрический синтез нейро-фаззи моделей. Проведены эксперименты по параметрическому синтезу нейро-фаззи сетей, подтверждающие эффективность разработанного метода.

Ключевые слова: мультиагентный метод, нейро-фаззи сеть, нейро-фаззи модель.

Постановка проблемы и анализ литературы. В настоящее время нейро-фаззи сети эффективно применяются в различных прикладных областях, связанных с управлением, прогнозированием и диагностикой, что вызвано их эффективностью, прозрачностью и относительной дешевизной при реализации. В частности, нейро-фаззи сети нашли широкое применение в таких областях человеческой деятельности как: медицинская и техническая диагностика, финансовый менеджмент, распознавание образов, разведка ископаемых, управление компьютерными сетями, управление технологическими процессами, анализ веб-контента в Internet и т.д. [1 – 5].

Архитектура нейро-фаззи сети изоморфна нечеткой базе знаний. Пусть задана обучающая выборка $\langle X, Y \rangle$, где $X = \{x_1^1, x_2^1, \dots, x_k^1, \dots, x_1^m, x_2^m, \dots, x_k^m\}$ – независимые переменные, $Y = \{y^1, y^2, \dots, y^m\}$ – выходная переменная, зависимость которой от независимых переменных исследуется. Пусть $P = \{p_1, p_2, \dots, p_n\}$ – настраиваемые параметры нейро-фаззи сети. Тогда задача параметрического синтеза нейро-фаззи сети заключается в поиске таких значений $P^* = \{p_1^*, p_2^*, \dots, p_n^*\}$, при которых выполняется следующее условие:

$$\varepsilon[FN(P^*)] \rightarrow \min, \quad (1)$$

где $\varepsilon[FN(P^*)]$ – ошибка прогнозирования нейро-фаззи сети FN , у которой параметры P принимают значения P^* . Ошибка определяется на основе исходных значений Y и рассчитанных значений Y^* , полученных при помощи нейро-фаззи сети.

В нейро-фаззи сетях используются дифференцируемые реализации треугольных норм (умножение и вероятностное ИЛИ), а также гладкие функции принадлежности [6]. Применение гладких функций принадлежности вызвано тем, что для настройки нейро-фаззи сетей применяются методы

обучения нейронных сетей, основанные на технике обратного распространения ошибки. Однако, как известно, метод обратного распространения ошибки обладает недостатками, связанными с долгой работой и ограничениями, накладываемыми на архитектуру сети (в данном случае, функции принадлежности должны быть дифференцируемыми).

В связи с этим актуальной является разработка новых методов, позволяющих выполнять настройку нейро-фаззи сетей, которые не обладают указанными недостатками. Одной из новых ветвей методов оптимизации являются методы коллективного интеллекта (Swarm Intelligence) [7, 8]. К таким методам относятся: метод муравьиных колоний [9, 10], метод пчелиной колонии [11], метод оптимизации с использованием роя частиц [12] и другие [13, 14]. Данные методы уже эффективно применяются для решения различных задач оптимизации: задач коммивояжёра, календарного планирования, динамической маршрутизации, нахождения оптимума многомерной функции, транспортных задач и т.д.

В данной статье исследуется возможность применения методологии коллективного интеллекта для настройки нейро-фаззи сетей. Поскольку задачу параметрического синтеза нейро-фаззи сети можно отнести к задаче непрерывной оптимизации, то для решения данной задачи предлагается исследовать метод пчелиной колонии. Данный метод представляет собой мультиагентный метод с прямой связью между агентами.

Цель статьи – разработка метода параметрического синтеза нейро-фаззи сетей, основанного на мультиагентном подходе с прямой связью между агентами. Для достижения цели необходимо решить следующие задачи:

- проанализировать особенности архитектуры нейро-фаззи сетей;
- исследовать метод пчелиной колонии: его основные принципы, особенности, преимущества и недостатки;
- разработать метод параметрического синтеза нейро-фаззи сетей, основанный на моделировании агентов с прямой связью между ними, учитывающий недостатки базового метода, и с применением дополнительных процедур, позволяющих устранить эти недостатки;
- программно реализовать разработанный метод настройки параметров нейро-фаззи сетей;
- провести эксперименты по решению тестовых задач и сравнить результаты, полученные с помощью предложенного метода, с результатами, полученными путём применения традиционного подхода для настройки нейро-фаззи сетей.

Архитектура нейро-фаззи сетей. На практике наиболее часто примеряют нейро-фаззи сети, которые реализуют систему нечеткого вывода Мамдани или Сугено в виде нейронной сети прямого распространения [5, 6].

Рассмотрим структуру нейро-фаззи сети Мамдани для определения параметров, которые требуется настраивать при параметрическом синтезе. Данная сеть состоит из пяти слоёв.

Первый слой образуют элементы входного вектора.

Второй слой содержит нечёткие термы, которые соответствуют входным переменным, и рассчитывает принадлежности входного вектора к каждому из нечётких термов:

$$\mu_{jp}(x_i) = \frac{1}{1 + \left(\frac{x_i - b_i^{jp}}{c_i^{jp}} \right)^{2a_i^{jp}}}, \quad i = \overline{1, N}, \quad p = \overline{1, k_j}, \quad j = \overline{1, m}, \quad (2)$$

где $\mu_{jp}(x_i)$ – функция принадлежности переменной x_i терму p -ой строчки j -го правила; a_i^{jp} , b_i^{jp} , c_i^{jp} – параметры функции принадлежности.

На третьем слое находятся строки-конъюнкции антецедентов (посылок) правил нечёткой базы знаний, каждый узел этого слоя реализует оператор T -нормы и выдаёт степень соответствия входного вектора сопоставляемому узлу-правилу.

Четвёртый слой объединяет правила в классы d_j , $j = \overline{1, m}$, и вычисляет степень принадлежности входного вектора соответствующим термам выходной переменной, каждый узел этого слоя реализует оператор T -конормы. Функция принадлежности входного вектора j -му терму d_j выходной переменной y определяется как: $\mu_{d_j} = \max_{p=1, k_j} \{w_{jp} \min_{i=1, N} [\mu_{jp}(x_i)]\}$.

Пятый слой содержит один нейрон, который объединяет функции принадлежности к нечётким термам выходной переменной и выполняет операцию дефазификации.

Число узлов в нейро-фаззи сети определяется следующим образом: слой 1 – по количеству элементов входного вектора; слой 2 – по количеству нечётких термов в базе знаний (n); слой 3 – по количеству строк-конъюнкций в базе знаний; слой 4 – по количеству классов, на которые разбивается диапазон выходной переменной (m).

Дугам графа присваиваются такие веса: единица (дуги между первым и вторым слоями); функции принадлежности входа нечёткому терму (дуги между вторым и третьим слоями); веса правил (дуги между третьим и четвёртым слоями); единица (дуги между четвёртым и пятым слоями). Пороги нейронов считаются равными нулю.

Таким образом, можно отметить, что настраиваемыми параметрами нейро-фаззи сети можно считать параметры функций принадлежности (для входных переменных и для выходной переменной) и веса правил. Следовательно, задачей параметрического синтеза нейро-фаззи сети является

нахождение таких значений параметров, при которых достигается наилучшее прогнозирование при использовании данной сети.

Метод пчелиной колонии. Метод пчелиной колонии является мультиагентным эвристическим итеративным методом случайного поиска с прямой связью между агентами, основанный на моделировании коллективного интеллекта колонии пчёл. Одной из главных особенностей коллективного поведения моделируемых агентов является разделение функций между различными агентами: занятыми фуражирами и разведчиками. Занятые фуражиры – это агенты, которые закреплены за определёнными областями пространства поиска и разрабатывают их, и, таким образом, направлены на улучшенное изучение найденных областей, а разведчики, в свою очередь, выполняют поиск новых перспективных областей пространства поиска, в которых может находиться оптимум. За счёт такого разделения труда достигается эффективная работа всей группы агентов. При этом агенты информируют других агентов об областях при помощи прямой связи, которая достигается путём моделирования виляющего танца, в результате свободные агенты (незанятые фуражиры) могут стать занятыми фуражирами.

Формальное описание такого поведения агентов с прямой связью между ними может быть приведено в терминах исчисления взаимодействующих систем (Calculus of Communicating Systems, CCS) [11].

CCS – математическая модель (формальный язык), предназначенная для описания процессов, обычно применяемая при изучении параллелизма. CCS содержит набор термов, операторов и аксиом, которые используются для описания и управления составленными выражениями. Выражения характеризуют элементы параллельной системы, а управление этими выражениями показывает, как ведёт себя система. Центральным элементом в CCS является уникально именованный агент, который обладает специфическим поведением. Поведение агента определяется множеством событий и действий, которые может выполнять агент. Множество событий, выполняемых агентом, описывается с помощью оператора "действие", который обозначается как ".". Другим важным оператором в CCS является оператор "+", который является оператором выбора. Этот оператор используется в случае, если описывается участие агента в одном из нескольких альтернативных действий.

CCS [11] работает с агентами и с действиями, которые представляют собой возможность перехода из одного состояния в другое. Таким образом, колония может быть представлена в виде связного графа, вершинами которого являются агенты, а ребрами – действия. Например, агент может быть представлен в следующем виде:

Разведчик_b = хороший_b(s).Поиск_b(s) + плохой_b.Незанятый_b.

Представленный агент *b* является разведчиком, который ищет область с оптимумом. В случае, если он будет *плохо* искать оптимум, то он станет

незанятым фуражиром, если же он будет *хорошо* искать, то он продолжит *Поиск_b(s)* некоторой области с оптимумом *s*.

Тогда процесс фуражировки можно формализовать с помощью CCS в следующем виде:

Использовать_b(s) = κ_{b,s}.Поиск_b(s),

Поиск_b(s) = нектар_{b,s}.Удачно_b(s) + ничего_s.Неудачно_b(s),

Удачно_b(s) = из_{b,s}.Вербовка_b(s),

Неудачно_b(s) = из_{b,s}.Незанятый_b,

Вербовка_b(s) = танец(b, s).Вербовка_b(s) + покинуть_b.Использовать_b(s),

Незанятый_b = танец(b', s).Использовать_b(s) + исследовать_b.Разведчик_b,

Разведчик_b = хороший_b(s).Поиск_b(s) + плохой_b.Незанятый_b.

В данном описании *b* – уникальный идентификатор одного агента, *b'* – идентификатор другого агента, *s* – источник нектара.

Таким образом, с помощью предложенной формализации моделируется поведение агентов с прямой связью между ними. Незанятый агент остаётся свободным до тех пор, пока не получит сигнал от занятого фуражира посредством обмена информацией; если такой сигнал получен, то агент перемещается к соответствующей области с оптимумом; в противном случае – он может сам стать разведчиком через некоторое время; если в предполагаемой области есть оптимум, то агент разрабатывает его и моделирует обмен информацией.

В результате анализа различных применений данного мультиагентного метода с прямой связью между агентами [15 – 20] можно выделить следующие его преимущества:

- метод не склонен к закливанию в локальных оптимумах, поскольку основан на случайном поиске;
- поиск лучшего решения основывается на решениях агентов всей колонии;
- метод может применяться в динамических приложениях, поскольку способен адаптироваться к изменениям окружающей среды;
- метод не выдвигает никаких требований к целевой функции;
- метод может использоваться для решения как дискретных, так и непрерывных задач оптимизации.

К недостаткам метода можно отнести:

- достаточно высокую итеративность метода;
- трудности теоретического анализа процесса получения решений, обусловленные тем, что поиск решения имеет стохастическую природу;
- априорную неопределённость времени поиска, хотя сходимость гарантируется;
- зависимость метода от настройки параметров, подбираемых экспериментально.

Таким образом, метод является эффективным и может быть рекомендован для решения задачи параметрического синтеза нейро-фаззи сетей, поскольку он не обладает недостатками традиционных методов, связанных с накладываемыми ограничениями на архитектуру сети. Однако рассматриваемый мультиагентный подход с прямой связью между агентами обладает и недостатками, которые должны быть учтены при разработке соответствующего метода параметрического синтеза нейро-фаззи сетей.

Параметрический синтез нейро-фаззи сетей на основе метода пчелиной колонии. Как было отмечено ранее, задача параметрического синтеза нейро-фаззи сети может рассматриваться как задача многомерной оптимизации, поскольку требуется найти значения параметров архитектуры нейро-фаззи сети, при которой достигается минимум ошибки прогнозирования такой сети. К настраиваемым параметрам нейро-фаззи сети относятся:

- параметры функций принадлежности термов входных переменных;
- веса правил;
- параметры функций принадлежности термов выходной переменной (в случае параметрического синтеза нейро-фаззи сети типа Мамдани).

Поскольку функции принадлежности обычно определяются 2 – 4 параметрами, а при этом каждой входной переменной соответствует несколько нечётких термов, а соответственно, и несколько функций принадлежности, то общее количество настраиваемых параметров будет значительным. В связи с этим в традиционный мультиагентный метод с прямой связью между агентами предлагается ввести дополнительные процедуры, которые позволили бы разрабатываемому методу лучше и быстрее изучать пространство поиска, а также быстрее сходиться к оптимальному решению:

- процедура имитации отжига [21], позволяющая со временем отбирать всё меньшее количество агентов для анализа полученных ими решений, что позволит повысить сходимость к оптимальному решению;
- применение операторов генетического поиска [22], в частности, оператора скрещивания, что позволит более детально изучить пространство поиска.

Разработанный метод параметрического синтеза нейро-фаззи сетей, основанный на мультиагентном подходе с прямой связью между агентами, моделирующий поведение пчёл, можно представить в виде последовательности следующих шагов.

Шаг 1. Задаются начальные параметры работы метода: обучающая выборка, по которой будет рассчитываться ошибка прогнозирования нейро-фаззи сети; начальное количество агентов-разведчиков B_s ; допустимая ошибка прогнозирования ε^* ; максимальная энергия; коэффициент α ; начальная температура T_{init} ; конечная температура T_{final} ; максимальное количество итераций $iter_{max}$. Также задаются параметры, связанные с архитектурой нейро-фаззи сети: количество настраиваемых переменных $argCnt$; минимальный

$minRange$ и максимальный $maxRange$ пределы для каждой из переменных, т.е. область, в которой можно искать возможные решения.

Шаг 2. Создаются начальные агенты-разведчики.

Шаг 2.1. Для каждого начального агента-разведчика создаётся случайное решение:

$$bee_i.x_j = rand \cdot (\max Range_j - \min Range_j) + \min Range_j, \forall i = \overline{1, B_s}, j = \overline{1, argCnt},$$

где bee_i – i -й агент-разведчик; $rand$ – случайное число из интервала $[0, 1]$.

Шаг 2.2. Для полученных случайных решений рассчитывается полезность данного источника нектара как значение ошибки прогнозирования соответствующей нейро-фаззи сети:

$$bee_i.profitability = \varepsilon [FN(bee_i.x_1, bee_i.x_2, \dots, bee_i.x_{argCnt})], \forall i = \overline{1, B_s}. \quad (3)$$

Устанавливается текущее количество итераций: $iter = 1$; устанавливается количество агентов-разведчиков: $B = B_s$; устанавливается текущая температура: $T = T_{init}$.

Шаг 3. Выбираются рабочие агенты, т.е. такие агенты, на базе которых будут создаваться новые агенты с помощью процедуры скрещивания.

Шаг 3.1. Определяется агент $best$ с наибольшей полезностью.

Шаг 3.2. Процедура имитации отжига [21]. Агент относится к рабочим агентам $workBee$, если выполняется условие:

$$e^{\frac{|bees_i.profitability - best.profitability|}{T}} > rand, \forall i = \overline{1, B}. \quad (4)$$

Шаг 4. Скрещивание. Поскольку реальные пчёлы-разведчики при выборе источника нектара пользуются также генетическим материалом (в биологии ещё не изучено, каким именно образом разведчики выбирают одни цветки и пропускают другие, то есть предполагается, что разведчики основываются на генетическом опыте), то с помощью процедуры скрещивания моделируется именно этот момент поведения пчёл. Для скрещивания используются ранее отобранные с помощью процедуры имитации отжига рабочие агенты $workBee$ и лучший агент за все итерации $best$. Новые агенты создаются в два этапа: на базе решений рабочих агентов и на базе решения лучшего агента.

Шаг 4.1. Создание новых агентов на базе рабочих агентов $workBee$:

$$newWorkBee_i.x_j = workBee_i.x_j \pm rand \cdot (workBee_i.x_j - best.x_j), \forall i = \overline{1, B_w}, j = \overline{1, argCnt},$$

где B_w – количество рабочих агентов $workBee$; знак преобразования "+" или "-" выбирается случайным образом.

Шаг 4.2. Создание новых агентов на базе лучшего агента $best$:

$$newWorkBee_i.x_j = best.x_j \pm rand \cdot (workBee_i.x_j - best.x_j), \forall i = \overline{1, B_w}, j = \overline{1, argCnt}.$$

Шаг 4.3. Для всех новых агентов производится корректировка полученных решений, поскольку полученные таким образом значения переменных могут выходить за пределы $[minRange; maxRange]$.

Шаг 4.4. Рассчитывается полезность полученных решений:

$$newWorkBee_i \cdot profitability = \varepsilon [FN(newWorkBee_i \cdot x_1, \dots, newWorkBee_i \cdot x_{argCnt})], \forall i = \overline{1, B_n},$$

где B_n – количество созданных при скрещивании агентов $newWorkBee$.

Шаг 4.5 Выбирается новый лучший агент $best$.

Шаг 5. Моделирование обмена информацией между агентами. К возможному обмену информацией допускаются рабочие агенты $workBee$, агенты, созданные путём скрещивания, $newWorkBee$, лучший агент за все итерации $best$. Моделирование обмена информацией происходит в несколько этапов. В результате данного моделирования выбираются те агенты, которые за счёт предоставления информации о найденных ими решениях выполняют вербовку других агентов для исследования области с данным решением.

Шаг 5.1. Выполняется нормирование полезностей агентов, допущенных к возможности выполнения танца. Нормирование выполняется следующим образом:

$$np_i = 1 - \frac{bee_i \cdot profitability}{profitability_{\max} - profitability_{\min}}, \quad (5)$$

где np_i – нормированная полезность i -го агента; $profitability_{\max}$ и $profitability_{\min}$ – максимальная и минимальная полезности среди всех агентов, соответственно.

Шаг 5.2. Добавление шумов к полученным нормированным полезностям и их корректировка:

$$np_i = \begin{cases} 1, & \text{если } np_i + w_i > 1; \\ np_i + w_i, & \text{если } e_n < np_i + w_i < 1; \\ 0, & \text{если } 0 < np_i + w_i < e_n, \end{cases} \quad (6)$$

где w_i – шум в полезности. Шум равномерно распределён в интервале $(-w, +w)$. Значение w выбирается экспериментально (предлагается $w = 0,1$); e_n – минимальный порог полезности. Минимальный порог выбирается экспериментально (предлагается $e_n = 0,1$).

Шаг 5.3. Определение достоинства решения каждого агента:

$$L_i = \max \{ np_i - \eta \cdot \overline{np}, 0 \},$$

где L_i – достоинство решения i -го агента; η – коэффициент, управляющий влиянием величины \overline{np} на L_i ; \overline{np} – среднее значение нормированной полезности всех агентов, которые допущены к моделированию обмена информацией:

$$\overline{np} = \frac{1}{B_c} \sum_{i=1}^{B_c} np_i,$$

где B_c – количество агентов, допущенных к обмену информацией.

Шаг 5.4. Выбор тех агентов, которые за счёт обмена информацией выполняют вербовку других агентов для исследования найденного ими решения. Агент считается выполнившим вербовку, если:

$$\frac{L_i}{\beta} > \gamma \cdot \overline{np}, \quad \forall i = \overline{1, B_c},$$

где $\beta > 0$ – коэффициент, понижающий влияние достоинства решения на вероятность выполнения вербовки; $\gamma \in (0; 1)$ – пороговый коэффициент, определяющий, на сколько выше должно быть достоинство решения данного i -го агента относительно средней полезности решений всех B_c агентов.

Шаг 6. Выбирается агент с лучшим решением *best*.

Шаг 7. Перезапуск агентов. Создаются агенты, которые будут рассматриваться как агенты-разведчики для следующей итерации.

К новым агентам-разведчикам будут относиться:

- агенты, выполнившие посредством танца вербовку, лучший агент;
- агенты, которые стали занятыми фуражирами вследствие вербовки.

Поскольку такие агенты должны выполнять улучшенное изучение уже существующей области с решением, то при создании решений для данных агентов должны учитываться решения завербовавших их агентов. В связи с этим для завербованных агентов решение создаётся следующим образом:

$$x_j = \text{dancedBee}.x_j + \text{range} \cdot \text{rand} - \frac{\text{range}}{2}, \quad \forall j = \overline{1, \text{argCnt}},$$

где *range* – это предел, в котором величина переменной может отклоняться от значения данной j -ой переменной в решении агента, моделировавшего обмен информацией; *dancedBee*. – агенты, решение которых создаётся случайным образом:

$$x_j = \text{rand} \cdot (\max \text{Range}_j - \min \text{Range}_j) + \min \text{Range}_j, \quad \forall j = \overline{1, \text{argCnt}}.$$

Также для всех созданных агентов рассчитывается полезность выбранного решения.

Шаг 8. Обновление параметров метода:

- увеличивается количество итераций: $\text{iter} = \text{iter} + 1$;
- изменяется текущая температура: $T = \alpha \cdot T$;
- изменяется предел *range*: $\text{range} = \text{range} \cdot \frac{\text{iter}_{\max} - \text{iter}}{\text{iter}_{\max}}$.

Шаг 9. Проверка условий на останов. Проверка условий на останов считается успешной, если выполняется хотя бы одно из условий:

- максимальное количество итераций достигнуто: $\text{iter} = \text{iter}_{\max}$;
- текущая температура равна конечной температуре: $T = T_{\text{final}}$;
- ошибка прогнозирования лучшей нейро-фаззи сети находится в допустимых рамках: $\varepsilon_{\text{best}} < \varepsilon^*$.

Если проверка на останов дала успешный результат, тогда выполняется переход на шаг 10, в противном случае – на шаг 3.

Шаг 10. Останов.

Эксперименты и результаты. Для проверки и исследования работоспособности и эффективности разработанного мультиагентного метода параметрического синтеза нейро-фаззи сетей с прямой связью была выполнена программная реализация данного метода. Для выполнения параметрического синтеза нейро-фаззи сети при помощи метода обратного распространения ошибки использовалась библиотека, реализующая его.

Для экспериментов использовалась сеть Сугено, параметры которой настраивались при помощи традиционного метода обратного распространения ошибки, и сеть Мамдани, которая настраивалась с помощью предложенного мультиагентного метода параметрического синтеза нейро-фаззи сетей. Выполнялась аппроксимация следующей двухмерной функции:

$$z = x^2 \cdot \sin(y - 1), \quad x \in [-7; 3], \quad y \in [-4,5; 1,7].$$

График рассматриваемой функции представлен на рис. 1.

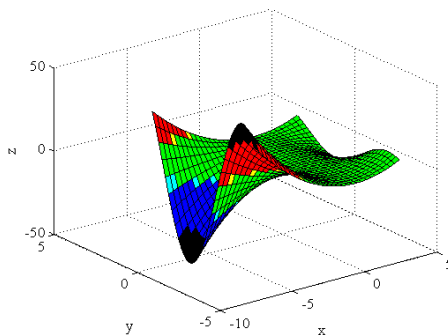


Рис. 1. График функции z

При этом входные и выходные переменные рассматриваются как лингвистические переменные, которые определяются следующими термами-множествами:

- для x и y : низкий, средний, высокий;
- для z : низкий, ниже среднего, средний, выше среднего, высокий.

Для формализации термов использовалась симметричная гауссовская функция принадлежности. При этом для нейро-фаззи сети типа Мамдани для выходной переменной использовалась эта же функция принадлежности, в то время как для нейро-фаззи сети типа Сугено в связи с ограничениями на её архитектуру использовалась линейная функция принадлежности.

Графики полученных аппроксимирующих зависимостей приведены на рис. 2.

Дисперсии, показывающие различия между тестовой выборкой и значениями целевой функции, рассчитанные на основе полученных нейро-фаззи сетей типа Сугено и Мамдани, составили 0,1388 и 0,0272, соответственно. Следовательно, можно сделать вывод, что в обоих случаях отклонение полученной зависимости от аппроксимируемой зависимости является незначительным, при этом отклонение нейро-фаззи сети типа Мамдани, настроенной с использованием предложенного мультиагентного метода меньше, что свидетельствует об эффективности разработанного мультиагентного метода с прямой связью между агентами для параметрического синтеза нейро-фаззи сетей.

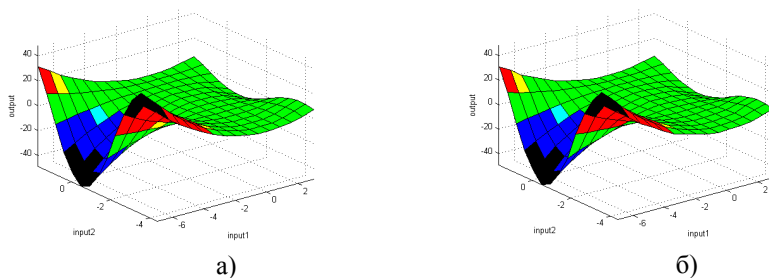


Рис. 2. Графическое представление полученных аппроксимирующих зависимостей для: а) нейро-фаззи сети типа Сугено; б) нейро-фаззи сети типа Мамдани

Представленные графики также подтверждают, что полученные зависимости незначительно отличается от аппроксимируемой.

Выводы. В работе решена актуальная задача параметрического синтеза нейро-фаззи сетей на основе мультиагентного метода оптимизации с прямой связью между агентами, который эффективно позволяет решать многомерные задачи непрерывной оптимизации. Однако, в связи с большой размерностью решаемой задачи было решено добавить дополнительные процедуры в метод, позволяющие повысить сходимость к лучшему решению, а также позволяющие более детально исследовать пространство поиска.

Научная новизна работы заключается в том, что впервые предложен мультиагентный метод с прямой связью между агентами, позволяющий выполнять параметрическую идентификацию нейро-фаззи сетей Мамдани и Сугено, что достигается за счёт применения мультиагентного подхода с дополнительными процедурами, позволяющими отбрасывать бесперспективные области поиска и более детально исследовать новые области пространства поиска.

Практическая ценность работы заключается в разработке программного обеспечения, реализующего предложенный метод, и позволяющего осуществлять параметрический синтез нейро-фаззи сетей.

Результаты проведенных экспериментов подтвердили эффективность разработанного мультиагентного метода с прямой связью между агентами для параметрического синтеза нейро-фаззи сетей, поскольку полученные аппроксимирующие зависимости характеризовались точностью, которая была не хуже аппроксимирующих зависимостей, полученных при помощи применения традиционных методов параметрической идентификации нейро-фаззи сетей.

- Список литературы:** 1. *Zadeh L.* Fuzzy sets // *Information and Control*. – 1965. – № 8. – P. 338–353. 2. *Mamdani E.H., Assilian S.* An Experiment in Linguistic Synthesis with Fuzzy Logic Controller // *Int. J. Man-Machine Studies*. – 1975. – Vol. 7. – № 1. – P. 1–13. 3. *Bellman R.E., Zadeh L.A.* Decision-Making in Fuzzy Environment // *Management Science*. – 1970. – № 4. – P. 141–160. 4. *Holmblad L.P., Ostergaard J.J.* Control of a Cement Kiln by Fuzzy Logic // *Fuzzy Information and Decision Processes*. – Amsterdam: North-Holland, 1982. – P. 389–400. 5. *Jang J.R.* ANFIS: Adaptive-Network-Based Fuzzy Inference System // *IEEE Trans. Systems & Cybernetics*. – 1993. – Vol. 23. – P. 665–685. 6. *Nauck D., Klawonn F., Kruse R.* Foundations of Neuro-Fuzzy Systems. – Chichester: John Wiley & Sons, 1997. – 305 p. 7. *Beni G., Wang J.* Swarm Intelligence // *Annual Meeting of the Robotics Society: Proceedings of Seventh International Conference*. – Tokyo: RSJ Press, 1989. – P. 425–428. 8. *Dorigo M.* Optimization, Learning and Natural Algorithms. – Milano: Politecnico di Milano, 1992. – 140 p. 9. *Dorigo M., Gambardella L.M.* Ant Colony System: A cooperative learning approach to the traveling salesman problem // *IEEE Transactions on Evolutionary Computation*. – 1997. – № 1 (1). – P. 53–66. 10. *Олейник А.А.* Сравнительный анализ методов оптимизации на основе метода муравьиных колоний // *Комп'ютерне моделювання та інтелектуальні системи: Збірник наукових праць / За ред. Д.М. Пізи, С.О. Субботіна*. – Запоріжжя: ЗНТУ, 2007. – С. 147–159. 11. *Sumpter D.T., Broomhead D.S.* Formalising the Link between Worker and Society in Honey Bee Colonies // *Lecture Notes In Computer Science: Proceedings of the First International Workshop on Multi-Agent Systems and Agent-Based Simulation*. – MABS '98 LNAI, 1998. – P. 95–110. 12. *Subbotin S.A., Oleynik A.A.* PSO with Control of Velocity Change for Feature Selection // *Inductive Modeling: Proceedings of 2nd International Conference ICIM'2008*, 2008. – Kyiv, 2008. – P. 81–83. 13. *Passino K.M.* Biomimicry of bacterial foraging for distributed optimization and control // *IEEE Control System Magazine*, 2002. – № 3 (22). – P. 52–67. 14. *De Castro L.N., Von Zuben F.J.* Artificial Immune Systems. Part I. Basic Theory And Applications / L. N. De Castro. – Technical Report No. Rt Dca 01/99. – Brazil: Feec/Unicamp, 1999. – 95 p. 15. *Chong S.C., Low M.H.* A Bee Colony Optimization Algorithm to Job Shop // *Winter Simulation Conference: Proceedings of the 38th conference on Winter simulation*. – Monterey: Monterey Press, 2006. – P. 1954–1961. 16. *Lučić P., Teodorović D.* Bee System: Modeling Combinatorial Optimization Transportation Engineering Problems by Swarm Intelligence // *Transportation Analysis: Proceedings of the Triennial Symposium TRISTAN IV*. – Sao Miguel: Azores Press, 2001. – P. 441–445. 17. *Teodorović D., Dell'Orco M.* Bee Colony Optimization – a Cooperative Learning Approach to Complex Transportation Problems // *Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini-EURO Conference and 10th Meeting of EWGT*. – 2005. – Poznan: Publishing House of the Polish Operational and System Research, 2005. – P. 51–60. 18. *Quijano N., Passino K.M.* Honey Bee Social Foraging Algorithms for Resource Allocation: Theory and Application. – Columbus: Publishing house of the Ohio State University, 2007. – 39 p. 19. *Nakrani S., Tovey C.* On Honey Bees and dynamic allocation in an internet server colony // *Adaptive Behavior*. – 2004. – № 12. – P. 223–240. 20. *Passino K.M., Seeley T.D.* Modeling and analysis of nest-site selection by honey bee swarms: The speed and accuracy trade-off // *Behavioral Ecology and Sociobiology*. – 2006. – № 59. – P. 427–442. 21. *Kirkpatrick S., Gelatt C.D., Vecchi M.P.* Optimization by simulated annealing // *Science*. – 1983. – Vol. 220. – P. 671–680. 22. *Gen M., Cheng R.* Genetic algorithms and engineering design. – New Jersey: John Wiley & Sons, 1997. – 352 p.