

Федеральное государственное автономное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

На правах рукописи



Сарин Константин Сергеевич

**МЕТОДОЛОГИЯ ПОСТРОЕНИЯ ИНТЕРПРЕТИРУЕМЫХ
НЕЧЕТКИХ КЛАССИФИКАТОРОВ, ОСНОВАННЫХ НА
ПРАВИЛАХ**

Специальность 2.3.8 – «Информатика и информационные процессы»

Диссертация на соискание ученой степени доктора технических наук

Научный консультант:

доктор технических наук, профессор

Ходашинский Илья Александрович

Томск 2025

Оглавление

Введение	5
1 Аналитический обзор методов построения нечетких классификаторов на основе анализа данных.....	18
1.1 Нечеткие системы.....	18
1.2 Нечеткий классификатор типа Min-Max	23
1.3 Генетические нечеткие системы	35
1.4 Метод трехэтапного построения нечетких классификаторов.....	47
1.5 Критерий интерпретируемости для построения нечетких систем.....	53
1.6 Постановка проблемы	56
1.7 Краткие выводы по разделу.....	62
2 Построение интерпретируемых нечетких классификаторов, основанных на правилах	65
2.1 Методология построения интерпретируемых нечетких классификаторов, основанных на правилах.....	65
2.2 Задача смешанной многокритериальной оптимизации.....	71
2.3 Метаэвристический алгоритм смешанной многокритериальной оптимизации	77
2.4 Дискретный алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений	86
2.5 Адаптированный алгоритм градиентного спуска для оптимизации параметров нечетких классификаторов.....	95
2.6 Метод разделения данных для построения прогностических моделей машинного обучения	102
2.7 Методика построения интерпретируемых нечетких классификаторов....	116

2.8	Краткие выводы по разделу.....	129
3	Вычислительные эксперименты и сравнение с аналогами.....	132
3.1	Нахождение фронта Парето тестовых функций алгоритмом смешанной многокритериальной оптимизации	133
3.2	Нахождение оптимума тестовых функций дискретным алгоритмом оптимизации	144
3.3	Построение нечетких классификаторов с применением метода разделения данных	154
3.4	Настройка параметров нечетких классификаторов адаптированным алгоритмом градиентного спуска.....	165
3.5	Построение нечетких классификаторов.....	169
3.6	Краткие выводы по разделу.....	182
4	Внедрение результатов диссертационной работы	185
4.1	Прогнозирование эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19.....	185
4.2	Синтез согласующих цепей на основе смешанной многокритериальной оптимизации	191
4.3	Синтез секции ступенчатого аттенюатора с помощью дискретного алгоритма оптимизации.....	206
4.4	Краткие выводы по разделу.....	208
	Заключение.....	210
	Список литературы.....	217
	ПРИЛОЖЕНИЕ А. Свидетельства о государственной регистрации программ для ЭВМ	250
	ПРИЛОЖЕНИЕ Б. Акты о внедрении (использовании) результатов диссертационного исследования	252

ПРИЛОЖЕНИЕ В. Признаки набора данных для прогнозирования эффективности реабилитации.....	258
---	------------

Введение

Актуальность темы исследования и степень ее разработанности. Системы искусственного интеллекта (СИИ), основанные на методах машинного обучения, нашли широкое применение во многих сферах жизнедеятельности человека, в том числе и таких критически важных как здравоохранение, энергетика и безопасность [1]. Поскольку решения, принимаемые такими системами, являются решениями с высокими ставками и влияют на жизни людей, появляется необходимость понимания того, как они были получены [2, 3]. Отсутствие такой возможности может повлиять на уровень доверия человека к результатам решений и вызвать опасения по поводу причинения вреда [4, 5]. Это может замедлить общественное принятие и внедрение СИИ.

В Национальной стратегии развития искусственного интеллекта на период до 2030 года, утвержденной указом Президента Российской Федерации от 10.10.2019 № 490 (ред. от 15.02.2024), отмечается, что отсутствие понимания того, как искусственный интеллект достигает результатов, является одной из причин низкого уровня доверия к современным технологиям искусственного интеллекта и может стать препятствием для их развития. В национальном стандарте Российской Федерации ГОСТ Р 59276-2020 «Системы искусственного интеллекта. Способы обеспечения доверия» одним из факторов снижения качества СИИ и доверия к ним указывается недостаточная понятность и объяснимость. В качестве способа достижения доверия предлагается использовать интеллектуальные алгоритмы обработки данных, обеспечивающие принятие системой объяснимых решений.

В последние годы произошел рост количества систем принятия решений, основанных на нейронных сетях глубокого обучения. Эмпирический успех моделей глубокого обучения обусловлен сочетанием эффективных алгоритмов обучения и их огромного параметрического пространства, которое может включать сотни слоев и миллионы параметров [6]. Это является причиной того,

что такие СИИ рассматриваются как сложные модели «черного ящика» [7], исключая прозрачность или поиск понимания механизма, на основе которого работает модель [8]. В Национальной стратегии развития искусственного интеллекта отмечается, что алгоритмы работы нейронных сетей крайне сложны для интерпретации и, следовательно, результаты их работы могут быть подвергнуты сомнению и отменены человеком.

Решение указанной выше проблемы предлагается осуществлять с помощью объяснимого искусственного интеллекта (ОИИ) [9]. СИИ обладает свойством объяснимости, если способна предоставлять доказательства или обоснование человеку в поддержку своего решения [10, 11]. Взаимодействие человека и СИИ играет ключевую роль в ОИИ. Полученные системой знания должны использоваться для предоставления объяснений; кроме того, пользователи должны иметь возможность оспаривать и задавать вопросы системам, чтобы понять обоснование решения в соответствии со своими конкретными потребностями [12]. СИИ, обладающие способностью объяснить человеку результат своего решения, также называют интерпретируемыми системами [13].

Нечеткие системы имеют большой потенциал в разработке решений ОИИ. Благодаря наличию базы продукционных правил они способны выразить закономерности в данных в ориентированной на человека форме, используя термины естественного языка [14]. Такая возможность позволяет предоставить понятные объяснения используемых системой знаний, а также детали, касающиеся процесса вывода при принятии решений [15]. С помощью теории нечеткой логики, лежащей в основе таких систем, была установлена связь между числовыми решениями и правилами на естественном языке. Таким образом, численные операции, осуществляющие процедуру принятия решения в нечетких системах, интерпретируются логическими рассуждениями во вполне понятной человеку лингвистической форме. Это побудило использовать нечеткие методы для разработки ОИИ [16].

Решением проблемы построения нечетких систем занимались А.Н. Аверкин, И.З. Батыршин, М.В. Бобырь, А.В. Боженюк, В.В. Борисов, Л.А.

Гладков, М.И. Дли, А.С. Катасёв, С.М. Ковалев, Ю.И. Кудинов, В.В. Курейчик, Т.М. Леденева, И.А. Мочалов, В.Г. Синюк, В.И. Сырямкин, В.Л. Хацкевич, И.А. Ходашинский, А.В. Язенин, Н.Г. Ярушкина, Р. Angelov, J.C. Bezdek, O. Cordon, D. Dubois, D. Filev, A. Gonzalez, S. Guillaume, F. Herrera, H. Ishibuchi, B. Kosko, F. Kruse, E.H. Mamdani, J.M. Mendel, S. Oh, W. Pedrycz, H. Prade, P.K. Simpson, M. Sugeno, T. Takagi, H. Tanaka, R.R. Yager, T. Yasukawa, L. Wang, L. Zadeh и др.

Нечеткие классификаторы являются нечеткими системами, осуществляющими процесс классификации или нахождения класса объекта по известным значениям его признаков [17, 18]. Продукционные правила таких систем связывают значения признаков объекта, представленных в лингвистической форме, с их классом и позволяют объяснить человеку результат своего решения. Анализ известных методов построения нечетких классификаторов выявил, что при высокой точности полученные модели имеют недостатки интерпретируемости, а это является критичным для систем ОИИ. Нечеткие классификаторы типа Min-Max содержат большое количество правил и формы функций принадлежности, которые затруднительно интерпретировать в лингвистические термины. Генетические нечеткие системы помимо большого числа правил имеют весовые коэффициенты правил, что вызывает трудности для объяснения полученного результата. Трехэтапный метод построения нечетких систем также генерирует функции принадлежности, вызывающие сложности при интерпретации в лингвистические термины. Нейро-нечеткие системы содержат в консеквентной части правил не метку класса, а функцию от значений признаков образца, затрудняющую объяснение результата.

На основании вышеизложенного можно сделать вывод о том, что разработка методов и алгоритмов построения интерпретируемых нечетких классификаторов для систем ОИИ, позволяющих объяснить человеку результат своего решения, является актуальной проблемой, требующей исследований для ее решения.

Цель и задачи исследования. Цель исследования заключается в повышении эффективности обнаружения закономерностей в данных с помощью

разработки и применения методологии построения интерпретируемых нечетких классификаторов, основанных на правилах.

Для достижения поставленной цели необходимо решить следующие основные задачи.

1. Провести аналитический обзор методов построения нечетких классификаторов.

2. Разработать методологию построения интерпретируемых нечетких классификаторов, основанных на правилах, которая определяет совокупность методов анализа данных и их взаимодействие.

3. Разработать и программно реализовать эффективные алгоритмы однокритериальной и многокритериальной оптимизации для поиска оптимума в смешанном и дискретном пространстве.

4. Разработать метод разделения данных на обучающую и валидационную выборки, способный минимизировать потери характеристических свойств данных при разделении и повысить точность моделей машинного обучения.

5. Разработать алгоритм оптимизации параметров на основе градиентного спуска для повышения точности нечетких классификаторов, построенных метаэвристическими алгоритмами.

6. Разработать методику построения нечетких классификаторов на основе предложенной методологии, способную учитывать критерии точности и интерпретируемости и использующую разработанные алгоритмы и методы.

7. Провести вычислительные эксперименты для оценки точности и интерпретируемости нечетких классификаторов, построенных с помощью предлагаемой методики, и сравнить с известными интерпретируемыми прогностическими моделями.

8. Применить результаты исследования для прогнозирования эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19 и синтеза согласующих цепей.

Объектом исследования является процесс построения систем ОИИ на основе анализа данных.

Предметом исследования являются методы и алгоритмы построения интерпретируемых нечетких классификаторов, основанных на правилах.

Методы исследования. В диссертационной работе применялись методы оптимизации, интеллектуального анализа данных, теории нечетких множеств, линейной алгебры, теории алгоритмов, теории вероятностей и математической статистики.

Достоверность результатов обеспечивается следующими основными факторами: 1) строгостью применения математических методов; 2) использованием общепринятых тестов и общедоступных наборов данных для проведения численных экспериментов по оптимизации и построению нечетких систем; 3) проведением экспериментов на лицензионном программном обеспечении MATLAB, результаты которых не противоречат результатам, полученным другими исследователями.

Научная новизна полученных результатов. В диссертации получены следующие новые научные результаты.

1. Предложена методология построения интерпретируемых нечетких классификаторов, основанных на правилах, определяющая совокупность методов анализа данных и их взаимодействие. Отличительной особенностью методологии является совместное применение методов разделения данных на обучающую и валидационную выборки, метаэвристических методов смешанной многокритериальной оптимизации для одновременной структурной и параметрической оптимизации классификаторов и стратегий выбора классификатора из сформированного Парето-множества. Методология позволяет разрабатывать интерпретируемые нечеткие классификаторы, сопоставимые по точности с существующими аналогами.

2. Разработан метаэвристический алгоритм многокритериальной оптимизации для поиска в смешанном дискретно-непрерывном пространстве. Отличительной особенностью алгоритма является совместное применение метаэвристики «кукушкин поиск» для формирования непрерывных элементов решения и генетического оператора скрещивания для формирования дискретных

элементов решения. Алгоритм позволяет улучшить эффективность поиска в смешанном пространстве по сравнению с широко используемым генетическим алгоритмом с недоминируемой сортировкой NSGAII.

3. Разработан метаэвристический алгоритм однокритериальной оптимизации на основе распределения вероятностей значений переменных для поиска в дискретном пространстве. Особенностью алгоритма является построение распределения вероятностей с помощью трансформации целевых значений решений в их весовые коэффициенты. Алгоритм позволяет улучшить эффективность поиска в бинарном пространстве по сравнению с широко используемыми генетическим алгоритмом и алгоритмом роящихся частиц.

4. Для оптимизации параметров нечеткого классификатора адаптирован алгоритм градиентного спуска, отличительной особенностью которого является представление нечеткого классификатора в виде системы разделяющих функций и нормирование их значений в целевой функции оптимизации. Алгоритм позволяет улучшить точность классификаторов, построенных метаэвристическим алгоритмом.

5. Предложен новый метод разделения данных при построении прогностических моделей машинного обучения, основанный на применении бинарного метаэвристического алгоритма для формирования обучающей и валидационной выборок близких по характеристикам к исходным данным. Метод позволяет улучшить точность и интерпретируемость нечетких классификаторов, построенных метаэвристическим алгоритмом смешанной многокритериальной оптимизации.

6. Разработана методика построения нечетких классификаторов с соблюдением компромисса «точность – интерпретируемость», отличительной особенностью методики является применение новых методов и алгоритмов, а именно: 1) смешанной многокритериальной оптимизации с алгоритмом на основе совместного применения метаэвристики «кукушкин поиск» и генетического оператора скрещивания для одновременной структурной и параметрической оптимизации классификаторов по критериям точности и интерпретируемости, 2)

метода разделения данных на обучающую и валидационную выборки дискретным алгоритмом оптимизации на основе распределения вероятностей с трансформацией целевых значений, 3) адаптированного алгоритма градиентного спуска для параметрической оптимизации классификаторов, настраивающей параметры функций принадлежности. Методика позволяет при сопоставимой точности улучшить интерпретируемость классификаторов по сравнению с известными прогностическими моделями, такими как генетические нечеткие системы FARC-HD и деревья решений CART.

Теоретическая значимость работы заключается в развитии технологии построения нечетких систем интеллектуального анализа данных. Методика построения нечетких классификаторов является инструментом извлечения знаний из данных и интерпретации результатов, облегчающих понимание и объяснение процессов принятия решений, связанных с классификацией объектов. Алгоритмы смешанной многокритериальной оптимизации и дискретной оптимизации могут применяться для решения задач глобальной параметрической оптимизации и комбинаторной оптимизации. Метод разделения данных может быть использован для борьбы с переобучением и выбора подходящей по сложности структуры при построении прогностических моделей другого типа, например, нейронных сетей, деревьев решений, моделей на основе дискриминантного анализа и метода опорных векторов.

Практическая значимость работы подтверждается использованием полученных в ней результатов в разработке программных средств:

– алгоритм смешанной многокритериальной оптимизации использовался для разработки программных средств синтеза согласующих цепей с заданной формой амплитудно-частотной характеристики;

– алгоритм дискретной оптимизации на основе распределения вероятностей с трансформацией целевых значений внедрен в экспериментальную версию интеллектуальной САПР «50ohm Tech Circuit Studio»;

– методика построения нечетких классификаторов применялась в разработке программных средств для построения интерпретируемых моделей

прогнозирования эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19.

Разработанные методология, методы и алгоритмы использованы при выполнении следующих проектов.

1. РФФ, № 24-21-00168 «Методы и алгоритмы построения нечетких адаптивных классификаторов для решения задач кибербезопасности» 2024-2025 гг.

2. РФФ, № 22-21-00021 «Интерпретируемый нечеткий классификатор рукописных данных для диагностики нейродегенеративных заболеваний» 2022-2023 гг.

3. НИР «Создание информационной системы защищенного управления гетерогенными сетями и устройствами, образующими инфраструктуру «интернета вещей», выполненной в рамках государственного задания Министерства образования и науки Российской Федерации на 2017-2019 гг., проект № 2.3583.2017/4.6.

4. НИР «Теоретические основы человеко-машинных интерфейсов», выполненной в рамках государственного задания Министерства образования и науки Российской Федерации на 2017-2019 гг., проект № 8.9628.2017/8.9.

5. РФФИ, № 16-07-00034 «Методы и инструментальные средства построения самообучающихся систем, основанных на нечетких правилах» 2016–2018 гг.

На защиту выносятся приведенные ниже положения.

1. Методология построения позволила сформировать интерпретируемые нечеткие классификаторы, основанные на правилах и сопоставимые по точности с существующими аналогами.

Соответствует п.13 специальности: разработка и применение методов распознавания образов, кластерного анализа, нейро-сетевых и нечетких технологий, решающих правил, мягких вычислений при анализе разнородной информации в базах данных.

2. Алгоритм смешанной многокритериальной оптимизации позволил статистически значимо (на уровне $\alpha=0,05$) улучшить результаты поиска по сравнению с широко используемым генетическим алгоритмом с недоминируемой сортировкой NSGA II на общепринятых тестах ZDT и DTLZ. Значение обратного расстояния поколений в среднем уменьшилось на 95,47%. Степень доминирования увеличилась в среднем на 70,92%.

Соответствует п. 1 специальности: разработка компьютерных методов и моделей описания, оценки и оптимизации информационных процессов и ресурсов, а также средств анализа и выявления закономерностей на основе обмена информацией пользователями и возможностей и используемого программно-аппаратного обеспечения.

3. Алгоритм дискретной оптимизации позволил статистически значимо (на уровне $\alpha=0,05$) улучшить эффективность поиска в бинарном пространстве по сравнению с широко используемыми генетическим алгоритмом и алгоритмом роящихся частиц на общепринятых унимодальных и мультимодальных эталонных тестовых функциях. В среднем отклонение от оптимального значения целевой функции уменьшилось в 4,3 раза по сравнению с генетическим алгоритмом и в 10,1 раза по сравнению с бинарным алгоритмом роящихся частиц.

Соответствует п. 1 специальности: разработка компьютерных методов и моделей описания, оценки и оптимизации информационных процессов и ресурсов, а также средств анализа и выявления закономерностей на основе обмена информацией пользователями и возможностей и используемого программно-аппаратного обеспечения.

4. Адаптированный алгоритм градиентного спуска позволил статистически значимо (на уровне $\alpha=0,05$) повысить точность нечетких классификаторов, построенных метаэвристическим алгоритмом смешанной многокритериальной оптимизации «кукушкин поиск» с генетическим оператором скрещивания на 38 общедоступных наборах данных. Максимальное увеличение точности составило 5,1%.

Соответствует п. 1 специальности: разработка компьютерных методов и моделей описания, оценки и оптимизации информационных процессов и ресурсов, а также средств анализа и выявления закономерностей на основе обмена информацией пользователями и возможностей и используемого программно-аппаратного обеспечения.

5. Метод разделения данных при построении нечетких классификаторов многокритериальным метаэвристическим алгоритмом на 38 общедоступных наборах данных позволил статистически значимо (на уровне $\alpha=0,05$) повысить точность классификации по сравнению с построением без разделения данных. Максимальное увеличение точности составило 6,5%.

Соответствует п.7 специальности: разработка методов обработки, группировки и аннотирования информации, в том числе, извлеченной из сети интернет, для систем поддержки принятия решений, интеллектуального поиска, анализа.

6. Метод разделения данных при построении нечетких классификаторов многокритериальным метаэвристическим алгоритмом на 38 общедоступных наборах данных позволил статистически значимо (на уровне $\alpha=0,05$) повысить интерпретируемость классификаторов путем уменьшения количества правил, количества признаков и общего количества нечетких терминов по сравнению с построением без разделения данных.

Соответствует п.7 специальности: разработка методов обработки, группировки и аннотирования информации, в том числе, извлеченной из сети интернет, для систем поддержки принятия решений, интеллектуального поиска, анализа.

7. Методика построения нечетких классификаторов на 38 общедоступных наборах данных позволила при сопоставимой точности статистически значимо (на уровне $\alpha=0,05$) повысить интерпретируемость классификаторов путем уменьшения количества правил в среднем на 39,2%, количества признаков в среднем на 8%, общего количества нечетких терминов в среднем на 49,8% по сравнению с нечеткими классификаторами FARC-HD, и количества правил в

среднем на 81,9%, количества условий в правиле в среднем на 54,5% по сравнению с классификаторами на основе деревьев решений CART.

Соответствует п.13 специальности: разработка и применение методов распознавания образов, кластерного анализа, нейро-сетевых и нечетких технологий, решающих правил, мягких вычислений при анализе разнородной информации в базах данных.

Внедрение результатов диссертационного исследования. Результаты диссертационного исследования использованы в следующем.

1. В разработке программных средств для прогнозирования эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19. Внедрены в Томском научно-исследовательском институте курортологии и физиотерапии (Томском НИИКиФ).

2. В разработке программных средств для синтеза согласующих цепей. Внедрены в научно-исследовательском институте микроэлектронных система (НИИ МЭС) и ООО «Микроэлектронные системы».

3. В разработке интеллектуальной САПР «50ohm Tech Circuit Studio», позволяющей автоматически синтезировать структуры и параметры СВЧ интегральных схем. Внедрены в ООО «50ом Тех.».

4. В учебном процессе кафедры компьютерных систем в управлении и проектировании (КСУП) Томского государственного университета систем управления и радиоэлектроники (ТУСУР).

Апробация работы. Результаты работы представлялись и обсуждались на следующих конференциях.

– Международная IEEE Сибирская конференция по управлению и связи (SIBCON 2017), г. Астана, 2017 г.

– XVI Национальная конференция по искусственному интеллекту с международным участием (КИИ 2018), г. Москва, 2018 г.

– III Международная научная конференция «Интеллектуальные информационные технологии в технике и на производстве» (ИТИ-III), г. Сочи, 2018 г.

– Международная IEEE Сибирская конференция по управлению и связи (SIBCON 2019), г. Томск, 2019 г.

– Международная научно-методическая конференция «Современное образование: интеграция образования, науки, бизнеса и власти», г. Томск, 2023 г., 2025 г.

– XXXVII Международная научная конференция «Математические методы в технике и технологиях», г. Казань, 2024 г.

– Международная научно-практическая конференция «Электронные средства и системы управления», г. Томск, 2017 г., 2023 г., 2024 г.

– XXX Байкальская Всероссийская конференция с международным участием «Информационные и математические технологии в науке и управлении», г. Иркутск, 2025 г.

Публикации. По результатам исследования опубликовано 59 печатных работ, из которых в рекомендованных ВАК РФ периодических изданиях – 31. Двадцать работ индексируются в базах научных публикаций SCOPUS и WoS. Получены 8 свидетельств о государственной регистрации программ для ЭВМ.

Личный вклад автора. Автором проводилась постановка цели и задач научного исследования. Все результаты работы, составляющие научную новизну и сформулированные в положениях, выносимых на защиту, получены автором лично. Автором лично разработаны методология построения интерпретируемых нечетких классификаторов, алгоритм смешанной многокритериальной оптимизации, дискретный алгоритм оптимизации, метод разделения данных, адаптированный алгоритм градиентного спуска для настройки параметров нечетких классификаторов, методика построения интерпретируемых нечетких классификаторов, а также программные средства, реализующие разработанные алгоритмы, метод разделения, методику построения. Кроме этого, автором разработаны программные средства, осуществляющие синтез цепей согласования и прогнозирования эффективности реабилитации. Также автором проведены вычислительные эксперименты для оценки эффективности разработанных алгоритмов оптимизации, метода разделения данных, адаптированного алгоритма

градиентного спуска и методики построения интерпретируемых нечетких классификаторов. Вычислительный эксперимент для оценивания работы алгоритма смешанной многокритериальной оптимизации при синтезе согласующих цепей выполнен совместно с *Михаилом Владимировичем Черкашиным*. Формирование первичных данных для прогнозирования эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19 выполнено совместно с сотрудниками Томского НИИКиФ. Результаты по отдельным направлениям получены совместно с соавторами научных публикаций.

Благодарности. Автор выражает глубокую благодарность научному консультанту, профессору кафедры КСУП ТУСУРа, д.т.н. профессору *Илье Александровичу Ходашинскому* за помощь и поддержку в выполнении настоящей работы. Автор выражает благодарность коллегам доценту кафедры КСУП, к.т.н. *Михаилу Олеговичу Светлакову*, доценту кафедры КСУП, к.т.н. *Марине Борисовне Бардамовой* и сотрудникам лаборатории интеллектуальных систем за обсуждение исследования, ряд ценных советов и замечаний.

Структура и объем диссертации. Диссертация содержит введение, 4 раздела, заключение, список литературы из 289 наименований и приложений. Объем работы с приложениями составляет 260 с., в том числе 72 рисунка и 40 таблиц.

1 Аналитический обзор методов построения нечетких классификаторов на основе анализа данных

В разделе представлен аналитический обзор современных методов построения нечетких классификаторов. Рассмотрено понятие нечеткой системы и нечеткого классификатора. Определены преимущества и недостатки нечетких классификаторов типа Min-Max, генетических нечетких систем и метода трехэтапного построения. Определены показатели интерпретируемости. Формализовано понятие классификатора, а также архитектура нечеткого классификатора, используемого в работе.

1.1 Нечеткие системы

Для принятия решений человек использует две формы знаний: объективное и субъективное [19]. В первой форме знание точно определено и может быть оценено количественно, например, с помощью построения уравнения, описывающего исследуемый процесс или явление. Во втором случае знание представляет лингвистическую информацию, которую обычно невозможно оценить традиционными математическими методами. В таком виде знания могут выражаться не количественно, а принимать нечеткие лингвистические значения. В таблице 1.1 приведены некоторые характеристики объектов из области инженерии и их значения, определяемые нечеткими лингвистическими терминами.

В науке и технике долгое время уделяли внимание объективным знаниям, определяя точные математические модели, а субъективные знания обычно игнорировались. Такие знания человек использует, в том числе, для решения сложных и неформализованных проблем, поэтому их использование было бы весьма полезным. С появлением теории нечетких множеств [20] субъективные

знания удалось формализовать. Благодаря этому, рассуждения с нечеткими лингвистическими терминами стали объектом исследования нечеткой логики, обрели форму математической модели и нашли применение в системах управления и принятия решений [21-30].

Таблица 1.1 – Характеристики, используемые в инженерии, и их нечеткие значения [19]

Характеристика	Используемые значения лингвистических терминов
Пропускная способность	Узкополосная, широкополосная
Размытие	Небольшое, достаточное, большое
Корреляция	Низкая, средняя, высокая, полная
Количество ошибок	Большое, среднее, малое, очень большое, не очень большое, очень маленькое, почти нулевое
Частота	Низкая, высокая, сверхвысокая
Разрешение	Низкое, среднее, высокое
Сэмплирование	Низкочастотное, среднечастотное, высокочастотное, очень высокочастотное
Стабильность сигнала	Устойчивый (слабо затухающий, сильно затухающий, чрезмерно затухающий, критически затухающий), неустойчивый

Особенностью нечеткой логики является использование в рассуждениях лингвистических переменных [31, 32]. Значениями этих переменных являются термины естественного языка, которые определяются нечеткими множествами. Формально лингвистическая переменная определяется набором пяти элементов [31]:

$$X = (\chi, T(\chi), U, G, M),$$

где χ – название переменной; $T(\chi)$ – множество терминов, определяющее совокупность ее лингвистических значений; U – универсальное множество; G – синтаксическое правило, которое порождает термины множества $T(\chi)$; M – семантическое правило, которое каждому лингвистическому значению X ставит в соответствие его смысл $M(X)$, где $M(X)$ обозначает нечеткое подмножество U . Лингвистические переменные входят в состав элементарного нечеткого высказывания, которое имеет следующий вид:

$$X \text{ есть } L \text{ или } X = L$$

где L лингвистический термин из множества $T(\chi)$. Логическое значение, являющееся степенью истинности высказывания, определяется на основании функции принадлежности $\mu: U \rightarrow [0, 1]$. Данная функция определяет степень принадлежности каждого элемента из U нечеткому множеству, представляющему термин L . Таким образом, степень истинности высказывания принимает вещественное значение из диапазона $[0, 1]$. Над нечеткими высказываниями определены правила для выполнения логических операций конъюнкции, дизъюнкции, отрицания и импликации [31, 33].

Использование субъективных знаний с лингвистическими терминами и теории нечеткой логики дало возможность осуществлять построение математических моделей принятия решений, называемых нечеткими системами [34]. В настоящем исследовании используется следующее рабочее определение данного понятия. Нечеткие системы – это системы принятия решений, основанные на базе нечетких продукционных правил и нечетком логическом выводе. Под термином «система» обычно понимается набор взаимодействующих компонентов с четко определенной структурой и организованных как сложное целое, которое можно отличить от «внешней» среды [35]. Система взаимодействует со средой через так называемые входы и выходы. В литературе такие понятия как нечеткая система, нечеткая модель, система, основанная на нечетких правилах, нечеткий контроллер используются взаимозаменяемо в зависимости от типа приложения [36].

Типичная структура нечеткой системы представлена на рисунке 1.1 [37-39]. Система отображает четкие входные значения в четкие выходы, которые могут быть представлены числовыми значениями или словами, например в случае классификации. Система содержит базу правил и три функциональных блока: фаззификатор, нечеткий вывод и дефаззификатор. Фаззификатор отображает четкие входные значения в нечеткие множества, чтобы определить степень истинности высказываний логического вывода. Далее определяется нечеткое множество, сформированное правилом нечеткого вывода на основе входных

значений и базы правил. После этого в блоке дефаззификации полученное в выводе нечеткое множество преобразуется в четкое значение.

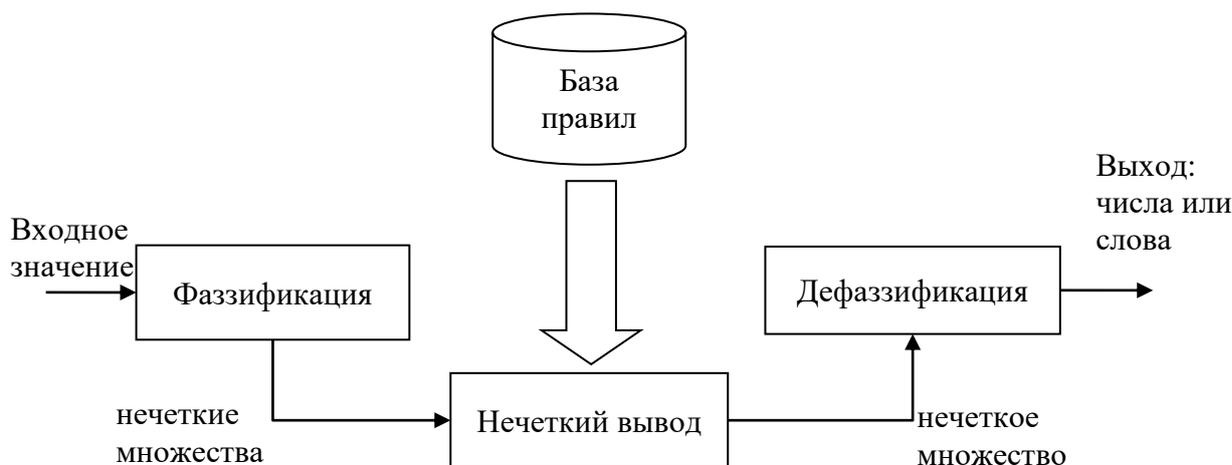


Рисунок 1.1 – Структура нечеткой системы

База нечетких правил может быть предоставлена экспертами или получена с помощью анализа данных. Правила представляют собой нечеткие импликации вида «ЕСЛИ-ТО». Часть «ЕСЛИ» импликации называется антецедентом, а часть «ТО» называется консеквентом. Правила в нечетких системах имеют три различных структуры, отличающиеся консеквентной частью. Рассмотрим данные структуры.

1. Антецедентная и консеквентная части правила содержат нечеткие высказывания и имеют следующий вид [23]:

ЕСЛИ x_1 есть A_1 И x_2 есть A_2 И ... x_n есть A_n ТО y есть B .

2. Антецедентная часть правила содержит нечеткие высказывания, а консеквентная содержит функцию от входных переменных системы, чаще всего используются линейная функция [40, 41]:

ЕСЛИ x_1 есть A_1 И x_2 есть A_2 И ... x_n есть A_n ТО $y = f(\mathbf{x})$,

где в случае линейной функции консеквентная часть определяется выражением

$$y = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n.$$

3. Антецедентная часть правила содержит нечеткие высказывания, а консеквентная четкое высказывание C , принимающее истинностное значение 0 или 1 [42-44]:

ЕСЛИ x_1 есть A_1 И x_2 есть A_2 И ... x_n есть A_n ТО C .

Правила такой структуры в основном применяются в нечетких системах, осуществляющих классификацию, где C соответствует высказыванию, например, «класс = B ».

Практические применения нечетких систем включают системы объяснимого искусственного интеллекта, а также системы принятия решений с проблемными областями, для которых полное математическое описание недоступно, или где использование точных моделей экономически невыгодно или крайне неудобно [35]. Доказано, что нечеткие системы являются универсальными аппроксиматорами, в том смысле, что нечеткие модели могут аппроксимировать любую непрерывную функцию с любой степенью точности на компактной области (замкнутой и ограниченной в конечномерном пространстве) [45, 46]. В то время как строгий математический анализ обеспечивает прочную теоретическую основу для нечеткого моделирования, практическое построение нечеткой модели по-прежнему требует внимания. Добиться универсального способа построения нечеткой системы, способной аппроксимировать любую непрерывную функцию с любой степенью точности, в практическом плане не удалось.

Настоящее исследование посвящено проблеме построения интерпретируемых нечетких систем, осуществляющих классификацию объекта и называемых нечеткими классификаторами. Далее в разделе рассмотрены известные методы построения нечетких классификаторов, приведены основные преимущества и недостатки.

1.2 Нечеткий классификатор типа Min-Max

В [47] предложен нечеткий классификатор, основанный на гипербоксах, получивший название нечеткий классификатор типа Min-Max (Fuzzy Min-Max, FMM) [48]. Гипербокс – это нечеткое отношение, определяющее на пространстве признаков область, геометрически похожую на гиперпараллелепипед. Для задания такой области достаточно указать две точки гипербокса, которые содержат минимальные и максимальные значения координат фигуры (рисунок 1.2). Данные точки совпадают с нижним и верхним углами, лежащими на диагонали.

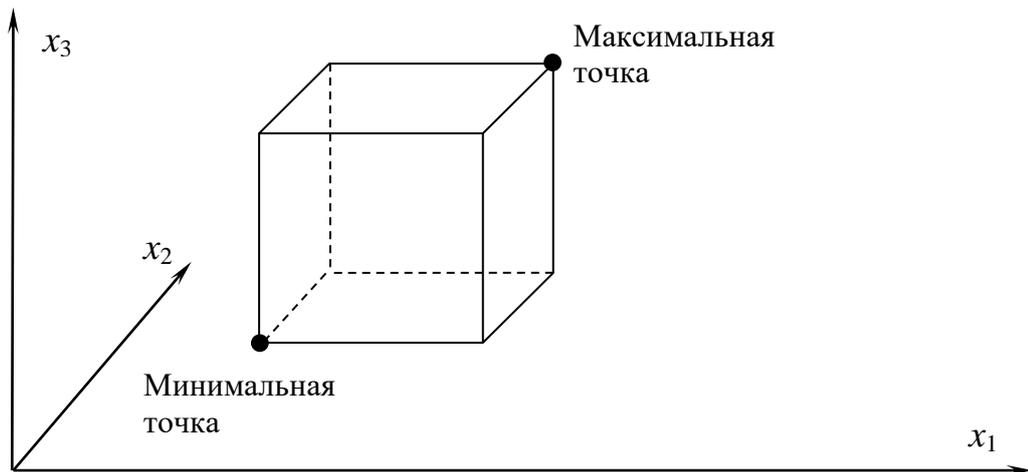


Рисунок 1.2 – Гипербокс в пространстве \mathcal{R}^3

С каждым гипербоксом связан один класс, т.е. предполагается, что экземпляр со значениями признаков, попадающих внутрь гипербокса, является экземпляром данного класса. Функция принадлежности формирует нечеткие границы этого пространства и задается следующим образом:

$$\mu(\mathbf{x}) = \frac{1}{2n} \sum_{i=1}^n \dots \left(\max \left(0, 1 - \max \left(0, \gamma \min \left(1, x_i - w_i \right) \right) \right) + \max \left(0, 1 - \max \left(0, \gamma \min \left(1, v_i - x_i \right) \right) \right) \right),$$

где γ – параметр чувствительности, который регулирует, насколько быстро уменьшается степень принадлежности отношения по мере увеличения расстояния от экземпляра x до гипербокса.

На рисунке 1.3 представлен пример функции принадлежности в двумерном пространстве с $\gamma = 4$. Область, находящаяся в границах минимальной и максимальной точек, имеет степень принадлежности, равную 1, и образует прямоугольное плато. За границами этой области степень принадлежности постепенно уменьшается. При проекции на координаты пространства данных x_1, \dots, x_n образуются функции принадлежности координат, с которыми могут быть связаны лингвистические термины, употребляемые в проблемной области. На рисунке 1.4 показана проекция функции рисунка 1.3 на координату признака x_1 , которая образуют функцию принадлежности лингвистического термина «Среднее».

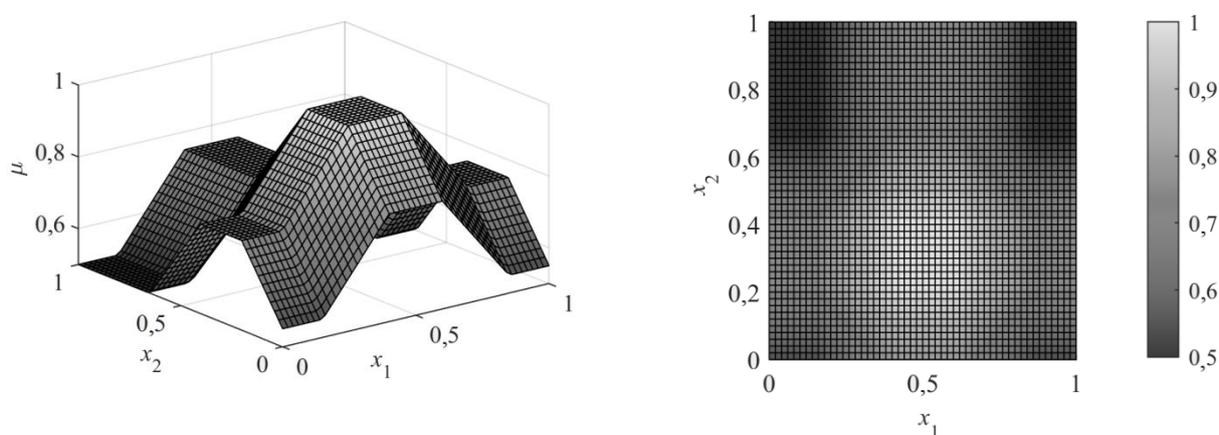


Рисунок 1.3 – Функция принадлежности нечеткого отношения в пространстве \mathfrak{R}^2 , формируемом гипербоксом с минимальной точкой $V = (0,4, 0,2)$ и максимальной точкой $W = (0,6, 0,4)$

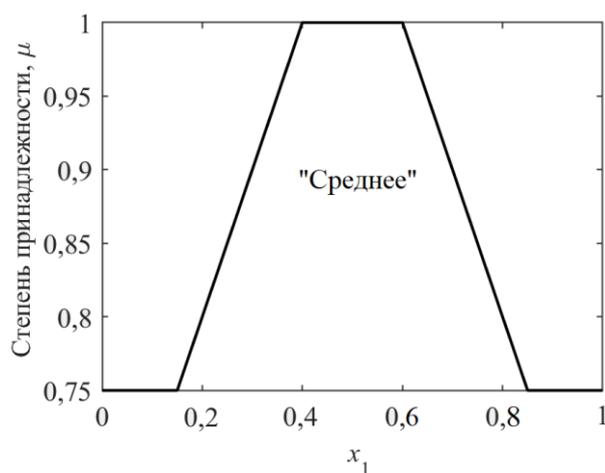


Рисунок 1.4 – Проекция функции принадлежности нечеткого отношения на координату признака x_1 , которая образует функцию принадлежности для лингвистического термина «Среднее»

Также для понимания работы классификатора удобно представлять его в виде трехслойной нейронной сети, показанной на рисунке 1.5. Входной слой имеет n элементов и соответствует признакам классифицируемого объекта. Элементы входного слоя соединены с элементами скрытого слоя, представляющего слой гипербоксов из m элементов. Данное соединение настраивается с помощью алгоритма обучения, которым находятся минимальная V_j и максимальная точка W_j точка каждого гипербокса B_j ($j=1, \dots, m$), а также соответствующий ему класс. Скрытый слой соединен с выходным слоем. Это соединение предполагает вычисления степени принадлежности объекта каждому гипербоксу и вычисления разделяющих функций g_i ($i=1, \dots, c$), как степеней принадлежности классу. Из-за схожести архитектуры нечеткого классификатора типа Min-Max с трехслойной нейронной сетью, они получали также название нечеткие нейронные сети типа Min-Max.

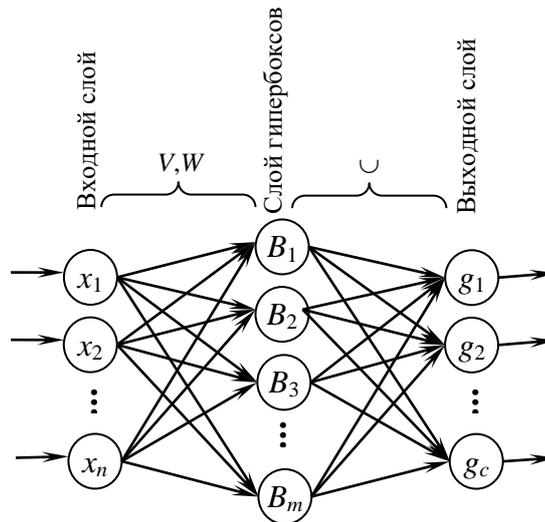


Рисунок 1.5 – Трехслойная нейронная сеть, представляющая нечеткий классификатор типа Min-Max

Построение классификатора фактически сводится к построению гипербоксов и их привязки к определенному классу. Каждый гипербокс определяет нечеткое правило, которое может быть интерпретировано для гипербокса B_j класса ω_k следующим образом:

$$\text{ЕСЛИ } x_1 = A_{1j} \text{ И } x_2 = A_{2j} \text{ И } \dots \text{ И } x_n = A_{nj} \text{ ТО класс} = \omega_k,$$

где A_{ij} – лингвистический термин, функция принадлежности которого образована проекцией гипербокса B_j на координату i . Множество гипербоксов определяют базу правил классификатора.

Алгоритм обучения. Алгоритм обучения нечеткого классификатора, предложенный Симпсоном [47], является эвристическим и относится к алгоритмам, которые адаптируют классификатор в режиме онлайн. *Адаптация классификатора в режиме онлайн* говорит о том, что классификатор имеет возможность дообучиться с приходом нового экземпляра данных без необходимости полного переобучения как со старой, так и с новой информацией (автономная адаптация) [49, 50]. В ходе работы алгоритма на вход последовательно подаются экземпляры обучающих данных. Следует отметить, что значение признаков экземпляров данных масштабируются в диапазон $[0, 1]$.

Таким образом, пространство экземпляров, а также функций принадлежности будет ограничено n -мерным единичным кубом I^n .

Процесс обучения начинается с выбора экземпляра данных и поиска гипербокса такого же класса, который может быть *расширен* при необходимости. Необходимость расширения потребуется, если полученный экземпляр имеет степень принадлежности меньше 1. Если гипербокс не удовлетворяет критериям расширения, то создается новый гипербокс из полученного экземпляра данных с соответствующим классом. Минимальная и максимальная точка в таком случае будут совпадать. Одним из последствий расширения может быть *перекрытие* гипербоксов. Перекрытие не является проблемой, когда происходит между гипербоксами, представляющими один и тот же класс. Перекрытие между гипербоксами, представляющие разные классы, устраняется с помощью процесса *сжатия*. Сжатие устраняет только перекрытие между теми частями гипербоксов, которые имеют полное членство, т.е. степень принадлежности, равную 1. Между неединичными значениями частей каждого из гипербоксов перекрытия допускаются и их устранять не следует. Далее выберется следующий экземпляр, и работа повторяется.

Таким образом, работа алгоритма обучения разделена на три этапа, которые применяются к каждому поступающему экземпляру данных.

1. Этап расширения: определяется ближайший к экземпляру гипербокс того же класса. Если гипербокс после расширения не превысит параметра максимального размера, то он расширяется для покрытия экземпляра. Если превышает, то экземпляр становится новым гипербоксом.

2. Этап проверки на перекрытие: определяется, перекрывает ли расширенный или добавленный гипербокс гипербоксы других классов.

3. Этап сжатия: Если существует перекрытие между гипербоксами разных классов, оно устраняется таким образом, чтобы изменения в гипербоксах были минимальными.

Модификации без процедуры сжатия. В [51] было замечено, что процесс сжатия чрезмерно устраняет части гипербоксов. Исключение этих частей

означает, что вклад в обучение данных, содержащихся в этих областях, сводится к нулю. Если обучение нейронной сети предполагает только один проход по данным, то это необратимая потеря, которая проявляется в ухудшении качества классификации. Поэтому было предложено убрать процесс сжатия, но при этом создавать гипербоксы двух типов: гипербоксы включения, которые содержат экземпляры, принадлежащие одному и тому же классу, и гипербоксы исключения, которые содержат экземпляры, принадлежащие двум или более классам, таким образом, представляя спорные области пространства шаблонов. Гипербокс включения представляет данные, принадлежащие классу, а гипербокс исключения представляет собой область перекрытия между разными классами. Экземпляр данных, попадающий в зону перекрытия, классифицируется как пустой класс, т. е. класс не предлагается. Производительность этого метода снижается, когда объем гипербоксов исключения сравним с объемом гипербоксов включения, поскольку доля выборок данных, которые классифицируются как пустой класс, может стать неприемлемо высокой. С этими двумя типами гипербоксов каждое нечеткое множество классов представляется как объединение гипербоксов включения одного и того же класса за вычетом объединения гипербоксов исключения. Такая модификация получила название классификатор включения/исключения (EFC).

Дальнейшее развитие данной классификатор получил в модификации, названной классификатор типа Min-Max с компенсаторной нейронной архитектурой (FMCN) [52]. Здесь гипербоксы исключения были заменены на компенсаторные нейроны. Такое введение вдохновлено рефлекторной системой человеческого мозга. Компенсаторные нейроны активны только в том случае, если образец данных попадает в перекрывающуюся область двух гипербоксов, представляющих разные классы, т. е. когда существует неоднозначность в отношении определения класса или степени принадлежности экземпляров данных. Вычисление степени принадлежности экземпляра классу определяется здесь уже с учетом значения компенсаторных нейронов. Этот вариант демонстрирует более высокую эффективность в решении проблемы перекрытия классов, чем предыдущие модели [53]. Структура FMCN усложняется за счет

использования компенсаторных нейронов, что увеличивает количество узлов в ее скрытом слое.

Многоуровневый нечеткий классификатор типа Min-Max (MLF), предложенный в [54], также не использует процедуру сжатия. Данный классификатор представляет собой многоуровневую древовидную структуру. Каждый узел дерева представляет собой отдельный классификатор с гипербоксами небольшого размера. Конечный результат формируется путем объединения выходов этих классификаторов. При сравнении [54] данный классификатор показал наименьшую чувствительность к параметру максимального размера гипербокса.

Модификации с процедурой сжатия. Первым значительным расширением оригинального FMM является общая нечеткая нейронная сеть типа Min-Max (GFMM) представленная в [55]. Данная сеть может одновременно обрабатывать помеченные и непомеченные экземпляры данных, комбинируя контролируемое и неконтролируемое обучение в единый алгоритм. Такая особенность позволяет использовать GFMM в трех различных режимах: классификация, кластеризация и гибридный режим (обучающие данные представляют собой смесь помеченных и непомеченных данных). Модифицирована нечеткая функция принадлежности гипербокса. В отличие от функции FMM, здесь степень принадлежности уменьшается с увеличением расстояния от гипербокса. В GFMM предложено адаптивное изменение параметра максимального размера гипербокса на этапе обучения. Но данный подход предполагает неоднократный проход по обучающим данным, чтобы подобрать приемлемое значение. Эмпирически было установлено, что GFMM производит меньше гипербоксов по сравнению с FMM и демонстрирует более низкие показатели ошибочной классификации [56].

Мохаммед и Лим [57] представили модель усовершенствованного FMM (EFMM) для преодоления ряда ограничений FMM. В ней вводятся три эвристики для улучшения этапа обучения FMM, а именно модифицированные процедуры расширения, теста на перекрытие и сжатия. Во-первых, процедура расширения обновляется с использованием нового правила расширения, которое использует

ограничения на размер каждого измерения, а не суммирование всех измерений. Использование нового правила расширения приводит к уменьшению областей перекрытия между гипербоксами, принадлежащим разным классам. Во-вторых, правила проверки перекрытия расширены, чтобы охватить все случаи перекрытия. Классификатор FMM учитывал только четыре варианта перекрытия, в этой работе было продемонстрировано, что существует еще пять, которые раньше никак не учитывались. Наконец, вводится новая процедура сжатия для устранения всех перекрывающихся случаев. EFMM сокращает все перекрывающиеся измерения, а не только минимально перекрывающееся измерение, как в FMM. Эти новые эвристические правила делают EFMM более точным, чем предыдущие варианты. Сложность выделена как основная проблема EFMM из-за использования правила расширения.

В [58] представлен вариант FMM, обозначаемый как K_n FMM, в котором применяется принцип k -ближайших правил к традиционному FMM. FMM рассматривает гипербокс с максимальным нечетким значением функции принадлежности для расширения. K_n FMM имеет k гипербокс с k максимальными значениями нечеткой функции принадлежности. Классический FMM генерирует новый гипербокс, когда гипербокс с наибольшим нечетким значением функции принадлежности не соответствует критериям расширения для добавления текущего шаблона. Однако в K_n FMM гипербокс со следующим максимальным значением нечеткой функции принадлежности проверяется на расширение, если гипербокс с наибольшим значением нечеткой функции принадлежности не соответствует критериям расширяемости. Эта процедура продолжается для k гипербокс, имеющих k наибольших значения нечеткой функции принадлежности. Генерируется новый гипербокс для добавления текущей обучающей выборки только в том случае, если все k гипербокс не удовлетворяют условию расширения.

В [59] предлагается нечеткий классификатор PFMM. Во-первых, в нем изменен критерий для выбора расширяемого гипербокса. Если существует несколько кандидатов на расширение с одинаковыми максимальными значениями

функции принадлежности, данный критерий позволяет выбрать один. Во-вторых, предложен новый набор тестов на перекрытие для рассмотрения всех типов перекрывающихся регионов. В-третьих, представлен новый набор правил сжатия для урегулирования перекрывающихся областей. В-четвертых, предложена новая стратегия гипербокса для уменьшения сложности системы. Эмпирически на 5 наборах данных из области медицины было установлено, что предложенный PFMM более точно классифицирует наборы данных, чем FMM, EFMM и KnFMM. Более того, PFMM создает меньшее количество гипербоксов по сравнению с данными аналогами.

Концептуальное решение для использования новых функций принадлежности в FMM представлено в работе [60]. Здесь функция принадлежности формируется на основании центроида, который определяется «кучностью» обучающих данных, и евклидова расстояния от данного центроида.

Модификация этапа расширения проведена в [61-64]. Было предложено расширять не ближайший гипербокс к образцу данных, а выбрать среди всех гипербоксов класса образца тот, который не превысит максимального размера после расширения и имеет минимальное расстояние. Эксперименты для решения задач кибербезопасности, таких как обнаружение спама, фишинговых сайтов и атак на сетевые соединения показали улучшение точности классификации по сравнению с оригинальным алгоритмом. В [65] для нахождения параметра максимального размера гипербокса предложено использование регрессионной модели, которая на основе метапризнаков данных определяет значение параметра.

Модификации с использованием пакетного режима обучения. Помимо модификаций, направленных на онлайн обучение, были предложены модификации, которые использовали пакетное обучение. Алгоритм пакетного обучения использует все экземпляры обучающих данных. В таком случае классификаторы теряют способность онлайн-адаптации. Время обучения таких алгоритмов обычно больше, но и точность прогнозов становится выше. Азад и Джа [66] предположили, что серия шагов расширения и сжатия приводит к изменению размеров гипербоксов и влияет на производительность

прогностической модели. Поэтому авторы предложили использовать генетические алгоритмы для оптимизации минимальных и максимальных значений гипербоксов, сгенерированных исходной FMM. Такой классификатор получил аббревиатуру FMMGA. В следующей работе [67] они использовали оптимизацию роящихся частиц вместо генетических алгоритмов, и добились лучшей производительности.

В [68] предложен классификатор на основе случайных гипербоксов (Random Hyperboxes RH). Данный классификатор является ансамблем, построенным из отдельных классификаторов на основе гипербоксов. Каждый такой гипербокс обучен на случайных подмножествах выборок и признаков обучающего набора данных. Эксперименты показали преимущества в точности классификации. Однако значительно улучшенная прогностическая эффективность предлагаемого метода RH достигается ценой потери интерпретируемости, что характерно для ансамблевых методов.

В работе [69] предлагается операция оптимизации, которая может устранить проблему перекрытия гипербоксов. Реализуется такая операция с помощью глубоких слоев, где их количество определяется сразу после устранения перекрытия между гипербоксами.

Резюмируя вышеприведенные модификации, сформирована таблица 1.2, где классификаторы разделены по типам изменений.

Оценка эффективности модификаций. Оценка эффективности работы предложенных модификаций в основном проводилась авторами в сравнении с оригинальным FMM. В [51] классификатор EFC на наборе данных Iris из репозитория UCI университета в Ирвине (<https://archive.ics.uci.edu/>) продемонстрировал точность в пределах 97%–100%. Значение точности FMM при этом находилось в пределах 92%–97,33%. Количество гипербоксов для EFC принимало значение в пределах 4–42, для FMM 16–56.

Классификатор FCN в работе [52] проверялся на 4-х реальных наборах данных из UCI и показал лучшие результаты точности по сравнению с FMM и GFMM. Многоуровневый классификатор MLF в [54] показал лучшие результаты

точности на 12 из 14 наборов данных из UCI. В сравнении также принимали участия GFMM, EFC, FMCN.

Таблица 1.2 – Модификации нечеткого классификатора типа Min-Max

Тип модификации	Описание	Аббревиатура
Без процедуры сжатия	Нечеткий классификатор типа Min-Max включения/исключения	EFC [51]
	Нечеткий классификатор типа Min-Max с компенсаторной нейронной архитектурой	FMCN [52]
	Многоуровневый нечеткий классификатор типа Min-Max	MLF[54]
С процедурой сжатия	Нечеткий классификатор типа Min-Max	FMM [47]
	Общая нечеткая нейронная сеть типа Min-Max	GFMM [55]
	Усовершенствованный нечеткий классификатор типа Min-Max	EFMM [57]
	Нечеткий классификатор типа Min-Max k ближайших гипербоксов	KnFMM [58]
	Нечеткий классификатор типа Min-Max с измененным критерием расширения	PFMM [59]
	Нечеткий классификатор типа Min-Max с функцией принадлежности на основе центроида	FMMMF [60]
Пакетное обучение	Нечеткий классификатор типа Min-Max с генетическим алгоритмом	FMMGA [66]
	Нечеткий классификатор типа Min-Max с алгоритмом роящихся частиц	FMMGA [67]
	Ансамблевый классификатор на основе случайных гипербоксов	RH [68]
	Нечеткий классификатор типа Min-Max на основе глубокой нейронной сети	FMMNN [69]

Практически все сравнения проводились на небольшом количестве наборов данных, и результат сравнений отчасти зависел от его выбора. В [68] сравнивались точности FMM, GFMM, EFMM, KnFMM и RH на 20 наборах данных из репозитория UCI. Наилучшего результата достиг классификатор RH. В [70] сравнивались по точности и количеству гипербоксов FMM, GFMM, EFMM, KnFMM на 16 наборах данных из репозитория UCI. Классификатором с минимальным количеством гипербоксов оказался GFMM, а с лучшей точностью EFMM. Количество гипербоксов (правил) в классификаторах достигало до 804 для GFMM и до 5531 для EFMM. Также в данной работе приведены сравнения точностей классификации с такими популярными классификаторами как деревья решений, метод опорных векторов, k -ближайших соседей и наивный байесовский

классификатор. Статистические сравнения показали, что есть различия в эффективности классификаторов Min-Max и популярных алгоритмов машинного обучения на этих 16 наборах экспериментальных данных. Причем средний ранг оказался выше у метода опорных векторов и алгоритма k -ближайших соседей, и ниже у деревьев решений и наивного байесовского классификатора. Данные результаты говорят, что точность классификатора Min-Max вполне конкурента с популярными классификаторами.

Результаты и выводы. Обзор классификатора FMM и его модификаций показал, что данная модель является весьма полезным инструментом классификации с инкрементальной парадигмой обучения, которая требует однопроходной процедуры через экземпляры данных. Данный классификатор обладает такими важными свойствами как адаптация в режиме онлайн, нелинейная граница разделимости между признаковыми пространствами классов, получение мягких и жестких решений классификации, интерпретируемость полученных результатов и быстрое время обучения. Адаптация классификатора обеспечивает изучение новых экземпляров данных без потери информации, извлеченной из предыдущих экземпляров. FMM не подвержен катастрофическому забыванию, также известному как дилемма стабильности-пластичности, в отличие от популярных нейронных сетей. Данный недостаток связан с неспособностью классификатора сохранять предыдущую информацию, полученную на старых обучающих данных, когда новые данные обрабатываются этим классификатором. Свойства нелинейной разделимости позволяет FMM построить произвольную нелинейную границу решения с произвольной степенью точности для разделения экземпляров данных разных классов.

Классический FMM, предложенный Симпсоном, имеет некоторые проблемы, связанные с процессом обучения, которые некоторые исследователи пытались устранить. К таким проблемам относятся: излишнее удаление части гипербоксов в процессе сжатия, увеличение перекрывающихся областей гипербоксов разных классов после растяжения, невозможность обнаружить все случаи перекрытия. Для устранения этих недостатков были предложены

модифицированные варианты классификатора. Как оригинальный, так и модифицированный классификатор достигают точности, сопоставимой с популярными моделями классификации. Несмотря на это остались существенные недостатки, связанные с интерпретируемостью модели.

1. Классификаторы содержат большое количество правил во всех вариантах модификаций FMM, их значение доходит до нескольких тысяч. Это негативно сказывается на интерпретируемости моделей, и анализе моделируемого процесса или явления.

2. Предложенные классификаторы не содержат механизма, позволяющего сократить число признаков. Данное обстоятельство также негативно отражается на интерпретируемости модели.

3. Количество нечетких множеств на признак соответствует числу гипербоксов (правил). Данное количество может достигать нескольких тысяч. В таких обстоятельствах невозможно интерпретировать нечеткое множество в лингвистический термин, а, следовательно, нечеткое правило не может быть представлено на естественном языке.

1.3 Генетические нечеткие системы

Построение нечетких классификаторов на основе анализа данных методами машинного обучения можно рассматривать как проблему оптимизации или поиска [71]. Генетический алгоритм – это хорошо известный и широко используемый метод глобального поиска, позволяющий исследовать большое пространство решений, требующих только вычисления целевых значений [72]. Алгоритм способен находить близкие к оптимальным решения в сложных пространствах поиска. Структура кодирования решения позволяет включать в поиск априорные знания о классификаторе, например, такие как число правил, переменных, параметров функции принадлежности [71]. Генетические нечеткие системы (ГНС) – это нечеткие системы, дополненные процессом обучения,

основанного на эволюционных вычислениях [71]. При построении ГНС исследователи изучают механизмы кодирования и оптимизации компонентов нечетких систем. Кодирование происходит в виде генетических векторов и генетической популяции, а оптимизация в виде адаптации структуры и параметров нечеткой системы [73, 74].

Архитектура классификатора ГНС в рассмотренных работах была впервые предложена в [42]. Нечеткий классификатор состоит из «ЕСЛИ-ТО» правил с нечеткими высказываниями в антецедентной части, метками классов в консеквентной части, определяющими класс образца данных, а также весовым коэффициентом:

ЕСЛИ $x_1 = A_{1j}$ И ... И $x_n = A_{nj}$ ТО *класс* = c_j с весом RW_j ,

где $j = 1, \dots, r$; r – количество правил; $\mathbf{x} = (x_1, \dots, x_n)$ – входные переменные, соответствующие признакам классифицируемого экземпляра \mathbf{x} ; A_{ij} – нечеткий термин (нечеткое множество), характеризующий i -ю переменную в j -ом правиле ($i = 1, \dots, n$); c_j – метка класса в j -м правиле, RW_j – весовой коэффициент правила, принимающий участие в нечетком выводе.

Кодирование классификатора. Центральным аспектом использования генетического алгоритма для автоматического построения нечеткого классификатора является то, что процесс проектирования можно рассматривать как задачу поиска в пространстве моделей, например, в пространстве наборов правил, посредством кодирования модели в хромосоме. С точки зрения оптимизации поиск подходящей нечеткой модели эквивалентен кодированию ее в виде структуры параметров, а затем поиску значений параметров, которые дают оптимальное значение для конкретной целевой функции. Следовательно, первым шагом в разработке ГНС является решение, какие части нечеткого классификатора подлежат оптимизации с помощью генетического алгоритма, и кодирование их в хромосомы.

Каждая переменная x_i связана с нечетким разбиением своей области. Разбиение представляет нечеткие множества, ассоциируемые с лингвистическими

терминами, и может быть рассмотрено как подход дискретизации для непрерывных пространств. Определение нечетких разбиений имеет решающее значение в нечетком моделировании, а степень детализации разбиения играет важную роль для точности классификации [75]. Определить нечеткие множества в разбиении можно с помощью экспертов или с помощью какого-либо алгоритма. На рисунке 1.6 показано равномерное нечеткое разбиение пространства переменной x с помощью функций принадлежности треугольного типа на три множества. Каждое множество характеризует нечеткий термин «Малое», «Среднее», «Большое». В [43, 44] частями классификатора, которые подлежат оптимизации, являются нечеткие термины для каждой переменной правил. Алгоритм оптимизации должен подобрать соответствующие термины из нечеткого разбиения для переменных таким образом, чтобы повысить эффективность классификации. Что касается консеквентной части, то нахождение ее не требуется, поскольку класс правила может быть определен с помощью эвристики [43, 44] на основании нечетких множеств антецедентной части и обучающих данных. Также не требуется нахождения и весового коэффициента правила, он тоже может быть определен с помощью эвристической процедуры [43, 44].

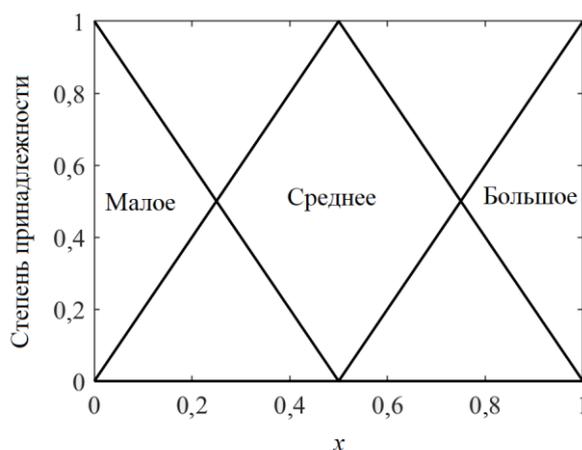


Рисунок 1.6 – Нечеткое разбиение множества значений переменной на три нечетких множества, определяющих термины «Малое», «Среднее», «Большое»

Если численно закодировать функции принадлежности нечеткого разбиения, то антецедентная часть любого правила также может быть

закодирована. Например, если функции принадлежности термина «Малое» присвоен код 1, функции термина «Среднее» присвоен код 2, функции термина «Большое» присвоен код 3, а отсутствие переменной в правиле кодируется значением 0, то следующее правило для трехмерного признакового пространства

$$\text{ЕСЛИ } x_1 = \text{«Малое» И } x_3 = \text{«Большое» ТО класс} = c_j$$

кодируется строкой «103». Таким образом, любое правило может быть закодировано с использованием номеров функций принадлежности. Данная кодировка может быть использована для кодирования хромосомы в генетическом алгоритме. При этом существует два подхода кодирования – Мичиганский и Питтсбургский [76-78]. Эти названия они получили от местонахождения научных коллективов, где впервые были применены.

В Мичиганском подходе хромосома представляет отдельное нечеткое правило, а классификатор определяется множеством правил, сформированных в популяции. Таким образом, по завершении работы генетического алгоритма классификатор собирается из всех элементов популяции. Количество хромосом в популяции определяет максимальное количество правил в классификаторе. В Питтсбургском подходе хромосома кодирует классификатор целиком. На рисунке 1.7 представлена схема такой кодировки. Хромосома разбивается на отрезки генов, представляющие нечеткие правила. Каждый ген в отрезке определяет код функции принадлежности нечеткого разбиения, которая задействована в антецедентной части правила для соответствующей переменной. Для термина A_{ij} код функции принадлежности обозначен как $\text{№}A_{ij}$. Длина хромосомы зависит от максимального количества правил в классификаторе r , размерности пространства данных n и имеет значение $n \cdot r$. После работы алгоритма в найденном решении может оказаться меньше r правил, поскольку если в правиле все гены будут закодированы символом «0», то это будет означать, что правило не используется.

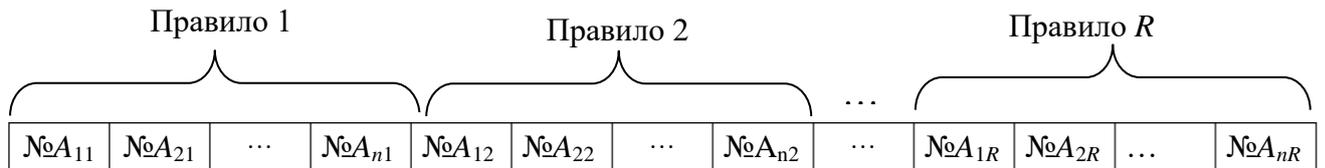


Рисунок 1.7 – Схема кодирования хромосомы в Питтсбургском подходе

Рассмотрим следующую базу правил, на примере которой будет продемонстрирована кодировка для Мичиганского и Питтсбургского подходов.

ЕСЛИ $x_1 = \text{«Малое»}$ И $x_2 = \text{«Среднее»}$ ТО *класс* = 1

ЕСЛИ $x_1 = \text{«Большое»}$ И $x_3 = \text{«Малое»}$ ТО *класс* = 1

ЕСЛИ $x_2 = \text{«Малое»}$ И $x_3 = \text{«Большое»}$ ТО *класс* = 2

При Мичиганском подходе данная база будет представлять популяцию хромосом длины 3. Размер популяции не должен быть меньше количества правил. Если размер популяции имеет значение, например, 5, то база правил может быть закодирована таким вариантом как на рисунке 1.8.

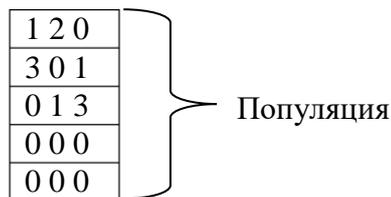


Рисунок 1.8 – Вариант кодирования нечеткого классификатора в Мичиганском подходе

Кодировка классификатора при Питтсбургском подходе будет содержаться в одной хромосоме. Если максимальное количество правил в классификаторе будет иметь значение 5, то хромосома будет иметь $5 \cdot 3 = 15$ генов. Один из вариантов может быть таким:

120 000 301 000 013.

В итоге построение классификатора сводится к нахождению популяции, в случае Мичиганского подхода, или хромосомы, в случае Питтсбургского подхода, имеющих максимальное значение критерия качества классификации. Такая задача относится к классу NP-трудных задач, решение которых находится полным

перебором возможных значений. Есть методы, позволяющие находить близкие к оптимальным значения без полного перебора. К таким методам относится генетический алгоритм. Преимущество Мичиганского подхода заключается в низких вычислительных затратах, поскольку построение классификатора и оценка эффективности его работы для вычисления целевой функции выполняется один раз за поколение. В случае же Питтсбургского подхода за одно поколение необходимо построить и оценить количество классификаторов, соответствующее размеру популяции. Основным недостатком Мичиганского подхода является различие между целевой функцией алгоритма и эффективностью классификатора в целом [79]. Целевой функцией является качество правила, что не всегда соответствует качеству классификатора в целом.

Приведенная выше кодировка приводит к большому количеству вариантов выбора классификатора. Для пространства из пяти признаков с тремя нечеткими множествами на каждый признак, при максимальном количестве правил в классификаторе – пяти – формируются $C_{243}^5 = 6\,774\,333\,588$ вариантов. С ростом числа признаков и нечетких множеств происходит экспоненциальный рост вариантов, что приводит к затруднению поиска оптимума генетическим алгоритмом. Чтобы сократить число вариантов, в [80] предложили предварительно сгенерировать множество правил-кандидатов, обладающих высокой интерпретируемостью (содержащих не более 2-х или 3-х нечетких терминов) и прогностической способностью, и проводить поиск подмножества правил с помощью бинарного генетического алгоритма. Каждый ген в хромосоме алгоритма определяет наличие в базе классификатора правила, соответствующего порядку гена: 0 – правило отсутствует, 1 – правило присутствует. Размер хромосомы определяется количеством предварительно сгенерированных правил.

Многоцелевые генетические нечеткие системы. В [76, 77, 78, 44, 43] построение классификатора генетическим алгоритмом осуществлялось как одноцелевая задача классификации, где целевой функцией выступала ошибка классификации. Поскольку классификатор необходим еще и как инструмент извлечения знаний, необходимо получать базу правил с небольшим числом

правил и используемых в них нечетких терминов. Поэтому задача построения нечеткого классификатора, исходя из требования интерпретируемости, формулируется как многокритериальная задача оптимизации. Помимо минимизации критерия ошибки классификации, должны быть оптимизированы показатели интерпретируемости. В [81] применяются три целевых значения: минимизация ошибки классификации, минимизация количества правил и минимизация среднего количества нечетких терминов в правиле.

Детализация нечеткого разбиения пространства признаков. В [82] отмечено, что одной из проблем построения нечетких классификаторов является уровень детализации нечеткого разбиения пространства признаков, также называемый гранулярностью. Обычно она ассоциируется с количеством терминов (нечетких множеств), на которое разбито пространство. Гранулярность может влиять как на точность, так и на интерпретируемость классификатора, и выбор подходящей гранулярности имеет решающее значение. В [82] авторы указали, что не существует единственной оптимальной гранулярности.

Самая простая стратегия в выборе детализации состоит в том, чтобы зафиксировать одну априорную гранулярность и создать равномерные нечеткие разбиения для всех переменных [43, 44]. На рисунке 1.9 показана равномерная гранулярность пространства признаков с разными уровнями детализации: двух, трех, четырех и пяти нечетких множеств. Нечеткие множества определяются функциями принадлежности треугольного типа. В [83] представлен метод подбора единого уровня равномерной грануляции для каждой переменной (признака). Сначала генерируется набор возможных нечетких правил эвристической процедурой с несколькими уровнями грануляции. Далее выбирается единый уровень гранулярности для каждой переменной в соответствии с частотой используемых уровней и значения критерия доверия правила. Затем данный уровень используется для повторного извлечения множества нечетких правил.

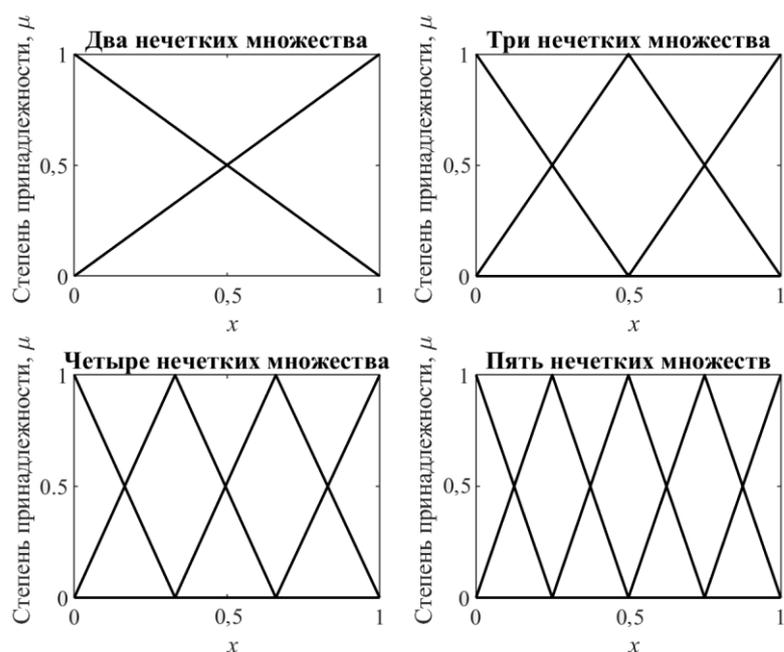


Рисунок 1.9 – Уровни детализации пространства признака нечеткими множествами

В [84] нечеткое разбиение пространства признаков осуществлялось с помощью алгоритма дискретизации САИМ [85]. Данный алгоритм преобразует непрерывные значения атрибутов в набор интервалов и присваивает каждому интервалу дискретное значение. Причем целью является минимизация количества дискретных интервалов при максимальной взаимозависимости между метками классов и дискретными значениями атрибутов, чтобы предотвратить чрезмерную потерю информации во время дискретизации. Полученные интервалы далее переводятся в функции принадлежности нечетких терминов.

Многоцелевые генетические нечеткие системы с настройкой функций принадлежности. Для повышения точности классификации в [86-88] предложен метод, который допускает боковое смещение функций принадлежности нечетких терминов. Такое смещение (рисунок 1.10) позволяет повысить точность классификации при сохранении исходных форм функций принадлежности, причем повышение точности происходит без значительной потери интерпретируемости нечетких терминов. Параметр смещения α представляет собой число в интервале, который позволяет смещать функцию принадлежности влево и вправо так, чтобы не накрывать функции принадлежности других терминов. Для термина рисунка 1.10 с равномерным разбиением из трех нечетких

множеством значений параметра смещения будет находиться в диапазоне $[-0,25, 0,25]$.

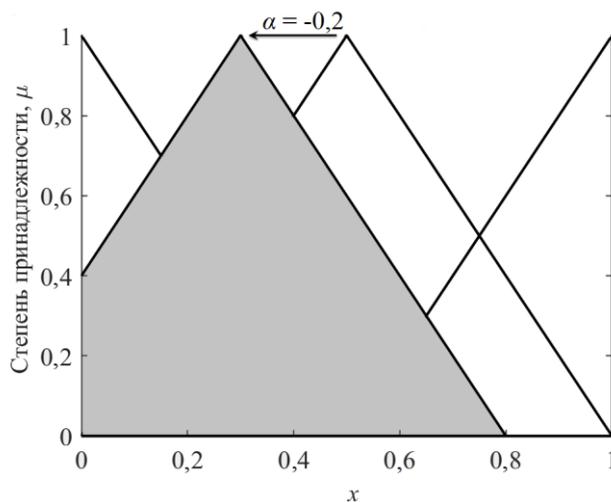


Рисунок 1.10 – Боковое смещение функций принадлежности

Параметр смещения также добавляется в хромосому для поиска оптимального значения смещения. Таким образом, работа генетического алгоритма осуществляется в режиме смешанной оптимизации, т.е. когда в хромосоме присутствуют как дискретные гены, так и непрерывные.

Оценка эффективности генетических нечетких систем. В [86] проведена оценка эффективности нескольких генетических нечетких систем для решения задачи классификации. Сравнивались системы FARC-HD [86, 87], 2SLAVE [89], FH-GBML [90], SGERD [91].

Нечеткие классификаторы FARC-HD [86, 87] разработаны для решения проблемы экспоненциального роста нечетких правил, с которой обычно сталкиваются при использовании полного сочетания лингвистических терминов в антецедентной части. Алгоритм построения позволяет уменьшить масштабируемость и сложность процесса классификации благодаря упрощению структуры и повышению интерпретируемости. Построение состоит из трех этапов. Первый этап – извлечение нечетких правил с помощью дерева поиска, которое позволяет перечислить все возможные варианты правил. Второй этап – предварительный отбор правил-кандидатов. Данный этап позволит уменьшить количество извлеченных правил, путем отбора наиболее значимых для

классификации. Третий этап – получение компактного набора нечетких правил с высокой точностью классификации из множества правил-кандидатов с помощью генетического алгоритма.

Классификаторы 2SLAVE [89] являются модификацией алгоритма SLAVE [92] с процедурой отбора признаков. Алгоритм обучения основан на генетическом алгоритме, в котором каждая хромосома представляет правило (Мичиганский подход), и находит набор нечетких правил. Хромосомы соревнуются в каждом запуске генетического алгоритма, и выбирается лучшее правило за запуск. Глобальное решение формируется лучшими правилами, полученными при многократном запуске алгоритма.

Алгоритм обучения нечеткого классификатора FH-GBML [90] состоит из двух этапов. Первый этап используется для генерации производительных нечетких правил, а второй используется для поиска комбинаций сгенерированных нечетких правил. Этот метод одновременно использует несколько нечетких разделов с различной степенью детализации для извлечения нечетких правил. Для нечетких множеств применяются функции принадлежности треугольного типа.

Классификатор SGERD [91] генерирует заранее определенное количество правил на один класс. На каждой итерации алгоритма хромосомы родители и их соответствующие потомки соревнуются за выбор лучших правил для каждого класса. Этот метод также одновременно использует несколько нечетких разбиений с различной степенью детализации.

Проведен эксперимент на 26 наборах данных из различных проблемных областей по построению и оценке эффективности классификаторов [86]. Статистическое сравнение показателей точности и интерпретируемости выявило значимое преимущество классификатора FARC-HD. В качестве показателя точности использовался процент правильно классифицированных образцов, а показателей интерпретируемости – число правил и среднее число терминов в правиле.

Проведено сравнение классификатора FARC-HD с другими типами интерпретируемых классификаторов, основанных на правилах: деревьями

решений [93]; классификаторами, основанными на ассоциации (CBA) [94]; классификаторами на основе множественных ассоциативных правил (CMAR) [95]; классификаторами на основе предиктивных ассоциативных правил (CPAR) [96]. Статистическое сравнение по критерию точности показало значимое преимущество FARC-HD по сравнению с альтернативными классификаторами. Сравнение интерпретируемости по показателю числа правил также выявило преимущество классификатора FARC-HD.

Результаты и выводы. Обзор ГНС показал эффективность работы реализаций данного классификатора. Высокая точность и интерпретируемость позволяет использовать его не только для классификации, но и для анализа и исследования проблемных областей. Применение генетического алгоритма позволило решить сложную комбинаторную задачу при построении нечеткого классификатора, а именно извлечь из огромного количества нечетких правил подмножество, определяющее базу правил классификатора. Чтобы понизить объем множества вариантов нечетких правил, исследователи предложили предварительно генерировать множество из небольшого количества. Такое множество состоит из правил, обладающих высокой интерпретируемостью (содержат небольшое количество нечетких терминов) и высокой прогностической способностью. Данное решение позволило существенно сократить пространство поиска вариантов классификатора, повысило скорость построения и точность его работы.

Предложены многоцелевые алгоритмы оптимизации, чтобы при построении классификатора ориентироваться не только на точность его работы, но и на критерий интерпретируемости. Количество правил и среднее значение нечетких терминов в правиле применялись в качестве таких критериев. Результат работы алгоритма возвращает набор классификаторов, находящихся на фронте Парето. В данном наборе содержатся недоминируемые решения, и пользователь, исходя из требований своей задачи, выбирает подходящее компромиссное решение. Такой подход позволил существенно повысить уровень интерпретируемости классификаторов.

Следующим решением в повышении эффективности построения стало применение смешанной оптимизации. Наряду с выбором функций принадлежности для antecedентов правил было предложено подбирать их параметры путем горизонтального сдвига без изменения формы самой функций. Это позволило повысить уровень точности классификатора без существенной потери интерпретируемости. Результаты экспериментов показали высокую точность классификации по сравнению с аналогичными методами классификации, основанными на правилах, такими как деревья решений, нечеткие ассоциативные правила.

Несмотря на достигнутые результаты, остались некоторые существенные недостатки в исследованиях ГНС, которые приведены ниже.

1. Классификаторы для данных больших размерностей (30 и более признаков) содержат большое количество правил (доходит до ста и выше). Такое число правил затрудняет интерпретируемость модели.

2. Применение весовых коэффициентов в правилах также понижает уровень интерпретируемости классификатора, поскольку их использование в объяснении результата весьма затруднительно.

3. В рассмотренных работах для построения классификатора применялся только генетический алгоритм, хотя есть аналогичные метаэвристические методы, превышающие эффективность поиска, например некоторые алгоритмы роевого интеллекта.

4. При построении классификатора с помощью смешанной оптимизации применялось только боковое смещение функций принадлежности. Другие варианты изменения параметров функций принадлежности не исследовались, а это потенциально должно повысить точность классификации при увеличении интерпретируемости (например, уменьшении числа правил).

1.4 Метод трехэтапного построения нечетких классификаторов

Процесс построения нечеткого классификатора, согласно методу трехэтапного построения, осуществляется поэтапно, где на каждом этапе проводятся работы по созданию или настройке одной из его частей, что напоминает процесс строительства дома, где кирпичик за кирпичиком, начиная от фундамента дома и заканчивая кровлей, создается здание. Для этого применяются методы и алгоритмы, ориентированные на работу с определенными элементами классификатора [97]. В [97-102] выделяются этапы отбора информативных признаков, генерации структуры и оптимизации параметров. Помимо классификаторов с помощью данного метода осуществляется построение нечетких аппроксиматоров [103-108]. Схема метода представлена на рисунке 1.11.

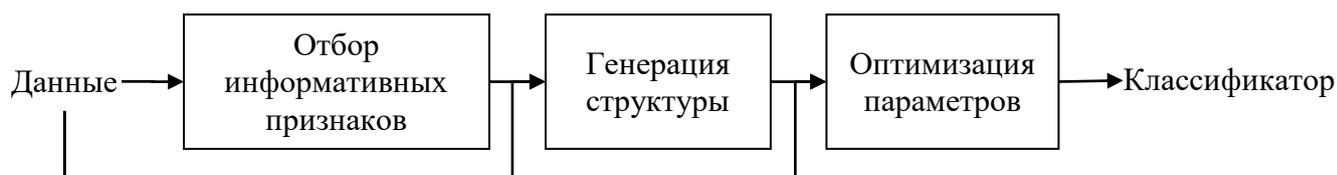


Рисунок 1.11 – Схема метода трехэтапного построения нечеткого классификатора

На этапе отбора признаков проводится анализ данных на предмет выявления подмножества информативных признаков, наиболее полезных для построения классификатора. Выявление данного подмножества помогает снизить признаковое пространство классификатора и тем самым повысить уровень интерпретируемости. Также отсечение неинформативных признаков может обеспечить повышение точности и скорости обучения. Кроме этого, в прикладных задачах исключается необходимость в дополнительных затратах по извлечению значений отсеченных признаков.

На этапе генерации структуры формируются структурные свойства системы. Определяется число правил и вид функций принадлежности, устанавливаются предварительные значения параметров. На этапе оптимизации параметров структурные свойства остаются неизменными, а параметры функций

принадлежности изменяются таким образом, чтобы повысить точность прогноза модели.

Продукционные правила в классификаторах метода трехэтапного построения имеют такую же структуру как в генетических нечетких системах, но с отсутствием весового коэффициента правила, что повышает интерпретируемость модели:

$$\text{ЕСЛИ } x_1 = A_{1j} \text{ И } \dots \text{ И } x_n = A_{nj} \text{ ТО класс} = c_j,$$

где $j = 1, \dots, r$; r – количество правил; $\mathbf{x} = (x_1, \dots, x_n)$ – входные переменные, соответствующие признакам классифицируемого экземпляра \mathbf{x} ; A_{ij} – нечеткий термин (нечеткое множество), характеризующий i -ю переменную в j -ом правиле ($i = 1, \dots, n$); c_j – метка класса в j -м правиле.

Отбор информативных признаков. Методы отбора признаков можно разделить на две группы: фильтры и обертки [100, 109]. Методы фильтрации основаны на обобщающих свойствах данных и не используют построение прогностических моделей. Данные методы могут быть основаны, например, на вычислении корреляции, взаимной информации или энтропии. Методы обертки для отбора признаков используют прогностическую модель, чтобы оценить эффективность множества признаков в ходе процедуры отбора. Здесь широко применяются жадные [102, 110] и бинарные метаэвристические алгоритмы [111-114].

Методы фильтров и оберток имеют свои достоинства и недостатки. Достоинствами методов, основанных на фильтрах, являются их масштабируемость и более высокая скорость выполнения. Общий недостаток заключается в том, что отсутствие взаимодействия с классификатором и игнорирование учета зависимости между признаками снижают эффективность отбора признаков. Преимущество методов оберток заключается в том, что они работают совместно с конкретным алгоритмом классификации и учитывают синергетический эффект от совместного использования отобранных признаков. Недостатками методов оберток являются более высокий риск переобучения и

большие временные затраты по причине необходимости многократного построения прогностических моделей и оценки их эффективности.

В [115] для построения нечетких классификаторов применялся алгоритм отбора признаков на основе взаимной информации [116]. В [117] для медицинских данных использовались алгоритм отбора ReliefF [118, 119] и ранжирование признаков с помощью статистического теста хи-квадрат [120, 121]

Жадная стратегия алгоритма отбора признаков в нечеткой системе Такаги-Сугено [122] была адаптированная для построения нечетких классификаторов в [102, 110]. Здесь после построения классификатора на всем признаковом пространстве итерационно удаляется по одному признаку до тех пор, пока не уменьшится точность классификации. Если точность падает, то работа алгоритма прекращается, и оставшееся множество признаков выбирается для построения классификатора.

Бинарные метаэвристические алгоритмы широко применяются при отборе признаков в построении нечетких классификаторов. Задача отбора признаков сводится к бинарной оптимизации, где элемент вектора решений принимает только два значения, указывающие на присутствие или отсутствие соответствующего признака. При этом помимо эволюционных алгоритмов, способных работать в дискретном пространстве поиска, применяются и алгоритмы роевого интеллекта, пространство поиска которых является непрерывным. Их применение осуществляется благодаря переводу поиска из непрерывного пространства в дискретное пространство с помощью специальных методов бинаризации [123-125]. В [99] представлен алгоритм отбора признаков при построении нечетких классификаторов на основе роевой метаэвристики, вдохновленной поведением ласточек. В [126] модифицирован для отбора признаков алгоритм «кукушкин поиск», вдохновленный гнездовым паразитизмом некоторых видов кукушек. При построении нечетких классификаторов применялся популяционный алгоритм с адаптивной памятью, стратегия поиска которого основывается на прошлом опыте выбора лучших решений [127].

Генерация структуры. Простым и эффективным алгоритмом генерации структуры в трехэтапном методе построения является алгоритм экстремумов класса [128]. Идея алгоритма заключается в разбиении пространства признаков на нечеткие множества, которые ограничены минимальным и максимальным значениями образцов обучающих данных. Алгоритм формирует только одно правило на каждый класс. С точки зрения интерпретируемости это является положительным результатом, но высокая точность прогноза не всегда может быть достигнута.

Также для генерации структуры в трехэтапном методе применяются алгоритмы кластеризации. Кластеры удобно трансформировать в правила нечеткой системы, поскольку границы кластера определяют границы разбиения пространства признаков, которое можно представить в виде разбиения нечетких множеств. Таким образом, каждый кластер формирует правило и нечеткое множество для каждой переменной в правиле. В [101, 129, 130] для генерации структуры применяется субтрактивная кластеризация [131]. Преимущества в данной кластеризации заключается в том, что количество кластеров не задается перед началом запуска алгоритма. Алгоритм сам его определяет в процессе своей работы. Тем не менее, необходимо задавать параметры, которые влияют на количество кластеров. В [129, 130, 132] предложены методы для подбора данных параметров при генерации структуры нечетких классификаторов. В [115] для генерации структуры применялся нечеткий алгоритм кластеризации FCM (Fuzzy C-Means) [133]. Данный алгоритм позволяет соотнести экземпляр данных каждому кластеру с некоторой степенью принадлежности, которая определяется значением из диапазона $[0, 1]$.

Оптимизация параметров. Для оптимизации параметров в методе трехэтапного построения широко применяются метаэвристические алгоритмы оптимизации [134]. С их помощью настраиваются параметры функций принадлежности лингвистических терминов правил таким образом, чтобы был улучшен критерий качества классификации. На данном этапе были применены алгоритмы эволюционного и роевого интеллекта. Благодаря возможности

обходить локальные оптимумы и отсутствию требования к дифференцируемости целевой функции метаэвристики позволяют найти значение параметров, которые значительно увеличивают точность классификации систем, созданных на этапе генерации структуры.

Методы дифференциальной эволюции применялись в [135] для настройки параметров функции принадлежности. Алгоритм «стадо криля», имитирующий перемещение антарктического криля в процессе поиска пищи, использовался для оптимизации параметров нечетких систем в [136, 137]. С помощью метаэвристики на основе гнездового паразитизма некоторых видов кукушек настраивались параметры нечетких классификаторов в [101, 138]. Алгоритмы оптимизации гравитационного поиска, основанного на законах гравитации и взаимодействии масс, сорняков, имитирующего процесс роста сорняков, обезьян, моделирующий поведение обезьян в горной местности для поиска пищи, применялись на этапе оптимизации параметров для построения нечетких классификаторов аутентификации пользователя [139]. Генетический алгоритм также применялся для оптимизации параметров функций принадлежности [140]. В [141] предложено улучшение алгоритмов оптимизации для настройки параметров нечетких систем с помощью различных схем генерации начальной популяции.

Оценка эффективности. В [97] проведена оценка эффективности нечетких классификаторов, полученных с помощью метода трехэтапного построения. Проведено сравнение с генетическими нечеткими системами и классификатором на основе деревьев решений [93]. Проводились эксперименты на 13 наборах данных. На этапах построения метода использовались различные метаэвристические алгоритмы и алгоритмы генерации структуры. Всего получено 32 варианта классификатора. Вариантом метода с лучшей точностью классификации оказался классификатор, генерация структуры которого осуществлялась алгоритмом экстремумов классов, отбор признаков – алгоритмом «серые волки», а оптимизация параметров – алгоритмом «стая птиц». Точность классификации варианта также оказалась лучше по сравнению с генетическими нечеткими системами и деревьями решений.

Результаты и выводы. Метод трехэтапного построения нечеткого классификатора предлагает широкий спектр реализаций, поскольку на этапах метода могут применяться различные алгоритмы оптимизации и кластеризации. Применяя алгоритмы отбора признаков, можно уменьшать количество переменных модели и тем самым упростить ее структуру и увеличить интерпретируемость. Кроме этого, применяя методы кластеризации, можно регулировать количество нечетких правил. Сравнение точности классификаторов данного метода с другими интерпретируемыми моделями показало высокие результаты.

Тем не менее, в данном методе есть существенный недостаток. Генерируемые с помощью метода функции принадлежности нечетких множеств могут приобретать форму, сильно отличающуюся от формы функций принадлежности при равномерном разбиении пространства признака. Данное обстоятельство затрудняет сопоставление функций принадлежности с лингвистическими терминами и тем самым усложняет возможность сформулировать нечеткое правило на естественном языке. Это понижает уровень интерпретируемости модели.

На рисунке 1.12а показано равномерное разбиение пространства на три нечетких множества, определенных функциями принадлежности треугольного типа. С каждым нечетким множеством можно связать лингвистические термины «Малое», «Среднее», «Большое». На рисунке 1.12б показан вариант разбиения пространства признака классификатора, полученного с помощью метода трехэтапного построения. Связать лингвистические термины с полученными функциями принадлежности вызывает трудности.

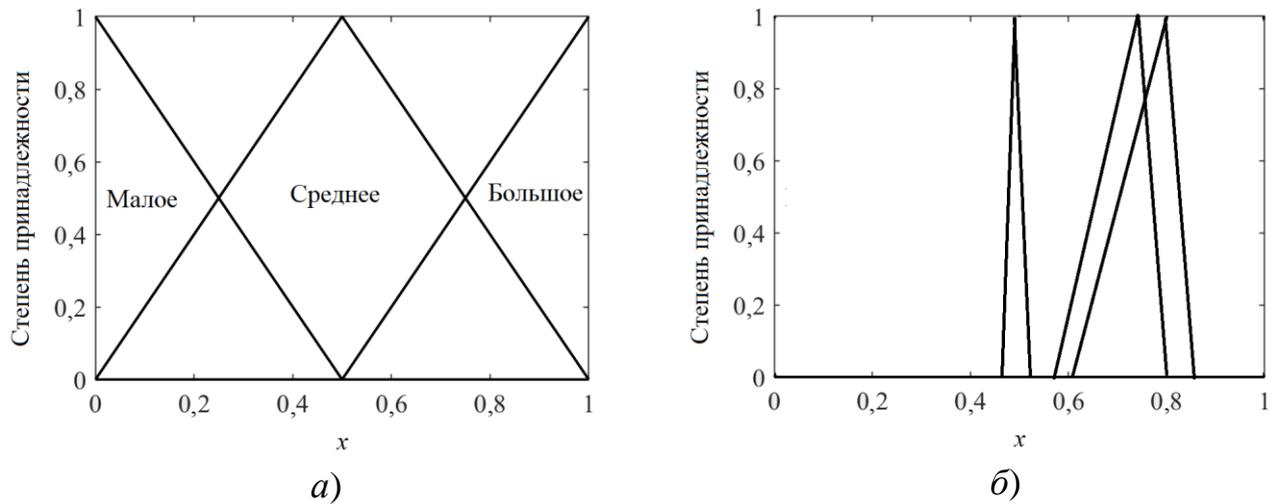


Рисунок 1.12 – Разделение пространства признака на нечеткие множества: *а)* – равномерное разбиение, *б)* – разбиение трехэтапным методом построения [97]

1.5 Критерий интерпретируемости для построения нечетких систем

Одной из задач интеллектуального анализа данных является обнаружение знаний. В классической статье [142] эта задача сформулирована как «нетривиальный процесс выявления действительных, новых, потенциально полезных и в конечном итоге понятных закономерностей в данных». Фигурирующую «понятность» в данном определении связывают с интерпретируемостью моделей машинного обучения. Наряду с точностью, определяющей адекватность модели, интерпретируемость является важным свойством, которое позволяет объяснить моделируемый процесс или явление и тем самым повысить доверие к результатам прогноза. Дать точное определение интерпретируемости и количественно ее измерить весьма затруднительно, на этот вопрос нет однозначного ответа [143-145]. В [144] в качестве альтернативы этому понятию употребляются также термины постижимость (comprehensibility) и понятность (understandability). Тем не менее, проведены попытки не только дать определение, но и измерить ее количественно. В настоящей работе представлены две точки зрения на интерпретируемость, а именно неформальное и формальное представления.

Неформальное представление интерпретируемости. В [144] предложили адаптировать понятие интерпретируемости для моделей искусственного интеллекта: модель интерпретируется в терминах человека, если язык модели можно перевести на язык человека. Также в [146] было высказано утверждение, что интерпретируемость можно оценить путем измерения сложности перевода с описания модели, основанное на нечеткой логике, на объяснение модели, основанное на естественном языке. Исходя из этого, интерпретируемость можно ассоциировать с возможностью объяснить работу модели на естественном языке. Вариант такого перевода для нечетких систем – это наглядное представление базы нечетких правил, которое бы позволило специалисту-эксперту перевести базу на естественный язык.

В [147] представлено изображение базы правил нечеткого классификатора для диагностики болезни Паркинсона по статичным изображениям спиралей и меандров [148]. В [149-151] получены изображения базы правил для классификатора диагностики болезни Паркинсона по динамическим рукописным данным. На рисунке 1.13 показано такое представление для популярной проблемы классификации цветков ириса (ирисы Фишера) [152]. Графическое изображение базы правил представляет собой матрицу, где строки образуют нечеткие правила R_1, \dots, R_4 , а столбцы лингвистические переменные. Каждая лингвистическая переменная ассоциируется с признаком классифицируемого объекта: x_1 – длина чашелистика, x_2 – ширина чашелистика, x_3 – длина лепестка, x_4 – ширина лепестка. Переменная принимает значение из множества терминов $T = \{\text{«Малое»}, \text{«Среднее»}, \text{«Большое»}\}$ согласно нечеткому разбиению, представленному на рисунке 1.6. На этом же рисунке показаны соответствующие этим значениям функции принадлежности. На пересечении правила и переменной изображена функция принадлежности, характеризующая значение лингвистической переменной правила. Последний столбец матрицы содержит значение консеквента правила, т.е. класс образца данных.

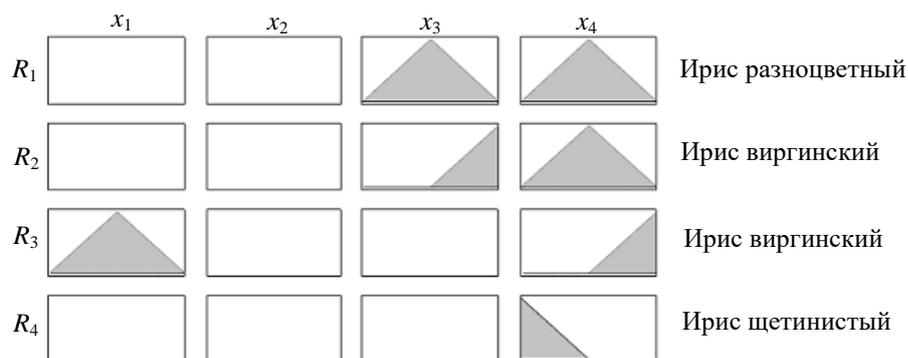


Рисунок 1.13 – Графическое представление базы правил нечеткого классификатора

Ассоциируя представленные функции принадлежности с соответствующим лингвистическим значением из множества терминов T , можно получить правила на естественном языке, которые позволят определить вид ириса по измерениям чашелистика и лепестка. Полученные правила представлены ниже.

Правило 1. ЕСЛИ длина лепестка «средняя» И ширина лепестка «средняя», ТО данный цветок является ирисом разноцветным.

Правило 2. ЕСЛИ длина лепестка «большая» И ширина лепестка «средняя», ТО данный цветок является ирисом виргинским.

Правило 3. ЕСЛИ длина чашелистика «средняя» И ширина лепестка «большая» ТО данный цветок является ирисом виргинским.

Правило 4. ЕСЛИ ширина лепестка «маленькая», ТО данный цветок является ирисом щетинистым.

Формальное представление интерпретируемости. Формальная точка зрения предполагает количественную оценку интерпретируемости. В [153] предложили классификацию оценок интерпретируемости нечетких систем, учитывающую различные их аспекты. С одной стороны оценки можно проводить на уровне базы правил или уровне нечетких разбиений, с другой стороны сама интерпретируемость может быть рассмотрена на уровне сложности, т.е. количественной оценки составных элементов, или на уровне семантики, т.е.

оценки соответствия моделируемой действительности. Типичными показателями интерпретируемости нечетких систем являются: количество правил, количество признаков, среднее количество нечетких терминов в правилах, общее количество нечетких терминов в базе правил [153, 154]. Данные показатели будут применяться в настоящей работе для оценки интерпретируемости нечетких классификаторов.

1.6 Постановка проблемы

В подразделе формализовано понятие классификатора, а также архитектура нечеткого классификатора, используемого в работе. Формализация архитектуры необходима для формирования элементов структуры и параметров при его построении. Определены критерии эффективности нечеткого классификатора и их численные показатели для оценки работы алгоритмов и методов построения.

Каноническое представление классификатора. Выделение объектов, принадлежащих конкретному классу, называется классификацией [18]. Она является ключевым элементом многих инженерных решений. Системы управления и принятия решений используют классификацию для определения отношений между входными и выходными данными. При этом каждый объект \mathbf{x} определяется конечным набором характеристик (признаков) и представляет собой вектор их значений $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Дудой и Хартом [17] введено каноническое описание классификатора как устройства, определяющего класс объекта \mathbf{x} среди множества классов $\{\omega_1, \omega_2, \dots, \omega_M\}$ посредством системы *разделяющих функций* $g_k(\mathbf{x}), k = 1, \dots, M$. Это представление изображено на рисунке 1.14.

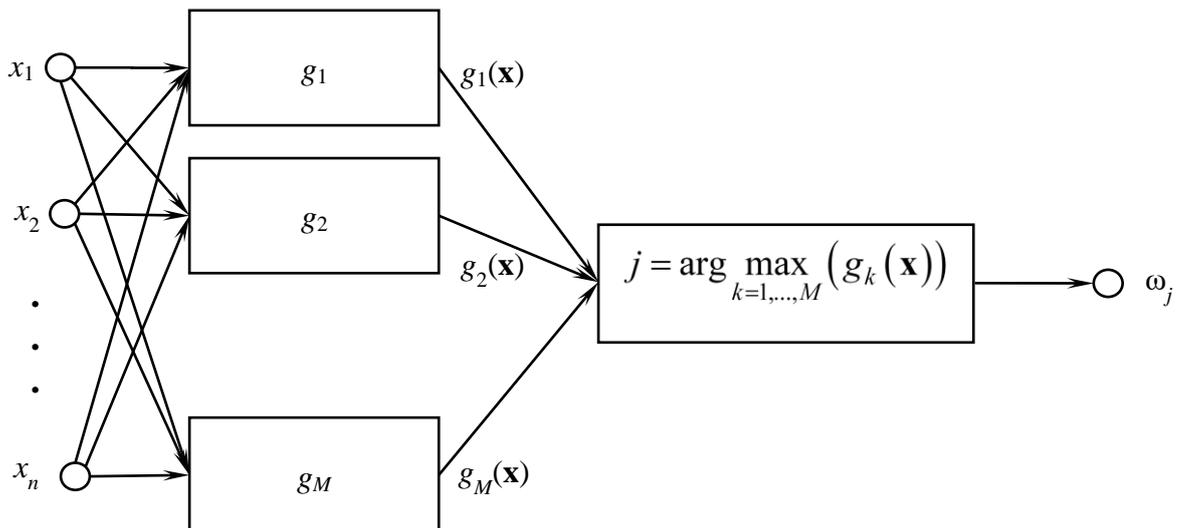


Рисунок 1.14 – Каноническая схема классификатора

Согласно этому представлению, классификатор для объекта \mathbf{x} выбирает класс ω_j если выполняется следующее условие

$$g_j(\mathbf{x}) > g_k(\mathbf{x}) \text{ для всех } k \neq j.$$

То есть с каждым классом связана функция, определяющая действительное значение по признакам объекта. Класс объекта определяется классом разделяющей функции с максимальным значением, которое называют степенью уверенности в выборе класса. Таким образом, классификатор вычисляет M разделяющих функций и определяет класс объекта путем наибольшего значения соответствующей функции. Исходя из канонического описания, классификатор определен тогда, когда определены разделяющие функции для каждого класса.

Архитектура нечеткого классификатора. В настоящем исследовании используется нечеткий классификатор, в посылках продукционных правил которого содержатся значения признаков объекта, выраженные в лингвистической форме, а в заключении содержится метка класса. Данные правила являются вполне понятными для человека и выражают связь между признаками и классом, что является необходимым для объяснения результата прогноза. Впервые правила такого вида для нечетких классификаторов были использованы Х. Ишибучи в [42].

Основой нечеткого классификатора является база нечетких правил следующего вида:

Правило R_j : ЕСЛИ $x_1 = A_{1j}$ И ... И $x_n = A_{nj}$ ТО класс = c_j ,

где $j = 1, \dots, r$; r – количество правил в базе; (x_1, \dots, x_n) – входные переменные; A_{ij} – нечеткий термин, характеризующий i -ю переменную в j -ом правиле; $c_j \in \{“\omega_1”, “\omega_2”, \dots, “\omega_M”\}$ – метка класса в заключении j -го правила.

Для нахождения класса образца данных $\mathbf{x} = (x_1, x_2, \dots, x_n)$ необходимо определить значения разделяющих функций каждого класса. В настоящем исследовании при нахождении значения разделяющей функции k -го класса применяются два вида агрегации правил. В первом случае агрегация правил основана на функции максимум [42]. Значения разделяющих функций определяется следующим образом:

$$g_k(\mathbf{x}) = \max_{c_j = \omega_k} \left(\prod_{i=1}^n \mu_{ij}(x_i) \right), \quad (1.1)$$

где $k = 1, \dots, M$; μ_{ij} функция принадлежности нечеткого множества термина A_{ij} . Т.е. степень уверенности в выборе k -го класса определяется максимальным значением степени принадлежности объекта \mathbf{x} нечеткому множеству, сформированному операцией дизъюнкции нечетких множеств A_{ij} antecedentной части правил. Рассматриваются только те правила, которые имеют метку k -го класса в консеквентной части – “ ω_k ”. В качестве t -нормы в работе используется функция произведения.

Во втором случае агрегация правил определяется суммой [155]. Значения разделяющих функций рассчитывается согласно следующему выражению:

$$g_k(\mathbf{x}) = \sum_{c_j = \omega_k} \prod_{i=1}^n \mu_{ij}(x_i), \quad (1.2)$$

т.е. степень уверенности в выборе k -го класса определяется суммой степеней принадлежности объекта x нечетким множествам, образованным конъюнкцией терминов посылок правил с метками k -го класса в заключении.

Формализация нечетких терминов осуществляется с помощью функций принадлежности, т.е. с каждым термином A_{ij} ассоциируется функция принадлежности $\mu_{ij}(x_i)$, определенная на множестве значений x_i . В работе использовались функции принадлежности гауссова типа, содержащие два параметра – среднее значение s и среднеквадратичное отклонение σ :

$$\mu(x) = e^{-\frac{(x-s)^2}{2\sigma^2}}. \quad (1.3)$$

Для понимания архитектуры нечеткого классификатора, используемого в работе, представим его в виде трехслойной нейронной сети. На рисунке 1.15 показан нечеткий классификатор с двумя входными значениями $x = (x_1, x_2)$ и двумя выходными значениями, g_1 и g_2 , определяющими степень уверенности в выборе класса 1 и класса 2 соответственно. На первом слое вычисляются степени принадлежности входных значений нечетким множествам, которые определяют нечеткие термины правил A_{ij} . Пространство входных значений разбито на три нечетких множества, которые определяются функциями принадлежности. На втором слое (слой правил) для каждого правила определяется степень принадлежности входного объекта x нечеткому множеству, образованному конъюнкцией нечетких терминов посылки правила. Для этого используется t -норма, и в настоящей работе она представлена функцией произведения. База правил классификатора на рисунке имеет четыре правила, поэтому на слое четыре узла. Первые три правила имеют метку класса 1 в консеквентной части, а четвертое метку класса 2. На последнем слое (слой агрегации правил) вычисляются степени уверенности в выборе класса на основе полученных значений t -норм правил предыдущего слоя и выбранной функции агрегации (выражения (1.1) и (1.2)).

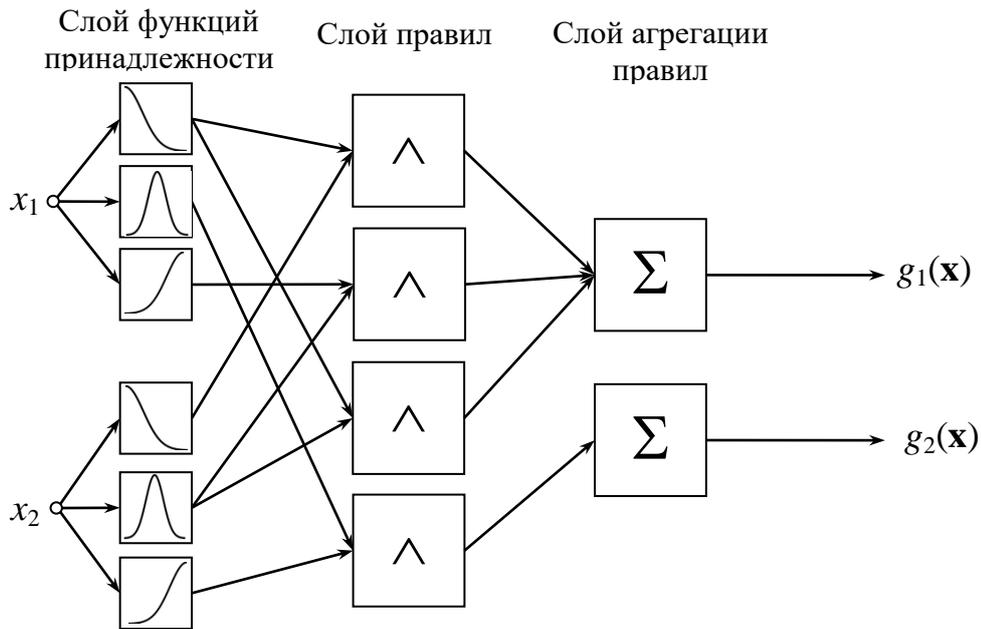


Рисунок 1.15 – Представление нечеткого классификатора в виде нейронной сети

Исходные данные. Для построения классификаторов предоставлена размеченная выборка образцов данных, называемая исходными данными или таблицей наблюдений, состоящая из значений признаков образцов и их метки класса:

$$Data = \{(\mathbf{x}_p, l_p) \mid p = 1, 2, \dots, m\}, \quad (1.4)$$

где m – количество образцов исходных данных, $\mathbf{x}_p = (x_{1p}, x_{2p}, \dots, x_{np})$ вектор входных данных (признаков) p -го образца в n -мерном пространстве, где x_{ip} значение i -й переменной (признака) p -го образца; l_p – метка класса p -го образца, $l_p \in \{\omega_1, \omega_2, \dots, \omega_M\}$.

Построение классификатора. Целью построения нечеткого классификатора является достижение высокой точности и интерпретируемости полученных моделей. Таким образом, проблема построения сводится к проблеме многокритериальной оптимизации, которую можно представить следующим образом:

$$\begin{cases} \max_{\mathbf{x}} (Fun_1^1(\mathbf{x}), Fun_2^1(\mathbf{x}), \dots, Fun_1^2(\mathbf{x}), Fun_2^2(\mathbf{x}), \dots) \\ \mathbf{x} \in FC \end{cases},$$

где $Fun_1^1(\mathbf{x}), Fun_2^1(\mathbf{x}), \dots$ – показатели точности классификатора \mathbf{x} ; $Fun_1^2(\mathbf{x}), Fun_2^2(\mathbf{x}), \dots$ – показатели интерпретируемости классификатора \mathbf{x} ; FC – возможное множество классификаторов. Формула приведена в предположении, что большее значение показателей является лучшим.

В подразделе 2.7 проблема оптимизации для построения нечеткого классификатора будет сформулирована содержательно для задачи смешанной многокритериальной оптимизации с учетом конкретных показателей эффективности и формой кодирования классификатора вектором решений.

Критерии эффективности. Для оценки эффективности построенных классификаторов и сравнения с аналогами в работе учитываются критерии точности и интерпретируемости. Для оценки точности применяется процент правильно классифицированных образцов выборки данных, не используемых для построения классификатора [156]:

$$acc = \frac{n_{true}}{n_{all}} \cdot 100\% ,$$

где n_{true} – число правильно классифицированных образцов тестовых данных, n_{all} – число образцов тестовых данных. Для оценки интерпретируемости нечетких классификаторов используется несколько показателей [153]: количество правил $\#R$, количество признаков (входных переменных) $\#F$, общее количество нечетких терминов в базе правил $\#T$, среднее количество нечетких терминов в антецедентной части (посылке) правила $\#D$.

Вычислительные эксперименты по построению и оценке эффективности нечетких классификаторов в настоящей работе проводятся согласно методу десятикратной перекрестной проверки [156]. Случайным образом набор данных разбивается на десять частей, каждая из которых содержит 10% образцов набора данных. Девять частей используется в качестве исходных данных для построения

и одна для тестирования. Таким образом, получаем десять пар исходных и тестовых данных. Для каждой пары выполняется три построения классификатора. Показатели критериев эффективности определяются средним значением по 30 построениям.

1.7 Краткие выводы по разделу

В разделе представлен аналитический обзор существующих методов, подходов и алгоритмов анализа данных для построения нечетких классификаторов. Рассмотрено важное свойство прогностических моделей – интерпретируемость. Определены критерии интерпретируемости. Формализована проблема построения интерпретируемых нечетких классификаторов. Ниже приведены основные выводы по разделу.

1. Рассмотрено понятие нечеткой системы как системы принятия решений, основанной на базе нечетких продукционных правил и нечетком логическом выводе. Представлена типичная структура нечетких систем, состоящая из базы продукционных правил и функциональных блоков: фаззификации, нечеткого вывода и дефаззификации. Рассмотрены три типичных структуры продукционных правил, используемых в нечетких системах. Представлены известные методы построения нечетких классификаторов, приведены основные преимущества и недостатки. Нечеткие классификаторы типа Min-Max используют инкрементную парадигму обучения, что позволяет адаптировать классификатор к вновь пришедшим данным без повторного обучения. Интерпретируемость данных классификаторов весьма затруднительна, поскольку они содержат большое количество правил, которое доходит до тысяч и приводит к невозможности интерпретировать нечеткие множества в лингвистические термины. Кроме этого, отсутствует механизм отбора признаков. В генетических нечетких системах формируются нечеткие множества, близкие к равномерному разделению, что дает возможность их интерпретации в лингвистические термины. Большое количество

правил и их весовые коэффициенты в данных классификаторах усложняют интерпретируемость. В методе трехэтапного построения формируются классификаторы с небольшим количеством правил, и используется отбор признаков. В то же время нечеткие множества в правилах по своей форме могут сильно отличаться от формы равномерного разбиения. Это затрудняет их интерпретацию в лингвистические термины и, соответственно, представление правил на естественном языке.

2. Рассмотрено важное свойство прогностических моделей – интерпретируемость. Интерпретируемость позволяет объяснить моделируемый процесс или явление и тем самым повысить доверие к результатам прогноза. Описано неформальное и формальное представление интерпретируемости. В неформальном представлении интерпретируемость ассоциируется с возможностью объяснить работу модели человеку на естественном языке. Для нечетких систем наглядное представление базы нечетких правил в графическом виде позволяет специалисту-эксперту перевести базу правил на естественный язык и понять, почему модель приняла то или иное решение. Формальное представление интерпретируемости характеризуется введением численных показателей. Для нечетких систем рассмотрены численные показатели, оценивающие интерпретируемость моделей.

3. Представлена в формализованном виде проблема построения интерпретируемого нечеткого классификатора. Рассмотрено каноническое представление классификатора в виде системы разделяющих функций. Представлена архитектура нечеткого классификатора, исследуемого в работе, в посылках продукционных правил которого содержатся значения признаков объекта, выраженные в лингвистической форме, а в заключение приведена метка класса. Описана процедура вывода, с помощью которой на основании базы нечетких правил и значений признаков объекта осуществляется его классификация. Определены показатели критериев точности и интерпретируемости нечетких классификаторов. В качестве показателя точности в работе используется процент правильно классифицированных образцов, а

используемые показатели интерпретируемости определяют количество правил, количество признаков, общее количество нечетких терминов в базе правил и среднее количество нечетких терминов в антецедентной части правила. Описана процедура десятикратной перекрестной проверки, согласно которой будут проводиться вычислительные эксперименты для эмпирической оценки эффективности нечетких классификаторов.

2 Построение интерпретируемых нечетких классификаторов, основанных на правилах

В разделе представлены новые научные результаты, полученные автором в рамках диссертационного исследования. Результаты направлены на решение проблемы построения интерпретируемых моделей машинного обучения, позволяющих объяснить человеку результаты своего прогноза. Такими моделями выступают нечеткие классификаторы. Предложена методология построения нечетких классификаторов, определяющая совокупность и взаимодействие методов анализа данных. Разработан алгоритм смешанной многокритериальной оптимизации для построения классификатора с соблюдением компромисса «точность–интерпретируемость». Разработан метод разделения данных на обучающую и валидационную выборки для сохранения их близости к исходным данным и минимизации потери информации при разделении. Разработан алгоритм дискретной оптимизации, применяемый в реализации метода разделения данных. Адаптирован алгоритм градиентного спуска, позволяющий настраивать параметры функций принадлежности нечеткого классификатора. В итоге предложена методика построения нечетких классификаторов, основанная на методологии построения и использующая разработанные алгоритмы и метод разделения. В разделе 3 приведены вычислительные эксперименты и оценена эффективность разработанных алгоритмов оптимизации, метода разделения данных и методики построения нечетких классификаторов.

2.1 Методология построения интерпретируемых нечетких классификаторов, основанных на правилах

Аналитический обзор методов построения нечетких классификаторов, проведенный в разделе 1, выявил недостатки, связанные с низким уровнем

интерпретируемости полученных моделей. В настоящем подразделе предложена методология построения нечетких классификаторов, основанных на правилах. Методология определяет систему методов анализа данных, а также их взаимодействие для построения интерпретируемых нечетких классификаторов, и учитывает выявленные недостатки.

Предложенная методология основана на следующих концептуальных положениях.

1. *Построение классификаторов осуществляется многокритериальными методами оптимизации.* Помимо точности важным критерием классификатора является интерпретируемость, что необходимо учитывать при построении. Поскольку построение нечетких классификаторов на основе данных методами машинного обучения можно рассматривать как проблему оптимизации или поиска [71], то в качестве инструмента построения выступают многокритериальные методы оптимизации, в частности метаэвристические алгоритмы.

2. *При построении применяется смешанная оптимизация.* Построение структуры классификатора с помощью методов оптимизации должно осуществляться одновременно с оптимизацией параметров. Построение структуры отдельно от оптимизации параметров, как в методе трехэтапного построения [97], не позволяет получить множество недоминируемых по критериям точности и интерпретируемости классификаторов. Это достижимо при применении методов смешанной оптимизации, позволяющей работать с переменными дискретного и непрерывного типа. Дискретными переменными можно закодировать структуру классификатора, а непрерывными переменными значения параметров функций принадлежности. Кодирование классификатора вектором решений смешанной оптимизации показано в подразделе 2.7.

3. *Разделение исходных данных на обучающую и валидационную выборки с минимизацией потерь характеристических свойств.* Многокритериальные методы оптимизации позволяют получить множество недоминируемых по критериям точности и интерпретируемости классификаторов. Для выбора модели

из этого множества необходимо оценить эффективность классификации и выбрать наиболее приемлемую. Чтобы исключить переобучение и выбрать наиболее эффективную модель с точки зрения обобщающей прогностической способности, необходимо проводить оценку на данных, не участвующих в построении. Таким образом, исходные данные должны быть разделены на обучающую и валидационную выборки. Случайное разделение может привести к изменению характеристических свойств данных в выборках и тем самым понизить обобщающую прогностическую способность, особенно для данных с малым количеством образцов. Поэтому при разделении необходимо уменьшить потерю информации и сохранить близость характеристических свойств выборок к исходным данным.

В генетических нечетких системах [83, 84, 88], построенных с помощью многокритериальных эволюционных методов, разделение данных не проводится. Построение осуществляется с помощью всех исходных данных и выбирается модель с лучшей точностью классификации на исходных данных. Такое решение может привести к выбору переобученной модели.

4. *Применение методов оптимизации, основанных на производных, для дообучения классификаторов.* Современные методы построения нечетких классификаторов отдают предпочтение метаэвристическим алгоритмам, поскольку они способны осуществлять поиск глобального оптимума, в отличие от детерминированных методов. Детерминированные методы, основанные на производных, осуществляют поиск локального оптимума с более высокой точностью по сравнению с метаэвристическими. Совместное применение методов при построении нечетких классификаторов позволит увеличить точность моделей.

5. *Функции принадлежности нечетких множеств интерпретируются лингвистическими терминами.* Форма и положение функций принадлежности построенных классификаторов должны обеспечивать их интерпретацию в лингвистические термины. Для этого необходимо соблюдение следующих условий: 1) нечеткие множества должны быть нормализованы; 2) расположение ядер нечетких множеств должно быть близко к равномерному покрытию области

значений признаков; 3) области значений признаков не должны содержать участки с низкими степенями принадлежности (близкими или равными 0). В подразделе 2.7 представлена методика построения нечетких классификаторов, где соблюдаются данные условия при построении классификаторов смешанной многокритериальной оптимизацией и алгоритмом градиентного спуска.

На рисунке 2.1 представлена контекстная диаграмма IDEF0 верхнего уровня методологии построения нечеткого классификатора. Исходные данные используются для построения нечетких классификаторов. Процесс построения осуществляется программными средствами и экспертом под управлением методов обработки и анализа данных. Используемые методы приведены ниже.

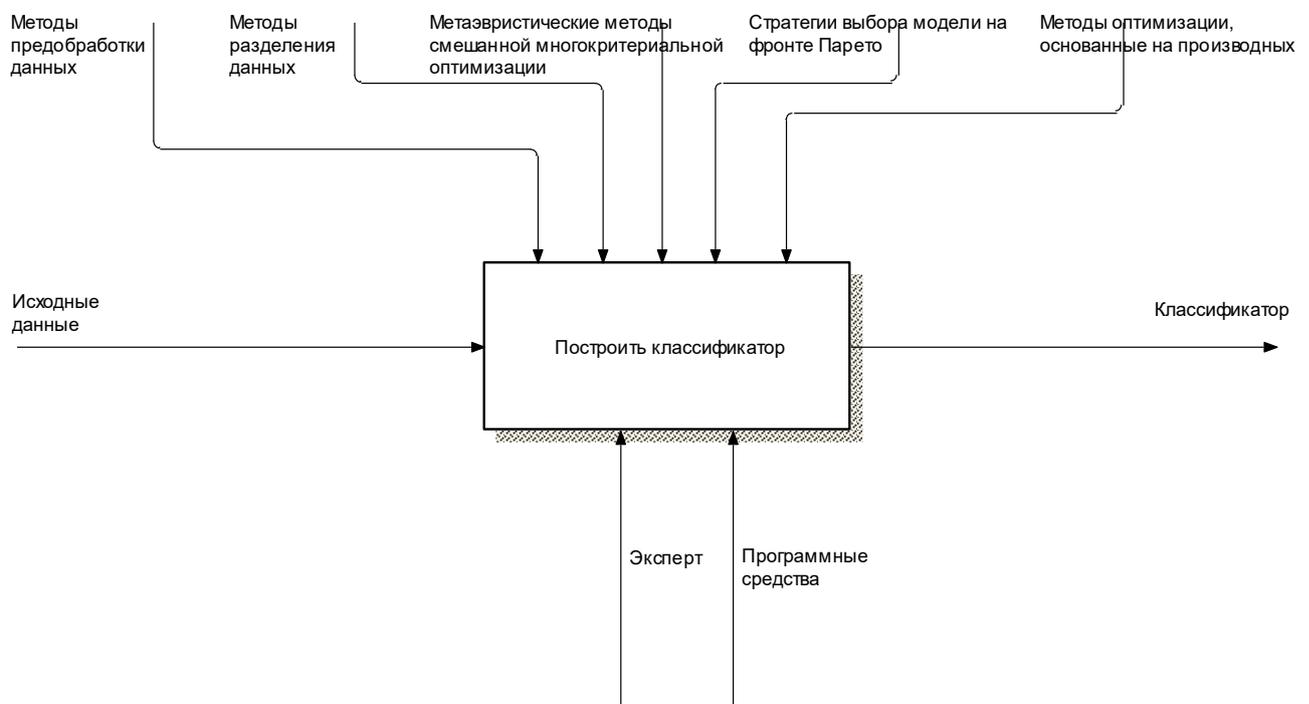


Рисунок 2.1 – Контекстная диаграмма IDEF0 верхнего уровня методологии построения интерпретируемых нечетких классификаторов

Методы предобработки данных преобразовывают исходные данные к виду, позволяющему улучшить показатели качества построенных моделей. Исходные данные часто содержат дефекты, которые усложняют их анализ и снижают качество работы алгоритмов построения. Типичными методами предварительной обработки данных являются нормализация, отбор информативных признаков, очистка данных от выбросов, устранение дисбаланса классов [157-159].

Методы разделения данных формируют из исходных данных обучающую и валидационную выборки. Современные методы машинного обучения разделяют данные при построении моделей случайным образом [156], без учета сохранения характеристическим свойств, или осуществляют построение без разделения [83, 84, 88], что чревато проблемой переобучения. В настоящем исследовании предложен метод разделения (подраздел 2.6), который позволяет получить выборки близкие по характеристикам к исходным данным.

Метаэвристические методы смешанной многокритериальной оптимизации позволяют решать оптимизационные задачи с несколькими целевыми значениями в пространстве с переменными дискретного и непрерывного типа. Решение задач такого типа сосредоточено на использовании эволюционных методов и, в частности, генетического алгоритма. В настоящем исследовании разработан новый алгоритм, позволяющий решать такие задачи. Он основан на гибридизации методов эволюционного и роевого интеллекта (подраздел 2.3).

Стратегии выбора модели на фронте Парето определяют выбор классификатора из множества недоминируемых по критериям точности и интерпретируемости классификаторов, построенных метаэвристическими методами многокритериальной оптимизации. Данный выбор осуществляется экспертом, который основывается на требованиях к полученному результату. В то же время здесь могут быть использованы и стандартные методы выбора решений многокритериальных задач оптимизации, такие как метод идеальной точки, метод последовательных уступок и ранжирование критериев [160].

Методы оптимизации, основанные на производных, позволяют решать оптимизационные задачи и улучшить решения, найденные с помощью метаэвристических методов. Применение данных методов для настройки параметров нечетких классификаторов, построенных с помощью метаэвристических методов, позволит улучшить прогностическую способность моделей. Типичными методами оптимизации, основанными на производных, являются метод Ньютона, градиентного спуска, Левенберга–Марквардта, сопряженных градиентов [161, 162]. В настоящем исследовании для настройки

параметров нечетких классификаторов адаптирован алгоритм градиентного спуска и представлен в подразделе 2.5.

На рисунке 2.2 представлена диаграмма декомпозиции построения классификатора, указаны процессы, в которых участвуют приведенные выше методы и их взаимодействие. Сначала исходные данные подвергаются преобразованию с помощью методов предобработки данных, устраняющих дефекты, влияющие на качество анализа. Преобразование осуществляется программными средствами. Затем данные подвергаются разделению с целью получения обучающей и валидационной выборок. Это проводится методами разделения и осуществляется программными средствами.

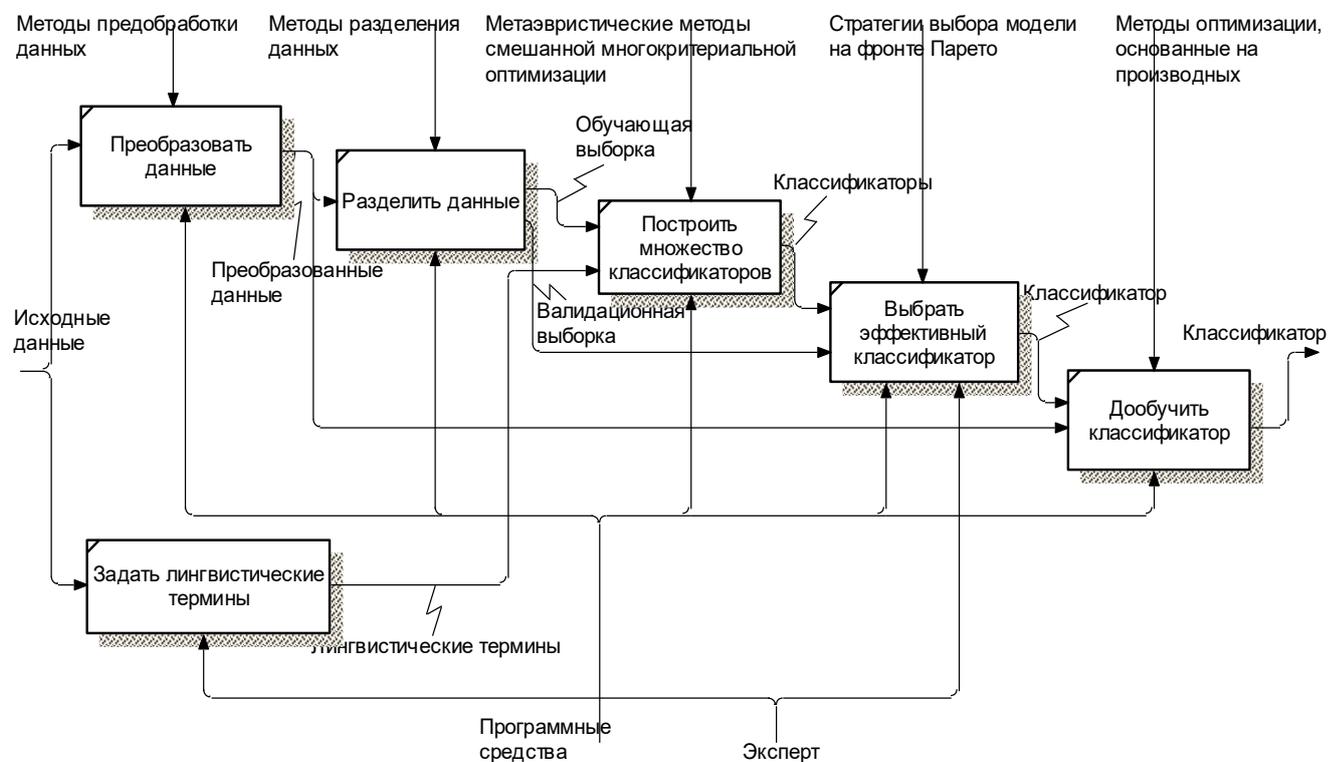


Рисунок 2.2 – Диаграмма IDEF0 методологии построения нечеткого классификатора

Под управлением метаэвристических методов смешанной многокритериальной оптимизации программные средства осуществляют построение множества недоминируемых по критериям точности и интерпретируемости нечетких классификаторов, используя обучающие данные. Предварительно эксперт должен определить лингвистические термины, которыми оцениваются значения признаков данных, и их количество для каждого признака

объекта. Количество является фиксированным и не изменяется в процессе построения. Термины будут употребляться в правилах классификаторов. Далее из полученного множества выбирается наиболее эффективный классификатор. Данную процедуру может осуществлять эксперт или программные средства на основе стратегий выбора модели на фронте Парето и с использованием валидационной выборки. После этого выбранная модель дообучается с помощью методов, основанных на производных, путем настройки параметров функций принадлежности. Выполняют данный процесс программные средства.

На основе предложенной методологии разработана методика построения нечетких классификаторов, основанных на правилах, которая представлена в подразделе 2.7. В методике используются методы и алгоритмы, разработанные в настоящем исследовании. Для разделения данных применяется оригинальный метод на основе бинарной оптимизации (подраздел 2.6), которая осуществляется авторским дискретным алгоритмом (подраздел 2.4). Построение множества классификаторов проводится авторским метаэвристическим алгоритмом смешанной многокритериальной оптимизации (подраздел 2.3). Дообучение классификатора проводится с помощью адаптированного алгоритма градиентного спуска, также разработанного автором в рамках настоящего исследования (подраздел 2.5).

2.2 Задача смешанной многокритериальной оптимизации

В подразделе рассматривается многокритериальная задача оптимизации для поиска в смешанном пространстве, т.е. пространстве, имеющем дискретные и непрерывные переменные. Многие прикладные проблемы сводятся к решению данной задачи. Приведена формальная постановка задачи, а также инструменты ее решения, среди которых наиболее эффективными являются метаэвристические алгоритмы. В подразделе 2.3 представлен новый метаэвристический алгоритм, решающий данную задачу и основанный на гибридизации генетического

алгоритма и метаэвристики «кукушкин поиск». В подразделе 2.7 показано, что к многокритериальной смешанной оптимизации сводится задача построения нечетких классификаторов, учитывающая в своем построении компромисс между точностью и интерпретируемостью полученных моделей. Предложенная в подразделе 2.7 методика построения нечетких классификаторов включает в себя применение данного алгоритма.

Применение методов оптимизации встречается практически во всех сферах деятельности человека, где целью является максимально увеличить или уменьшить какие-либо критерии эффективности. Например, в науке и технике алгоритмы оптимизации применяются при построении прогностических моделей на основе нейронных сетей [163, 164] и нечетких систем [103], где критерием эффективности выступает точность прогнозируемых результатов. Минимизация конфликтных ситуаций являются целью для прокладывания маршрутов самолетов [165]. В экономике задача определения портфеля опционов связана с получением максимальной прибыли [166, 167]. В медицине задача выбора показателей здоровья связана с минимизацией ошибки выставления диагноза на основе этих показателей [168, 169].

Реальные приложения требуют одновременного достижения нескольких целей, причем цели обычно конфликтуют между собой. Это приводит к задаче нахождения баланса между целями. Такая проблема называется задачей многокритериальной оптимизации. Результатом решения этой задачи является множество компромиссных решений, называемых оптимальными по Парето. Суть решений заключается в том, что если одно из них превосходит другое по какому-либо критерию, то обязательно будет существовать критерий, по которому решение будет уступать. Другими словами, любое улучшение при выборе решения по одному критерию должно привести к ухудшению, по крайней мере, одного, другого критерия. Если множество оптимальных по Парето решений получено, то подходящее решение из этого набора обычно выбирается на основе дополнительного критерия. Если значения всех переменных задачи многокритериальной оптимизации принадлежат непрерывному множеству, то эта

задача называется задачей непрерывной многокритериальной оптимизации. Если же значения переменных принадлежат подмножеству целых чисел, то задача называется задачей целочисленной многокритериальной оптимизации. Когда присутствуют как целочисленные, так и вещественные значения, то это задача смешанной многокритериальной оптимизации [170]. Последний случай возникает во многих реальных приложениях. Смешанные переменные определенно увеличивают сложность пространства поиска и усложняют решение задач оптимизации [171]. Следовательно, необходимо разработать эффективные алгоритмы решения смешанной многокритериальной оптимизации.

Метаэвристические алгоритмы успешно применялись для решения задач непрерывной и дискретной оптимизации [172, 173]. В отличие от традиционных методов данные алгоритмы не подвергнуты «застреванию» в локальных оптимумах, в меньшей степени зависят от исходных отправных точек и не ограничены видом целевой функции. Несмотря на высокую актуальность, метаэвристическим алгоритмам смешанной многокритериальной оптимизации было уделено недостаточное внимание.

В смешанной оптимизации пространство поиска может быть разделено на непрерывную и дискретную области, которые соответствуют непрерывной и дискретной компонентам вектора поиска переменных соответственно. Общая форма проблемы смешанной многокритериальной оптимизации может быть представлена следующим образом:

$$\min_{\mathbf{x}} (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))$$

$$\mathbf{x} = (x_1^d, x_2^d, \dots, x_k^d, x_1^c, x_2^c, \dots, x_l^c),$$

где $f_i(\mathbf{x})$ – i -я целевая функция ($i=1, \dots, m$), \mathbf{x} – вектор поиска, x_1^d, \dots, x_k^d – дискретные компоненты вектора поиска, x_1^c, \dots, x_l^c – непрерывные компоненты вектора поиска.

Многокритериальная оптимизация направлена на одновременное улучшение нескольких критериев. Вместо поиска единственного оптимума ищется набор лучших компромиссных решений, который называется оптимальными по Парето. Эти решения представляют собой наилучший компромисс между различными целями.

Модифицированный алгоритм роящихся частиц для многокритериальной оптимизации, предложен в [174]. Поскольку оптимум представляют множество недоминируемых решений (фронт Парето), в данной работе предлагается использовать внешний архив для их хранения. На каждой итерации алгоритма, недоминируемые решения попадают в этот архив и затем используются для перемещения частиц на следующей итерации. Пространство поиска разбивается на гиперкубы, которые адаптируются количеством содержащихся в них частиц. Данный метод использовался для модификации многих метаэвристических алгоритмов роевого интеллекта, чтобы проводить поиск по нескольким критериям. Модификация метаэвристики «стая волков» приведена в [175] и [176], в работе [177] был изменен алгоритм «муравьиного льва»,

Для решения задач смешанной многокритериальной оптимизации применяют два типа методов. Первый тип – это традиционные детерминированные методы оптимизации, а второй тип основан на стохастических, недетерминированных алгоритмах. Для работы с традиционными методами сначала преобразуют многокритериальную задачу к однокритериальной. Для этого могут использоваться метод взвешенных сумм, метод ϵ -ограничений, метод скаляризации с взвешенными ограничениями [170]. Затем применяются традиционные методы оптимизации, такие как метод градиентного спуска, симплекс метод, метод ветвей и границ. В работе [178] был применен метод ветвей и границ. Традиционные методы оптимизации имеют ограничения, например, целевые функции должны быть непрерывны, дифференцируемы или выпуклы. Кроме этого, традиционным методам необходимо аналитическое представление целевой функции, т.е. они не способны решать оптимизационную проблему и «черного ящика» [179].

Среди стохастических алгоритмов использовались метаэвристические алгоритмы эволюционного и роевого интеллекта. Один самых популярных алгоритмов многокритериальной оптимизации, генетический алгоритм с недоминируемой сортировкой NSGAII, способен работать со смешанными данными [180]. Дискретные элементы вектора решений могут быть представлены не только в бинарном виде, как в каноническом генетическом алгоритме, но и в произвольном дискретном виде [181], поскольку генетические операторы не требуют числовую природу элемента вектора решений.

Эволюционный алгоритм оценивания вероятностных распределений применялся в работе [182]. Особенностью данного алгоритма, в отличие от других стратегий эволюционного поиска, например таких как генетический алгоритм, является то, что эволюция от поколения к поколению осуществляется с помощью оценки вероятностных распределений наиболее приспособленных особей и генерации на их основе новых особей. Это позволяет избежать использования оператора скрещивания и сократить количество параметров алгоритма [183, 184]. Благодаря использованию вероятностных распределений может осуществляться генерация как непрерывных, так и дискретных элементов векторов решений. Многокритериальность поиска оптимума осуществлялась согласно методу скаляризации целевых значений, предложенного в [185]. Согласно этому методу, многокритериальная задача оптимизации сводится к некоторому числу однокритериальных (скалярных) подзадач. В качестве целевой функции используется функция Чебышева, объединяющая целевые значение в один критерий. Для оценки эффективности в работе применены тесты ZDT и DTLZ, разработанные для многокритериальных непрерывных алгоритмов. Дискретные переменные создавались путем квантования непрерывных значений, причем использовались несколько вариантов долей дискретных переменных.

Алгоритм смешанной многокритериальной оптимизации роящихся частиц предложен в [186]. Для работы со смешанными данными использован подход работы [187]. Пространство поиска разделено на непрерывную и дискретную области, которые соответствуют вещественной и дискретной компонентам

вектора решения. Перемещение частицы в обеих областях проводится согласно каноническому алгоритму, т.е. как в непрерывном пространстве. Чтобы перевести положение частицы в дискретную область используются локальные гиперкубы, каждая вершина которого соответствуют одному из вариантов дискретных значений. Набор дискретных значений ближайшей к частице вершине, согласно евклидовому расстоянию, будет соответствовать значению дискретных переменных. Для обеспечения многокритериальности, используются наборы локальных и глобальных решений. При перемещении частицы выбираются случайным образом решения из этих наборов. Кроме этого, предложен оригинальный метод сохранения многообразия популяции частиц, основанный на выборе глобальной частицы с помощью введенной метрики разнообразия. Алгоритм был опробован для задачи инженерного проектирования, а именно оптимизации компоновки ветряных электростанций.

Другой вариант алгоритма роящихся частиц предложен в [188]. Поддержка многокритериальности в этом алгоритме основана на идее, предложенной в работе [189]. Лучшие недоминируемые решения, найденные в процессе выполнения алгоритма, записываются в архив. При генерации новых решений, глобальный лидер случайным образом выбирается из этого архива. Дискретные переменные представляются в двоичной кодировке, где элемент вектора решения принимает значение 0 или 1. Таким образом, каждая дискретная переменная может быть представлена набором бинарных элементов вектора решения. Работа алгоритма с такими бинарными элементами проводится согласно [190] с помощью трансформационных функций, переводящих непрерывные значения в 0 или 1. Работа с вещественными переменными проходит согласно каноническому алгоритму. Результат работы сохраняется в архиве. Созданный алгоритм был применен для проектирования инженерных сооружений.

2.3 Метаэвристический алгоритм смешанной многокритериальной оптимизации

В подразделе представлен алгоритм MICS (Mixed-Integer Cuckoo Search) [191, 192], который является алгоритмом многокритериальной оптимизации для поиска оптимума в смешанном пространстве, образованном дискретными и непрерывными переменными. В разделе 3 оценена работа алгоритма в модифицированных тестах ZDT и DTLZ со смешанными переменными. Результаты вычислительного эксперимента показали высокую эффективность предлагаемого алгоритма на комплексных оценках сходимости и многообразия. Алгоритм был применен в методике построения интерпретируемых нечетких классификаторов, представленной в подразделе 2.7. С помощью данного алгоритма ведется построение структуры и оптимизация параметров с соблюдением компромисса между критериями точности и интерпретируемости моделей. В Приложении А представлено свидетельство о регистрации программы, реализующей данный алгоритм.

Алгоритм MICS основан на метаэвристике «кукушкин поиск» [193, 194] и генетическом алгоритме. Компонент вектора поиска, соответствующий непрерывным переменным, предлагается подбирать с помощью алгоритма «кукушкин поиск», а дискретный компонент с помощью генетического оператора скрещивания. Такое применение обусловлено тем, что алгоритмы роевого интеллекта показали лучшие результаты поиска в непрерывном пространстве. Это достигается благодаря стратегии поиска, связанной с движением материальной частицы. Применяются такие характеристики как расстояние, скорость и направление движения частиц, что неотъемлемо связано с непрерывным изменением величин [195]. Алгоритм «кукушкин поиск» относится к алгоритмам роевого интеллекта, и экспериментальные исследования показали его преимущества по сравнению с каноническим алгоритмом роящихся частиц [193]. Генетические алгоритмы, наоборот, в своей стратегии поиска не используют

аспекты, связанные с расстоянием, и операторы генетического алгоритма, согласно метафоре, работают в дискретном пространстве.

Оптимизационный алгоритм «кукушкин поиск». Оптимизационный алгоритм «кукушкин поиск» (АКП) – метаэвристика, построенная на основе имитации способа размножения кукушки, когда она находит недавно построенные гнезда и подкладывает в них свои яйца (заменяет своими), которые в итоге могут быть выброшены хозяином гнезда. В основе алгоритма лежат три следующих правила:

1) кукушка откладывает по одному яйцу в случайно выбранное гнездо, которое представляет собой решение;

2) часть лучших решений будет перенесена в следующее поколение;

3) количество гнезд фиксировано и есть вероятность того, что хозяин может обнаружить чужое яйцо; в этом случае хозяин может выбросить яйцо из гнезда или вовсе отказаться от гнезда и построить новое на новом месте.

С математической точки зрения первое правило представляет собой случайный процесс перемещения в пространстве поиска, причем отличительной особенностью АКП является то, что движение по пространству осуществляется с помощью полетов Леви. Процесс полета Леви является случайным блужданием, которое характеризуется серией скачков, обусловленных функцией плотности вероятности с «толстыми» хвостами, за счет которых вероятность значительных отклонений от среднего больше, чем у нормального распределения. Плотность распределения вероятности полетов Леви $L(s)$ задается следующим выражением:

$$L(s) = \frac{1}{\pi} \int_0^{\infty} \cos(\tau s) e^{-\alpha \tau^{\beta}} d\tau,$$

где параметр α характеризует отклонение плотности и в данной работе устанавливается в значение 1, параметр β принимает значения из интервала $[1, 2]$.

На рисунке 2.3 показаны плотности распределения вероятности нормального распределения со средним значением 0 и отклонением 1 ($N(0,1)$) и полетов Леви с $\beta=3/2$.

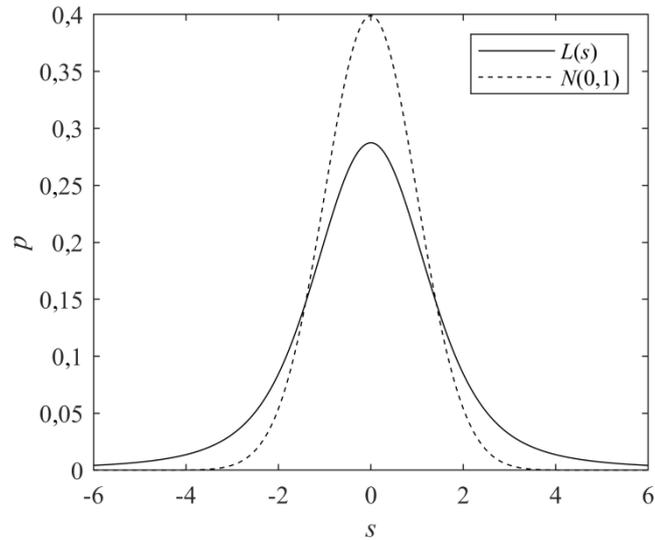


Рисунок 2.3 – Плотности распределения вероятностей

Исследования показывают [196, 197], что полеты Леви могут максимально увеличить эффективность поисков ресурсов в неопределенной среде. Полеты Леви были обнаружены среди траекторий поиска добычи питания альбатросов, паукообразных обезьян и плодовых мушек. Даже люди, охотники-собиратели племени Ju/'hoansi, могут делать пути с траекториями полетов Леви. На рисунке 2.4 показаны траектории движения из 100 шагов с длиной шага, соответствующей полетам Леви и нормальному распределению. В траектории движения с полетами Леви заметны чередования шагов малого и большого размера.

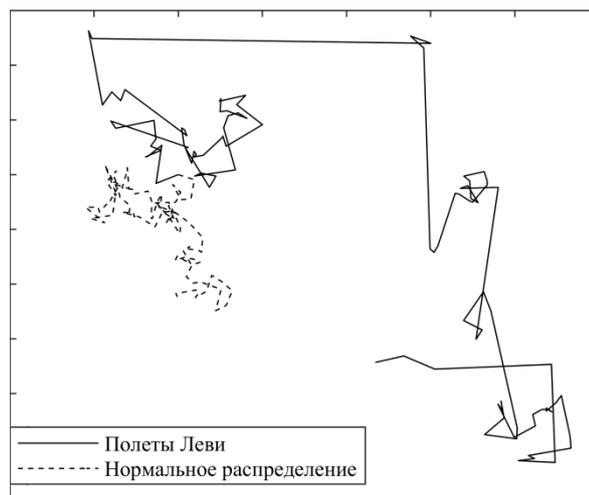


Рисунок 2.4 – Траектории движения с величиной шага согласно полетам Леви и нормальному распределению

Существует несколько способов генерации полетов Леви, но один из самых эффективных и в то же время простых является использование алгоритма,

предложенного в [198] для симметричного стабильного распределения Леви. Здесь «симметричного» означает, что шаги могут быть положительные и отрицательные согласно плотности вероятности на рисунке 2.3. В соответствии с этим алгоритмом длина шага вычисляется следующим образом:

$$Levi(\gamma) = \gamma \cdot u / v^{1/\beta}, \quad (2.1)$$

где γ – коэффициент прыжка полетов Леви и в данной работе он выставлен в значение $0,1 \cdot L$ (L – размер области поиска) согласно рекомендации в [194]; u, v – нормально распределенные случайные величины: $u \sim N(0; \sigma_u^2)$, $v \sim N(0; \sigma_v^2)$, $\sigma_v^2 = 1$,

$$\sigma_u^2 = \left(\frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma((1 + \beta)/2) \cdot 2^{(\beta-1)/2}} \right)^{1/\beta},$$

$\Gamma(x)$ – гамма-функция; параметр β в данной работе выставлен в значение $3/2$.

Второе правило, лежащее в основе алгоритма, по своей сути соответствует элитарности, т.е. лучшие решения передаются следующему поколению. Такая стратегия помогает улучшить сходимость алгоритма. Третье правило можно рассматривать как мутацию, при которой с определенной вероятностью отбрасываются наихудшие решения и генерируются новые решения, похожие на имеющиеся.

Модификация для многокритериальной оптимизации. Алгоритмы многокритериальной оптимизации, в отличие от однокритериальной оптимизации, находят не единственное решение, а множество недоминируемых по Парето решений. Решение A доминирует B по Парето (символически обозначается $A \prec_F B$), если A не уступает B по всем критериям, и, по крайней мере, по одному превосходит. Математическая запись этого определения выглядит следующим образом:

$$A \prec_F B = \forall i (f_i(A) \leq f_i(B)) \wedge \exists j (f_j(A) < f_j(B)).$$

Область пространства доминируемых решений решением A при двух критериях f_1 и f_2 показана на рисунке 2.5а. Множество всех недоминируемых решений образует фронт Парето (обозначается F_1), и говорят, что такие решения имеют ранг 1. Множество решений, над которыми доминируют только решения из F_1 , образуют множество F_2 . Такие решения имеют ранг 2. Множество решений, над которыми доминируют только решения из F_1 и F_2 , образуют множество F_3 и получают ранг 3 и т.д.. На рисунке 2.5б изображены множества решений, имеющих ранги 1, 2 и 3. Таким образом, целевой функцией для алгоритма оптимизации выступает ранг Парето, который должен быть минимизирован.

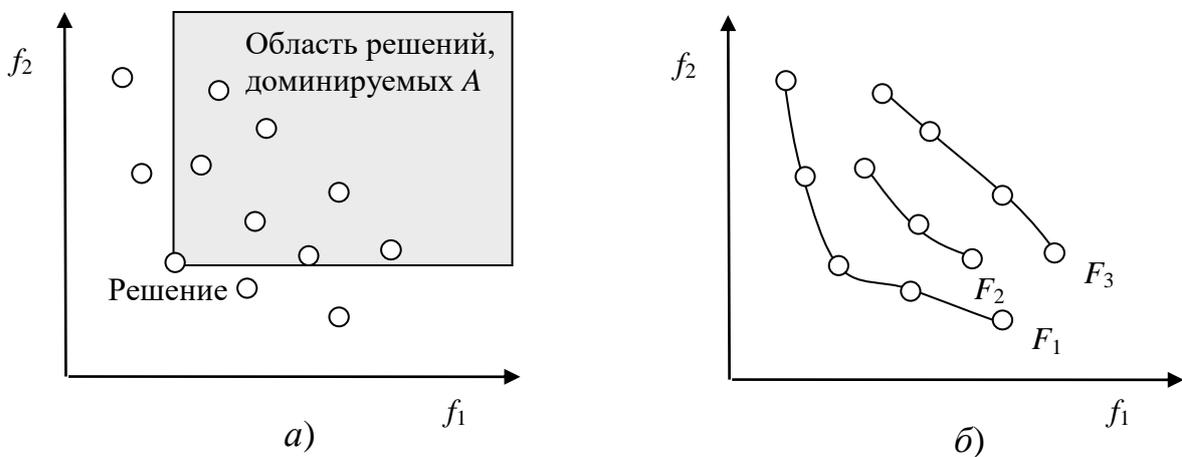


Рисунок 2.5 – Области решений оптимизации с двумя критериями а) область доминирования решения A ; б) фронты Парето решений с разными рангами

В работе [180] предложен алгоритм недоминируемой сортировки, определяющий ранги Парето для популяции в генетическом алгоритме. В то же время не все решения, имеющие одинаковый ранг, являются равнозначными. Желательно, чтобы они были равномерно распределены вдоль всей границы фронта. Для этого вводится специальная метрика разреженности, измеряющая расстояние между решениями с одинаковым рангом Парето. Вычисляется она с помощью Манхэттенского расстояния по всем критериям между правым и левым соседом решения. Решения, находящиеся на концах фронта, имеют максимальное значение разреженности. Чем больше данное расстояние, тем выше рангом решение. На рисунке 2.6 показаны решения одного ранга Парето. Решение A имеет ранг разреженности выше, чем B , поскольку $A_1 + A_2 > B_1 + B_2$. Таким образом,

решения с одинаковым рангом Парето могут быть ранжированы согласно метрике разреженности.

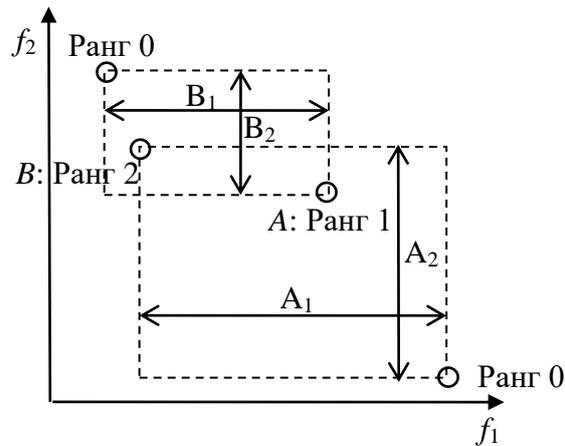


Рисунок 2.6 – Выставление рангов решений с учетом разреженности

Поскольку при генерации новой популяции в роевых алгоритмах учитывается лучшее глобальное решение, то в случае многокритериального подхода в настоящей работе предлагается выбирать его случайным образом на фронте Парето. Выбор должен быть осуществлен таким образом, чтобы решения с большим рангом разреженности имели большую вероятность быть выбранными, поскольку преобладает необходимость осуществлять поиск в их окрестностях. Здесь могут быть применены методы селекции генетического алгоритма [72], где функцией приспособленности выступает ранг разреженности. Поскольку в алгоритме «кукушкин поиск» лучшие решения популяции не могут быть потеряны в ходе его работы, то нет необходимости создавать архив для сохранения их значений.

Алгоритм смешанной многокритериальной оптимизации. На рисунке 2.7 представлена схема предлагаемого алгоритма. Итерационный процесс начинается с генерации новых решений, после которого проводится их обновление, путем изменения значений некоторых переменных. Далее извлекаются лучшие решения для генерации новых решений на следующей итерации. Новая итерация начинается после обновления текущей популяции с помощью рангов недоминируемой сортировки и сортировки разреженности.

Процесс повторяется заданное число раз. После завершения из популяции извлекаются недоминируемые по Парето решения.



Рисунок 2.7 – Схема алгоритма MICS

Ниже представлено подробное пошаговое описание алгоритма.

Вход: Размер популяции N , число итераций $MaxIter$, вероятность удаления гнезда p_a .

Выход: R – множество решений фронта Парето.

Начало

Шаг 1. Инициализация.

Случайным образом сгенерировать популяцию решений $Pop = \{Pop_1, Pop_2, \dots, Pop_N\}$. Популяция новых решений $Pop^{new} = \{\emptyset\}$. Инициализировать популяцию лучших решений $Pop^{best} = Pop$. Значения целевых функций популяции Pop установить в максимум $f_{ij}^{pop} = \infty$, где $i = 1, \dots, N, j = 1, \dots, m$.

Шаг 2. Инициализировать счетчик итераций $t = 1$. Начало итерационного процесса.

Шаг 3. Генерация новых решений.

Формируется популяция новых решений Pop^{new} на основе Pop и Pop^{best} .

Для элементов векторов решений, которые соответствуют дискретным значениям, применить генетический оператор скрещивания над Pop и Pop^{best} . В настоящей работе применялось равномерное скрещивание, когда значение гена потомка выбирается случайным образом от соответствующих генов родителей. Схема представлена на рисунке 2.8.

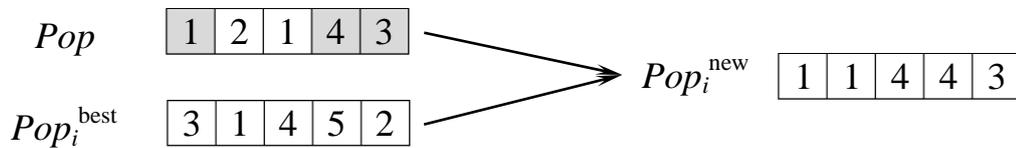


Рисунок 2.8 – Генерация дискретных элементов нового вектора решений

Для элементов векторов решений, которые соответствуют вещественным значениям, выполнить «случайное перемещение кукушки». Текущее значение изменить на величину полета Леви согласно (2.1), размер области поиска определяется расстоянием до лучшего решения из Pop^{best} :

$$Pop_i^{new} = Pop_i + Levi(0,1 \cdot (Pop_i^{best} - Pop_i)),$$

где $i = 1, \dots, N$.

Шаг 4. Изменить новые решения (мутация).

Изменение значения некоторых элементов Pop^{new} .

Изменить значения элементов векторов решений Pop^{new} с вероятностью p_a . Новые значения выбираются случайным образом из области значений элемента вектора. Данный шаг аналогичен оператору мутации генетического алгоритма.

Шаг 5. Вычислить значения целевых функций для каждого решения Pop^{new} .

$$f_{ij}^{new} = f_j(Pop_i^{new}),$$

где $i = 1, \dots, N, j = 1, \dots, m, f_j(\cdot)$ – j -я целевая функция.

Шаг 6. Отбор лучших решений на фронте Парето (Селекция).

Формируется множество лучших решений $Pop^{best} = \{Pop_1^{best}, Pop_2^{best}, \dots, Pop_N^{best}\}$.

Для популяции $Pop \cup Pop^{new}$ на основе значений целевых функций f^{new} и f^{pop} определить ранг Парето с помощью недоминируемой сортировки и ранг разреженности $rank_c$ каждого элемента. Сформировать из элементов фронта Парето $F_1 = \{b_1, b_2, \dots, b_k\}$ множество из N решений $Pop^{best} = \{Pop_1^{best}, Pop_2^{best}, \dots, Pop_N^{best}\}$. Поскольку предпочтения отдаются решениям с высшим рангом разреженности, то в качестве инструмента выбора может быть применена, например, рулеточная селекция. Вероятность выбора решения b_i ($i=1, \dots, k$) из F_1 в Pop^{best} в таком случае может быть рассчитана следующим образом:

$$P(b_i) = \frac{2(k - rank_c(b_i))}{k(k+1)}.$$

Шаг 7. Обновление популяции.

Обновить популяцию Pop с учетом новых решений Pop^{new} . Для этого применяется процедура, предложенная в [180]. Схема процедуры представлена на рисунке 2.9. Новую популяцию Pop будут составлять N решений из множества $Pop \cup Pop^{new}$, которое содержит $2 \cdot N$ элементов. Решения выбираются согласно рангам недоминирования Парето. Сначала извлекаются решения F_1 , затем F_2 и т.д. Если в процессе извлечения получилось так, что выбрана только часть какого-либо множества F_i (на рисунке 2.9 это множество F_3), то извлекаться должны

решения, имеющие высший ранг разрежённости. Т.е. решения из F_i выбираются согласно сортировке разрежённости. После необходимо обновить значения целевых функций f^{pop} согласно новым элементам популяции Pop .

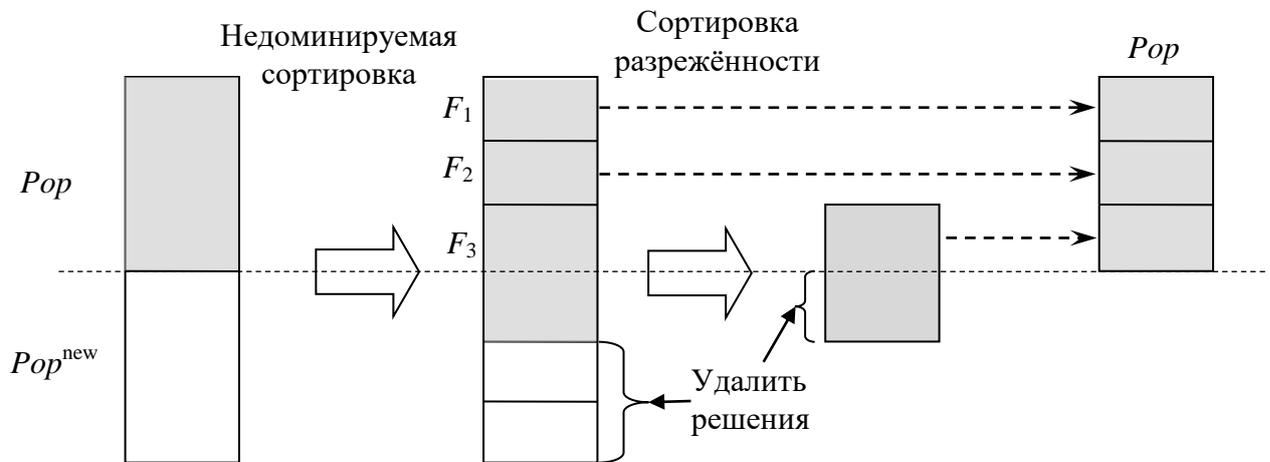


Рисунок 2.9 – Обновление популяции с помощью недоминируемой сортировки и сортировки разрежённости [180]

Шаг 8. Проверка остановки алгоритма.

Если $t < MaxIter$ то $t = t + 1$ и перейти на *Шаг 3*, иначе перейти на *Шаг 8*.

Шаг 9. Извлечь в R недоминируемые решения из популяции Pop .

Конец

2.4 Дискретный алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений

В подразделе предложен новый метаэвристический алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений PDT (Probability Distribution with Transformation) [199, 200, 201] для поиска оптимума в дискретном пространстве. В основе алгоритма лежит формирование распределений вероятностей для значений переменных вектора поиска. Данные распределения используются для генерации новых решений и формируют направление поиска. В разделе 3 оценена работа алгоритма в бинарном пространстве поиска для нахождения оптимума тестовых функции, применяемых

для оценки эффективности алгоритмов оптимизации. Вычислительный эксперимент показал высокую эффективность алгоритма по сравнению с генетическим алгоритмом и алгоритмом роящихся частиц. Практическое применение алгоритм обрел в методике построения интерпретируемых нечетких систем, о чем представлено в подразделе 2.7. В методике алгоритм используется для поиска в бинарном пространстве и с помощью него осуществляется разделение исходных данных на обучающую и валидационную выборки с минимизацией потерь информации при разделении. В Приложении А представлено свидетельство о регистрации программы, реализующей данный алгоритм.

Бинарная оптимизация. Комбинаторная оптимизация занимается проблемами нахождения оптимума с дискретными значениями возможных решений. Одним из частных случаев этой проблемы является бинарная оптимизация, в которой элементы вектора решения могут принимать только два значения. Практическое применение таких задач весьма обширно. В области медицины бинарная оптимизация применялась для диагностики опухолей головного мозга [202], нахождения подмножеств согласованных признаков при прогнозировании эффективности реабилитации пациентов после коронавирусной инфекции [117], классификации сложных заболеваний [203], классификации аритмии по электрокардиограмме [204]. В экономической сфере для выбора издателей журналов при размещении рекламы [205], планировании рабочего процесса [206], планировании выпуска новой версии программного обеспечения [207], проектировании производственных ячеек [208]. В науке и технике бинарная оптимизация использовалась для нахождения подмножества информативных признаков при построении прогностических систем [100, 111, 126], восстановлении нагрузки в первичных распределительных сетях [209], диагностики неисправности энергосистем [210], решении проблемы позиционирования антенны [211], проектировании сварных балок [212], разделении аппаратного и программного обеспечения во встроенных системах [213]. В [190] отмечается, что помимо чисто комбинаторных задач, задачи с

вещественными числами могут быть представлены в двоичном виде и решены в дискретном числовом пространстве.

Для решения задач бинарной оптимизации применяют два типа методов. Первый тип – это традиционные детерминированные методы оптимизации, а второй тип основан на стохастических, недетерминированных алгоритмах. Традиционными являются методы релаксации, Лагранжа, ветвей и границ, целочисленное программирование [214, 215]. К сожалению, эти методы являются трудозатратными и предназначены для решения задач небольших размерностей, что на практике встречается очень редко. Кроме этого, большинство традиционных методов требуют аналитическое задание целевой функции, а также ее дифференцируемость и неразрывность. Задачи большой размерности с множеством локальных оптимумов значительно ухудшают поиск традиционных методов.

Вышеперечисленные проблемы устраняют недетерминированные методы, представляемые метаэвристическими алгоритмами [172, 173]. В отличие от традиционных методов данные алгоритмы не подвергнуты «застреванию» в локальных оптимумах, в меньшей степени зависят от исходных отправных точек, не ограничены видом целевой функции и способны решать оптимизационную проблему «черного ящика» [179].

В [216] доказано, что не существует эвристического алгоритма, который мог бы работать достаточно эффективно для решения всех задач оптимизации. Разработанные в настоящее время алгоритмы дают удовлетворительные результаты при решении некоторых задач, но не всех. Поэтому в этой области ведутся активные исследования, в результате чего предлагаются новые эвристические алгоритмы.

Наиболее популярные алгоритмы бинарной оптимизации относятся к алгоритмам роевого интеллекта. Подобно эволюционным они основаны на механизмах природы и представляют собой модель скоординированного поведения объектов, которые могут быть представителями флоры, фауны или физическими объектами. Эволюционные вычисления основаны на конкуренции и

естественном отборе, тогда как роевой интеллект опирается главным образом на сотрудничество объектов [217].

Большинство алгоритмов роевого интеллекта разработаны для непрерывной оптимизации и для того, чтобы осуществлять поиск в бинарном пространстве, применяются механизмы адаптации, называемые бинаризацией [125]. Самым популярным методом бинаризации является использование трансформационных функций, которые переводят непрерывные значения элементов векторов решений в значения из диапазона $[0, 1]$. Затем применяется правило бинаризации, при котором решение преобразуется в бинарное значение из множества $\{0, 1\}$. С помощью функций трансформации были адаптированы алгоритмы роящихся частиц [190, 218], искусственных водорослей [219], шимпанзе [220], роя сальпов [221], стаи китов [222]. В [218] были исследованы различные варианты функций трансформации для алгоритма роящихся частиц. Лучшая сходимость была достигнута алгоритмом с V-образной функцией трансформации.

Метод бинаризации на основе модификации алгебраических операций преобразует вещественные операторы, используемых в формулах перемещения частиц, в их логические аналоги, что позволяет оперировать бинарными решениями. Например, вместо сложения используется операция дизъюнкции, а вместо умножения – конъюнкция. С помощью данного метода были адаптированы алгоритмы роящихся частиц [223], мозгового штурма [210], роста деревьев [224], летучих мышей [223], непрерывной муравьиной колонии [205], черной дыры [225], кукушкин поиск [226].

Квантовый метод бинаризации тоже преобразует операторы непрерывного алгоритма. В этом методе каждое допустимое решение имеет позицию и вектор квантования, который содержит вероятности принять значение 1 для соответствующего элемента решения. Вектор квантования обновляется с учетом положений глобальных и локальных лидеров. Используя данный метод, были адаптированы алгоритмы роящихся частиц [227], искусственных водорослей [207], гравитационный поиск [228], муравьиной колонии [229].

Среди алгоритмов эволюционного интеллекта для бинарной оптимизации широко применялся генетический алгоритм [230-232]. Также были предложены его модификации, так, например, в работе [233] представлен гибрид на основе генетического алгоритма и алгоритма роящихся частиц. Сначала оба алгоритма находят решения независимо друг от друга, а затем результаты объединяются с помощью метода средневзвешенной комбинации. После этого применяется локальный поиск для нахождения окончательного решения.

Оценка эффективности алгоритмов в большинстве исследований проводилась при решении определенных прикладных задач. Для объективной оценки работы алгоритмов применяют тестовые функции, которые позволяют определить эффективность при нахождении оптимума различных целевых функций, например, унимодальных, мультимодальных, овражных, разрывных, выпуклых, вогнутых. При бинарной оптимизации применяют тестовые функции для поиска в непрерывном пространстве. Бинарное пространство поиска образуют путем дискретизации непрерывных значений и их бинарном кодировании [218, 234-236].

Дискретный алгоритм оптимизации. Разработанный дискретный алгоритм является итерационным, где на каждой итерации формируется вероятностная модель. Модель определяет вероятность появления конкретного дискретного значения для переменной. Вероятности формируются на основе частоты появления дискретного значения каждой переменной среди решений популяции, причем меньшее значение целевой функции должно увеличивать вклад решения в повышение вероятности. Для этого вводятся трансформационные функции, которые переводят значение целевой функции решения популяции в весовой коэффициент. На рисунке 2.10 представлена схема алгоритма.



Рисунок 2.10 – Схема алгоритма PDT

На этапе инициализации определяется начальная популяция решений P_{op} случайным или иным образом. Далее рассчитываются весовые коэффициенты решений популяции w . Весовой коэффициент принимает значение из диапазона $[0, 1]$, чем меньше целевое значение решения, тем больше значение w . Для того, чтобы сформировать весовые коэффициенты из целевых значений предлагается использовать функции трансформации. В качестве таких функций, например,

могут быть использованы следующие: T_L – линейная функция, T_S – сигмоида, T_Q – квадратичная функция, T_T – гиперболический тангенс. Графики функций представлены на рисунке 2.11. Область определения функций ограничена отрезком $[f_{\min}, f_{\max}]$, где f_{\min} и f_{\max} максимальное и минимальное значение целевой функции в популяции.

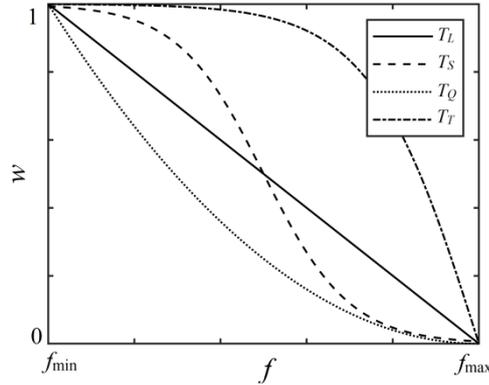


Рисунок 2.11 – Графики функций трансформаций для перевода целевых значений в веса решений

Аналитические выражения функций трансформации показаны ниже.

$$T_L(f) = \frac{f_{\max} - f}{f_{\max} - f_{\min}},$$

$$T_S(f) = \frac{1}{1 + e^{(10(f - f_{\min}) / (f_{\max} - f_{\min}) - 5)}},$$

$$T_Q(f) = \left(\frac{f - f_{\min}}{f_{\max} - f_{\min}} \right)^2,$$

$$T_T(f) = \left| \operatorname{th} \left(4 \left(\frac{f - f_{\min}}{f_{\max} - f_{\min}} - 1 \right) \right) \right|.$$

После получения весовых коэффициентов решений рассчитывается распределение вероятностей для каждой переменной, состоящей из вероятностей получить переменной определенное дискретное значение. Вероятности

рассчитываются на основе значений переменных и весовых коэффициентов решений.

С помощью полученных распределений генерируется новая популяция и подвергается мутации, чтобы предотвратить преждевременную сходимость. Далее популяция обновляется лучшими решениями текущей и новой популяции. После этого снова рассчитываются весовые коэффициенты решений, и продолжается новый цикл. По завершении заданного количества итераций из популяции выбирается лучшее решение. Рассматривается проблема оптимизации, в которой минимизируется критерий эффективности.

Ниже представлено пошаговое описание алгоритма.

Вход: Установить размер популяции N , число итераций $MaxIter$, вероятность мутации p_a и функцию трансформации T . Обозначим l_{jk} k -е дискретное значение j -й переменной.

Выход: R найденное решение.

Начало

Шаг 1. Инициализация.

Случайным или иным образом сгенерировать популяцию решений $Pop = [Pop_1, Pop_2, \dots, Pop_N]$ и вычислить соответствующее целевое значение $f = [f_1, f_2 \dots, f_N]$.

Шаг 2. Инициализировать счетчик итераций $t = 1$. Начало итерационного процесса.

Шаг 3. Вычислить весовые коэффициенты решений популяции Pop с помощью функции трансформации T : $w_i = T(f_i)$, где $i = 1, \dots, N$.

Шаг 4. Вычислить распределения вероятностей.

Для каждого дискретного значения k переменной j определить сумму весовых коэффициентов решений Pop , которые принимают данное дискретное значение k . Обозначим такую сумму S_{jk} , где $j = 1, \dots, n$, $k = 1, \dots, m$.

$$S_{jk} = \sum_{i=1}^N w_i \cdot \begin{cases} 1, & \text{если } Pop_{ij} == l_{jk} \\ 0, & \text{иначе} \end{cases} .$$

Вычислить эмпирическую вероятность появления k -го значения переменной j :

$$P_{jk} = \frac{S_{jk}}{\sum_{k=1}^m S_{jk}}.$$

Шаг 5. Сгенерировать новые решения.

Формируется популяция новых решений Pop^{new} на основе вероятностей P каждой переменной:

$$Pop_{ij}^{new} = l_{jk},$$

где k удовлетворяет условию $p_{k-1} < rand(0,1) \leq p_k$,

$$p = \left[0, P_{j1}, P_{j1} + P_{j2}, \dots, \sum_{k=1}^m P_{jm} \right],$$

$$j = 1, \dots, n, k = 1, \dots, m.$$

Шаг 6. Изменить новые решения (мутация).

Изменить значения элементов векторов решений Pop^{new} с вероятностью p_a . Новые значения выбираются случайным образом из области значений элемента вектора решений.

Шаг 7. Вычислить значение целевых функций f_i^{new} , где $i = 1, \dots, N$, для каждого решения Pop^{new} .

Шаг 8. Обновить популяцию Pop путем выбора N лучших решений из множества решений $Pop \cup Pop^{new}$. Обновить значения целевых функций f_i согласно решениям Pop .

Шаг 9. Проверка остановки алгоритма.

Если $t < MaxIter$ то $t = t + 1$ и перейти на *Шаг 3*, иначе перейти на *Шаг 10*.

Шаг 10. Извлечь в R лучшее решение из популяции Pop .

Конец

2.5 Адаптированный алгоритм градиентного спуска для оптимизации параметров нечетких классификаторов

В подразделе предложен адаптированный алгоритм градиентного спуска для настройки параметров нечеткого классификатора [237]. Данный алгоритм входит в состав методики построения нечеткого классификатора, представленной в подразделе 2.7, и применяется после построения классификаторов метаэвристическим алгоритмом. Исходные значения параметров, найденные с помощью метаэвристики, могут быть улучшены более точным локальным поиском градиентного спуска. Такое дообучение параметров классификаторов может повысить точность их прогнозов.

Современные методы построения нечетких классификаторов отдают предпочтение метаэвристическим алгоритмам, поскольку они способны осуществлять поиск глобального оптимума, в отличие от детерминированных, которые сосредоточены на локальном поиске. Также данные алгоритмы не требуют анализа целевой функции и ее дифференцируемости, что является затруднительным для нечетких классификаторов. Фактически целевая функция представляется «черным ящиком», для которого определяются только входные и выходные значения [179]. Но в детерминированных алгоритмах, основанных на дифференцировании, точность поиска локального оптимума в окрестности заданной точки значительно выше. Поэтому комбинированный поиск на основе метаэвристического и детерминированного алгоритмов потенциально может повысить эффективность поиска глобального оптимума.

Целевая функция оптимизации. Задача алгоритма градиентного спуска заключается в нахождении параметров функций принадлежности, увеличивающих эффективность классификации. Обычно эффективность классификации оценивается с помощью критериев непригодных для дифференцирования, например, таких как процент правильно классифицированных экземпляров, F -мера. Применение градиентных методов

оптимизации, в том и числе и алгоритма градиентного спуска, требует дифференцируемости целевой функции. Для нахождения параметров нейронных сетей с помощью алгоритма градиентного спуска (называемого обратным распространением ошибки) в качестве целевой функции используют сумму квадратов отклонений значений разделяющих функции и ожидаемых для экземпляров исходных данных, которая должна быть минимизирована [238]:

$$E = \frac{1}{2} \sum_{p=1}^m \sum_{k=1}^M (g_k(\mathbf{x}_p) - y_{pk})^2,$$

где (\mathbf{x}_p, l_p) – p -й образец исходных данных $Data$ (выражение (1.4)), y_{pk} ожидаемое значение разделяющей функции k -го класса для p -го образца. Если образец принадлежит классу ω_h , т.е. $l_p = \omega_h$, то ожидаемые значения разделяющих функций данного образца определяются следующим образом:

$$y_{pk} = \begin{cases} 1, & \text{если } k = h \\ 0, & \text{иначе} \end{cases}, k = 1, \dots, M.$$

Использование данной целевой функции для нахождения параметров нечеткого классификатора сталкивается со следующими проблемами. Первая проблема заключается в том, что наибольшие значения степени уверенности g_i в выборе класса объекта будут сильно отличаться от 1 и близиться к значению 0, особенно при больших признаковых пространствах объектов. Это обусловлено использованием t -нормальной функции «произведения». Таким образом, чтобы уменьшить значение целевой функции алгоритм оптимизации сосредоточится на уменьшении значений g_i для неправильных классов, игнорируя правильные. Вторая проблема появляется при выборе в качестве агрегации правил функции «суммы», согласно (1.2). В этом случае значения степеней уверенности в выборе класса могут быть больше 1, хотя ожидаемые значения ограничены диапазоном $[0, 1]$. Чтобы избавиться от вышеприведенной проблемы при нахождении параметров нечеткого классификатора предлагается изменить целевую функцию

оптимизации с помощью нормирования значений разделяющих функций путем деления на их сумму:

$$E = \frac{1}{2} \sum_{p=1}^m \sum_{k=1}^M \left(\frac{g_k(\mathbf{x}_p)}{\sum_{k=1}^c g_k(\mathbf{x}_p)} - y_{pk} \right)^2. \quad (2.2)$$

При выборе такого критерия увеличится разность между значениями степеней уверенности разных классов, а значения будут находиться в диапазоне $[0, 1]$.

Параметры нечеткого классификатора и вектор решений. С помощью алгоритма градиентного спуска предлагается оптимизировать параметры функций принадлежности нечеткого классификатора. Согласно (1.3), это среднее значение s и среднеквадратичное отклонение σ . Количество параметров, настраиваемых алгоритмом, определяется произведением числа нечетких множеств, на которые разбиты каждая переменная, и числом параметров функции принадлежности. Классификатор, представленный на рисунке 1.15 (подраздел 1.6), имеет 6 функций принадлежности гауссова типа и два параметра на каждую функцию. Таким образом, общее число параметров $6 \cdot 2 = 12$. Вектор решений для задачи оптимизации будет выглядеть следующим образом $\mathbf{P} = (s_1, \sigma_1, s_2, \sigma_2, \dots, s_6, \sigma_6)$.

Вычисление градиента целевой функции. Основой метода градиентного спуска является понятие градиента целевой функции. Градиент представляет собой вектор частных производных функции по ее переменным и показывает направление в сторону ее наискорейшего возрастания. Например, для задачи поиска параметров нечеткого классификатора, представленного на рисунке 1.15, градиент имеет следующий вид:

$$\nabla E = \left(E'_{s_1}, E'_{\sigma_1}, E'_{s_2}, E'_{\sigma_2}, \dots, E'_{s_6}, E'_{\sigma_6} \right).$$

Далее будут найдены формулы для вычисления частных производных по параметрам s и σ классификатора. Применяя правила дифференцирования

сложной, степенной и дробной функции по переменной b к целевой функции E (выражение (2.2)), получаем следующее значение частной производной.

$$E'_b = \sum_{p=1}^m \sum_{k=1}^M \left(\left(\frac{g_k(\mathbf{x}_p)}{\sum_{k=1}^c g_k(\mathbf{x}_p)} - y_{pk} \right) \frac{g_k(\mathbf{x}_p)'_b \left(\sum_{k=1}^c g_k(\mathbf{x}_p) \right) - \left(\sum_{k=1}^c g_k(\mathbf{x}_p)'_b \right) g_k(\mathbf{x}_p)}{\left(\sum_{k=1}^c g_k(\mathbf{x}_p) \right)^2} \right). \quad (2.3)$$

Чтобы рассчитать данное значение необходимо вычислить значения производных разделяющих функций $g_k(\mathbf{x}_p)'_b$, где $k = 1, \dots, M$. Рассмотрим два случая, когда применяется агрегация правил согласно (1.1) и согласно (1.2).

Агрегация правил функцией максимума (1.1).

Поскольку производные определяются по параметрам функций принадлежности, введем следующие обозначения. Если правило-«победитель» (обозначим переменной h номер такого правила) содержит в antecedентной части функцию принадлежности с параметром, по которому проводится дифференцирование, то значение производной по параметру среднего s функции принадлежности вычисляется следующим образом:

$$g_k(\mathbf{x}_p)'_s = \max_{c_j = \omega_k} \left(\prod_{i=1}^n \mu_{ij}(x_{ip}) \right)'_s = \left(\prod_{i=1}^n e^{-\frac{(x_{ip}-s_{ih})^2}{2\sigma_{ih}^2}} \right)'_s,$$

$$g_k(\mathbf{x}_p)'_s = \frac{x_{dp}-s}{\sigma^2} e^{-\frac{(x_{dp}-s)^2}{2\sigma^2}} \prod_{i=1, i \neq d}^n e^{-\frac{(x_{ip}-s_{ih})^2}{2\sigma_{ih}^2}} = \frac{x_{dp}-s}{\sigma^2} \prod_{i=1}^n e^{-\frac{(x_{ip}-s_{ih})^2}{2\sigma_{ih}^2}},$$

$$g_k(\mathbf{x}_p)'_s = \frac{x_{dp}-s}{\sigma^2} g_k(\mathbf{x}_p), \quad (2.4)$$

где d обозначает индекс переменной, по параметру функции принадлежности которой проводится дифференцирование, σ – среднее квадратичное отклонение функции принадлежности, в которой находится параметр s .

Аналогично определяется значение производной по параметру среднеквадратичного отклонения:

$$g_k(\mathbf{x}_p)'_{\sigma} = \frac{(x_{dp} - s)^2}{\sigma^3} e^{-\frac{(x_{dp} - s)^2}{2\sigma^2}} \prod_{i=1, i \neq d}^n e^{-\frac{(x_{ip} - s_{ih})^2}{2\sigma_{ih}^2}} = \frac{(x_{pd} - s)^2}{\sigma^3} g_k(\mathbf{x}_p), \quad (2.5)$$

где s – параметр среднего значения функции принадлежности, в которой находится отклонение σ .

Если правило-«победитель» не содержит в antecedентной части функцию принадлежности с параметром, по которому проводится дифференцирование то значение производной равно 0.

Агрегация правил функцией суммы (1.2).

Обозначим множество номеров правил k -го класса, содержащих в antecedентной части функцию принадлежности с параметром b , как R_k^b , а номер соответствующей переменной в правиле как d . Значение производной по параметру среднего s функции принадлежности определяется следующим образом:

$$g_k(\mathbf{x}_p)'_s = \left(\sum_{\substack{j, \\ c_j = \omega_k}} \prod_{i=1}^n \mu_{ij}(x_{ip}) \right)'_s = \sum_{j \in R_k^s} \left(\prod_{i=1}^n e^{-\frac{(x_{ip} - s_{ij})^2}{2\sigma_{ij}^2}} \right)'_s,$$

$$g_k(\mathbf{x}_p)'_s = \sum_{j \in R_k^s} \left(\frac{x_{dp} - s}{\sigma^2} e^{-\frac{(x_{dp} - s)^2}{2\sigma^2}} \prod_{i=1, i \neq d}^n e^{-\frac{(x_{ip} - s_{ij})^2}{2\sigma_{ij}^2}} \right)'_s = \frac{x_{dp} - s}{\sigma^2} \sum_{j \in R_k^s} \prod_{i=1}^n e^{-\frac{(x_{ip} - s_{ij})^2}{2\sigma_{ij}^2}}, \quad (2.6)$$

где σ – среднеквадратичное отклонение функции принадлежности, в которой находится параметр s . Значение производной по параметру среднеквадратичного отклонения σ будет иметь следующий вид:

$$g_k(\mathbf{x}_p)'_{\sigma} = \sum_{j \in R_k^{\sigma}} \left(\frac{(x_{pd} - s)^2}{\sigma^3} e^{-\frac{(x_{pd} - s)^2}{2\sigma^2}} \prod_{i=1, i \neq d}^n e^{-\frac{(x_{ij} - s_{ij})^2}{2\sigma_{ij}^2}} \right) = \frac{x_{pd} - s}{\sigma} g_k(\mathbf{x}_p)'_s, \quad (2.7)$$

где s – среднее значение функции принадлежности, в которой находится параметр отклонения σ .

Алгоритм градиентного спуска. Схема алгоритма оптимизации параметров на основе градиентного спуска приведена на рисунке 2.12. Вектор поиск, состоящий из параметров функций принадлежности, обозначим $\mathbf{P}=(p_1, p_2, \dots, p_n)$. Для поиска минимума целевой функции E (выражение 2.2), необходимо изменять параметры нечеткого классификатора в направлении противоположном направлению градиента. Причем при изменении параметры не должны выйти за границы их значений. Границы являются параметрами алгоритма и задаются до выполнения его работы. На схеме верхняя граница обозначена вектором $\mathbf{U}=(U_1, U_2, \dots, U_n)$, а нижняя вектором $\mathbf{L}=(L_1, L_2, \dots, L_n)$. Если параметры выходят за границы требуемого диапазона, то принимают граничные значения. Сам спуск является итерационным процессом, в котором вектор \mathbf{P} изменяется следующим образом:

$$\mathbf{P} = \mathbf{P} - h\nabla E(\mathbf{P}), \quad (2.8)$$

где h – длина шага в направлении спуска, а ∇E – градиент, который вычисляется согласно выражениям (2.3), (2.4), (2.5), (2.6) и (2.7). Длина шага в предлагаемом варианте выбирается методом дробления, т.е. итерации согласно (2.8) проводятся до тех пор, пока функция E убывает, а точность классификации $acc(\mathbf{P})$ не уменьшается. Если убывание прекращается или точность уменьшается, то шаг h уменьшается вдвое, вычисляются параметры \mathbf{P} с новым шагом, и снова проверяется условие. Дробление шага продолжается до тех пор, пока не будет выполнено условие. Итерационный процесс прекращается по достижению длины шага h минимального допустимого значения ϵ , которое задается до выполнения работы алгоритма.

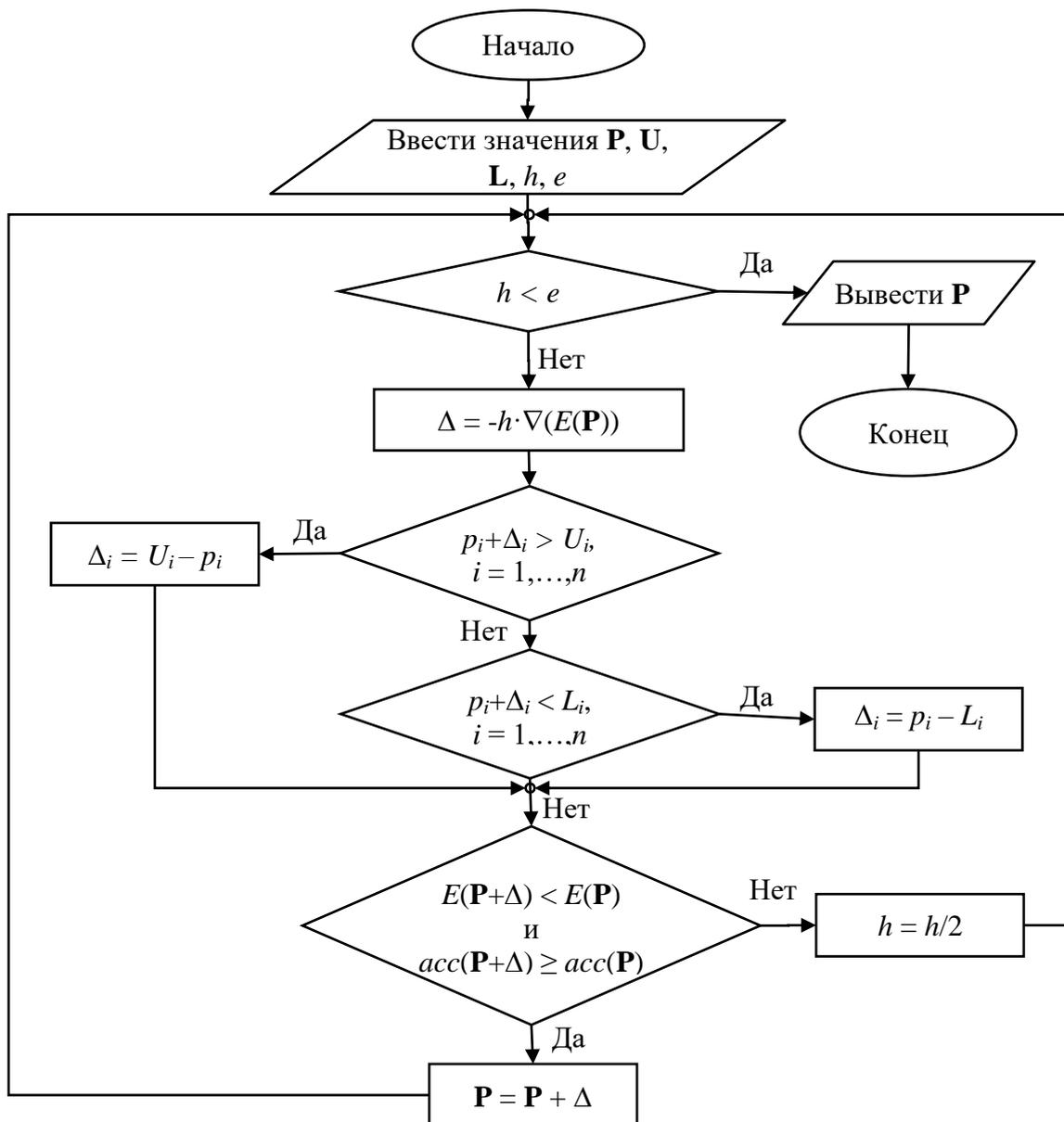


Рисунок 2.12 – Схема алгоритма градиентного спуска для оптимизации параметров нечеткого классификатора

Введение дополнительного условия для дробления шага, а именно уменьшения точности классификации, связано с предположением, что целевая функция оптимизации (2.2) в полной мере не может соответствовать точности классификации. Например, уменьшение значения целевой функции может не повысить точность классификации, или наоборот, увеличение значения целевой функции может не повлечь за собой уменьшение точности. Данное условие было добавлено, поскольку целью алгоритма является повышение точности. В экспериментальной части исследования (подраздел 3.4) проведен вычислительный эксперимент по оптимизации параметров нечетких

классификаторов адаптированным алгоритмом градиентного спуска и проведены сравнения точности классификации с применением и без применения данного условия.

2.6 Метод разделения данных для построения прогностических моделей машинного обучения

В подразделе представлен метод разделения данных на обучающую и валидационную выборки для построения прогностических моделей, направленный на сохранение близости выборок к исходным данным [239]. Под понятием близости подразумевается сохранение минимального отклонения характеристик признаков данных. В качестве таких характеристик в настоящем исследовании применяются среднее значение и среднеквадратичное отклонение. Сохранение близости позволяет уменьшить потерю информации при построении моделей по сравнению со случайным разделением, и тем самым повысить обобщающую прогностическую способность. С помощью обучающих данных осуществляется построение множества альтернативных моделей с различными по сложности структурными характеристиками, а с помощью валидационных выбирается наиболее точная модель. Предлагаемый метод применяется в методике построения интерпретируемых нечетких систем, представленной в подразделе 2.7. В экспериментальной части исследования (подраздел 3.3) представлен вычислительный эксперимент по построению нечетких классификаторов с разделением данных предлагаемым методом. Применение метода показало повышение точности классификации и интерпретируемости по сравнению со случайным разделением и без разделения исходных данных.

Метод разделения данных. Важной задачей машинного обучения является задание сложности структуры прогностической модели. Понятие сложности структуры связано с количественным составом компонентов модели, определяющих общее количество ее параметров [240-242]. Например, сложность

нейронной сети может быть определена количеством слоев и нейронов в них [243]. Сложность моделей на основе деревьев решений определяется максимальной глубиной дерева [244]. Сложность нечетких систем, например, может быть определена количеством правил или общим количеством используемых терминов [153]. Чем выше сложность модели, тем больше составных компонентов и параметров включено в нее. Большая часть вводимых мер сложности зависят от модели и сравнения этих мер возможны только между моделями одного типа [242, 243, 245].

Сложные по структуре модели, как правило, имеют более высокую точность прогноза, в то время как простые модели за счет меньшего количества параметров не всегда способны в достаточной мере определить зависимость между признаками и целевым значением. Кроме этого, сложные модели часто требуют больше вычислительных ресурсов, поэтому на выбор уровня сложности так же могут повлиять практические соображения, такие как доступная вычислительная мощность и временные ограничения. Также сложность модели неразрывно связана с понятием интерпретируемости, особенно в таких моделях как нечеткие системы и деревья решений [246]. Если интерпретируемость имеет решающее значение, то предпочтение отдается простым моделям, которые являются более интерпретируемыми и облегчают понимание и объяснение процессов принятия решений, связанных с ними [247].

Если целью построения прогностической модели является высокая точность, это не означает, что для ее достижения необходимо выбрать высокую сложность. Когда модель слишком сложна, она способна очень точно делать прогноз для обучающих данных, улавливая небольшие изменения значений признаков. Таким образом, модель становится чрезмерно адаптированной к конкретным данным и, как следствие, теряет обобщающую способность, понижая точность за пределами обучающего набора данных. Такое поведение модели машинного обучения получило название переобучение [248, 249]. Причиной переобучения часто является высокая сложность модели, поэтому важно правильно подобрать ее уровень для предотвращения падения обобщающей

способности. На рисунке 2.13 показано множество моделей в виде точек, недоминируемых по критериям сложности и точности, с различными значениями сложности структуры и соответствующими точностями в виде ошибки классификации на обучающих данных. Модель с лучшей обобщающей способностью не обязательно является моделью с лучшей точностью на обучающих данных, как это показано на рисунке.

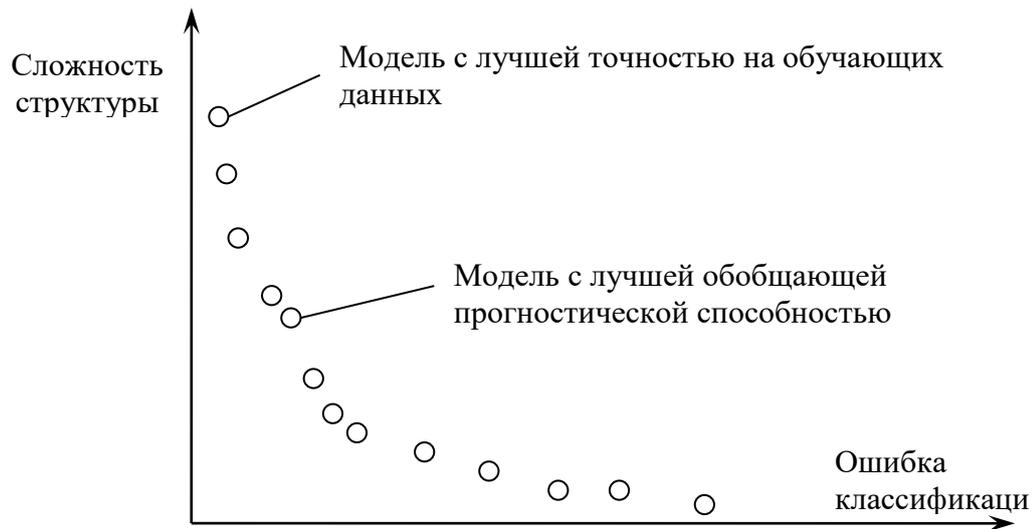


Рисунок 2.13 – Множество недоминируемых по двум критериям прогностических моделей

Обычно сложность структуры задается гиперпараметрами модели до ее построения. В таком случае, чтобы подобрать подходящую по сложности структуру, исключающую переобучение, проводят поиск, состоящий из серии запусков алгоритма построения модели и проверки их эффективности с различными комбинациями гиперпараметров. При этом построение модели проводится на обучающей выборке, а проверка эффективности на валидационной, которые создаются случайным образом из исходного набора [250]. Чтобы такой итерационный поиск гиперпараметров был эффективным, применяют специальные методы поиска и обнаружения эффективной комбинации [251, 252]. Наиболее популярными являются решетчатый поиск, случайный поиск и байесовская оптимизация. Решетчатый поиск является полным перебором фиксированного набора значений гиперпараметров для поиска оптимальной

комбинации [253]. Случайный поиск отличается от решетчатого поиска тем, что поиск выполняется не полным перебором комбинаций, а случайным образом, учитывая ограничения времени на выполнение и ресурсы [254]. Байесовская оптимизация проводит поиск на основе модели гауссова процесса [255].

Перечисленные методы такого итерационного подхода являются весьма трудозатратными, поскольку после каждой предложенной комбинации в процессе поиска необходимо запустить алгоритм обучения и проверить эффективность модели. Этот недостаток отсутствует в другом подходе подбора сложности структуры модели. Согласно этому подходу, формирование структуры происходит в процессе работы алгоритма обучения вместе с оптимизацией параметров, а уменьшение величины ее сложности является одной из целей построения. Здесь алгоритмом построения является многокритериальный алгоритм оптимизации, где целями являются увеличение точности прогноза и уменьшение сложности структуры модели. Алгоритм возвращает множество недоминируемых по Парето моделей согласно этим критериям.

Данный подход был применен в [83, 84, 88] для построения нечетких классификаторов. Для этого применялся многокритериальный эволюционный алгоритм оптимизации, где помимо увеличения критерия точности так же целью являлось уменьшение критерия сложности структуры модели, а именно количества нечетких правил и общего количества нечетких терминов. При выборе классификатора с лучшей прогностической способностью среди моделей, построенных алгоритмом, авторы выбирали модель с лучшей точностью, оцениваемой на обучающих данных. Валидационная выборка для выбора модели не использовалась. Для построения применялись все исходные данные, а это, как было показано выше, может быть причиной выбора переобученной модели, чрезмерно адаптированной к обучающим данным.

Чтобы бороться с переобучением в обоих подходах необходимо разделять исходные данные на обучающую и валидационную выборки. Обычно разделение проводят случайным образом, что может повлечь потерю важной информации для моделирования наблюдаемого процесса или явления, особенно при малом

количестве образцов. Это негативно отразится на обобщающей способности прогностической модели. Необходимо сохранить близость выборок к исходным данным. Под близостью в настоящей работе подразумевается минимальное отклонение характеристик полученных выборок от исходных данных. В качестве таких характеристик могут выступать среднее значение и среднеквадратичное отклонение в признаках данных.

Для решения указанной выше проблемы в настоящем исследовании предлагается метод разделения исходных данных на обучающую и валидационную выборки с помощью бинарных метаэвристических алгоритмов. Метод позволяет сохранить близость выборок к исходным данным, минимизировать потерю информации по сравнению со случайным разделением и может быть применен для построения моделей как в итерационном подходе с подбором гиперпараметров, так и с помощью многокритериальной оптимизации.

В настоящем подразделе представлен новый метод, позволяющий провести разделение с сохранением близости. Для того чтобы минимизировать отклонение характеристик признаков в методе применяется бинарный алгоритм оптимизации, который формирует обучающую и валидационную выборки, близкие к исходной. Введение валидационной выборки позволит исключить выбор переобученной модели среди моделей с различными по сложности структурными свойствами и увеличит обобщающую способность. Схема построения прогностических моделей с применением разделения данных представлена на рисунке 2.14.



Рисунок 2.14 – Схема построения прогностических моделей с разделением данных

Бинарный алгоритм оптимизации. В общем виде работа метаэвристических популяционных алгоритмов оптимизации представляет итерационный процесс, в котором генерируются новые решения и отбираются лучшие из имеющихся и новых. Генерация новых решений осуществляется на основе стратегии работы алгоритма, связанной с выбором метафоры (образчика наблюдаемого процесса или явления живой и неживой природы). Итерационный процесс продолжается до наступления определенных условий, которыми могут выступать, например, заданное количество итераций, достижение необходимого результата или отсутствие улучшения целевых значений на протяжении определенного количества итераций.

Применяемые в методе алгоритмы являются бинарными, т.е. поиск оптимального решения осуществляется в бинарном пространстве, когда переменная может принимать только два дискретных значения. Изменения, которым необходимо подвергнуть алгоритм, связаны с ограничениями, накладываемыми на отношение количества образцов обучающей и валидационной выборки. Данное отношение является параметром метода и задается до начала его выполнения. После генерации новых решений необходимо их корректировать, чтобы удовлетворять ограничениям. Таким образом, схема бинарного метаэвристического алгоритма, применяемого в методе, будет иметь вид, представленный на рисунке 2.15. Серым цветом выделен новый оператор модификации решения, необходимый для работы метода.

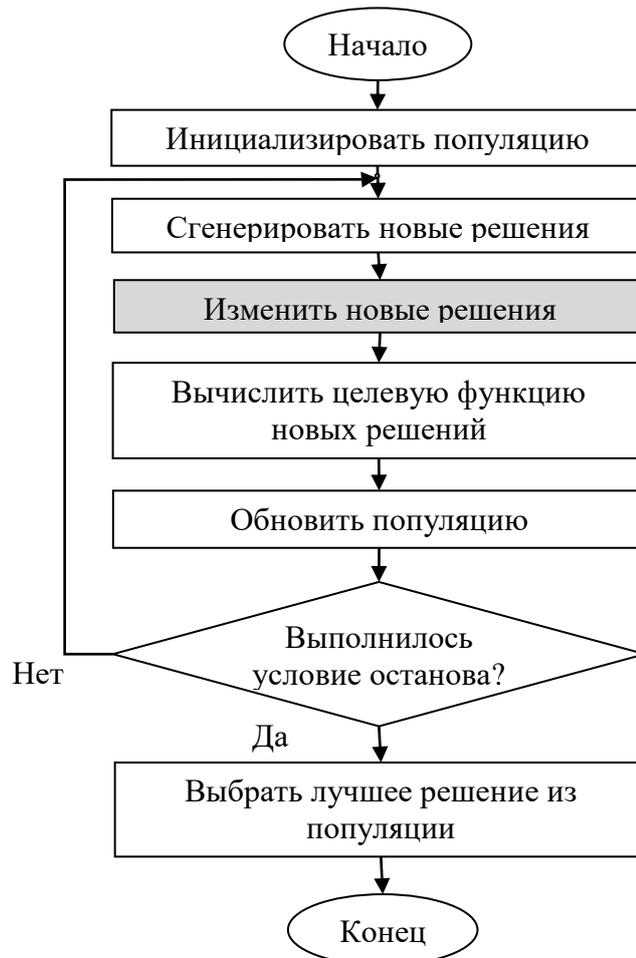


Рисунок 2.15 – Схема метаэвристического популяционного алгоритма, применяемого для разделения данных

Вектор решений оптимизации. Как уже было указано выше, оптимизация проводится в бинарном пространстве поиска, когда переменные могут принимать только два дискретных значения. Условимся, что такими значениями будут 0 и 1. С каждой переменной связан образец исходной выборки, и если значение переменной 0, то соответствующий образец будет включен в обучающую выборку, а если 1, то в валидационную. Количество элементов в векторе будет соответствовать числу образцов в исходной выборке данных. Пример вектора решений представлен на рисунке 2.16.

Образец 1	Образец 2	Образец 3	Образец 4	Образец 5	Образец 6	Образец 7	...	Образец t
1	1	0	0	1	1	0	...	0

Рисунок 2.16 – Вектор поиска метода разделения данных

Изменения вектора решений. Изменения, которыми должны подвергаться векторы решений популяции, касаются ограничений, накладываемым на обучающие и валидационные выборки. Ограничения связаны с удовлетворением отношения между количеством образцов выборок. Отношение должно быть фиксированным, и задается до начала выполнения метода. В процессе работы метаэвристического алгоритма получаемые сгенерированные векторы решений могут не удовлетворять ограничениям, поэтому необходима их корректировка. Она проводится на данном этапе метода.

Пусть необходимо получить выборки с отношением образцов обучающих и валидационных данных как $a:b$. Сначала определим количество образцов в каждой из выборок, которые необходимо получить. Обозначим буквой c количество образцов в обучающей выборке, тогда количество образцов в валидационной будет $m-c$, где m количество образцов в исходной выборке. Необходимо чтобы соотношение образцов выборок удовлетворяло следующему условию:

$$\frac{c}{m-c} = \frac{a}{b}.$$

Выразим из вышеприведенного выражения c и округлим его значение. Получим необходимое количество образцов для обучения:

$$c = \text{round}\left(\frac{am}{a+b}\right).$$

Если выборки предполагаются стратифицированные, т.е. отношение между числом образцов классов в полученных выборках должно соответствовать исходной выборке, то необходимо определить количество обучающих и валидационных образцов каждого класса:

$$c_i = \text{round}\left(\frac{am_i}{a+b}\right),$$

где c_i – количество образцов i -го класса в обучающей выборке, m_i – всего количество образцов i -го класса в исходном наборе данных. Количество валидационных образцов i -го класса будет соответствовать значению $m_i - c_i$.

Далее проводится корректировка значений векторов популяции таким образом, чтобы число элементов, имеющих значение 0 (обучающие образцы), соответствовало c , а число элементов, имеющих значение 1 (валидационные образцы), соответствовало $m - c$. Обозначим r число элементов вектора решений, имеющих значение 0. Процедура корректировки будет выглядеть следующим образом.

Начало

Шаг 1. Если число элементов, имеющих значение 0, равно r и $r > c$, то случайным образом выбирается k -й элемент вектора со значением 0, и его значение меняется на 1:

$$X_k = 1, k = \text{rand}\{p_1^1, p_2^1, \dots, p_r^1\},$$

где p_1^1, \dots, p_r^1 – индексы элементов со значением 1. Если $r < c$, то случайным образом выбирается элемент вектора со значением 1 и его значение меняется на 0:

$$X_k = 0, k = \text{rand}\{p_1^0, p_2^0, \dots, p_r^0\},$$

где p_1^0, \dots, p_r^0 – индексы элементов со значением 0.

Шаг 2. Если $r = c$, то завершить работу, иначе перейти на *Шаг 1*.

Конец

В случае стратифицированных выборок вышеприведенная процедура корректировки должна быть проделана отдельно для каждого множества элементов вектора решений, связанных с образцами только одного класса. Таким образом, процедура проводится столько раз, сколько классов имеют образцы исходной выборки. Ниже представлена измененная процедура корректировки в случае стратифицированной выборки.

Начало

Шаг 1. Инициализировать счетчик классов $i = 1$.

Шаг 2. Если число элементов класса ω_i , имеющих значение 0, равно r_i , и $r_i > c_i$, то случайным образом выбирается k -й элемент вектора со значением 0, и его значение меняется на 1:

$$X_k = 1, k = \text{rand}\{p_1^{1,i}, p_2^{1,i}, \dots, p_r^{1,i}\},$$

где $p_1^{1,i}, \dots, p_r^{1,i}$ – индексы элементов класса ω_i со значением 1. Если $r_i < c_i$, то случайным образом выбирается элемент вектора решений класса ω_i со значением 1, и его значение меняется на 0:

$$X_k = 0, k = \text{rand}\{p_1^{0,i}, p_2^{0,i}, \dots, p_r^{0,i}\},$$

где $p_1^{0,i}, \dots, p_r^{0,i}$ – индексы элементов со значением 0.

Шаг 3. Если $r_i = c_i$, то перейти на *Шаг 4*, иначе перейти на *Шаг 2*.

Шаг 4. Если все классы рассмотрены, то завершить работу, иначе $i = i + 1$ и перейти на *Шаг 2*.

Конец

Целевая функция оптимизации. Целью работы метода является разделение исходной выборки на обучающую и валидационную с сохранением близости полученных выборок к исходной без потери важной для обучения и валидации информации. Понятие близости в данной работе связано со схожими характеристиками признаков данных. В качестве меры близости предлагается использовать отклонение значений характеристик выборок, полученных в ходе работы метода, от исходной выборки данных. Такими характеристиками могут выступать статистические характеристики, например, среднее значение, среднеквадратичное отклонение (таблица 2.1).

Таблица 2.1 – Статистические характеристики признаков

Название характеристики признака	Формула
Среднее значение	$\mu = \frac{1}{m} \sum_{i=1}^m x_i$
Среднеквадратичное отклонение	$\sigma = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_i - \mu)^2}$

Поскольку целью работы метода является сохранение близости полученных выборок к исходной выборке, целевая функция должна количественно оценивать эту близость. Одним из вариантов меры близости выборок, а также целевой функции, является наибольшее суммарное отклонение значений характеристик признаков:

$$\varphi_1 = \max_{j=1, \dots, n} \left(\sum_{i=1}^{nf} \beta_i \frac{|F_i(X_j^{tra}) - F_i(X_j)| + |F_i(X_j^{val}) - F_i(X_j)|}{\Delta_{ij}^{\max}} \right).$$

В вышеприведенной формуле X_j , X_j^{tra} , X_j^{val} – это множества значений j -й переменной исходной, обучающей и валидационной выборок соответственно ($j=1, \dots, n$), $F_i(X)$ – i -я характеристика множества X ($i=1, \dots, nf$). Чтобы отклонения были одного порядка, их значения нормируются в диапазон $[0, 1]$ путем деления на максимальное значение Δ_{ij}^{\max} , которое может принять отклонение i -й характеристики j -й переменной. Коэффициенты β_i принимают значение из диапазона $[0, 1]$ и определяет важность близости i -й характеристики признака, определяется до начала работы метода. Меньшее значение целевой функции соответствуют лучшему значению близости, поэтому при нахождении выборок необходимо, чтобы данная функция была минимизирована.

Другой вариант целевой функции может быть сосредоточен только на увеличении близости обучающей выборки к исходной. За счет исключения близости валидационной выборки может быть повышена близость обучающей, и тем самым построена модель на данных, приближенных к исходным. Такой

вариант целевой функции, в которой учитывается только близость обучающей выборки, представлен ниже:

$$\varphi_2 = \max_{j=1, \dots, n} \left(\sum_{i=1}^{nf} \beta_i \frac{|F_i(X_j^{tra}) - F_i(X_j)|}{\Delta_j^{\max}} \right).$$

В приведенных целевых функциях рассчитывается максимальное суммарное отклонение по значениям характеристик признаков. Функции возвращают наихудший вариант отклонения характеристик одного из признаков и являются строгими. Если такое худшее значение не может быть улучшено, то работа алгоритма оптимизации не будет сосредотачиваться на уменьшении отклонения характеристик других признаков. Исходя из вышеприведенных рассуждений, еще двумя вариантами целевых функций могут быть модификации указанных выше, в которых максимальное суммарное отклонение по значениям признаков заменено на среднее значение. Такие варианты представлены ниже для функций φ_1 и φ_2 соответственно:

$$\varphi_3 = \text{mean}_{j=1, \dots, n} \left(\sum_{i=1}^{nf} \beta_i \frac{|F_i(X_j^{tra}) - F_i(X_j)| + |F_i(X_j^{val}) - F_i(X_j)|}{\Delta_j^{\max}} \right),$$

$$\varphi_4 = \text{mean}_{j=1, \dots, n} \left(\sum_{i=1}^{nf} \beta_i \frac{|F_i(X_j^{tra}) - F_i(X_j)|}{\Delta_j^{\max}} \right).$$

В настоящем исследовании в качестве характеристик признаков будут использованы среднее значение и среднеквадратичное отклонение. Для нормирования отклонений этих величин в целевой функции необходимо разделить их на максимально возможное значение, которое они могут принять. В случае отклонения средних значений, это будет разность между максимальным и минимальным значениями переменной, поскольку выполняются следующие условия:

$$\left| \mu(X_j^{tra}) - \mu(X_j) \right| \leq (x_j^{\max} - x_j^{\min}),$$

$$\left| \mu(X_j^{val}) - \mu(X_j) \right| \leq (x_j^{\max} - x_j^{\min}),$$

где x_j^{\max} и x_j^{\min} максимальное и минимальное значения множества X_j .

Для среднеквадратичного отклонения будут справедливы условия:

$$\left| \sigma(X_j^{tra}) - \sigma(X_j) \right| \leq \sqrt{\frac{1}{m} \sum_{i=1}^m (x_j^{\max} - x_j^{\min})^2},$$

$$\left| \sigma(X_j^{val}) - \sigma(X_j) \right| \leq \sqrt{\frac{1}{m} \sum_{i=1}^m (x_j^{\max} - x_j^{\min})^2}.$$

После алгебраических преобразований вышеприведенных неравенств, получаем следующее:

$$\left| \sigma(X_j^{tra}) - \sigma(X_j) \right| \leq (x_j^{\max} - x_j^{\min}),$$

$$\left| \sigma(X_j^{val}) - \sigma(X_j) \right| \leq (x_j^{\max} - x_j^{\min}).$$

Таким образом, для среднего значения и среднеквадратичного отклонения нормировка будет проводиться с помощью деления на разность между максимальным и минимальным значениями переменной:

$$\Delta_{ij}^{\max} = x_j^{\max} - x_j^{\min},$$

где $i = 1, 2, j = 1, \dots, n$.

С учетом выбранных характеристик получим выражения для вариантов целевых функций $\varphi_1 - \varphi_4$. Меньшее значение функций говорит об увеличении близости, таким образом, целью оптимизации является минимизация одного из вариантов целевой функций:

$$\varphi_1 = \max_{j=1, \dots, n} \left(\beta_1 \left(\frac{|\mu(X_j^{tra}) - \mu(X_j)| + |\mu(X_j^{val}) - \mu(X_j)|}{x_j^{\max} - x_j^{\min}} \right) + \dots \right. \\ \left. \dots + \beta_2 \left(\frac{|\sigma(X_j^{tra}) - \sigma(X_j)| + |\sigma(X_j^{val}) - \sigma(X_j)|}{x_j^{\max} - x_j^{\min}} \right) \right), \quad (2.9)$$

$$\varphi_2 = \max_{j=1, \dots, n} \left(\beta_1 \frac{|\mu(X_j^{tra}) - \mu(X_j)|}{x_j^{\max} - x_j^{\min}} + \beta_2 \frac{|\sigma(X_j^{tra}) - \sigma(X_j)|}{x_j^{\max} - x_j^{\min}} \right), \quad (2.10)$$

$$\varphi_3 = \text{mean}_{j=1, \dots, n} \left(\beta_1 \left(\frac{|\mu(X_j^{tra}) - \mu(X_j)| + |\mu(X_j^{val}) - \mu(X_j)|}{x_j^{\max} - x_j^{\min}} \right) + \dots \right. \\ \left. \dots + \beta_2 \left(\frac{|\sigma(X_j^{tra}) - \sigma(X_j)| + |\sigma(X_j^{val}) - \sigma(X_j)|}{x_j^{\max} - x_j^{\min}} \right) \right), \quad (2.11)$$

$$\varphi_4 = \text{mean}_{j=1, \dots, n} \left(\beta_1 \frac{|\mu(X_j^{tra}) - \mu(X_j)|}{x_j^{\max} - x_j^{\min}} + \beta_2 \frac{|\sigma(X_j^{tra}) - \sigma(X_j)|}{x_j^{\max} - x_j^{\min}} \right). \quad (2.12)$$

Для того чтобы определить, какой вариант целевой функции наиболее эффективен, проведен вычислительный эксперимент, представленный в подразделе 3.3. Предложенный метод разделения формирует обучающие и валидационные выборки для построения нечетких классификаторов с помощью метаэвристического алгоритма смешанной многокритериальной оптимизации. Классификаторы строились для каждого варианта целевой функции, после чего полученные классификаторы сравнивались на предмет точности классификации. Эффективность целевой функции определялась точностью построенных моделей.

Выбор модели. Алгоритм построения моделей создает множество моделей, имеющих различную структурную сложность. С точки зрения прогностической способности выбор эффективной модели будет заключаться в оценке критерия

эффективности каждой из них. Модель, показывающая лучший результат эффективности, будет выбрана в качестве результирующей. При этом предлагается два варианта выборки данных для выбора модели. Первый использует только валидационные данные. Второй вариант использует исходный набор, т.е. включает как обучающую, так и валидационную выборки. Сравнение эффективности этих вариантов для построения нечетких классификаторов приведено в разделе 3 с вычислительными экспериментами (подраздел 3.3).

В настоящем исследовании метод разделения применяется для построения нечеткого классификатора метаэвристическим алгоритмом смешанной многокритериальной оптимизации и входит в методику построения интерпретируемых нечетких классификаторов, представленной в подразделе 2.7. В качестве стратегии выбора модели из сформированного Парето-множества выбирается модель с лучшей точностью классификации на валидационной выборке. В экспериментальной части исследования (подраздел 3.3) наряду с четырьмя вариантами целевой функций алгоритма бинарной оптимизации применены два варианта выборок данных для выбора классификатора на фронте Парето. Таким образом, исследовано восемь вариантов работы предлагаемого метода для построения нечетких классификаторов. С помощью применения статистических тестов определены вид целевой функции и выборка данных, при которых достигается лучшая точность.

2.7 Методика построения интерпретируемых нечетких классификаторов

В подразделе представлена методика построения интерпретируемых нечетких классификаторов [256]. Методика разработана на основе предложенной методологии (см. подраздел 2.1) с применением алгоритмов оптимизации и метода разделения данных, представленных в данном разделе.

Исходными данными методики является размеченная выборка данных $Data = \{(\mathbf{x}_p, l_p) \mid p=1, \dots, m\}$, где каждый образец характеризуется набором числовых

признаков $\mathbf{x}_p=(x_{1p}, \dots, x_{np})$ и имеет целевое значение l_p – метку класса образца (см. подраздел 1.6). Результатом выполнения методики является база нечетких правил, формирующая построенный нечеткий классификатор (рисунок 2.17).

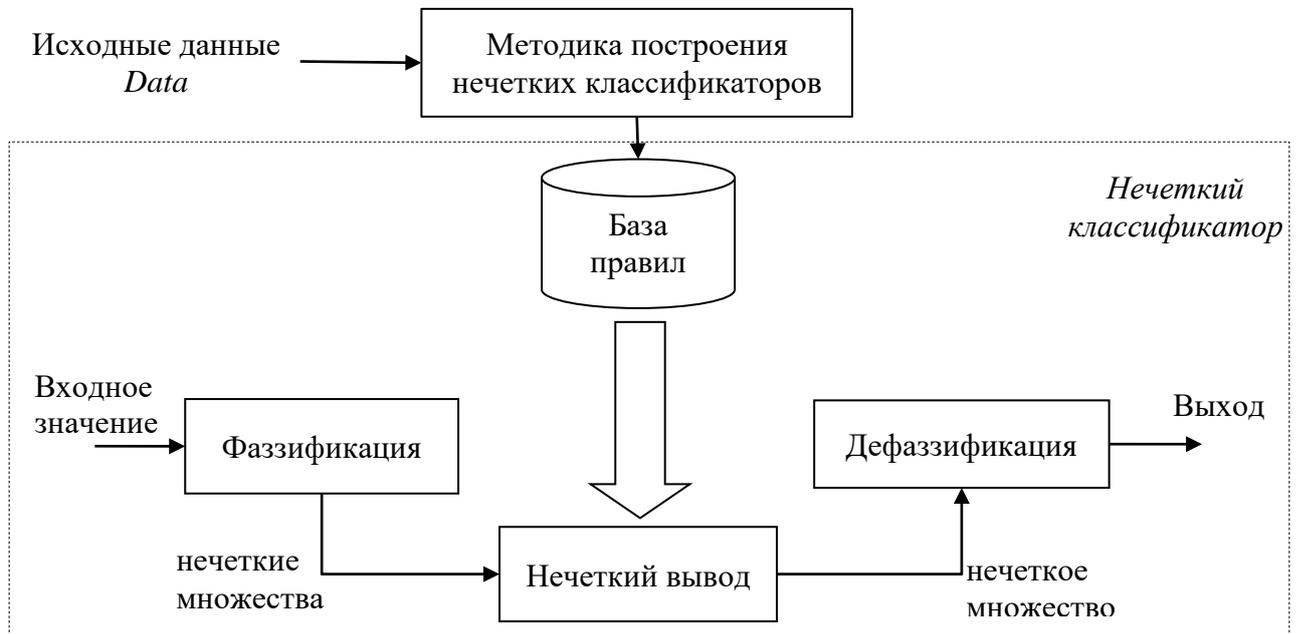


Рисунок 2.17 – Структура нечеткого классификатора, построенного предлагаемой методикой

Сначала данные подвергаются нормированию путем сведения значений всех признаков к интервалу $[0, 1]$. Далее применяется метод разделения данных, представленный в подразделе 2.6, для получения обучающей (tra) и валидационной (val) выборок, близких по статистическим характеристикам к исходной $Data$. В качестве алгоритма бинарной оптимизации для разделения данных в методе используется дискретный алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений, представленный в подразделе 2.4. После разделения данных проводится построение недоминируемого по критериям точности и интерпретируемости множества нечетких классификаторов с помощью метаэвристического алгоритма смешанной многокритериальной оптимизации, представленного в подразделе 2.3. После выбора подходящего классификатора проводится оптимизация параметров функций принадлежности адаптированным алгоритмом градиентного спуска,

который представлен в подразделе 2.5. На рисунке 2.18 показана схема предлагаемой методики. Детальное рассмотрение шагов описано ниже.

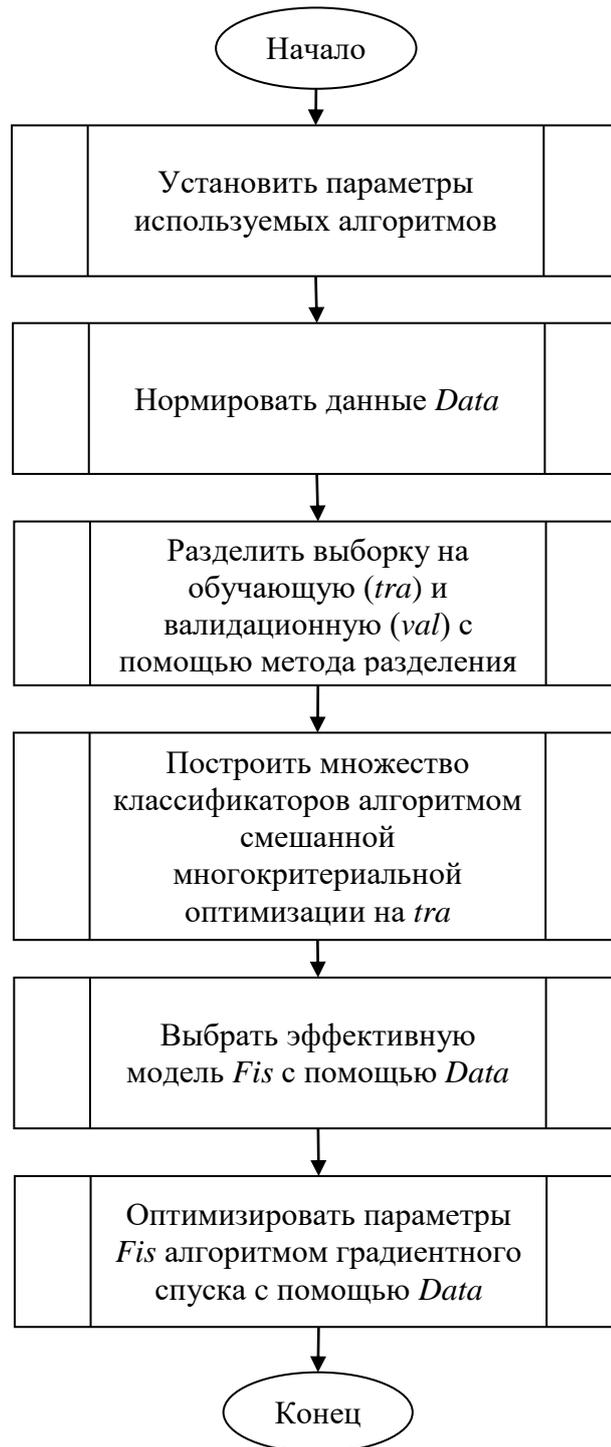


Рисунок 2.18 – Схема методики построения нечеткого классификатора

Установка параметров алгоритмов. На данном этапе определяются число лингвистических терминов для каждой переменной $\mathbf{h} = (h_1, h_2, \dots, h_n)$, и максимальное количество правил в классификаторе r . Проводится установка

параметров алгоритма смешанной многокритериальной оптимизации для построения классификаторов, дискретного алгоритма для разделения данных и модифицированного алгоритма градиентного спуска для настройки параметров функций принадлежности. В экспериментальной части работы приведены рекомендуемые значения параметров.

Нормирование данных. На данном этапе проводится корректировка значений входных данных, называемая нормированием. Проведение нормирования необходимо, поскольку значения признаков могут изменяться в очень большом диапазоне и отличаться друг от друга на несколько порядков. Такой дисбаланс может присутствовать как в значениях между признаками, так и в пределах одного признака. Это может вызвать неустойчивость работы прогностической модели и ухудшить результаты прогноза [156]. Значения всех признаков приводятся к интервалу $[0, 1]$ с помощью следующего преобразования

$$x_i^* = \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}},$$

где x_i – значение i -го признака образца, x_i^{\min} , x_i^{\max} минимальное и максимальное значение i -го признака в исходном наборе данных $Data$, $i=1, \dots, n$.

Разделение исходной выборки на обучающую и валидационную. Нормированные данные разделяются на обучающую и валидационную выборки с помощью метода разделения данных. Дискретный алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений применяется в методе для разделения данных. Алгоритм при поиске в бинарном пространстве показал статистически значимое улучшение показателей сходимости и стабильности на тестовых функциях по сравнению с генетическим алгоритмом и алгоритмом роящихся частиц. В экспериментальной части исследования (подраздел 3.3) показано, что вариант целевой функции (2.10) при разделении исходной выборки продемонстрировал улучшение точности построения классификаторов, поэтому он применяется в дискретном алгоритме.

Построение нечеткого классификатора с помощью многокритериального алгоритма смешанной оптимизации. На данном этапе формируется множество нечетких классификаторов с помощью алгоритма смешанной многокритериальной оптимизации. Алгоритм позволяет проводить поиск в смешанном пространстве, когда переменные могут принимать дискретные и непрерывные значения. Благодаря этому вектор поиска может одновременно кодировать структуру и параметры классификатора, и таким образом в процессе оптимизации осуществляется построение структуры и настройка параметров.

Исходными данными алгоритма являются обучающие данные tra . На выходе – полученные алгоритмом классификаторы, которые образуют множество недоминируемых решений по критериям точности и интерпретируемости. Далее будет показано, каким образом построение классификаторов сводится к задаче смешанной многокритериальной оптимизации.

Разделение пространства признаков на нечеткие множества.

Пространство каждого признака разделяется на нечеткие множества, количество которых определяется соответствующим элементом вектора \mathbf{h} . С каждым нечетким множеством ассоциируется нечеткий термин, употребляемый в базе правил для оценки значения признака.

В методике для нечетких множеств применяются функции принадлежности гауссова типа (1.3) с параметрами среднего значения s и отклонения σ функции. Начальные значения параметров равномерно покрывают признаковое пространство. Если пространство i -й переменной разбито на h_i нечетких множеств, то параметры функции j -го множества определяются согласно следующим выражениям:

$$s_{ij} = \frac{(x_i^{\max} - x_i^{\min})(j-1)}{h_i - 1}, \sigma_{ij} = \frac{(x_i^{\max} - x_i^{\min})}{3(h_i - 1)}, \quad (2.13)$$

где $i = 1, \dots, n$, $j = 1, \dots, h_i$, x_i^{\min} и x_i^{\max} минимальное и максимальное значения i -го признака исходных данных. Параметр алгоритма h_i , устанавливающий количество нечетких множеств при разделении пространства i -й переменной, задается до начала запуска. На рисунке 2.19 показаны функции принадлежности в случае деления пространства на три нечетких множества. За каждой функцией закреплен лингвистический термин «Малое», «Среднее» или «Большое», указывающий соответствующую интерпретацию нечеткого множества.

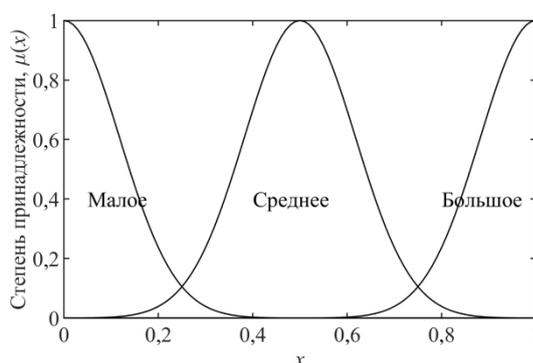


Рисунок 2.19 – Равномерное разбиение пространства признаков нечеткими множествами с функциями принадлежности гауссова типа

Боковое смещение и смещение охвата.

В настоящей работе для построения компактного и точного классификатора метаэвристический алгоритм будет использовать боковое смещение и смещение охвата функций принадлежности. В первом случае подбирается смещение параметра среднего значения функции принадлежности относительно равномерного разбиения, во втором случае – смещение отклонения. Такой подход, при малых смещениях от равномерного разбиения, позволит достичь лучшей точности по сравнению с равномерным разбиением, без существенной потери интерпретируемости.

В [257] вводится параметр α для функций принадлежности, имеющих боковое смещение относительно равномерного разбиения. С учетом этого параметра введено обозначение нечетких множеств, называемое двухкортежным лингвистическим нечетким представлением. Согласно этому обозначению, нечеткое множество может быть представлено парой (S, α) , где S – нечеткое

множество равномерного разбиения, а α – боковое смещение функции принадлежности. Значение α определяет долю расстояния до соседних центров функций принадлежности равномерного разбиения. На рисунке 2.20 показано смещение $\alpha = -0,5$ для нечеткого множества «Среднее».

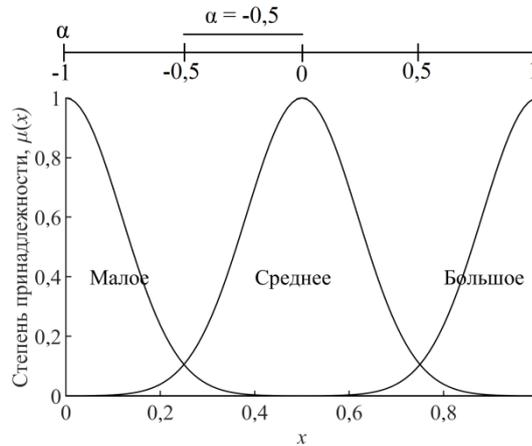


Рисунок 2.20 – Боковое смещение для нечеткого множества «Среднее»

Чтобы определить параметр среднего значения функции принадлежности нечеткого множества (S_{ij}, α_{ij}) необходимо проделать следующее преобразования:

$$s_{ij}^* = s_{ij} + \alpha_{ij} \frac{x_i^{\max} - x_i^{\min}}{h_i - 1},$$

где α_{ij} – боковое смещение параметра среднего для j -й функции принадлежности i -й переменной.

Преобразуем s_{ij} согласно (2.13) и получим следующее выражение:

$$s_{ij}^* = \frac{x_i^{\max} - x_i^{\min}}{h_i - 1} (j - 1 + \alpha_{ij}). \quad (2.14)$$

Помимо бокового смещения функций принадлежности в настоящей работе предлагается использовать смещение охвата для подбора значения среднеквадратичного отклонения. Для этого введем параметр β , который показывает долю от разности среднеквадратичных отклонений при h_i и h_i-1 равномерных разделений на нечеткие множества. Для определения нового

значения отклонения при заданном смещении β проводятся следующие преобразования:

$$\sigma_{ij}^* = \sigma_{ij} + \beta_{ij} \left(\frac{x_i^{\max} - x_i^{\min}}{3(h_i - 1)} - \frac{x_i^{\max} - x_i^{\min}}{3h_i} \right).$$

Преобразуем σ_{ij} согласно (2.13) и получим следующее выражение:

$$\sigma_{ij}^* = \frac{x_i^{\max} - x_i^{\min}}{3} \left(\frac{1 + \beta_{ij}}{h_i - 1} - \frac{\beta_{ij}}{h_i} \right) \quad (2.15)$$

Оба параметра α и β при малых значениях инициируют малые сдвиги, что не существенно отражается на потере интерпретируемости. В настоящей работе оба смещения изменяются в диапазоне $[-0,3, 0,3]$. С помощью выражений (2.14) и (2.15) они преобразуются в параметры функции принадлежности. На рисунке 2.21 показаны боковые смещения функции принадлежности нечеткого множества «Среднее» для $\alpha = -0,3$ и $\alpha = 0,3$, а также смещения охвата для $\beta = -0,3$ и $\beta = 0,3$.

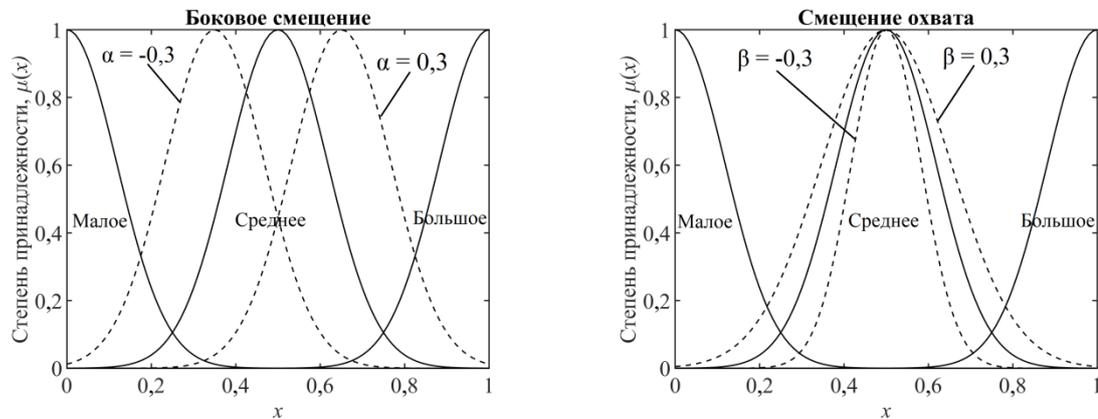


Рисунок 2.21 – Изменения функции принадлежности нечеткого множества «Среднее» при боковом смещении и смещении охвата

Кодирование классификатора.

Применение метаэвристического алгоритма предполагает векторное кодирование классификатора. Предлагаемая кодировка учитывает как структурные свойства, так и параметрические. Вектор поиска разделен на две части – целочисленную и вещественную (структура вектора представлена на

рисунке 2.22). В целочисленной части представлена структура классификатора. Каждый элемент части содержит целочисленный код нечеткого множества согласно равномерному разделению. Также помимо нечеткого множества, элемент вектора может содержать код, обозначающий отсутствие переменной в правиле. Будем обозначать это кодом 0. В случае деления пространства признака на три нечетких множества (рисунок 2.19), элемент вектора принимает значение из множества $\{0, 1, 2, 3\}$, где «0» говорит об отсутствии переменной в правиле, «1» соответствует термину «Малое», «2» – «Среднее», «3» – «Большое». Элементы разбиты на группы, представляющие одно правило. Порядок нечетких множеств в группах следует порядку переменных образцов данных. Обозначение μ_{ij} на рисунке соответствует коду функции принадлежности i -й переменной j -го правила, где $i = 1, \dots, n, j = 1, \dots, r$. Параметр r указывает на максимально возможное число правил в классификаторе и задается до запуска алгоритма. Размер целочисленной части соответствует $n \cdot r$ элементам.

Вещественная часть вектора решений содержит боковые смещения и смещения охвата каждого нечеткого множества пространства признаков. Элементы со смещениями чередуют друг друга и разбиты на группы параметров, представляющие нечеткие множества одного уровня равномерного деления (рисунок 2.22). Обозначение α_{ij} и β_{ij} на рисунке соответствуют боковому смещению и смещению охвата i -й переменной j -го нечеткого множества, где $i = 1, \dots, n, j = 1, \dots, h_i$. Размер вещественной части составляет $2(h_1+h_2+\dots+h_n)$ элементов. Размер всего вектора решения составляет $(n \cdot r + 2(h_1+h_2+\dots+h_n))$ элементов.

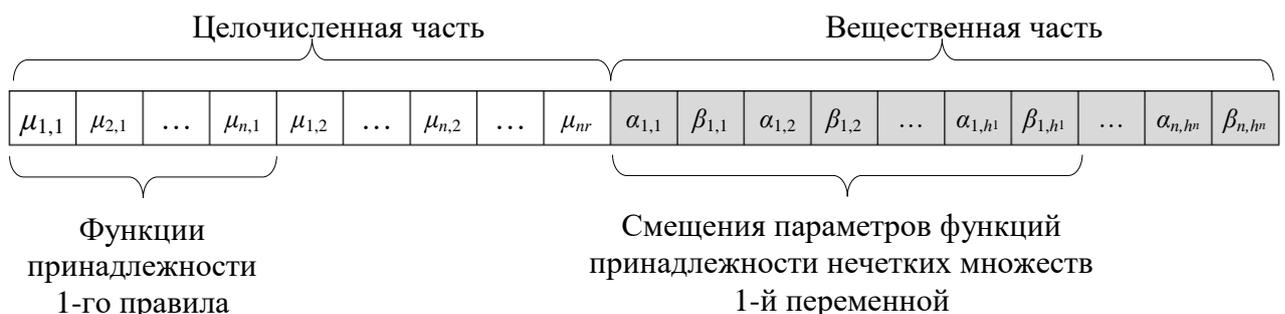


Рисунок 2.22 – Структура вектора поиска алгоритма смешанной оптимизации

Консеквентная часть правила, определяющая класс, не присутствует в векторе поиска, поскольку может быть вычислена на основе функций принадлежности antecedentной части и обучающих экземпляров данных. В [83, 258] представлена эвристика, с помощью которой определяется класс правила. Введем понятие классового веса правила, под которым подразумевается суммарная степень принадлежности образцов обучающей выборки одного класса нечеткому множеству, образованному antecedentом правила:

$$\eta_k = \sum_{l_p="o_k"} \prod_{i=1}^n \mu_i(x_{ip}),$$

где $k = 1, \dots, M$, M – число классов, μ_i – функция принадлежности i -й переменной в правиле, x_{ip} – значение i -й переменной образца \mathbf{x}_p обучающих данных tra , а l_p соответствующая метка класса. Класс правила определяется максимальным значением классового веса:

$$c = "o_j", \text{ где } j = \arg \max_{k=1, \dots, M} \{\eta_k\}.$$

Целочисленная часть вектора решений, представляющая структуру классификатора, может быть интерпретирована в базу правил. На рисунке 2.23 показан пример такой интерпретации. Проблемная область содержит два признака $n = 2$. Максимальное количество правил установлено в значение 4 ($r = 4$). Размер целочисленной части имеет $2 \cdot 4 = 8$ элементов. Количество нечетких множеств, на которое разбито пространство признаков, установлено в значение 3 ($h = 3$). Образованные нечеткие множества, представленные на рисунке 2.19, имеют следующие коды «Малое» = 1, «Среднее» = 2, «Большое» = 3. Отсутствие нечеткого множества, оценивающего переменную, кодируется 0. Фактически такая ситуация говорит об отсутствии переменной в правиле. Показанная на рисунке целочисленная часть вектора решений кодирует базу из трех правил, хотя размер рассчитан на 4 правила. На месте переменных одного из правил в векторе присутствуют коды 0, что говорит об отсутствии потенциального правила.

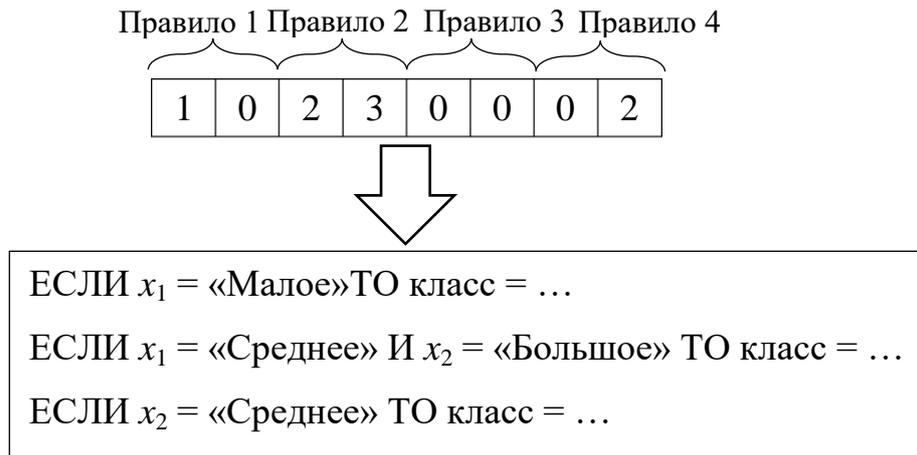


Рисунок 2.23 – Интерпретация базы правил при целочисленном кодировании

Диапазон изменения значений элементов вектора решения.

Границы, в пределах которых должны варьироваться элементы вектора решения $\mathbf{V}=(v_1, v_2, \dots, v_{n \cdot r + 2(h_1 + \dots + h_n)})$, определяются количеством нечетких множеств h_i , на которые разделено пространство признаков, а также диапазоном изменений бокового смещения α и смещения охвата β . На основании вышеизложенного, элементы вектора v_i , где $i = 1, 2, \dots, n \cdot r$, принимают целочисленные значения из множества $\{0, 1, \dots, h_j\}$, где

$$j = \begin{cases} n, & \text{если } \text{mod}(i, n) = 0 \\ \text{mod}(i, n), & \text{иначе} \end{cases},$$

а элементы с $n \cdot r + 1$ по $(n \cdot r + 2(h_1 + h_2 + \dots + h_n))$ принимают вещественные значения и ограничены диапазоном $[-0,3, 0,3]$.

Постановка задачи оптимизации.

Для того чтобы обеспечить компромисс в построенных классификаторах между интерпретируемостью и точностью, в работе алгоритма смешанной многокритериальной оптимизации предлагается использовать три целевых значения, а именно точность классификации, определяющей процент правильно классифицированных образцов, количество правил и общее количество используемых в базе правил нечетких терминов. Первый критерий должен быть максимизирован, два последних минимизированы. Таким образом, построение

классификатора сводится к задаче многокритериальной смешанной оптимизации следующего вида:

$$\left\{ \begin{array}{l} acc(\mathbf{V}) \rightarrow \max \\ \#R(\mathbf{V}) \rightarrow \min \\ \#C(\mathbf{V}) \rightarrow \min \\ v_i \in \{0,1,\dots,h_j\}, 1 \leq i \leq nr, j = \begin{cases} n, \text{если } \text{mod}(i,n) = 0 \\ \text{mod}(i,n), \text{иначе} \end{cases} \\ v_i \in [-0,3,0,3], nr + 1 \leq i \leq 2(nr + h_1 + \dots + h_n) \end{array} \right. ,$$

где \mathbf{V} вектор решения, $acc(\mathbf{V})$, $\#R(\mathbf{V})$ и $\#C(\mathbf{V})$ – точность, количество правил и общее количество нечетких терминов классификатора, закодированного вектором \mathbf{V} , соответственно. Представленная выше задача решается с помощью алгоритма кукушкин поиск с генетическим оператором скрещивания, предложенным в подразделе 2.3. Алгоритм возвращает множество недоминируемых по трем целям решений, представляющих нечеткие классификаторы.

Модификация решений в алгоритме оптимизации.

Для того чтобы повысить интерпретируемость и точность классификаторов, проведем модификацию алгоритма таким образом, чтобы полученные правила содержали не более трех нечетких терминов. Для этого в генерируемых алгоритмом решениях проводятся следующие изменения. После генерации новой популяции в каждой группе элементов целочисленной части решения, образующих правило, случайным образом выбираются три элемента, среди элементов не равных 0. Если таких элементов в группе три и менее, то модификация не проводится, иначе те элементы, которые оказались не выбранными, принимают значение 0, а значение остальных не меняются.

Выбор эффективной модели. Из полученного множества классификаторов выбирается классификатор, имеющий лучшую точность на выборке из обучающих и валидационных образцов, т.е. исходных данных *Data*. Именно такое

сочетание образцов показало наилучшую точность в ходе проведения эксперимента (см. подраздел 3.3).

Оптимизация параметров классификатора алгоритмом градиентного спуска. С помощью адаптированного алгоритма градиентного спуска выполняется настройка параметров функций принадлежности выбранного классификатора. При этом нужно обеспечить сохранность диапазона значений смещений α и β . Поскольку в алгоритме градиентного спуска проводится изменение параметров функций принадлежности, необходимо перевести диапазон изменения значений от смещений к данным параметрам. Используя преобразование (2.14), (2.15), нижние границы α и β в значении -0,3, а верхние в 0,3, получим нижние и верхние границы диапазона изменения параметров:

$$s_{ij}^{\min} = \frac{x_i^{\max} - x_i^{\min}}{h_i - 1} (j - 1,3), s_{ij}^{\max} = \frac{x_i^{\max} - x_i^{\min}}{h_i - 1} (j - 0,7),$$

$$\sigma_{ij}^{\min} = \frac{x_i^{\max} - x_i^{\min}}{3} \left(\frac{0,7}{h_i - 1} + \frac{0,3}{h_i} \right), \sigma_{ij}^{\max} = \frac{x_i^{\max} - x_i^{\min}}{3} \left(\frac{1,3}{h_i - 1} - \frac{0,3}{h_i} \right).$$

Используя вышеперечисленные границы и целевую функцию алгоритма градиентного спуска согласно выражению (2.2), сформулируем задачу оптимизации для настройки параметров функций принадлежности, которую решает адаптированный алгоритм:

$$\left\{ \begin{array}{l} E = \frac{1}{2} \sum_{p=1}^m \sum_{k=1}^c \left(\frac{g_k(\mathbf{x}_p, \mathbf{s}, \boldsymbol{\sigma})}{\sum_{k=1}^c g_k(\mathbf{x}_p, \mathbf{s}, \boldsymbol{\sigma})} - y_{pk} \right)^2 \rightarrow \min \\ s_{ij} \in \left[\frac{x_i^{\max} - x_i^{\min}}{h_i - 1} (j - 1,3), \frac{x_i^{\max} - x_i^{\min}}{h_i - 1} (j - 0,7) \right], i = 1, \dots, n, j = 1, \dots, h_i \\ \sigma_{ij} \in \left[\frac{x_i^{\max} - x_i^{\min}}{3} \left(\frac{0,7}{h_i - 1} + \frac{0,3}{h_i} \right), \frac{x_i^{\max} - x_i^{\min}}{3} \left(\frac{1,3}{h_i - 1} - \frac{0,3}{h_i} \right) \right], i = 1, \dots, n, j = 1, \dots, h_i \end{array} \right. ,$$

где g_k – значение разделяющей функции k -го класса ($k = 1, \dots, c$), \mathbf{x}_p – вектор значений признаков p -го образца обучающих данных ($p = 1, \dots, m$), y_{pk} – ожидаемое значение разделяющей функции k -го класса для p -го образца обучающих данных.

2.8 Краткие выводы по разделу

В разделе представлены новые научные результаты, полученные в рамках диссертационного исследования: методология построения нечетких классификаторов, алгоритмы оптимизации, метод разделения данных и методика построения нечетких классификаторов. Ниже приведены основные выводы по разделу.

1. Предложена методология построения нечетких классификаторов, основанных на правилах. Методология определяет совокупность и взаимодействие методов обработки и анализа данных для получения интерпретируемых моделей. Методология включает в себя методы предобработки данных, методы разделения данных, метаэвристические методы смешанной многокритериальной оптимизации, стратегии выбора решений на фронте Парето и методы оптимизации, основанные на производных.

2. Разработан гибридный алгоритм смешанной многокритериальной оптимизации, основанный на метаэвристике «кукушкин поиск» и генетическом операторе скрещивания. Для поиска в дискретном пространстве применяется генетический оператор, а для поиска в непрерывном пространстве – стратегии метаэвристики. Для осуществления многокритериального поиска на каждой итерации алгоритма применяется недоминируемая сортировка к элементам популяции алгоритма. Таким образом, лучшие недоминируемые по критериям оптимизации решения остаются в популяции и передаются на следующую итерацию. В разделе с экспериментальным исследованием представлена работа предложенного алгоритма в сравнении с генетическим алгоритмом с

недоминируемой сортировкой NSGAII на модифицированных тестах ZDT и DTLZ со смешанными переменными.

3. Разработан новый метаэвристический алгоритм оптимизации для поиска оптимума в дискретном пространстве. В основе алгоритма лежит формирование распределений вероятностей для значений переменных вектора поиска с помощью трансформации целевых значений решений в их весовые коэффициенты. Данные распределения используются для генерации новых решений и формируют направление поиска. В разделе с экспериментальным исследованием представлена работа алгоритма в сравнении с генетическим алгоритмом и бинарным алгоритмом роящихся частиц. Для эксперимента использовались восемнадцать тестовых унимодальных и мультимодальных функций.

4. Адаптирован алгоритм градиентного спуска для оптимизации параметров функций принадлежности нечетких классификаторов. В качестве целевой функции используется среднеквадратичное отклонение степеней уверенности разделяющих функций. Для корректной работы алгоритма степени уверенности нормируются. Найдены аналитические выражения градиентов для параметров среднего значения и отклонения функций принадлежности гауссова типа. Предлагаемый адаптированный алгоритм применяется в методике построения интерпретируемых нечетких систем и используется для тонкой настройки параметров. В разделе с экспериментальным исследованием представлена работа алгоритма для настройки параметров классификаторов, построенных многокритериальным алгоритмом оптимизации.

5. Разработан метод разделения данных на обучающую и валидационную выборки для построения прогностических моделей, направленный на сохранение близости выборок к исходным данным. Под понятием близости подразумевается сохранение минимального отклонения характеристик признаков. В работе качестве таких характеристик применяются среднее значение и среднеквадратичное отклонение. Сохранение близости позволяет уменьшить потерю информации по сравнению со случайным разделением, и тем самым

повысить обобщающую прогностическую способность модели. Введены меры близости. С помощью обучающих данных осуществляется построение множества альтернативных моделей с различными по сложности структурными характеристиками, а с помощью валидационных выбирается наиболее точная модель. Разделение данных осуществляется алгоритмом оптимизации, осуществляющим поиск в бинарном пространстве. Предлагаемый метод применяется в методике построения интерпретируемых нечетких систем. В экспериментальной части исследования проведены эксперименты по построению нечетких классификаторов многокритериальным алгоритмом с разделением данных предлагаемым методом. Проведено сравнение построенных моделей с разделением случайным образом и без разделения данных.

6. На основе методологии построения нечетких классификаторов разработана методика, включающая в себя применение разработанных алгоритмов оптимизации, адаптированного алгоритма градиентного спуска и метода разделения данных. Представлена кодировка классификатора в виде вектора решений для смешанной оптимизации. Построение классификаторов осуществляется с помощью алгоритма смешанной многокритериальной оптимизации на основе метаэвристики «кукушкин поиск» и генетического оператора скрещивания. Разделение данных на обучающую и валидационную выборки осуществляется с помощью метода разделения данных с применением дискретного алгоритма оптимизации на основе распределения вероятностей с трансформацией целевых значений. Параметры функций принадлежности лучшего классификатора подвергаются настройке с помощью адаптированного алгоритма градиентного спуска. В разделе с экспериментальным исследованием представлены результаты сравнения эффективности построенных с помощью данной методики классификаторов с альтернативными моделями.

3 Вычислительные эксперименты и сравнение с аналогами

В разделе представлены вычислительные эксперименты разработанных алгоритмов, метода разделения данных и методики построения нечетких классификаторов. Цель проведения экспериментов заключалась в оценке их эффективности и выполнении сравнения с аналогами. Работа алгоритма смешанной многокритериальной оптимизации и дискретного алгоритма на основе распределения вероятностей оценивалась при поиске оптимума тестовых функций, используемых для проверки эффективности оптимизационных алгоритмов. Метод разделения данных использовался в построении нечетких классификаторов различных предметных областей с последующей оценкой точности и интерпретируемости полученных моделей. Адаптированный алгоритм градиентного спуска применялся для корректировки параметров нечетких классификаторов, предварительно построенных метаэвристическим алгоритмом смешанной многокритериальной оптимизации. Эффективность алгоритма оценивалась путем сравнения точности классификаторов до и после корректировки параметров. В заключении представлены эксперименты для проверки методики построения нечетких классификаторов, включающей в себя вышеприведенные алгоритмы и метод разделения. Для того чтобы оценить значимость различий результатов в сравнении с аналогами, применены статистические тесты.

Сравнение с аналогами алгоритма смешанной многокритериальной оптимизации и дискретного алгоритма оптимизации проводилось статистическими тестами согласно [259, 260]. Сравнение эффективности нечетких классификаторов проводилась согласно [261-263]. Реализация и вычислительные эксперименты алгоритмов, метода и методики осуществлялись в виде программных средств на языке MATLAB в среде программирования MATLAB R2022b.

3.1 Нахождение фронта Парето тестовых функций алгоритмом смешанной многокритериальной оптимизации

В настоящем подразделе представлен вычислительный эксперимент для оценки эффективности работы метаэвристического алгоритма смешанной многокритериальной оптимизации (см. подраздел 2.3). Для проведения экспериментов использована методика работы [182]. Алгоритм оценивался на наборах модифицированных тестовых функций ZDT [264] и DTLZ [265], в которые включены непрерывные и дискретные переменные. Для сравнения эффективности проведен эксперимент работы генетического алгоритма с недоминируемой сортировкой NSGAII [180] в тех же условиях методики.

Дискретные переменные. Наборы тестовых функций ZDT и DTLZ содержат непрерывные переменные. Для того чтобы использовать их в смешанной оптимизации, некоторая часть переменных изменена на дискретные путем квантования непрерывных значений. Количество дискретных переменных n_D в тестах эксперимента рассчитывалось как $n_D = \text{round}(n \cdot \theta)$, где n – общее количество переменных, а θ – доля дискретных переменных от этого количества. Первые n_D переменных функций принимают дискретные значения.

Каждая дискретная переменная может принять 21 значение из множества допустимых значений. В эксперименте использовались два варианта таких множеств – равномерное множество Q_1 и неравномерное Q_2 . Значения элементов Q_1 равномерно распределены по области изменения соответствующей непрерывной переменной. Если область изменения $[0, 1]$, то $Q_1 = \{0, 0,05, 0,1, \dots, 1\}$. Элементы неравномерного множества значений Q_2 для области изменения $[0, 1]$ вычисляются следующим образом:

$$Q_2(i) = \begin{cases} 0, & i = 1 \\ 0,01, & i = 2 \\ (i-1) \cdot (i-2) \cdot 0,0025 + 0,045, & 3 \leq i \leq 21 \end{cases} .$$

Наглядное изображение полученных множеств значений дискретных переменных на отрезке $[0, 1]$ показано на рисунке 3.1.

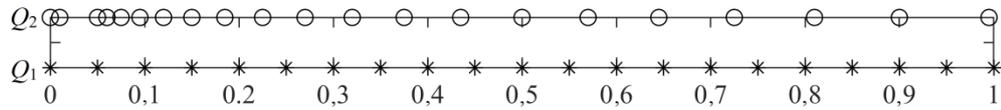


Рисунок 3.1 – Множества значений дискретных переменных

Критерии эффективности. Для оценки эффективности работы алгоритма применялись два критерия. Первый – обратное расстояние поколений (inverted generational distance, *IGD*) [266], применяемое для оценки работы алгоритмов многоцелевой оптимизации:

$$IGD = \frac{\sum_{v \in PF} d(v, U)}{|PF|},$$

где PF – множество точек целевого пространства, образующих фронт Парето и являющихся оптимумом тестовых функций; U – множество точек целевого пространства, соответствующих результату работы алгоритма; $d(v, U)$ – минимальное евклидово расстояние между точкой v и множеством точек U . *IGD* является комплексным оценочным критерием, который может отражать как сходимость, так и разнообразие результата алгоритма. Чем меньше *IGD*, тем выше эффективность.

В отличие от непрерывной оптимизации, фронт Парето смешанной оптимизации для тестовых функций *ZDT* и *DTLZ* будет содержать конечное значение точек. Количество точек на фронте будет зависеть от количества вариантов значений дискретных переменных, т.е. от $|Q_1|$ и $|Q_2|$, которое равно 21. В тестовых функциях *ZDT* два целевых значения, количество точек фронта равно значению 21. В тестах *DTLZ* три целевых значения, количество точек на фронте 21^2 . Значения элементов Q_1 и Q_2 подобраны таким образом, чтобы среди них присутствовали координаты оптимумов тестовых функций. На рисунке 3.2 показаны фронты Парето задач *ZDT2* и *DTLZ2*.

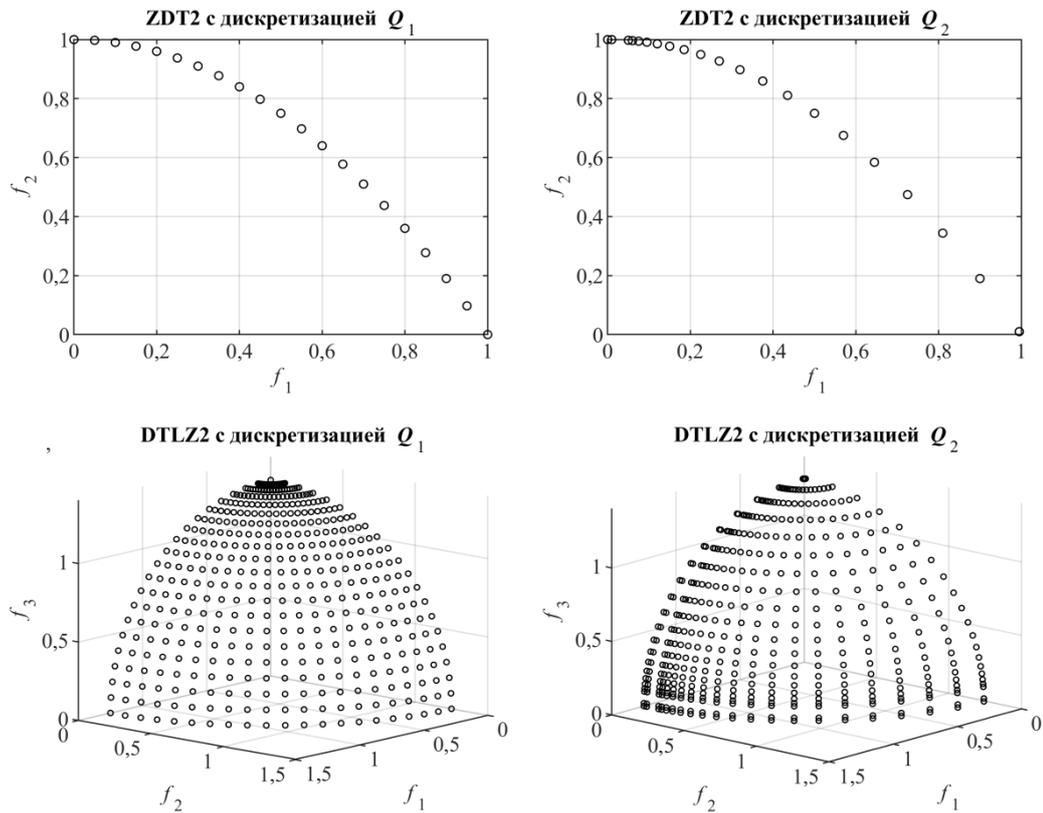


Рисунок 3.2 – Фронты Парето решений задач смешанной многокритериальной оптимизации ZDT2 и DTLZ2 с равномерным Q_1 и неравномерным Q_2 множествами значения дискретных переменных

Второй критерий – степень доминирования (dominate ratio, DR) [182]. Это относительный показатель, рассчитываемый при сравнении алгоритмов оптимизации. Допустим, сравниваются k алгоритмов многокритериальной смешанной оптимизации. Каждый алгоритм возвращает множество решений U_1, U_2, \dots, U_k . Объединим результаты оптимизации всех алгоритмов в одно множество $U = U_1 \cup U_2 \cup \dots \cup U_k$ и выделим из него множество недоминируемых решений U' . Степень доминирования i -го алгоритма определяет процентный состав решений U' входящих в U_i :

$$DR_i = \frac{|U_i \cap U'|}{|U'|} \cdot 100\%,$$

где $i = 1, \dots, k$. Фактически этот показатель определяет соотношение доминирования по Парето среди наборов решений всех сравниваемых алгоритмов. Чем больше значение, тем выше эффективность оцениваемого

алгоритма. DR также является комплексным оценочным критерием, который отражает не только сходимость, но и характеристики многообразия решений оцениваемого алгоритма. Когда приближенные фронты Парето двух алгоритмов перекрываются друг с другом, то DR вполне может сравнить их производительность, но, когда они не пересекаются, показатель не может точно оценить разницу в производительности между двумя алгоритмами.

Параметры вычислительного эксперимента. Для того чтобы адекватно оценить эффективность алгоритмов, их выполнение должно проводиться в одинаковых условиях. В связи с этим общие настройки алгоритмов должны быть одинаковыми. Размер популяции установлен в значение 220. Для тестов ZDT количество итераций имеет значение 100, количество целевых значений 2, количество переменных 30. Для тестов DTLZ вышеперечисленные параметры установлены в значения 1000, 3 и 12 соответственно. Область поиска всех переменных во всех задачах ограничивалась отрезком $[0, 1]$.

Параметры алгоритма MICS имели следующие значения. Коэффициент прыжка полетов Леви γ (выражение (2.1)) равномерно уменьшался на каждой итерации с 2,5 до 1,5 для тестов ZDT и с 0,3 до 0,05 для тестов DTLZ. Вероятность удаления гнезда $p_a = 0,05$.

Далее представлены значения параметров алгоритма NSGAII. В качестве оператора селекции выбрана рулеточная селекция. Оператор скрещивания представлен равномерным скрещиванием, причем для вещественных переменных применялось равномерное арифметическое скрещивание [267]. В качестве оператора мутации использовалась случайная мутация с вероятностью 0,05.

В экспериментах использовались три варианта долей дискретных переменных θ : 0,2, 0,5, и 0,8. Первые n_D переменные тестовых функций принимают дискретные значения, где n_D – количество дискретных переменных. Для каждого варианта сочетаний значения θ и множества дискретных значений (равномерное Q_1 или неравномерное Q_2) алгоритм запускался 20 раз. Используемые в эксперименте тестовые функции приведены в таблице 3.1, при

этом применены следующие обозначения: Q – множество значений дискретных переменных, n – количество переменных.

Таблица 3.1 – Тестовые функции эксперимента

№	Название теста	Целевые функции	Оптимальное решение
1	ZDT1	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \cdot \left(1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})}\right)$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1)$	$x_1 \in Q,$ $x_i = 0, i = 2, \dots, n$
2	ZDT2	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \cdot \left(1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2\right)$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1)$	$x_1 \in Q,$ $x_i = 0, i = 2, \dots, n$
3	ZDT3	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \cdot \left(1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})} - (f_1(\mathbf{x})/g(\mathbf{x}) \sin(10\pi f_1(\mathbf{x})))\right)$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1)$	$x_1 \in Q,$ $x_i = 0, i = 2, \dots, n$
4	ZDT4	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \cdot \left(1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})}\right)$ $g(\mathbf{x}) = 1 + 10(m-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$	$x_1 \in Q,$ $x_i = 0, i = 2, \dots, n$
5	ZDT6	$f_1(\mathbf{x}) = 1 - e^{-4x_1 \sin^6(6\pi x_1)}$ $f_2(\mathbf{x}) = g(\mathbf{x}) \cdot \left(1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2\right)$ $g(\mathbf{x}) = 1 + \left(9 \left(\sum_{i=2}^n x_i \right) / (n-1)\right)^{0,25}$	$x_1 \in Q,$ $x_i = 0, i = 2, \dots, n$
6	DTLZ1	$f_1(\mathbf{x}) = \frac{1}{2} x_1 x_2 (1 + g(\mathbf{x}))$ $f_2(\mathbf{x}) = \frac{1}{2} x_1 (1 - x_2) (1 + g(\mathbf{x}))$ $f_3(\mathbf{x}) = \frac{1}{2} (1 - x_1) (1 + g(\mathbf{x}))$ $g(\mathbf{x}) = 100 \left(10 + \sum_{i=3}^n \left((x_i - 0,5)^2 - \cos(20\pi(x_i - 0,5)) \right) \right)$	$x_1 \in Q, x_2 \in Q$ $x_i = 0,5, i = 3, \dots, n$

Продолжение Таблицы 3.1

№	Название теста	Целевые функции	Оптимальное решение
7	DTLZ2	$f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(x_1 \pi/2) \cos(x_2 \pi/2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(x_1 \pi/2) \sin(x_2 \pi/2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin(x_1 \pi/2)$ $g(\mathbf{x}) = \sum_{i=3}^n (x_i - 0,5)^2$	$x_1 \in Q, x_2 \in Q$ $x_i = 0,5, i = 3, \dots, n$
8	DTLZ3	$f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(x_1 \pi/2) \cos(x_2 \pi/2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(x_1 \pi/2) \sin(x_2 \pi/2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin(x_1 \pi/2)$ $g(\mathbf{x}) = 100 \left(10 + \sum_{i=3}^n \left((x_i - 0,5)^2 - \cos(20\pi(x_i - 0,5)) \right) \right)$	$x_1 \in Q, x_2 \in Q$ $x_i = 0,5, i = 3, \dots, n$
9	DTLZ5	$f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(\theta_1) \cos(\theta_2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(\theta_1) \sin(\theta_2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin(\theta_1)$ $g(\mathbf{x}) = \sum_{i=3}^n (x_i - 0,5)^2, \theta_1 = \frac{\pi}{2} x_1,$ $\theta_2 = \frac{\pi}{4(1 + g(\mathbf{x}))} (1 + 2x_2 g(\mathbf{x}))$	$x_1 \in Q, x_2 \in Q$ $x_i = 0,5, i = 3, \dots, n$
10	DTLZ6	$f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(\theta_1) \cos(\theta_2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(\theta_1) \sin(\theta_2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin(\theta_1)$ $g(\mathbf{x}) = \sum_{i=3}^n x_i^{0,1}, \theta_1 = \frac{\pi}{2} x_1, \theta_2 = \frac{\pi}{4(1 + g(\mathbf{x}))} (1 + 2x_2 g(\mathbf{x}))$	$x_1 \in Q, x_2 \in Q$ $x_i = 0, i = 3, \dots, n$

Результаты эксперимента. По результатам 20 запусков алгоритмов оценивались средние значения обратного расстояния поколений *IGD* и степени доминирования *DR*. Результаты оценок приведены в таблицах 3.2 и 3.3 соответственно.

Таблица 3.2 – Средние значения обратного расстояния поколений *IGD*

№	Тест	Квантование	Доля дискретных переменных					
			0,2		0,5		0,8	
			MICS	NSGAII	MICS	NSGAII	MICS	NSGAII
1	ZDT1	Равномерное Q_1	0	0,0366	0,0009	0,0318	0,0016	0,0172
2		Неравномерное Q_2	0	0,0288	0,0001	0,0212	0,0018	0,0124
3	ZDT2	Равномерное Q_1	0	0,0532	0	0,0484	0,0029	0,0333
4		Неравномерное Q_2	0	0,0493	0,0004	0,0348	0,0026	0,0216
5	ZDT3	Равномерное Q_1	0	0,0361	0,0006	0,0312	0,0015	0,0179
6		Неравномерное Q_2	0,0001	0,0267	0,0003	0,0219	0,0009	0,0122
7	ZDT4	Равномерное Q_1	0,0003	5,6805	0,0021	3,7562	0,0647	1,5311
8		Неравномерное Q_2	0	5,7010	0,0032	3,4023	0,0353	1,1729
9	ZDT6	Равномерное Q_1	0	2,9245	0,1212	2,7397	0,1218	2,2194
10		Неравномерное Q_2	0	2,8349	0,0001	2,6315	0,4335	2,2743
11	DTLZ1	Равномерное Q_1	0,0129	0,4898	0,0129	0,0108	0,0131	0,0130
12		Неравномерное Q_2	0,0121	0,1317	0,0076	0,0086	0,0075	0,0081
13	DTLZ2	Равномерное Q_1	0,0285	0,0264	0,0284	0,0275	0,0295	0,0286
14		Неравномерное Q_2	0,0251	0,0202	0,0255	0,0243	0,0264	0,0256
15	DTLZ3	Равномерное Q_1	0,0267	0,2519	0,0324	0,0311	0,0332	0,0329
16		Неравномерное Q_2	0,0213	0,4797	0,0286	0,0295	0,0296	0,0282
17	DTLZ5	Равномерное Q_1	0	0	0	0	0	0
18		Неравномерное Q_2	0	0	0	0	0	0
19	DTLZ6	Равномерное Q_1	0	0	0	0	0	0
20		Неравномерное Q_2	0	0	0	0	0	0
Среднее значение			0,0064	0,9386	0,0132	0,6425	0,0403	0,3724

Таблица 3.3 – Оценки средней степени доминирования *DR* (%)

№	Тест	Квантование	Доля дискретных переменных					
			0,2		0,5		0,8	
			MICS	NSGAII	MICS	NSGAII	MICS	NSGAII
1	ZDT1	Равномерное Q_1	100	0	100	0	99,52	0,48
2		Неравномерное Q_2	100	0	100	0	100	0
3	ZDT2	Равномерное Q_1	100	0	100	0	97,11	2,89
4		Неравномерное Q_2	100	0	100	0	100	0
5	ZDT3	Равномерное Q_1	100	0	100	0	96,25	3,75
6		Неравномерное Q_2	100	0	100	0	100	0
7	ZDT4	Равномерное Q_1	100	0	100	0	100	0
8		Неравномерное Q_2	100	0	100	0	100	0
9	ZDT6	Равномерное Q_1	100	0	100	0	100	0
10		Неравномерное Q_2	100	0	100	0	100	0
11	DTLZ1	Равномерное Q_1	95,25	4,75	65,21	34,79	70,18	35,79
12		Неравномерное Q_2	91,52	8,48	71,29	28,71	75,44	44,55
13	DTLZ2	Равномерное Q_1	62,96	37,04	61,38	38,74	62,66	58,88
14		Неравномерное Q_2	66,47	33,53	67,53	32,51	68,30	66,26
15	DTLZ3	Равномерное Q_1	92,82	7,18	67,68	32,32	69,50	38,21
16		Неравномерное Q_2	95,40	4,60	74,20	25,80	72,74	34,35
17	DTLZ5	Равномерное Q_1	100	0	100	0	100	50
18		Неравномерное Q_2	100	0	100	0	100	45

Продолжение Таблицы 3.3

№	Тест	Квантование	Доля дискретных переменных					
			0,2		0,5		0,8	
			MICS	NSGAII	MICS	NSGAII	MICS	NSGAII
19	DTLZ6	Равномерное Q_1	100	100	100	100	100	100
20		Неравномерное Q_2	100	100	100	100	100	100
Среднее значение			95,22	14,78	90,37	19,64	90,59	29,01

Для демонстрации некоторые решения, полученные алгоритмами, представлены на рисунках 3.3 и 3.4.

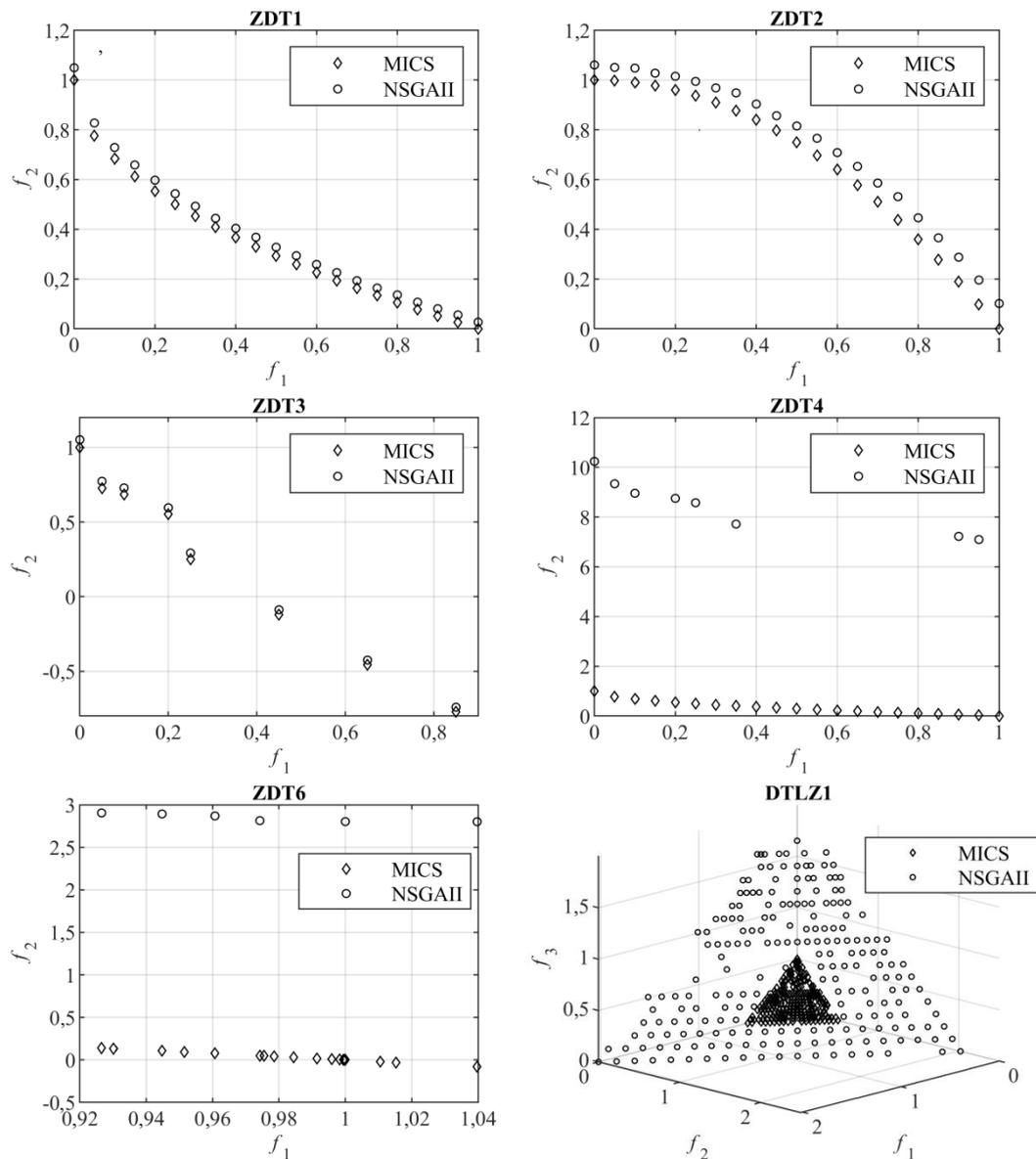


Рисунок 3.3 – Примеры найденных решений для тестов ZDT1–ZDT4, ZDT6, DTLZ1

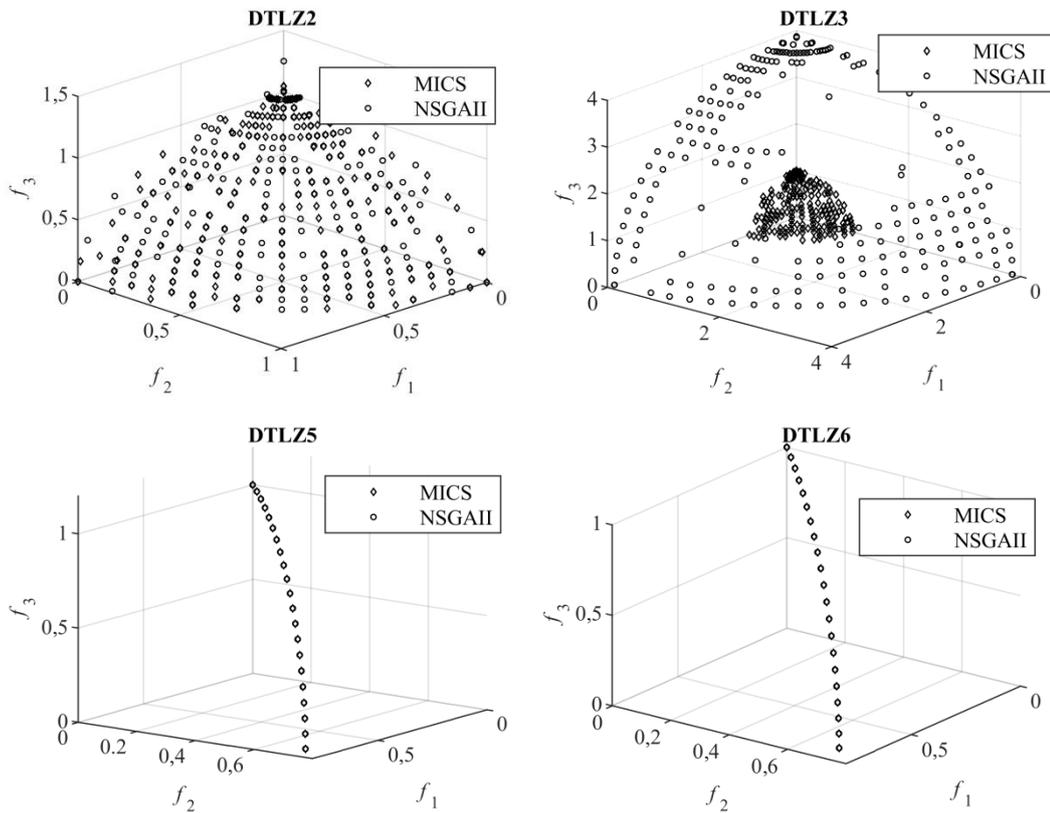


Рисунок 3.4 – Примеры найденных решений для тестов DTLZ2, DTLZ3, DTLZ5, DTLZ6

Выводы. Для доказательства сравнения эффективности работы алгоритмов необходимо выявить статистически значимую тенденцию в смещении критериев IGD и DR . Чтобы определить, значительно ли отличаются результаты эффективности предлагаемого алгоритма от NSGAII, воспользуемся парным статистическим тестом Уилкоксона [259, 268]. Цель теста – обнаружить существенные различия между двумя выборочными средними значениями IGD и DR . Нулевая гипотеза H_0 теста утверждает отсутствие значимых различий в оценках эффективности сравниваемых алгоритмов, а альтернативная H_1 утверждает обратное.

Тест знаковых рангов Уилкоксона применен к результатам оценок в таблицах 3.2 и 3.3, причем для IGD сравнения эффективности проводились не только между MICS и NSGAII, но и для одного и того же алгоритма с разными долями дискретных переменных. Такое сравнение позволит выявить тенденцию к увеличению или уменьшению критерия эффективности алгоритма с изменением количества дискретных переменных. Результаты асимптотической значимости

теста (p -value) для IGD представлены в таблице 3.4, а для DR в таблице 3.5. Поскольку меньшее значение IGD соответствует лучшему решению, то меньшая сумма рангов теста в таблице 3.4 говорит о большей эффективности алгоритма. Для DR , наоборот, большее значение соответствует лучшему решению, поэтому большая сумма рангов теста в таблице 3.5 говорит о большей эффективности алгоритма.

Таблица 3.4 – Статистическое сравнение алгоритмов MICS и NSGAII по критерию IGD

Сравниваемые алгоритмы		Доля дискретных переменных		Сумма рангов		p -value	Гипотеза H_0 ($\alpha=0,05$)
Алгоритм 1	Алгоритм 2	Алгоритм 1	Алгоритм 2	+	-		
MICS	NSGAII	0,2	0,2	3	133	<0,001	Отклоняется
MICS	NSGAII	0,5	0,5	16,5	119,5	0,0078	Отклоняется
MICS	NSGAII	0,8	0,8	18	118	0,0096	Отклоняется
MICS	MICS	0,2	0,5	13	92	0,0131	Отклоняется
MICS	MICS	0,2	0,8	10	126	0,0027	Отклоняется
MICS	MICS	0,5	0,8	1	135	<0,001	Отклоняется
NSGAII	NSGAII	0,2	0,5	133	3	<0,001	Отклоняется
NSGAII	NSGAII	0,2	0,8	133	3	<0,001	Отклоняется
NSGAII	NSGAII	0,5	0,8	119,5	16,5	0,0078	Отклоняется

Таблица 3.5 – Статистическое сравнение алгоритмов MICS и NSGAII по критерию DR

Сравниваемые алгоритмы		Доля дискретных переменных		Сумма рангов		p -value	Гипотеза H_0 ($\alpha=0,05$)
Алгоритм 1	Алгоритм 2	Алгоритм 1	Алгоритм 2	+	-		
MICS	NSGAII	0,2	0,2	171	0	<0,001	Отклоняется
MICS	NSGAII	0,5	0,5	171	0	<0,001	Отклоняется
MICS	NSGAII	0,8	0,8	171	0	<0,001	Отклоняется

Анализ таблицы 3.4 показывает статистически значимое преимущество алгоритма MICS над NSGAII по критерию IGD . Это преимущество обеспечивается на уровне значимости $\alpha = 0,05$. Значение критерия в среднем улучшается на 89,18%, 97,95% и 99,3% для 0,8, 0,5 и 0,2 долей дискретных переменных соответственно. Среднее улучшение критерия с учетом долей дискретных переменных по сравнению с NSGAII составило 95,47%.

Таблица 3.5 показывает статистически значимое преимущество по критерию DR алгоритма MICS на уровне значимости $\alpha = 0,05$. Критерий в

среднем превосходит на 61,58%, 70,73% и 80,44% для 0,8, 0,5 и 0,2 долей дискретных переменных соответственно. Среднее значение улучшения критерия по всем используемым в эксперименте долям дискретных переменных составило 70,92%.

Что касается влияния количества дискретных переменных на эффективность, то здесь алгоритмы проявили себя кардинально противоположно (см. таблицу 3.5). С увеличением числа дискретных переменных MICS проявил статистически значимую тенденцию к увеличению *IDG* на уровне значимости $\alpha = 0,05$. Алгоритм NSGAI, напротив, показал статистически значимое уменьшение *IDG* с ростом количества дискретных переменных. Такое поведение алгоритмов говорит о том, что генетический алгоритм более эффективен при поиске в дискретном пространстве. Стратегия «кукушкин поиск» эффективнее реализуется в непрерывном пространстве.

Предложенный алгоритм смешанной многокритериальной оптимизации MICS продемонстрировал значимый прирост эффективности на комплексных оценках сходимости и многообразия, таких как степень доминирования и обратное расстояние поколений, по сравнению с генетическим алгоритмом с недоминируемой сортировкой NSGAI. Достижение таких результатов получилось благодаря совместному применению генетического оператора скрещивания для генерации новых целочисленных элементов решений и стратегии «кукушкин поиск» для генерации новых элементов с непрерывными значениями. Используя генетические операторы в дискретном пространстве и шаги перемещений стратегий роевого интеллекта в непрерывном пространстве, можно добиться ощутимых результатов оптимизации в смешанном пространстве поиска.

3.2 Нахождение оптимума тестовых функций дискретным алгоритмом оптимизации

В настоящем подразделе представлены эксперименты с разработанным дискретным алгоритмом оптимизации на основе распределения вероятностей с трансформацией целевых значений (PDT). Алгоритм тестировался для бинарной проблемы оптимизации, т.е. когда переменные принимают только два значения. В экспериментальном исследовании использовались восемнадцать различных унимодальных и мультимодальных эталонных функций, широко применяемых для тестирования алгоритмов оптимизации [218, 234-236]. В таблице 3.6 представлены их характеристики, а на рисунках 3.5 и 3.6 графики в двумерном пространстве поиска. Функции f_1-f_{11} являются унимодальными, т.е. содержат только один глобальный оптимум. Функции $f_{12}-f_{18}$ являются мультимодальными и содержат один глобальный и множество локальных оптимумов, число которых экспоненциально растет с увеличением размерности задачи. Эксперимент проводился согласно методике работы [234].

Таблица 3.6 – Тестовые функции эксперимента

№	Целевая функция	Диапазон поиска	Значение оптимума функции
1	$f_1(\mathbf{x}) = \sum_{i=1}^n x_i^2$	[-100, 100]	0
2	$f_2(\mathbf{x}) = \sum_{i=1}^n ix_i^4$	[-2,56, 2,56]	0
3	$f_3(x) = \sum_{i=1}^n (x_i + 0,5)^2$	[-10, 10]	0
4	$f_4(\mathbf{x}) = \sum_{i=1}^n ix_i^4 + rand(0,1)$	[-2,56, 2,56]	0
5	$f_5(\mathbf{x}) = \max_{i=1}^n (x_i)$	[-100, 100]	0
6	$f_6(\mathbf{x}) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	[-100, 100]	0
7	$f_7(\mathbf{x}) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[-2, 2]	0

Продолжение Таблицы 3.6

№	Целевая функция	Диапазон поиска	Значение оптимума функции
8	$f_8(\mathbf{x}) = \sum_{i=1}^n ix_i^2$	[-10, 10]	0
9	$f_9(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$	[-10, 10]	0
10	$f_{10}(\mathbf{x}) = \sum_{i=1}^n x_i ^{i+1}$	[-1, 1]	0
11	$f_{11}(\mathbf{x}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	[-6, 6]	0
12	$f_{12}(\mathbf{x}) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$	[-10, 10]	-30
13	$f_{13}(\mathbf{x}) = 1 - \cos \left(2\pi \sqrt{\sum_{i=1}^n x_i^2} \right) + 0,1 \sqrt{\sum_{i=1}^n x_i^2}$	[-25, 25]	0
14	$f_{14}(\mathbf{x}) = 0,1 \left(\sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) + \dots \right.$ $\left. \dots (x_n - 1)(1 + \sin^2(2\pi x_n)) \right) + \sum_{i=1}^n u(x_i, 5, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	[-5,5]	0
15	$f_{15}(\mathbf{x}) = \sum_{i=1}^n (x_i - 10 \cos(2\pi x_i) + 10)$	[-2, 2]	0
16	$f_{16}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	[-10, 10]	0
17	$f_{17}(\mathbf{x}) = -20e^{\left(-0,2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right)} - e^{\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right)} + 20 + e$	[-3, 3]	0
18	$f_{18}(\mathbf{x}) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	[-15, 15]	-50,0929

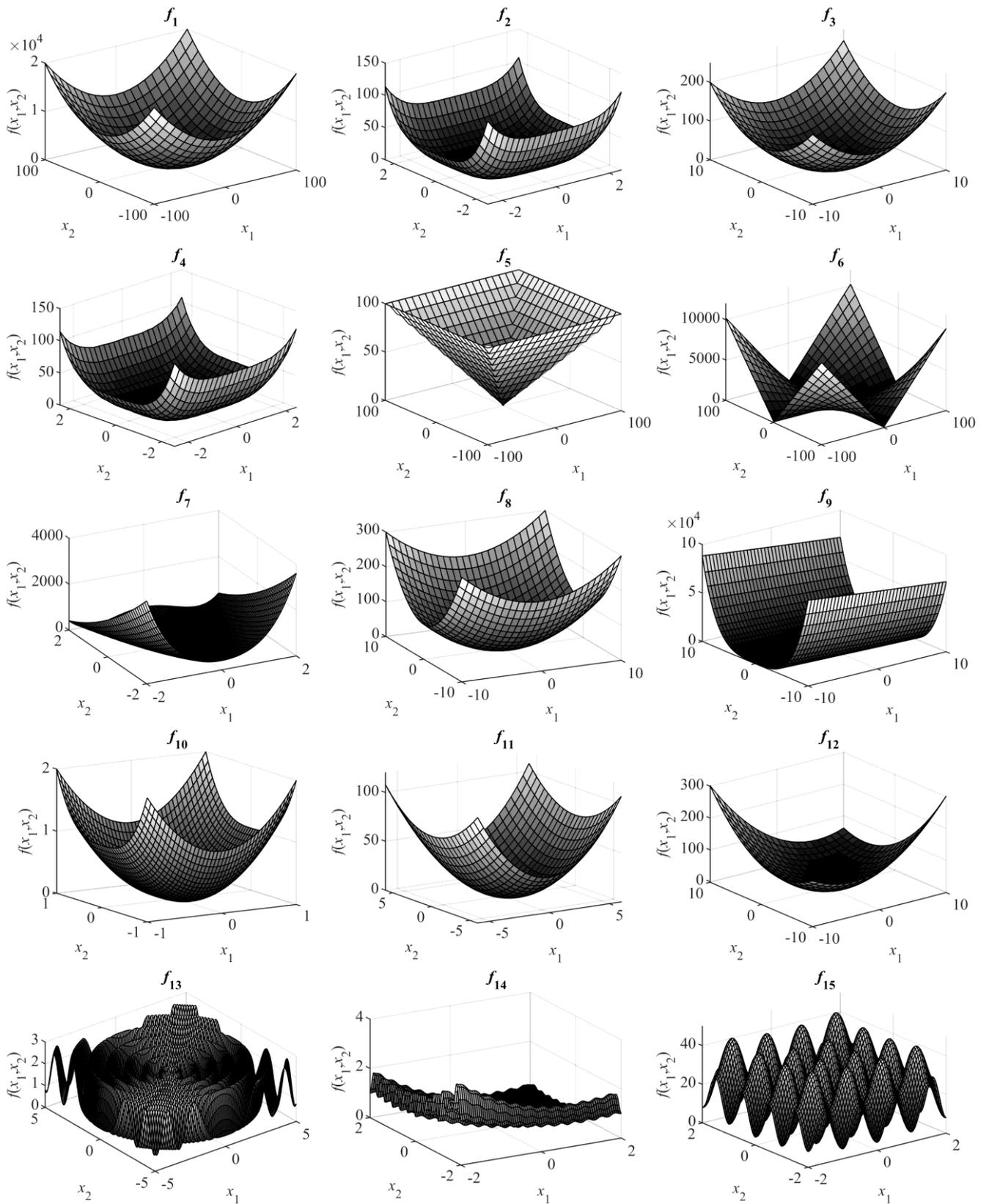


Рисунок 3.5 – Графики тестовых функций f_1 – f_{15} в двумерном пространстве поиска

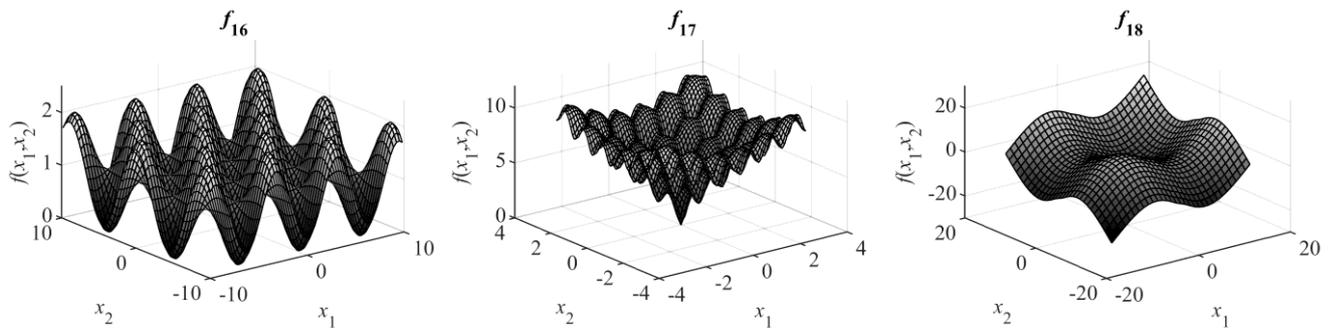


Рисунок 3.6 – Графики тестовых функций f_{16} – f_{18} в двумерном пространстве поиска

Дискретизация непрерывных значений. Поскольку алгоритм является дискретным и оперирует в эксперименте бинарными векторами решений, элементы которых принимают значение 0 или 1, проводится кодировка вещественных значений бинарным вектором. Процедура перевода бинарного вектора решения в значения вещественных переменных представлена на рисунке 3.7. Данная процедура выполняется всякий раз, когда алгоритму необходимо рассчитать значение целевой функции. Количество переменных в эксперименте имеет значение 5, количество битов для кодирования значения каждой переменной – 15. Таким образом, величина бинарного вектора решений составляет $5 \cdot 15 = 75$ элементов. Количество дискретных значений, которое может иметь каждая переменная, соответствует $N = 2^{15}$. Эти значения определяются с помощью равномерного квантования на диапазоне поиска переменной. Шаг дискретизации определяется следующим образом:

$$\Delta h = \frac{R_{\max} - R_{\min}}{N - 1},$$

где R_{\min} и R_{\max} – нижняя и верхняя граница диапазона значений переменной соответственно. Фактически, бинарное значение переменной – это бинарное представление порядкового номера дискретного значения на диапазоне $[R_{\min}, R_{\max}]$ с шагом дискретизации Δh .

Если переменная x кодируется бинарным вектором $[b_1, \dots, b_{15}]$, то вещественное значение этой переменной определяется следующим образом:

$$x = R_{\min} + \Delta h \cdot \sum_{i=1}^{15} b_i \cdot 2^{i-1}.$$

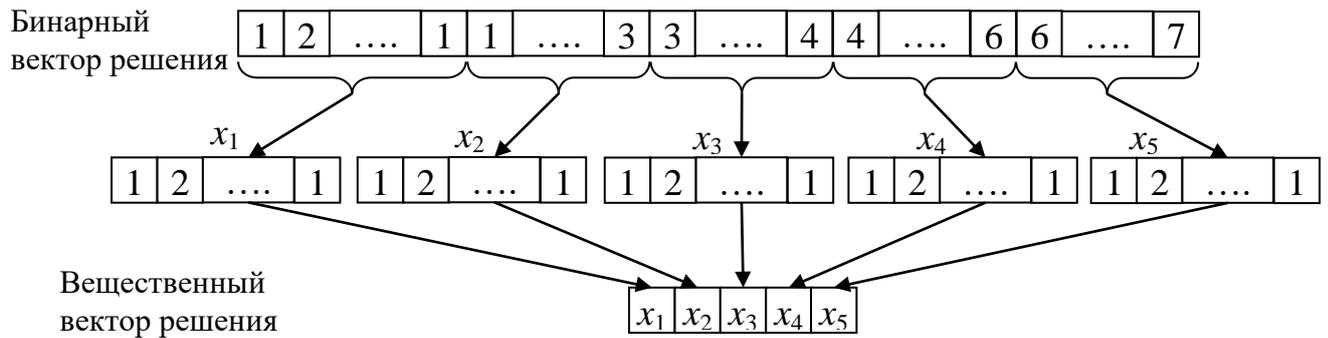


Рисунок 3.7 – Перевод бинарного вектора решения в непрерывный вектор для вычисления значения целевой функции

Критерии эффективности. Для оценки эффективности работы алгоритма применялись два критерия [259]. Первый оценивает сходимость алгоритма и определяется средним отклонением найденного целевого значения от фактического:

$$E = \frac{1}{n_{run}} \cdot \sum_{i=1}^{n_{run}} |f_i - f'|,$$

где n_{run} – количество запусков алгоритма, f_i – найденное алгоритмом значение целевой функции в i -м запуске, f' – фактическое значение оптимума целевой функции.

Второй критерий оценивает стабильность работы недетерминированного алгоритма и определяется среднеквадратичным отклонением найденного оптимума целевой функции:

$$STD = \sqrt{\frac{1}{n_{run}} \sum_{i=1}^{n_{run}} (f_i - M)^2},$$

где M – среднее значение целевой функции

$$M = \frac{\sum_{i=1}^{n_{run}} f_i}{n_{run}}.$$

Меньшее значение обоих критериев соответствует лучшему значению эффективности.

Кроме вышеприведенных критериев в ходе проведения эксперимента получены графики сходимости алгоритмов, позволяющие оценить скорость

сходимости стохастических алгоритмов и показывающие зависимость критерия E от итерации [218, 234, 235]. Значение E , приводимое на графиках, является средним значением по запускам алгоритма.

Выбор функции трансформации целевых значений. Для выбора функции трансформации целевых значений в весовые коэффициенты решений были использованы следующие функции: T_L – линейная функция, T_S – сигмоида, T_Q – квадратичная функция, T_T – гиперболический тангенс.

Алгоритм PDT с разными функциями трансформации был использован для поиска оптимума тестовых функций. Было осуществлено 30 запусков на каждой тестовой функции. Полученные значения критериев эффективности приведены в таблице 3.7.

Таблица 3.7 – Оценка эффективности алгоритма PDT с различными функциями трансформации

f	T_L		T_S		T_Q		T_T	
	E	STD	E	STD	E	STD	E	STD
f_1	0,000047	0,000000	0,000049	0,000014	0,000047	0,000000	0,000059	0,000034
f_2	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
f_3	0,402924	0,231313	0,454002	0,280296	0,478528	0,240684	0,498614	0,281871
f_4	0,005653	0,004306	0,006070	0,004650	0,005624	0,003659	0,005170	0,003590
f_5	0,041098	0,096504	0,030112	0,026176	0,042929	0,140588	0,037639	0,042328
f_6	0,015259	0,000000	0,015259	0,000000	0,015666	0,002229	0,015463	0,001114
f_7	2,629031	1,500038	2,511857	1,363161	3,214144	1,576402	3,036793	1,573228
f_8	0,000001	0,000000	0,000001	0,000000	0,000001	0,000000	0,000003	0,000003
f_9	0,885491	0,863594	0,850277	0,472212	0,803704	0,485206	0,835841	0,739384
f_{10}	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
f_{11}	0,000001	0,000000	0,000001	0,000000	0,000001	0,000000	0,000001	0,000002
f_{12}	0,239046	0,613565	0,870265	1,432694	0,885326	2,910895	0,362761	0,898002
f_{13}	0,119874	0,040684	0,149874	0,062972	0,163207	0,071839	0,123821	0,056141
f_{14}	0,057948	0,077372	0,058916	0,059387	0,053028	0,053763	0,033397	0,046236
f_{15}	0,331668	0,603435	0,199166	0,404835	0,398790	0,618058	0,099557	0,303747
f_{16}	0,036349	0,016301	0,028747	0,014743	0,029202	0,015521	0,034756	0,019396
f_{17}	0,000374	0,000041	0,000374	0,000041	0,000367	0,000000	0,000389	0,000069
f_{18}	0,407336	1,540216	0,405125	1,540966	0,610862	1,858916	0,000837	0,003885

Для улучшения оценки эффективности эволюционных алгоритмов в [259] отмечается, что необходимо проводить статистические тесты. Недостаточно сравнивать алгоритмы по значениям E и STD [260], необходимо провести статистический тест, чтобы доказать, что предлагаемый новый алгоритм

представляет собой значительное улучшение по сравнению с другими существующими методами.

Чтобы судить о том, отличаются ли статистически значимо результаты работы алгоритма с различными функциями трансформации друг от друга, был проведен непараметрический статистический тест Фридмана при уровне значимости $\alpha = 0,05$. Значения асимптотической значимости p -value, меньшие 0,05, можно рассматривать как убедительное свидетельство против нулевой гипотезы [259]. Тест Фридмана множественных сравнений не выявил отклонение нулевой гипотезы H_0 для обоих критериев. Гипотеза H_0 утверждает об отсутствии значимых различий между вариантами алгоритма с различными функциями трансформации. Асимптотическая значимость для критерия E соответствует значению p -value = 0,757, а для критерия STD значению p -value = 0,590. Таким образом, выбор рассмотренных функций трансформации существенно не повлияет на эффективность работы алгоритма. В дальнейшем будет использоваться линейная функция трансформации.

Параметры вычислительного эксперимента. Эффективность предлагаемого алгоритма PDT оценивалась в сравнении с такими популярными алгоритмами оптимизации как генетический алгоритм (GA) и бинарный алгоритм роящихся частиц (BPSO). Алгоритмы выполнялись в одинаковых условиях. Общие настройки имели следующие значения: размер популяции – 30, количество итераций – 100, количество переменных – 5, число бит на одну переменную – 15, количество запусков алгоритма на каждую тестовую функцию – 30. Специфичные параметры алгоритмов GA и BPSO были установлены в значения, рекомендованные в [218, 235]. Значения специфичных параметров приведены в таблице 3.8.

Таблица 3.8 – Значения параметров алгоритмов

Алгоритм	Параметр	Значение
PDT	Функция трансформации T	Линейная T_L
	Вероятность мутации p	0,05
GA	Вид селекции	Рулеточная
	Вид скрещивания (вероятность)	Одноточечный (0,9)
	Вид мутации (вероятность)	Равномерный (0,005)
BPSO	Коэффициенты C_1, C_2	2,2
	Вес инерции W	Линейно уменьшается с 0,9 до 0,4
	Максимальная скорость	6
	Функция трансформации	V-образная

Результаты эксперимента. В результате выполнения эксперимента были получены значения критериев эффективности каждого алгоритма. Данные значения приведены в таблице 3.9. Последняя строка таблицы содержит средние значения показателей. На рисунке 3.8 показаны графики сходимости алгоритмов, позволяющие оценить скорость сходимости. Графики представлены в логарифмической шкале по оси критерия сходимости, что позволяет более четко отследить скорость сходимости алгоритмов на протяжении всей их работы.

Таблица 3.9 – Оценки эффективности бинарных алгоритмов

f	GA		BPSO		PDT	
	E	STD	E	STD	E	STD
f_1	0,005349	0,028272	27,845964	37,522025	0,000047	0,000000
f_2	0,000000	0,000000	0,008293	0,018114	0,000000	0,000000
f_3	0,440548	0,254233	0,799580	0,513039	0,402924	0,231313
f_4	0,021143	0,046160	0,067826	0,054019	0,005653	0,004306
f_5	0,649841	1,354239	5,634938	3,111477	0,041098	0,096504
f_6	0,016683	0,004988	4,112874	3,333243	0,015259	0,000000
f_7	3,113280	2,491653	4,288055	2,188550	2,629031	1,500038
f_8	0,000009	0,000025	0,804053	0,875332	0,000001	0,000000
f_9	10,522179	24,073438	1,827357	1,320775	0,885491	0,863594
f_{10}	0,000000	0,000000	0,000166	0,000383	0,000000	0,000000
f_{11}	0,000004	0,000008	0,303480	0,574945	0,000001	0,000000
f_{12}	0,760543	1,068967	0,571626	0,780258	0,239046	0,613565
f_{13}	0,457784	0,259707	0,418610	0,159854	0,119874	0,040684
f_{14}	0,072918	0,062301	0,054331	0,053543	0,057948	0,077372
f_{15}	1,227600	1,001766	1,408564	0,734589	0,331668	0,603435
f_{16}	0,042970	0,016959	0,061572	0,021475	0,036349	0,016301
f_{17}	0,000583	0,000525	0,357644	0,145604	0,000374	0,000041
f_{18}	4,935197	6,147513	3,989981	3,238601	0,407336	1,540216
Среднее значение	1,24	2,05	2,92	3,04	0,29	0,31

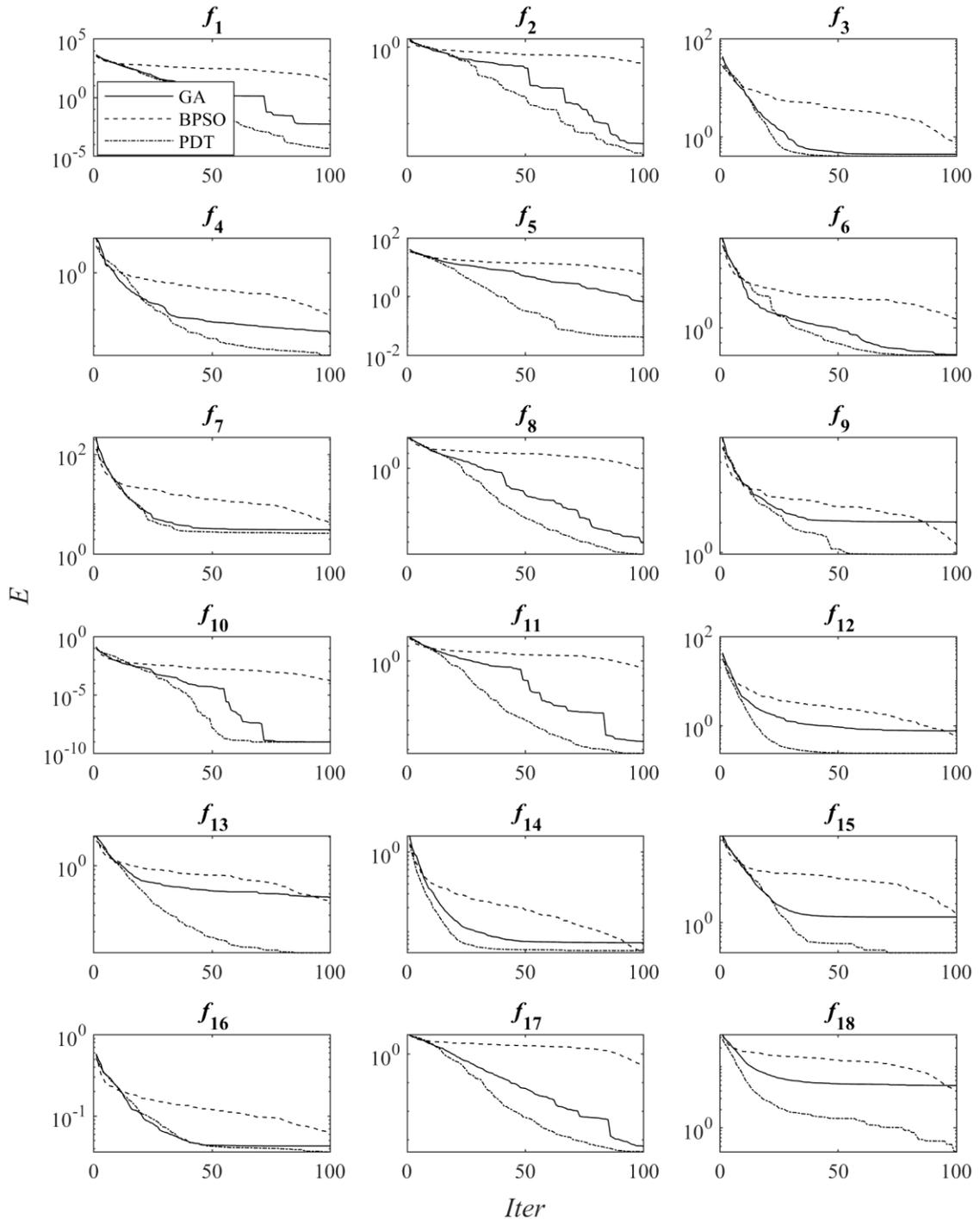


Рисунок 3.8 – Графики сходимости алгоритмов

Выводы. Чтобы определить, значительно ли отличаются результаты эффективности предлагаемого алгоритма от аналогов, воспользуемся парным статистическим тестом Уилкоксона [259]. Нулевая гипотеза H_0 теста утверждает об отсутствии значимых различий в оценках эффективности сравниваемых

алгоритмов, альтернативная гипотеза H_1 утверждает об обратном. В таблице 3.10 представлены результаты сравнения. Асимптотическая значимость для критериев E и STD при сравнении с генетическим алгоритмом и алгоритмом роящихся частиц оказалась меньше уровня значимости $\alpha = 0,05$. Сумма отрицательных рангов теста превалирует над положительными. Это говорит о том, что значения критериев алгоритма PDT статистически значимо меньше алгоритмов GA и BPSO на уровне значимости $\alpha = 0,05$.

Таблица 3.10 – Статистическое сравнение PDT и альтернативных алгоритмов

Критерий	Альтернативный алгоритм	Сумма рангов		p -value	Гипотеза $H_0(\alpha=0,05)$
		+	-		
E	GA	0	136	<0,001	Отклоняется
	BPSO	2	169	<0,001	Отклоняется
STD	GA	6	130	<0,001	Отклоняется
	BPSO	4	167	<0,001	Отклоняется

Анализ рисунка 3.9 показывает, что на начальных итерациях скорость алгоритма роящихся частиц для тестов f_6 , f_7 , f_9 , f_{15} и f_{16} оказывается выше остальных алгоритмов, но начиная примерно с пятнадцатой итерации она спадает. В целом же алгоритм на основе распределения вероятностей с трансформацией целевых значений опережает по скорости своих конкурентов.

Предложенный дискретный алгоритм на основе распределения вероятностей с трансформацией целевых значений показал статистически значимое улучшение показателей сходимости и стабильности, таких как отклонение от оптимума и среднеквадратичное отклонение целевых значений. Сравнения проводились с генетическим алгоритмом и бинарным алгоритмом роящихся частиц. Для эксперимента использовались восемнадцать тестовых унимодальных и мультимодальных функций. В среднем отклонение от оптимума уменьшилось в 4,3 раза по сравнению с генетическим алгоритмом и в 10,1 раза по сравнению с бинарным алгоритмом роящихся частиц. Полученные результаты говорят об эффективности предложенного алгоритма для оптимизации в бинарном пространстве.

3.3 Построение нечетких классификаторов с применением метода разделения данных

В подразделе 2.6 предложен метод разделения данных бинарным метавэвристическим алгоритмом. Согласно методу, исходные данные разделяются на обучающую и валидационную выборку с целью исключить переобучение, т.е. повысить обобщающую прогностическую способность модели, заключающуюся в точности прогнозов на образцах, не участвующих в обучении. На основе обучающей выборки проводится построение моделей, а с помощью валидационной осуществляется выбор наиболее эффективной по критерию точности.

В настоящем подразделе представлен эксперимент по применению данного метода при построении нечетких классификаторов с помощью алгоритма смешанной многокритериальной оптимизации MICS (см. подраздел 2.3). Процедура построения классификатора входит в состав методики построения (подраздел 2.7). Алгоритм возвращает множество недоминируемых по Парето классификаторов. Разделение данных на обучающую и валидационную выборки осуществлялось дискретным алгоритмом оптимизации на основе распределения вероятностей с трансформацией целевых значений, представленным в подразделе 2.4.

Целью эксперимента является проверка эффективности применения предлагаемого метода разделения данных при построении нечеткого классификатора по сравнению с построением без разделения, как предлагается в [83, 84, 88], где выбирается классификатор с лучшей точностью на обучающих данных, и с разделением данных случайным образом. Критериями эффективности выступают точность классификации, где показателем выступает процент верно классифицированных образцов, и интерпретируемость, где в качестве показателей используются количество правил, общее количество используемых в базе правил нечетких термов и количество признаков. Кроме этого, в результате эксперимента

необходимо определить вид целевой функции для бинарного алгоритма среди предложенных вариантов (2.9), (2.10), (2.11) и (2.12), и вариант формирования выборки данных для выбора модели на фронте Парето с лучшим значением критерия точности. Для формирования выборки предлагаются два варианта. Первый использует только валидационные данные. Второй вариант использует исходный набор, т.е. включает как обучающую, так и валидационную выборки.

Методика проведения вычислительного эксперимента. Методика проведения вычислительного эксперимента осуществлялась по схеме десятикратной перекрестной проверки и соответствует методике, приведенной в [83, 84, 86]. Для каждого из 10 разбиений данных метод запускался три раза. Таким образом, эффективность оценивались средним значением по тридцати запускам метода на каждом наборе данных. Вычислялись показатель точности – процент верно классифицированных образцов, и показатели интерпретируемости – количество правил, общее количество используемых в классификаторе нечетких терминов, количество используемых признаков. Согласно данной методике, оценивалась эффективность и альтернативных методов. Значения параметров при проведении эксперимента представлены в таблице 3.11.

Таблица 3.11 – Значения параметров при проведении эксперимента

	Название параметра	Значение параметра
Общие параметры	Отношение количества образцов $a:b$	3:1
	Количество нечетких множеств на признак h	Для всех признаков 3
	Максимальное количество терминов в правиле	3
	Диапазон изменений бокового смещения α и охвата β	[-0,3 0,3]
Параметры дискретного алгоритма	Размер популяции	30
	Число итераций	1000
	Функция трансформации T	Линейная T_L
	Вероятность мутации p	0,05
Параметры алгоритма смешанной оптимизации	Размер популяции	100
	Число итераций	5000
	Максимальное количество правил r	Кол-во классов \times 7
	Вероятность удаления гнезда p_a	0,05
	Коэффициент прыжка полетов Леви γ	Равномерно уменьшался с 0,4 до 0,05

Наборы данных. Для экспериментов были использованы 38 наборов данных из репозитория Knowledge Extraction based on Evolutionary Learning (KEEL, <http://keel.es>). В таблице 3.12 приведены характеристики этих наборов. Указаны количества образцов, признаков, классов, а также краткое описание проблемной области, представляющей набор. Три набора являются синтетическими, т.е. специально разработанными для проверки эффективности классификаторов, остальные собраны из реальных проблемных областей.

Таблица 3.12 – Наборы данных эксперимента. Обозначения: $\#F$ – количество признаков, $\#P$ – количество образцов, $\#C$ – количество классов

№	Название набора	$\#F$	$\#P$	$\#C$	Описание проблемной области
1	Appendicitis	7	106	2	Диагностика аппендицита по семи медицинским характеристикам.
2	Australian	14	690	2	Принять решение о выдаче кредитных карт.
3	Balance	4	625	3	Определить наклон весов при взвешивании.
4	Banana	2	5300	2	Определить принадлежность образца кластерам в форме банана.
5	Bands	19	365	2	Определить объект в виде цилиндрической ленты при ротোগравюрной печати.
6	Cleveland	13	297	5	Диагностика сердечных заболеваний.
7	Contraceptive	9	1473	3	Определить выбор метода контрацепции у женщин.
8	Dermatology	34	358	6	Дифференциальная диагностика эритематосквамозных заболеваний.
9	Ecoli	7	336	8	Определить место локализации белков в клетке.
10	Glass	9	214	7	Определить тип стекла по содержанию оксидов.
11	Haberman	3	306	2	Определить, выживет ли пациент после операции по удалению злокачественной опухоли груди.
12	Hayes-roth	4	160	3	Искусственный набор для проверки классификаторов.
13	Heart	13	270	2	Определить наличие или отсутствие заболеваний сердца.
14	Hepatitis	19	80	2	Определить, выживет ли пациент с гепатитом.
15	Ionosphere	33	351	2	Определить тип отраженного сигнала радара.
16	Iris	4	150	3	Определить вид цветка ириса по его измерениям.
17	Magic	10	19020	2	Определить какие изображения гамма-телескопа созданы первичными гамма-лучами, а какие адронными ливнями.
18	Mammographic	5	830	2	Определить доброкачественность или злокачественность опухоли при маммографическом исследовании.
19	Movement_libras	90	360	15	Определить тип движения руки по координатам траектории.
20	Newthyroid	5	215	3	Диагностика болезней щитовидной железы.
21	Page-blocks	10	5472	5	Определить тип блока страницы документа в процессе сегментации.

Продолжение Таблицы 3.12

№	Название набора	#F	#P	#C	Описание проблемной области
22	Phoneme	5	5404	2	Определить носовые и ротовые звуки речи.
23	Pima	8	768	2	Диагностика сахарного диабета у женщин племени индейцев Пима.
24	Ring	20	7400	2	Искусственный набор для проверки классификаторов.
25	Satimage	36	6435	7	Определить тип почвы по многоспектральным спутниковым снимкам поверхности Земли.
26	Sonar	60	208	2	Определить по отраженным сигналам сонара под разными углами тип объекта – камень или металлический цилиндр.
27	Spambase	57	4597	2	Определить по характеристикам электронного письма массовую, неперсонифицированную рассылку (электронный спам).
28	Spectfheart	44	267	2	Диагностика пациента по изображениям компьютерной томографии.
29	Tae	5	151	3	Оценка эффективности преподавания педагогом.
30	Thyroid	21	7200	3	Диагностика болезней щитовидной железы.
31	Titanic	3	2201	2	Определить выживаемость пассажира Титаника по его характеристикам.
32	Twonorm	20	7400	2	Искусственный набор для проверки классификаторов.
33	Vehicle	18	846	4	Определить тип транспортного средства по его силуэту.
34	Vowel	13	990	11	Определить устойчивый гласный звук из одиннадцати гласных звуков английского языка.
35	Wdbc	30	569	2	Диагностика злокачественной опухоли молочной железы.
36	Wine	13	178	3	Определить сорт вина по химическому составу.
37	Wisconsin	9	683	2	Диагностика рака молочной железы.
38	Yeast	8	1484	10	Определить место локализации клеток дрожжей среди десяти возможных альтернатив.

Результаты эксперимента. После проведения эксперимента были получены средние значения по запускам оценок эффективности построенных нечетких классификаторов. В таблице 3.13 приведены значения точности вариантов предлагаемого метода и альтернативных методов построения для классификаторов, где в качестве агрегации правил применялась функция максимума (выражение (1.1)). В таблице 3.14 приведены значения точности для классификаторов с агрегацией правил функцией суммы (выражение (1.2)). Для обозначения альтернативных методов в таблицах используется: МI – метод, согласно которому классификаторы обучаются на всем наборе данных, и выбирается модель с лучшей точностью на этом же наборе; Random – метод, в

котором обучающая и валидационная выборки формируются случайным образом; V – итоговый классификатор выбирается на основе валидационной выборки; $V+T$ – итоговый классификатор выбирается на основе всей исходной выборки; φ_1 – целевая функция (2.9); φ_2 – целевая функция (2.10); φ_3 – целевая функция (2.11); φ_4 – целевая функция (2.12).

Таблица 3.13 – Значение точности классификации (%) вариантов предлагаемого метода и альтернативных методов с агрегацией правил функцией максимума

№	Наборы данных	Альтернативы		Метод разделения данных								
		MI	Random		V				V+T			
			V	V+T	φ_1	φ_2	φ_3	φ_4	φ_1	φ_2	φ_3	φ_4
1	Appendicitis	84,70	82,27	83,09	82,58	84,21	82,30	82,33	84,45	86,00	83,79	82,58
2	Australian	83,91	85,02	84,59	84,83	85,07	85,27	84,69	84,69	84,73	84,73	84,83
3	Balance	87,73	82,61	82,61	89,28	88,58	88,54	88,05	89,23	88,32	88,91	88,16
4	Banana	73,53	76,26	76,33	77,40	77,52	77,03	74,92	77,32	80,04	77,03	74,92
5	Bands	67,33	63,73	65,47	64,85	65,56	64,32	65,98	66,94	66,67	65,77	66,09
6	Cleveland	54,93	55,92	57,06	56,83	55,76	56,12	57,68	54,13	56,37	54,30	56,16
7	Contraceptive	50,38	51,53	51,89	53,46	54,63	52,03	52,80	53,97	52,33	51,99	52,01
8	Dermatology	93,75	87,87	89,39	92,53	92,81	93,20	92,74	93,28	93,00	93,28	93,20
9	Ecoli	76,52	74,42	75,42	77,29	77,49	77,52	78,50	79,38	77,79	78,53	79,59
10	Glass	61,67	60,92	61,40	63,54	63,17	63,14	62,90	63,23	64,11	63,09	63,06
11	Haberman	71,67	72,30	73,93	72,98	71,77	71,99	72,43	72,08	72,96	72,63	72,63
12	hayes-roth	77,92	77,71	80,21	77,50	76,04	79,17	75,63	79,58	78,75	78,75	77,71
13	Heart	80,25	79,38	80,00	79,01	79,51	76,79	77,90	77,90	79,01	78,15	78,02
14	Hepatitis	85,19	83,87	85,11	82,86	84,79	82,56	82,27	83,79	83,79	80,25	80,11
15	Ionosphere	90,43	88,04	88,90	88,43	88,34	89,37	88,99	89,46	89,85	90,22	89,75
16	Iris	96,00	95,11	95,11	96,67	96,00	96,22	96,00	96,00	95,56	96,00	95,78
17	Magic	83,00	82,42	82,47	83,01	83,21	82,89	82,86	83,03	82,96	82,92	82,90
18	Mammographic	82,81	82,92	83,56	83,08	82,63	82,51	82,60	83,56	82,93	83,00	83,05
19	Movement_libras	44,63	37,69	38,24	43,06	44,07	43,98	42,50	42,69	44,26	45,28	45,09
20	Newthyroid	93,80	93,64	94,11	93,33	92,40	92,56	92,73	95,04	94,28	93,82	93,98
21	Page-blocks	94,47	94,26	94,32	94,38	94,36	94,37	94,48	94,55	94,38	94,46	94,59
22	Phoneme	79,33	79,54	79,61	79,75	79,48	79,31	79,74	79,78	79,58	79,41	79,89
23	Pima	74,20	76,14	75,19	74,62	75,19	74,19	74,10	74,28	74,54	74,54	75,54
24	Ring	81,19	80,84	81,02	86,60	86,74	86,51	86,89	86,75	86,98	86,59	86,98
25	Satimage	82,58	79,78	79,84	82,55	82,66	82,66	82,49	82,65	82,62	82,69	82,68
26	Sonar	75,54	73,38	73,37	73,83	71,75	72,71	75,11	72,21	73,46	73,52	72,86
27	Spambase	88,21	87,10	87,24	90,05	90,46	89,94	90,31	90,14	90,51	90,15	90,44
28	Spectfheart	79,04	78,41	79,17	79,93	79,06	78,68	78,68	79,45	78,56	76,96	77,68
29	Tae	51,41	49,74	53,31	50,64	51,54	50,83	49,54	53,50	55,03	53,47	52,38
30	Thyroid	94,26	94,26	94,31	94,29	94,32	94,33	94,29	94,32	94,33	94,32	94,32
31	Titanic	77,15	78,51	78,59	78,89	78,63	78,96	78,96	78,98	78,77	78,96	78,96
32	Twonorm	87,13	86,95	87,30	89,75	89,85	89,63	90,18	90,00	89,47	89,89	90,30
33	Vehicle	65,85	61,44	61,87	66,52	65,65	65,22	66,16	67,54	66,68	66,68	66,24
34	Vowel	42,79	33,50	34,01	44,24	43,43	42,29	42,79	44,28	43,87	42,73	42,63
35	Wdbc	94,84	94,72	94,55	92,91	93,61	94,85	93,73	93,78	94,02	94,96	94,55

Продолжение Таблицы 3.13

№	Наборы данных	Альтернативы			Метод разделения данных							
		MI	Random		V				V+T			
			V	V+T	φ_1	φ_2	φ_3	φ_4	φ_1	φ_2	φ_3	φ_4
36	Wine	94,00	94,54	93,78	93,42	93,22	93,98	93,43	93,59	94,17	93,76	93,43
37	Wisconsin	95,63	95,67	96,05	95,58	95,82	95,72	95,43	95,37	95,83	96,21	95,33
38	Yeast	53,17	52,25	52,68	53,17	52,97	53,30	52,97	53,44	53,73	53,44	53,33

Таблица 3.14 – Значение точности классификации (%) вариантов предлагаемого метода и альтернативных методов с агрегацией правил функцией суммы

№	Наборы данных	Альтернативы			Метод разделения данных							
		MI	Random		V				V+T			
			V	V+T	φ_1	φ_2	φ_3	φ_4	φ_1	φ_2	φ_3	φ_4
1	Appendicitis	85,06	82,24	81,64	83,91	81,94	84,09	81,06	85,79	85,58	85,36	83,27
2	Australian	84,88	85,22	85,75	84,83	85,17	84,44	85,56	85,46	85,60	85,12	85,65
3	Balance	90,55	89,86	90,02	90,82	90,98	90,61	90,93	90,82	90,66	90,93	91,04
4	Banana	75,91	76,21	76,19	76,61	76,38	76,83	76,35	76,61	76,31	76,79	76,45
5	Bands	66,17	65,04	65,84	63,54	64,34	65,46	64,54	65,49	68,11	65,70	67,26
6	Cleveland	56,16	56,17	56,70	57,55	56,93	56,96	57,91	53,94	55,69	54,24	53,79
7	Contraceptive	53,30	51,96	51,49	53,43	52,37	52,89	52,03	53,61	53,95	53,20	52,19
8	Dermatology	92,18	88,65	89,24	91,97	93,08	91,97	93,11	92,54	92,55	92,26	94,13
9	Ecoli	77,19	76,05	78,62	77,23	78,10	79,69	76,02	77,53	77,58	80,27	77,61
10	Glass	63,81	61,55	63,09	61,35	60,66	61,04	63,76	63,99	64,44	62,97	62,51
11	Haberman	71,33	73,40	73,70	72,21	72,65	72,53	72,97	71,65	71,45	72,74	72,19
12	hayes-roth	78,13	76,88	79,17	78,33	78,33	78,13	78,33	80,00	78,33	80,42	80,00
13	Heart	83,58	81,48	81,85	82,47	81,85	81,48	80,25	82,47	82,72	81,11	82,10
14	Hepatitis	84,27	84,83	87,47	83,19	81,98	81,09	84,72	83,13	86,71	82,22	83,58
15	Ionosphere	90,89	88,33	88,42	90,04	90,61	90,42	90,70	91,08	91,37	91,56	90,98
16	Iris	96,89	94,89	94,67	97,11	97,11	96,22	96,89	95,78	95,78	95,78	96,44
17	Magic	82,42	82,49	82,51	82,68	82,95	82,70	82,91	82,68	82,65	82,74	82,95
18	Mammographic	83,72	82,20	83,41	82,63	82,87	82,91	82,97	83,29	83,45	83,33	83,53
19	Movement_libras	43,70	39,54	40,65	43,61	44,91	43,70	44,17	45,37	45,56	45,09	45,83
20	Newthyroid	95,34	93,07	93,83	93,66	93,02	92,57	93,01	95,68	94,41	94,12	93,65
21	page-blocks	94,53	94,49	94,60	94,41	94,53	94,58	94,40	94,58	94,72	94,71	94,47
22	Phoneme	79,08	78,74	78,89	78,98	78,81	79,08	79,00	79,16	79,21	79,22	79,06
23	Pima	75,62	75,66	75,23	74,93	76,10	75,83	74,88	75,14	75,92	75,53	75,71
24	Ring	91,80	85,78	85,88	92,76	92,25	92,12	92,53	92,73	92,69	92,12	92,53
25	Satimage	82,81	80,60	80,96	82,81	82,62	82,64	82,81	82,80	82,85	82,92	82,94
26	Sonar	73,30	72,67	71,43	72,69	73,81	73,21	73,67	76,02	73,94	74,32	73,18
27	Spambase	89,70	85,69	85,84	89,78	89,93	89,66	90,04	89,82	89,91	89,73	90,23
28	Spectfheart	78,06	78,94	79,22	78,93	79,05	79,16	78,80	77,56	78,18	79,30	79,81
29	Tae	55,86	50,83	53,71	50,85	52,36	49,58	50,21	54,63	52,83	52,39	53,03
30	Thyroid	94,23	94,25	94,26	94,21	94,24	94,19	94,23	94,25	94,30	94,22	94,25
31	Titanic	78,83	78,53	78,57	78,68	78,60	78,54	78,53	78,78	78,69	78,72	78,59
32	Twonorm	93,61	88,99	89,27	93,53	93,62	93,52	93,39	93,66	94,04	93,53	93,55
33	Vehicle	66,25	61,71	61,35	65,33	65,96	66,55	65,53	66,12	66,20	67,37	66,36
34	Vowel	42,53	32,53	32,73	43,54	43,03	43,64	42,15	43,80	42,56	44,61	42,59
35	Wdbc	96,02	95,37	95,73	95,07	95,19	95,49	94,66	96,02	96,08	95,84	95,54
36	Wine	93,64	92,55	95,69	93,40	94,18	92,15	93,99	94,16	94,53	94,38	94,36
37	Wisconsin	96,07	95,43	95,97	95,81	96,11	95,71	95,86	95,67	96,12	95,17	95,96

Продолжение Таблицы 3.14

№	Наборы данных	Альтернативы			Метод разделения данных							
		MI	Random		V				V+T			
			V	V+T	φ_1	φ_2	φ_3	φ_4	φ_1	φ_2	φ_3	φ_4
38	Yeast	53,53	53,44	53,44	52,70	53,19	54,02	52,81	53,17	53,94	53,87	53,28

В таблице 3.15 приведены средние по запускам значения показателей интерпретируемости в ходе проведения эксперимента. Представлены результаты только для варианта метода с целевой функцией φ_2 (выражение (2.10)) и выбора итогового классификатора на Парето фронте с помощью всей исходной выборки, поскольку, как будет показано ниже, такое сочетание функции и выборки показало лучшую точность классификации. В таблице введены следующие обозначения: $\#R$ – количество правил, $\#T$ – общее количество нечетких терминов в классификаторе, $\#F$ – количество используемых признаков.

Для демонстрации скорости сходимости дискретного алгоритма при разделении данных на рисунке 3.9 представлены графики изменения значений целевой функции φ_2 (выражение (2.10)) по мере прохождения итераций алгоритма. Приведены средние значения целевой функции по тридцати запускам алгоритма. Для примера представлены графики наборов данных Hepatitis, Banana, Wine и Glass.

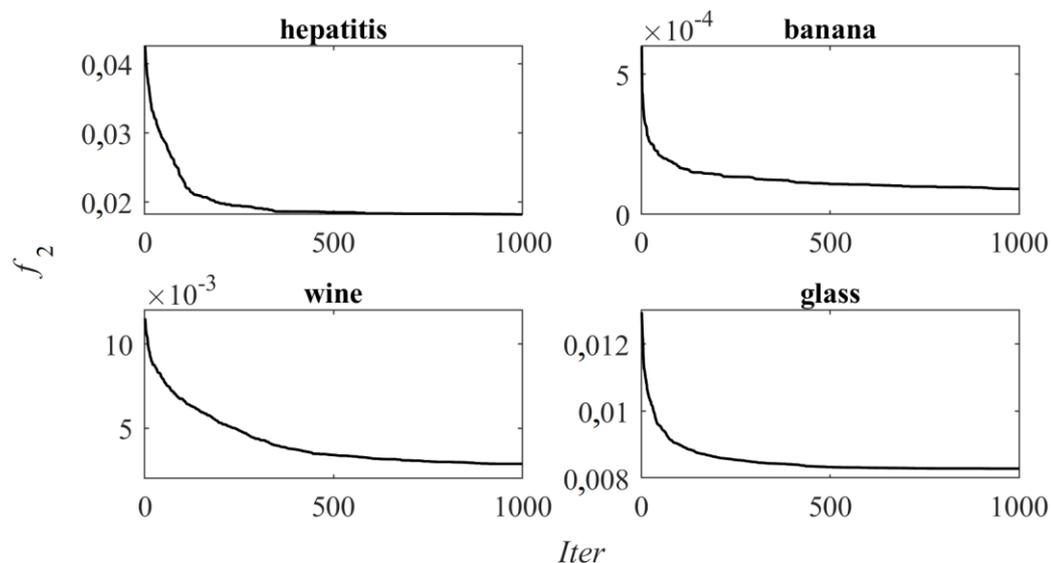


Рисунок 3.9 – Графики зависимости значений целевой функции от итерации при работе дискретного алгоритма при разделении данных

Таблица 3.15 – Значения показателей интерпретируемости в ходе проведения эксперимента

№	Набор данных	Агрегация максимум						Агрегация сумма					
		M1			Метод разделения			M1			Метод разделения		
		#R	#T	#F	#R	#T	#F	#R	#T	#F	#R	#T	#F
1	Appendicitis	3,03	4,97	3,90	2,83	4,27	3,67	2,47	3,77	3,43	2,27	3,20	3,10
2	Australian	6,90	12,53	8,90	6,80	11,00	8,27	5,90	9,10	7,30	4,87	7,57	6,50
3	Balance	2,50	5,80	4,00	2,43	4,93	4,00	3,60	4,33	4,00	2,53	4,67	4,00
4	Banana	4,50	6,73	2,00	5,00	8,50	2,00	5,37	5,93	2,00	4,90	5,97	2,00
5	Bands	9,00	15,27	11,03	8,80	15,57	11,33	7,17	12,17	9,67	7,50	12,47	9,57
6	Cleveland	10,83	27,00	11,70	11,20	26,93	11,40	7,57	19,37	10,30	8,53	20,77	10,37
7	Contraceptive	6,57	16,00	7,37	7,17	17,80	7,70	5,77	14,13	7,13	6,20	15,47	7,60
8	Dermatology	12,10	23,57	16,43	10,83	20,70	15,30	13,17	29,37	18,90	10,07	19,93	15,27
9	Ecoli	11,90	24,50	6,90	11,43	22,63	6,87	10,80	19,70	6,83	9,90	18,37	6,83
10	Glass	9,70	19,47	8,23	9,23	17,13	8,30	8,33	16,93	7,77	7,80	15,20	7,73
11	Haberman	3,17	4,40	2,43	3,00	3,80	2,40	3,57	4,67	2,47	2,60	3,57	2,57
12	Hayes-roth	7,67	11,37	3,57	7,47	10,67	3,57	7,20	8,70	3,17	7,63	9,17	3,37
13	Heart	7,73	13,80	9,37	7,53	13,10	9,00	6,93	8,73	7,47	6,33	7,97	6,87
14	Hepatitis	4,87	8,07	7,03	4,00	6,30	5,53	3,43	5,57	5,33	2,97	4,43	4,07
15	Ionosphere	8,80	13,20	11,33	8,60	11,90	10,43	6,40	8,97	7,90	5,60	8,00	7,40
16	Iris	3,67	4,37	2,63	3,33	4,00	2,63	3,07	3,13	2,07	3,23	3,60	2,40
17	Magic	7,77	15,77	8,07	8,10	16,53	8,20	5,60	10,13	6,57	5,10	9,13	5,90
18	Mammographic	4,63	7,53	4,27	4,87	7,87	4,40	4,37	5,57	3,83	3,73	5,10	3,67
19	Movement_libras	21,43	62,37	44,77	21,73	62,43	45,67	23,30	68,13	48,20	25,07	73,20	51,10
20	Newthyroid	4,27	7,77	4,47	3,57	6,00	3,97	4,40	7,20	4,03	3,73	6,10	3,90
21	Page-blocks	9,30	17,97	8,67	9,03	16,73	8,43	8,67	16,57	8,30	8,40	15,43	7,93
22	Phoneme	5,17	9,97	5,00	5,50	10,03	4,93	4,20	7,17	4,97	4,10	7,07	5,00
23	Pima	7,07	13,20	7,07	6,87	12,33	7,00	4,73	6,93	5,10	4,53	6,30	4,87
24	Ring	12,23	16,83	13,17	11,93	16,03	13,43	13,77	18,00	14,90	13,87	17,07	15,20
25	Satimage	15,00	38,10	23,40	14,37	36,50	23,53	17,97	48,10	26,93	18,50	49,47	26,63
26	Sonar	8,30	17,87	16,20	7,23	14,80	13,13	10,40	15,73	14,17	8,10	11,60	10,67
27	Spambase	8,10	15,87	14,53	8,73	16,80	15,33	9,43	16,80	14,40	9,63	17,80	15,23
28	Spectfheart	5,63	12,33	11,00	5,20	10,90	9,80	2,17	3,33	3,20	2,50	4,03	3,97
29	Tae	5,90	11,93	4,87	5,37	10,70	4,93	5,53	11,23	4,93	4,87	9,57	4,80
30	Thyroid	6,37	12,63	10,20	6,43	12,27	10,03	4,70	7,90	6,60	4,63	7,57	6,60
31	Titanic	3,00	3,00	3,00	2,63	3,10	3,00	3,00	3,90	3,00	2,47	3,47	3,00
32	Twonorm	11,43	29,80	18,43	11,77	29,57	18,10	13,87	22,03	18,63	13,70	20,33	18,10
33	Vehicle	9,70	22,87	13,30	9,87	23,23	12,83	9,57	23,17	13,30	8,83	20,53	12,70
34	Vowel	12,77	36,23	11,27	13,53	37,67	11,43	12,40	35,57	11,50	13,00	36,63	11,20
35	Wdbc	6,17	10,80	9,40	4,87	8,73	8,03	6,33	8,33	7,23	4,77	6,30	5,87
36	Wine	5,57	7,70	6,40	4,10	5,40	4,87	5,53	7,43	6,33	3,97	5,47	5,03
37	Wisconsin	5,80	9,73	6,80	5,30	7,97	6,23	4,83	6,70	5,90	3,40	5,07	4,73
38	Yeast	10,33	23,97	7,93	10,63	24,27	7,97	10,03	23,40	8,00	10,30	23,93	7,97

Выводы. Для того чтобы оценить значимость различий эффективности построения классификаторов, применяют статистический анализ, в частности непараметрические тесты [261, 262, 263]. Для парных сравнений результатов

эксперимента использовался тест знаковых рангов Уилкоксона, который основан на вычислении разности между двумя выборочными средними значений показателей эффективности. Для проведения теста использовался уровень значимости $\alpha=0,05$. Нулевая гипотеза H_0 утверждает об отсутствии статистической разницы в результатах эффективности показателей классификаторов обоих методов. Альтернативная гипотеза H_1 утверждает об обратном.

Статистический анализ таблицы 3.13 и таблицы 3.14 показывает статистически значимое преимущество точности классификации одного из вариантов предлагаемого метода по сравнению с построением без разделения данных на обучение и валидацию. Результаты теста приведены в таблицах 3.16 и 3.17. Сочетание целевой функции (2.10) для разделения исходной выборки и выбора наиболее точного классификатора на фронте Парето с помощью всей исходной выборки показало превосходящие результаты, т.е. точность классификации при использовании предлагаемого метода статистически значимо превосходит аналог. Такое сочетание также показало статистически значимое преимущество в точности классификации по сравнению со случайным разделением исходных данных на обучающие и валидационные. Результаты теста приведены в таблице 3.18. Данное сочетание будет использовано в дальнейшем для построения нечетких классификаторов.

Статистический анализ значений показателей интерпретируемости таблицы 3.15 не выявил различий в количестве нечетких правил для классификаторов с агрегацией на основе функции максимума, а для функции суммы количество правил оказалось значимо меньше. Кроме этого, количество нечетких терминов и признаков классификаторов, созданных предлагаемым методом, тоже оказалось значимо меньше. Данные результаты свидетельствуют о повышении уровня интерпретируемости. Результаты теста приведены в таблице 3.19.

Таблица 3.16 – Статистические сравнения результатов точности классификации с методом без разделения данных (таблица 3.13)

Пары сравнений	Сумма рангов		<i>p</i> -value	Гипотеза H_0 ($\alpha=0,05$)
	+	-		
MI и (V, φ_1)	294	409	0,3843	Принимается
MI и (V, φ_2)	307	396	0,50286	Принимается
MI и (V, φ_3)	351,5	389,5	0,77948	Принимается
MI и (V, φ_4)	346	320	0,84148	Принимается
MI и (V+T, φ_1)	216	487	0,04136	Отвергается
MI и (V+T, φ_2)	192,5	548,5	0,00988	Отвергается
MI и (V+T, φ_3)	238	465	0,08726	Принимается
MI и (V+T, φ_4)	285,5	455,5	0,2187	Принимается

Таблица 3.17 – Статистические сравнения результатов точности классификации с методом без разделения данных (таблица 3.14)

Пары сравнений	Сумма рангов		<i>p</i> -value	Гипотеза H_0 ($\alpha=0,05$)
	+	-		
MI и (V, φ_1)	476	227	0,0601	Принимается
MI и (V, φ_2)	367	336	0,8181	Принимается
MI и (V, φ_3)	403,5	226,5	0,14706	Принимается
MI и (V, φ_4)	373	257	0,34212	Принимается
MI и (V+T, φ_1)	295,5	407,5	0,4009	Принимается
MI и (V+T, φ_2)	188,5	552,5	0,0083	Отвергается
MI и (V+T, φ_3)	309,5	431,5	0,37886	Принимается
MI и (V+T, φ_4)	348	393	0,7414	Принимается

Таблица 3.18 – Статистическое сравнение точности классификации со случайным разделением исходных данных (таблицы 3.13 и 3.14)

Агрегация	Пары сравнений	Сумма рангов		<i>p</i> -value	Гипотеза H_0 ($\alpha=0,05$)
		+	-		
Максимум	Random(V) и (V+T, φ_2)	77,5	663,5	<0,001	Отвергается
	Random(V+T) и (V+T, φ_2)	171	570	0,00386	Отвергается
Сумма	Random(V) и (V+T, φ_2)	45	696	<0,001	Отвергается
	Random(V+T) и (V+T, φ_2)	165,5	575,5	0,00298	Отвергается

Таблица 3.19 – Статистическое сравнение интерпретируемости по таблице 3.19

Агрегация	Пары сравнений	Показатель	Сумма рангов		<i>p</i> -value	Гипотеза H_0 ($\alpha=0,05$)
			+	-		
Максимум	MI и (V+T, φ_2)	#R	498,5	242,5	0,06288	Принимается
		#T	580,5	160,5	0,00228	Отвергается
		#F	408,5	152,5	0,02202	Отвергается
Сумма	MI и (V+T, φ_2)	#R	562,5	178,5	0,00544	Отвергается
		#T	541,5	199,5	0,01314	Отвергается
		#F	412,5	148,5	0,01828	Отвергается

Предложенный метод разделения данных основан на разделении исходных данных на обучающую и валидационную выборки, при котором учитываются меры близости между выборками и исходными данными. В качестве мер близости выступают отклонения таких характеристик данных как среднее значение и среднеквадратичное отклонение. Разделение проводится с помощью бинарного метаэвристического алгоритма, увеличивающего близость между выборками. Предлагаемое разделение позволяет уменьшить потерю информации для обучения и валидации по сравнению со случайным разделением. Кроме этого, использование валидационной выборки уменьшает эффект переобучения, когда построенная модель теряет обобщающую способность, но показывает высокие результаты точности на обучающих данных.

В подразделе проведено экспериментальное исследование применения метода для построения нечетких классификаторов с помощью метаэвристического алгоритма смешанной многокритериальной оптимизации. Обучающие данные, полученные разделением, использовались алгоритмом для построения множества недоминируемых по Парето классификаторов. Валидационная выборка применялась для выбора модели с лучшей обобщающей способностью.

Экспериментальное исследование позволило получить следующие результаты.

1. Установлены вид целевой функции для разделения данных и вариант формирования выборки данных для выбора модели на Парето фронте, которые позволяют получить нечеткие классификаторы со статистически значимым преимуществом в точности классификации. Найденная целевая функция основана на максимальном отклонении характеристик обучающей выборки от исходной, без учета характеристик валидационной выборки. Для выбора модели с лучшей обобщающей способностью необходимо использовать все исходные данные.

2. Разделение исходных данных на обучающую и валидационную выборки на основе введенных мер близости и применение для этого бинарного

метаэвристического алгоритма продемонстрировало статистически значимый прирост точности по сравнению со случайным разделением исходных данных.

3. Использование обучающей выборки для построения нечеткого классификатора с помощью многокритериального алгоритма и выбора классификатора с лучшей обобщающей способностью с использованием валидационной выборки продемонстрировало статистически значимый прирост точности и интерпретируемости по сравнению с моделями, построенными без применения разделения данных. Максимальный прирост точности составил 6,5% на наборе данных banana.

Таким образом, предложенный метод разделения данных на примере построения нечетких классификаторов с помощью многокритериального алгоритма оптимизации позволил повысить точность и интерпретируемость полученных моделей. Результаты были достигнуты благодаря разделению данных на обучающую и валидационную выборки с учетом сохранения близости к исходным данным, что позволило уменьшить уровень переобучения моделей и сократить потери характеристических свойств данных при их разделении.

3.4 Настройка параметров нечетких классификаторов адаптированным алгоритмом градиентного спуска

Предложенный в подразделе 2.5 адаптированный алгоритм градиентного спуска позволяет изменить значения параметров функций принадлежности нечетких классификаторов таким образом, чтобы была минимизирована функция, определяющая сумму квадратов отклонений между нормированными значениями разделяющих функций и ожидаемыми их значениями (2.2). В настоящем подразделе представлен эксперимент по построению нечетких классификаторов с настройкой параметров функций принадлежности с помощью предлагаемого алгоритма. Целью эксперимента является проверка эффективности применения адаптированного алгоритма градиентного спуска для улучшения точности

классификации моделей, построенных с помощью алгоритма смешанной многокритериальной оптимизации с применением метода разделения данных.

Методика проведения вычислительного эксперимента. Методика проведения эксперимента осуществлялась по схеме десятикратной перекрестной проверки и соответствует методике, приведенной в [83, 84, 86]. Для каждого из 10 разбиений набора данных осуществлялось построение классификатора с помощью алгоритма смешанной многокритериальной оптимизации с применением метода разделения на обучающую и валидационную выборки. Затем параметры функций принадлежности полученного классификатора изменялись адаптированным алгоритмом градиентного спуска. Такая процедура осуществлялась три раза. Таким образом, эффективность оценивалась средним значением показателя по тридцати построениям и настройкам параметров для каждого набора данных. В качестве критерия эффективности оценивалась точность классификации, где показателем выступает процент правильно классифицированных образцов тестовой выборки.

В эксперименте использовались 38 наборов данных из репозитория KEEL (<http://keel.es>), информация о которых представлена в таблице 3.12. Настройка параметров адаптированным алгоритмом градиентного спуска проводилась после построения классификаторов, представленных в подразделе 3.3, т.е. построение осуществлялось с помощью алгоритма смешанной многокритериальной оптимизации, выборки для обучения и валидации формировались с помощью метода разделения данных (см. подраздел 2.6) с применением дискретного алгоритма на основе распределения вероятностей (см. подраздел 2.4). Параметры алгоритмов для предварительной настройки классификаторов указаны в таблице 3.11. Значения параметров адаптированного алгоритма градиентного спуска представлены в таблице 3.20. Нижние и верхние границы изменения параметров функций принадлежности L и U рассчитывались согласно (2.14) и (2.15), исходя из диапазона изменений бокового смещения α и охвата β , который имеет значение $[-0,3, 0,3]$. Алгоритм использовался как в варианте с дополнительным условием для дробления шага, так и без него. В подразделе 2.5 предлагалось к условию,

выполнение которого уменьшает шаг поиска h , добавить условие падения точности классификации.

Таблица 3.20 – Значения параметров адаптированного алгоритма градиентного спуска при проведении эксперимента

Название параметра	Значение параметра
Минимальная длина шага e	$1 \cdot 10^{-8}$
Начальное значение шага h	0,1
Диапазон изменений бокового смещения α и смещения охвата β	[-0,3 0,3]

Результаты эксперимента. После проведения эксперимента были получены средние значения оценок эффективности построенных нечетких классификаторов. В таблице 3.21 приведены значения точности без применения адаптированного алгоритма (обозначение MS) и с его применением. Вариант алгоритма с дополнительным условием дробления шага обозначен GD*, без дополнительного условия – GD.

Выводы. Для того чтобы оценить значимость различий эффективности классификаторов, построенных с применением и без применения адаптированного алгоритма градиентного спуска, использовался тест знаковых рангов Уилкоксона. Нулевая гипотеза теста H_0 утверждает об отсутствии статистической разницы в результатах эффективности показателей классификаторов. Альтернативная гипотеза H_1 утверждает об обратном. Для проведения теста применялся уровень значимости $\alpha = 0,05$. Результаты теста приведены в таблице 3.22. Применение адаптированного алгоритма градиентного спуска показало значимое преимущество как для архитектуры классификатора с агрегацией правил функцией максимума, так и для архитектуры с функцией суммы. Добавление в алгоритм условия падения точности классификации для дробления шага создало незначительный перевес в сторону улучшения точности. Суммы рангов свидетельствуют в пользу применения условия, но значения p -value не преодолело критического $\alpha=0,05$. Поэтому можно говорить о том, что добавление данного условия не приведет к статистически значимому улучшению точности классификации.

Таблица 3.21 – Значение точности классификации (%) после применения адаптированного алгоритма градиентного спуска

№	Наборы данных	Агрегация максимум			Агрегация сумма		
		MS	GD	GD*	MS	GD	GD*
1	Appendicitis	86,00	85,42	86,33	85,58	84,48	85,06
2	Australian	84,73	85,31	84,98	85,60	85,80	86,09
3	Balance	88,32	87,90	88,75	90,66	90,66	91,25
4	Banana	80,04	81,21	82,84	76,31	77,90	80,27
5	Bands	66,67	66,60	66,53	68,11	67,53	67,63
6	Cleveland	56,37	56,06	56,29	55,69	56,98	56,62
7	Contraceptive	52,33	52,91	53,61	53,95	54,07	54,11
8	Dermatology	93,00	94,12	94,02	92,55	93,94	94,03
9	Ecoli	77,79	79,66	79,57	77,58	80,68	80,18
10	Glass	64,11	67,01	66,05	64,44	65,59	65,03
11	Haberman	72,96	74,14	73,39	71,45	74,14	74,04
12	Hayes-roth	78,75	80,42	79,79	78,33	79,58	79,79
13	Heart	79,01	80,49	80,74	82,72	85,06	83,33
14	Hepatitis	83,79	86,64	87,53	86,71	85,46	85,98
15	Ionosphere	89,85	90,32	90,51	91,37	90,61	91,28
16	Iris	95,56	95,78	95,56	95,78	96,00	96,44
17	Magic	82,96	82,98	83,29	82,65	82,99	83,22
18	Mammographic	82,93	83,16	83,13	83,45	83,80	84,33
19	Movement_libras	44,26	49,07	49,35	45,56	50,09	50,00
20	Newthyroid	94,28	95,36	95,22	94,41	94,44	94,73
21	page-blocks	94,38	94,04	94,44	94,72	94,52	94,85
22	Phoneme	79,58	80,04	80,45	79,21	79,06	79,35
23	Pima	74,54	74,76	75,01	75,92	76,41	76,23
24	Ring	86,61	87,45	87,31	92,69	91,32	92,80
25	Satimage	82,62	83,09	82,92	82,85	83,50	83,19
26	Sonar	73,46	76,36	75,86	73,94	77,16	75,90
27	Spambase	90,51	83,81	90,71	89,91	85,42	89,99
28	Spectfheart	78,56	78,58	79,07	78,18	79,95	78,95
29	Tae	55,03	53,49	54,58	52,83	52,19	52,83
30	Thyroid	94,33	94,09	94,36	94,30	94,13	94,30
31	Titanic	78,77	78,72	78,77	78,69	78,78	78,82
32	Twonorm	89,47	91,23	90,18	94,04	93,06	94,08
33	Vehicle	66,68	67,63	67,35	66,20	67,54	67,66
34	Vowel	43,87	45,32	45,39	42,56	43,80	43,94
35	Wdbc	94,02	95,42	95,43	96,08	95,78	96,31
36	Wine	94,17	94,88	95,26	94,53	96,60	96,24
37	Wisconsin	95,83	96,26	95,97	96,12	95,97	96,07
38	Yeast	53,73	54,61	54,59	53,94	55,55	54,56

Таблица 3.22 – Статистическое сравнение точности классификации

Агрегация	Пары сравнений	Сумма рангов		p-value	Гипотеза H_0 ($\alpha=0,05$)
		+	-		
Максимум	MS и GD	130	611	<0,001	Отвергается
	MS и GD*	21,5	644,5	<0,001	Отвергается
	GD и GD*	283	458	0,2042	Принимается
Сумма	MS и GD	200	503	0,0220	Отвергается
	MS и GD*	75,5	590,5	<0,001	Отвергается
	GD и GD*	306	435	0,3472	Принимается

Применение алгоритма для настройки параметров функций принадлежности продемонстрировало статистически значимое увеличение точности нечетких классификаторов, построенных метаэвристическим алгоритмом смешанной многокритериальной оптимизации. Максимально достигнутое увеличение точности составило 5,1% на наборе данных Movement_libras.

3.5 Построение нечетких классификаторов

В настоящем подразделе представлены результаты экспериментов для оценки эффективности методики построения нечетких классификаторов, представленной в подразделе 2.7. В эксперименте проведена оценка точности классификации и интерпретируемости. Проведено сравнение с известными интерпретируемыми моделями машинного обучения, такими как нечеткие классификаторы FARC-HD [86, 87] и деревья решений CART [269].

Нечеткие классификаторы FARC-HD разработаны для решения проблемы экспоненциального роста нечетких правил, с которой обычно сталкиваются при использовании полного сочетания лингвистических терминов в antecedentной части. Алгоритм построения позволяет уменьшить масштабируемость и сложность процесса классификации благодаря упрощению структуры и повышению интерпретируемости. Построение состоит из трех этапов. Первый этап – извлечение нечетких правил с помощью дерева поиска, которое позволяет

перечислить все возможные варианты правил. Второй этап – предварительный отбор правил-кандидатов. Данный этап позволит уменьшить количество извлеченных правил, путем отбора наиболее значимых для классификации. Третий этап – получение компактного набора нечетких правил с высокой точностью классификации из множества правил-кандидатов с помощью генетического алгоритма.

Алгоритм CART позволяет построить бинарные деревья решений, где каждый узел имеет двух потомков. Процесс классификации заключается в движении по узлам дерева в тех направлениях, где выполняются условия, накладываемые на значение признака образца. Листья дерева определяют его класс. Благодаря древовидной структуре, деревья решений являются интерпретируемыми прогностическими моделями и могут быть представлены в виде продукционных правил. Алгоритм построения дерева формирует условия в узловых вершинах таким образом, чтобы минимизировать неопределенность, связанную с образцами различных классов обучающих данных. Мерой неопределенности выступает индекс Джини, основанный на относительных частотах появления образцов определенного класса.

Методика проведения вычислительного эксперимента. Построение нечетких классификаторов FARC-ND осуществлялось с помощью программного обеспечения для анализа данных KEEL 3.0 [270]. Параметры алгоритма построения выставлены в значения, рекомендованные разработчиками, и приведены в таблице 3.23. Использовались три нечетких термина на каждый признак для классификаторов FARC-ND и предлагаемой методики, что соответствует понимаемому человеку разделению пространства на термины «Малое», «Среднее» и «Большое».

Построение деревьев решений осуществлялось в среде MATLAB с помощью реализации алгоритма CART в библиотеке «Statistics and Machine Learning Toolbox». Параметры алгоритма подбирались с помощью байесовской оптимизации [271]. Диапазон изменения параметров представлен в таблице 3.23.

Параметры для проведения построения нечетких классификаторов с помощью предлагаемой методики представлены в таблице 3.24.

Таблица 3.23 – Параметры альтернативных интерпретируемых классификаторов

Модель классификации	Название параметра	Значение
FARC-HD	Количество лингвистических терминов на переменную	3
	Минимальный коэффициент поддержки (support)	0,05
	Максимальный коэффициент уверенности (confidence)	0,8
	Максимальное количество терминов в правиле	3
	Максимальное число итераций	15000
	Размер популяции	100
	Параметр альфа (α)	0,15
CART	Максимальное количество узлов в дереве	[1, к-во образцов – 1]
	Минимальное количество образцов в листовых узлах	[1, к-во образцов / 2]
	Максимальная глубина дерева	[1, максимальное количество узлов]

Таблица 3.24 – Параметры метода разделения и алгоритмов оптимизации в методике построения нечетких классификаторов

	Название параметра	Значение параметра
Общие параметры	Отношение количество образцов $a:b$ для разделения данных	3:1
	Количество лингвистических терминов на переменную h	Для всех признаков 3
	Максимальное количество терминов в правиле	3
	Диапазон изменений бокового смещения α и смещения охвата β	[-0,3 0,3]
Параметры дискретного алгоритма	Размер популяции	30
	Число итераций	1000
	Функция трансформации T	Линейная T_L
	Вероятность мутации p	0,05
Параметры алгоритма смешанной оптимизации	Размер популяции	100
	Число итераций	5000
	Максимальное количество правил r	Кол-во классов $\times 7$
	Вероятность удаления гнезда p_a	0,05
Параметры алгоритма градиентного спуска	Коэффициент прыжка полетов Леви γ	Равномерно уменьшался с 0,4 до 0,05
	Минимальная длина шага e	$1 \cdot 10^{-8}$
	Начальное значение шага h	0,1
	Диапазон изменений бокового смещения α и смещения охвата β	[-0,3 0,3]

Методика проведения эксперимента осуществлялась по схеме десятикратной перекрестной проверки и соответствует методике, приведенной в [83, 84, 86]. Для каждого из 10 разбиений данных построение классификаторов

осуществлялось три раза. Таким образом, эффективность оценивалась средним значением показателей по тридцати построениям для каждого набора данных. Определены показатель точности – процент правильно классифицированных образцов, и показатели интерпретируемости – количество правил, количество признаков, общее количество используемых в классификаторе нечетких терминов, количество нечетких терминов в правиле. Для деревьев решений CART показателю количества нечетких терминов в правиле соответствует количество условий в правиле дерева. В эксперименте использовались 38 наборов данных различных проблемных областей из репозитория KEEL (<http://keel.es>), информация о которых представлена в таблице 3.12.

Результаты эксперимента. После проведения эксперимента были получены средние значения оценок эффективности построенных прогностических моделей классификации. В таблице 3.25 представлены результаты классификаторов, построенных с помощью предлагаемой методики. Точность классификации обозначена *acc*, количество правил *#R*, количество признаков *#F*, общее количество нечетких терминов *#T*, количество нечетких терминов в правиле *#D*. Результаты работы классификаторов FARC-HD и деревьев решений CART приведены в таблице 3.26. На наборе данных Satimage не удалось провести эксперимент по построению классификатора FARC-HD, происходил сбой в программном обеспечении KEEL 3.0., поэтому результаты работы для этого набора в таблице указаны как NaN.

Таблица 3.25 – Значения показателей эффективности классификаторов, построенных с помощью предлагаемой методики

№	Набор данных	Агрегация максимум					Агрегация сумма				
		<i>acc</i>	<i>#R</i>	<i>#F</i>	<i>#T</i>	<i>#D</i>	<i>acc</i>	<i>#R</i>	<i>#F</i>	<i>#T</i>	<i>#D</i>
1	Appendicitis	86,33	2,83	3,67	4,27	1,81	85,06	2,27	3,10	3,20	1,52
2	Australian	84,98	6,80	8,27	11,00	1,97	86,09	4,87	6,50	7,57	2,07
3	Balance	88,75	2,43	4,00	4,93	2,21	91,25	2,53	4,00	4,67	1,72
4	Banana	82,84	5,00	2,00	8,50	1,78	80,27	4,90	2,00	5,97	1,54
5	Bands	66,53	8,80	11,33	15,57	2,19	67,63	7,50	9,57	12,47	2,47
6	Cleveland	56,29	11,20	11,40	26,93	2,78	56,62	8,53	10,37	20,77	2,81
7	Contraceptive	53,61	7,17	7,70	17,80	2,52	54,11	6,20	7,60	15,47	2,49
8	Dermatology	94,02	10,83	15,30	20,70	2,75	94,03	10,07	15,27	19,93	2,89
9	Ecoli	79,57	11,43	6,87	22,63	2,25	80,18	9,90	6,83	18,37	2,59

Продолжение Таблицы 3.25

№	Набор данных	Агрегация максимум					Агрегация сумма				
		<i>acc</i>	<i>#R</i>	<i>#F</i>	<i>#T</i>	<i>#D</i>	<i>acc</i>	<i>#R</i>	<i>#F</i>	<i>#T</i>	<i>#D</i>
10	Glass	66,05	9,23	8,30	17,13	2,09	65,03	7,80	7,73	15,20	2,57
11	Haberman	73,39	3,00	2,40	3,80	1,68	74,04	2,60	2,57	3,57	1,42
12	Hayes-roth	79,79	7,47	3,57	10,67	1,55	79,79	7,63	3,37	9,17	1,35
13	Heart	80,74	7,53	9,00	13,10	2,02	83,33	6,33	6,87	7,97	1,82
14	Hepatitis	87,53	4,00	5,53	6,30	1,91	85,98	2,97	4,07	4,43	2,05
15	Ionosphere	90,51	8,60	10,43	11,90	2,55	91,28	5,60	7,40	8,00	2,58
16	Iris	95,56	3,33	2,63	4,00	1,25	96,44	3,23	2,40	3,60	1,25
17	Magic	83,29	8,10	8,20	16,53	2,20	83,22	5,10	5,90	9,13	2,14
18	Mammographic	83,13	4,87	4,40	7,87	1,86	84,33	3,73	3,67	5,10	1,41
19	Movement_libras	49,35	21,73	45,67	62,43	2,99	50,00	25,07	51,10	73,20	2,99
20	Newthyroid	95,22	3,57	3,97	6,00	1,82	94,73	3,73	3,90	6,10	1,71
21	Page-blocks	94,44	9,03	8,43	16,73	2,47	94,85	8,40	7,93	15,43	2,69
22	Phoneme	80,45	5,50	4,93	10,03	2,24	79,35	4,10	5,00	7,07	1,69
23	Pima	75,01	6,87	7,00	12,33	1,95	76,23	4,53	4,87	6,30	1,54
24	Ring	87,31	11,93	13,43	16,03	1,97	92,80	13,87	15,20	17,07	1,99
25	Satimage	82,92	14,37	23,53	36,50	2,87	83,19	18,50	26,63	49,47	2,91
26	Sonar	75,86	7,23	13,13	14,80	2,92	75,90	8,10	10,67	11,60	2,86
27	Spambase	90,71	8,73	15,33	16,80	2,92	89,99	9,63	15,23	17,80	2,89
28	Spectfheart	79,07	5,20	9,80	10,90	2,69	78,95	2,50	3,97	4,03	2,76
29	Tae	54,58	5,37	4,93	10,70	2,25	52,83	4,87	4,80	9,57	2,25
30	Thyroid	94,36	6,43	10,03	12,27	2,60	94,30	4,63	6,60	7,57	2,71
31	Titanic	78,77	2,63	3,00	3,10	1,26	78,82	2,47	3,00	3,47	1,37
32	Twonorm	90,18	11,77	18,10	29,57	2,81	94,08	13,70	18,10	20,33	2,34
33	Vehicle	67,35	9,87	12,83	23,23	2,67	67,66	8,83	12,70	20,53	2,81
34	Vowel	45,39	13,53	11,43	37,67	2,82	43,94	13,00	11,20	36,63	2,87
35	Wdbc	95,43	4,87	8,03	8,73	2,40	96,31	4,77	5,87	6,30	2,39
36	Wine	95,26	4,10	4,87	5,40	1,37	96,24	3,97	5,03	5,47	2,09
37	Wisconsin	95,97	5,30	6,23	7,97	1,88	96,07	3,40	4,73	5,07	1,37
38	Yeast	54,59	10,63	7,97	24,27	2,40	54,56	10,30	7,97	23,93	2,64
Среднее значение		79,35	7,67	9,41	15,50	2,23	79,72	7,11	8,78	13,72	2,20

Таблица 3.26 – Значения показателей эффективности альтернативных моделей классификации

№	Набор данных	FARC-HD					CART			
		<i>acc</i>	<i>#R</i>	<i>#F</i>	<i>#T</i>	<i>#D</i>	<i>acc</i>	<i>#R</i>	<i>#F</i>	<i>#D</i>
1	Appendicitis	85,79	3,50	4,70	8,27	2,34	82,33	3,13	1,97	1,66
2	Australian	85,12	15,80	13,17	41,70	2,63	85,07	4,47	3,23	2,32
3	Balance	90,18	26,6	4,00	60,00	2,25	77,60	25,43	4,00	5,06
4	Banana	80,26	6,50	2,00	9,83	1,53	89,04	54,17	2,00	6,48
5	Bands	69,60	7,37	11,67	20,80	2,82	61,48	15,87	7,30	4,78
6	Cleveland	57,05	43,96	12,67	127,33	2,90	54,44	7,93	4,33	2,48
7	Contraceptive	48,68	15,93	8,87	44,63	2,80	56,71	12,87	4,47	4,37
8	Dermatology	95,80	15,20	18,47	35,53	2,34	94,67	10,10	8,07	4,23
9	Ecoli	80,87	20,93	7,00	55,07	2,63	77,49	12,23	4,77	4,15
10	Glass	67,37	13,30	8,93	33,90	2,54	69,54	17,30	7,57	4,69
11	Haberman	72,04	3,33	3,71	5,67	1,69	72,09	1,80	0,67	1,48

Продолжение Таблицы 3.26

№	Набор данных	FARC-HD					CART			
		<i>acc</i>	<i>#R</i>	<i>#F</i>	<i>#T</i>	<i>#D</i>	<i>acc</i>	<i>#R</i>	<i>#F</i>	<i>#D</i>
12	Hayes-roth	40,21	8,43	3,90	20,13	2,35	81,25	13,90	3,37	5,75
13	Heart	82,72	18,67	12,70	51,77	2,77	76,05	6,73	4,20	2,81
14	Hepatitis	84,36	9,60	13,43	23,27	2,43	80,93	2,60	1,50	1,76
15	Ionosphere	91,66	11,10	16,10	23,77	2,14	90,24	7,87	5,83	3,02
16	Iris	95,11	3,87	2,87	6,47	1,66	95,33	4,00	1,93	2,22
17	Magic	82,86	10,3	7,93	23,60	2,28	84,68	117,50	9,77	7,67
18	Mammographic	79,60	6,13	4,77	16,40	2,70	83,20	4,20	2,90	2,39
19	Movement_libras	63,33	52,00	59,97	154,00	2,96	64,91	43,43	29,67	6,12
20	Newthyroid	95,71	4,70	4,70	9,57	2,02	93,33	6,73	2,87	2,96
21	Page-blocks	92,95	5,30	5,73	11,10	2,08	96,81	28,97	8,03	6,28
22	Phoneme	79,93	8,23	5,00	12,77	1,55	86,84	247,17	5,00	10,77
23	Pima	75,80	9,07	7,67	24,03	2,65	72,87	9,37	4,13	3,24
24	Ring	93,87	18,13	18,23	33,83	1,86	89,12	207,23	20,00	13,79
25	Satimage	NaN	NaN	NaN	NaN	NaN	86,17	114,30	32,07	8,15
26	Sonar	79,55	9,60	17,77	25,50	2,66	75,87	8,27	6,87	3,35
27	Spambase	88,70	10,97	17,93	26,07	2,38	92,56	114,97	35,07	10,00
28	Spectfheart	79,42	13,00	1,47	1,80	1,00	78,04	2,23	1,20	1,41
29	Tae	55,67	7,20	4,87	17,00	2,35	56,81	19,33	4,93	5,37
30	Thyroid	94,09	3,20	4,03	7,07	2,22	99,60	14,17	7,77	5,42
31	Titanic	78,83	4,03	3,00	5,77	1,43	78,66	6,63	2,93	2,87
32	Twonorm	96,59	27,20	19,93	70,03	2,57	84,73	270,80	20,00	8,81
33	Vehicle	47,33	6,37	10,63	19,10	3,00	71,70	58,10	16,70	7,77
34	Vowel	40,24	19,83	11,90	57,00	2,87	78,59	88,87	12,60	7,61
35	Wdbc	94,31	5,03	8,77	9,63	1,90	93,26	7,70	5,40	3,35
36	Wine	96,62	4,70	6,70	9,73	2,07	93,04	5,43	3,87	2,64
37	Wisconsin	96,35	5,80	6,30	9,00	1,55	95,72	9,00	4,53	3,38
38	Yeast	56,97	11,63	7,10	31,50	2,70	56,88	21,57	5,87	5,54
Среднее значение		78,26	12,61	10,23	30,88	2,29	80,46	42,27	8,09	4,90

Чтобы оценить работу деревьев решений с количеством условий в правиле, совпадающим с количеством терминов в правиле у нечетких классификаторов, т.е. не более трех, был проведен эксперимент, где максимальное значение глубины дерева не превышало трех уровней. В таблице 3.27 приведены значения точности и показателей интерпретируемости.

Интерпретируемость нечетких классификаторов. База правил нечетких классификаторов является источником знаний и позволяет понять, каким образом класс образца данных определяется значениями его признаков. На рисунке 3.10 изображено графическое представление базы правил классификаторов, построенных для набора данных Wdbc. Набор данных содержит характеристики ядер клеток опухоли молочной железы, определенных по оцифрованному

изображению тонкоигольной аспирационной биопсии. Задача классификации заключается в том, чтобы определить, является ли обнаруженная опухоль доброкачественной или злокачественной. Набор данных содержит 30 признаков, 569 образцов и 2 класса: М – опухоль злокачественная, В – опухоль доброкачественная.

Таблица 3.27 – Значения показателей эффективности деревьев решений CART с глубиной не более трех уровней

№	Набор данных	acc	#R	#F	#D	№	Набор данных	Acc	#R	#F	#D
1	Appendicitis	81,45	4,50	3,10	2,20	20	Newthyroid	92,10	5,70	2,50	2,61
2	Australian	84,93	5,70	4,70	2,60	21	Page-blocks	95,94	7,00	4,00	2,86
3	Balance	66,08	6,40	3,90	2,74	22	Phoneme	76,15	4,40	2,70	2,37
4	Banana	81,11	7,00	2,00	2,86	23	Pima	73,32	4,70	2,40	2,36
5	Bands	57,66	6,00	4,50	2,65	24	Ring	73,73	4,00	3,00	2,25
6	Cleveland	53,21	7,40	5,20	2,91	25	Satimage	76,94	7,00	4,20	2,86
7	Contraceptive	50,79	4,70	2,90	2,47	26	Sonar	71,12	7,50	6,30	2,93
8	Dermatology	79,03	5,00	4,00	2,40	27	Spambase	87,84	8,00	6,50	3,00
9	Ecoli	79,17	7,10	4,10	2,85	28	Spectfheart	71,47	6,10	5,10	2,71
10	Glass	68,41	6,90	4,90	2,84	29	Tae	53,71	7,00	4,30	2,85
11	Haberman	74,44	6,10	2,60	2,66	30	Thyroid	99,06	5,00	4,00	2,60
12	Hayes-roth	60,00	4,00	3,00	2,25	31	Titanic	78,92	6,00	3,00	2,67
13	Heart	76,67	6,70	4,20	2,80	32	Twonorm	74,34	6,50	5,00	2,76
14	Hepatitis	80,94	5,50	3,80	2,53	33	Vehicle	63,72	7,10	4,70	2,87
15	Ionosphere	90,33	4,40	3,20	2,33	34	Vowel	34,55	7,40	3,90	2,91
16	Iris	95,33	3,90	1,90	2,19	35	Wdbc	92,44	6,30	4,90	2,71
17	Magic	79,09	4,80	2,70	2,31	36	Wine	89,90	6,40	4,30	2,76
18	Mammographic	83,77	4,40	3,30	2,32	37	Wisconsin	94,59	5,80	4,10	2,64
19	Movement_libras	34,72	7,80	6,30	2,97	38	Yeast	55,12	7,30	5,00	2,90

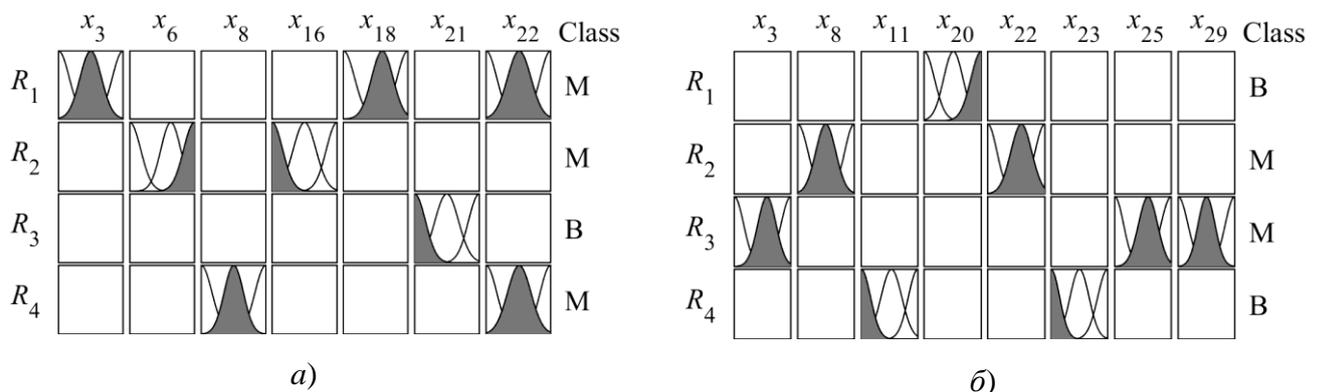


Рисунок 3.10 – Графическое представление базы правил нечетких классификаторов, построенных для набора данных Wdbc

На рисунке 3.10 представлены базы правил классификаторов, построенных с помощью предлагаемой методики в двух запусках построения. Полученные

классификаторы обладают одинаковой точностью $acc=96,49\%$. Следующие показатели интерпретируемости имеют одинаковое значение: $\#R=4$, $\#T=8$, $\#D=2$. Число признаков оказалось разным. На рисунке 3.10а $\#F=7$, а на 3.10б $\#F=8$. Следует отметить, что в исходном наборе 30 признаков. В строках на рисунке представлены правила R_1, \dots, R_4 , в столбцах представлены признаки. Последний столбец описывает класс правила. На пересечении правила и признака отображаются функции принадлежности лингвистических терминов «Малое», «Среднее», «Большое». Выбранный в правиле термин выделен заливкой. Если термины не отображаются, то в данном правиле соответствующий признак не используется. База правил нечеткого классификатора может быть представлена на естественном языке с учетом лингвистических терминов и названий признаков. Для рисунка 3.10а она выглядит следующим образом.

R_1 : ЕСЛИ «периметр 1-го ядра» средний И «количество вогнутых частей контура 2-го ядра» среднее И «текстура 3-го ядра» средняя ТО опухоль злокачественная.

R_2 : ЕСЛИ «компактность 1-го ядра» высокая И «компактность 2-го ядра низкая» ТО опухоль злокачественная.

R_3 : ЕСЛИ «радиус 3-го ядра» маленький ТО опухоль доброкачественная.

R_4 : ЕСЛИ «количество вогнутых частей контура 1-го ядра» среднее И «текстура 3-го ядра» средняя ТО опухоль злокачественная.

Аналогичным образом на естественный язык может быть переведена база правил рисунка 3.10б.

R_1 : ЕСЛИ «фрактальная размерность 2-го ядра» высокая ТО опухоль доброкачественная.

R_2 : ЕСЛИ «количество вогнутых частей контура 1-го ядра» среднее И «текстура 3-го ядра средняя» ТО опухоль злокачественная.

R_3 : ЕСЛИ «периметр 1-го ядра» средний И «гладкость 3-го ядра» средняя И «симметрия 3-го ядра средняя» ТО опухоль злокачественная.

R_4 : ЕСЛИ «радиус 2-го ядра» малый И «периметр 3-го ядра» малый ТО опухоль доброкачественная.

Дерево решений CART тоже формирует базу продукционных правил, где каждый узел дерева определяет условие в антецедентной части. На рисунке 3.11 представлено дерево решений, полученное в ходе проведения эксперимента для набора данных $Wdbc$. Дерево обладает следующими показателями эффективности: $acc=98,25\%$, $\#R=8$, $\#F=5$, $\#D=3,38$.

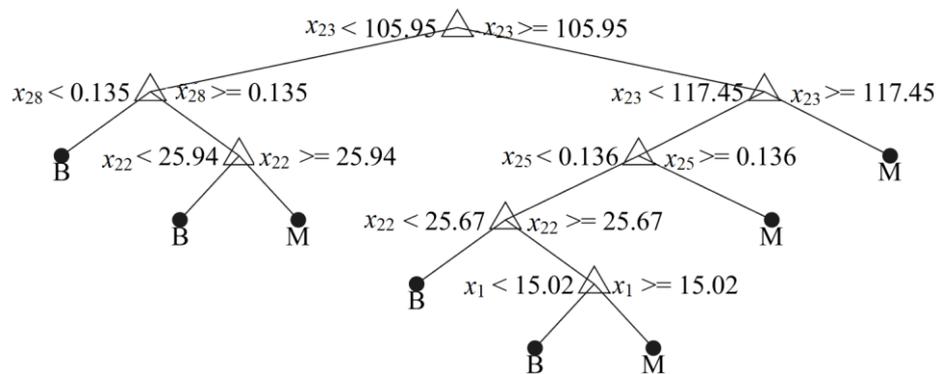


Рисунок 3.11 – Графическое изображение дерева решений CART для набора данных $Wdbc$

Полученная с помощью данного дерева база правил на естественном языке выглядит следующим образом.

R_1 : ЕСЛИ «периметр 3-го ядра» больше 105,95 И «количество вогнутых частей контура 3-го ядра» меньше 0,135 ТО опухоль доброкачественная.

R_2 : ЕСЛИ «периметр 3-го ядра» больше 105,95 И «количество вогнутых частей контура 3-го ядра» больше либо равно 0,135 И «текстура 3-го ядра» меньше 25,94 ТО опухоль доброкачественная.

R_3 : ЕСЛИ «периметр 3-го ядра» больше 105,95 И «количество вогнутых частей контура 3-го ядра» больше либо равно 0,135 И «текстура 3-го ядра» больше либо равно 25,94 ТО опухоль злокачественная.

R_4 : ЕСЛИ «периметр 3-го ядра» больше либо равен 105,95 И «периметр 3-го ядра» меньше 117,45 И «гладкость 3-го ядра» меньше 0,136 И «текстура 3-го ядра» меньше 25,67 ТО опухоль доброкачественная.

R_5 : ЕСЛИ «периметр 3-го ядра» больше либо равен 105,95 И «периметр 3-го ядра» меньше 117,45 И «гладкость 3-го ядра» меньше 0,136 И «текстура 3-го ядра» больше либо равна 25,67 И «радиус 1-го ядра» меньше 15,02 ТО опухоль доброкачественная.

R_6 : ЕСЛИ «периметр 3-го ядра» больше либо равен 105,95 И «периметр 3-го ядра» меньше 117,45 И «гладкость 3-го ядра» меньше 0,136 И «текстура 3-го ядра» больше либо равна 25,67 И «радиус 1-го ядра» больше либо равен 15,02 ТО опухоль злокачественная.

R_7 : ЕСЛИ «периметр 3-го ядра» больше либо равен 105,95 И «периметр 3-го ядра» меньше 117,45 И «гладкость 3-го ядра» больше либо равна 0,136 ТО опухоль злокачественная.

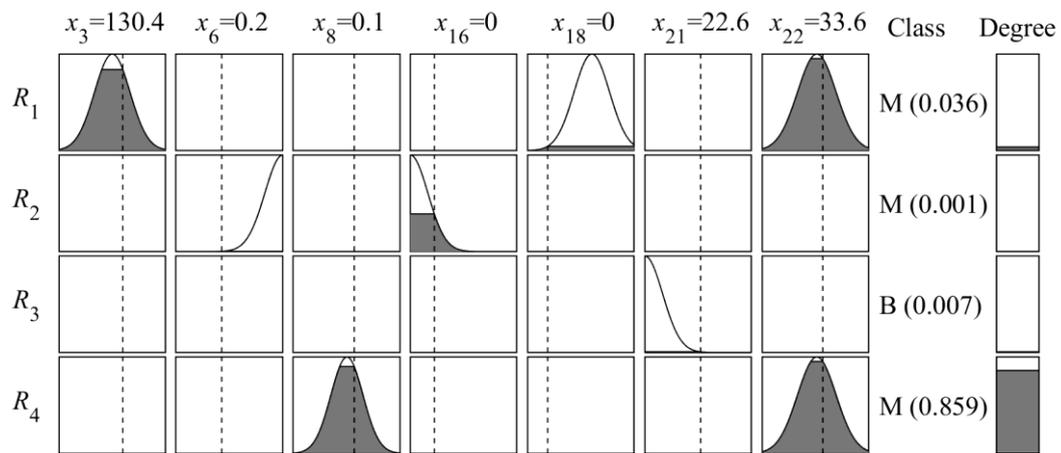
R_8 : ЕСЛИ «периметр 3-го ядра» больше либо равен 105,95 И «периметр 3-го ядра» больше либо равен 117,45 ТО опухоль злокачественная.

Для объяснения полученного результата при классификации образца данных нечетким классификатором нужно выделить правило, имеющее максимальную степень соответствия образцу. Формально степень соответствия в нечетком выводе определяется t-нормальной функцией, которая в данной работе является произведением значений степеней принадлежности признаков нечеткому множеству, связанному с соответствующим лингвистическим термином правила. Класс выделенного правила является результатом классификации образца.

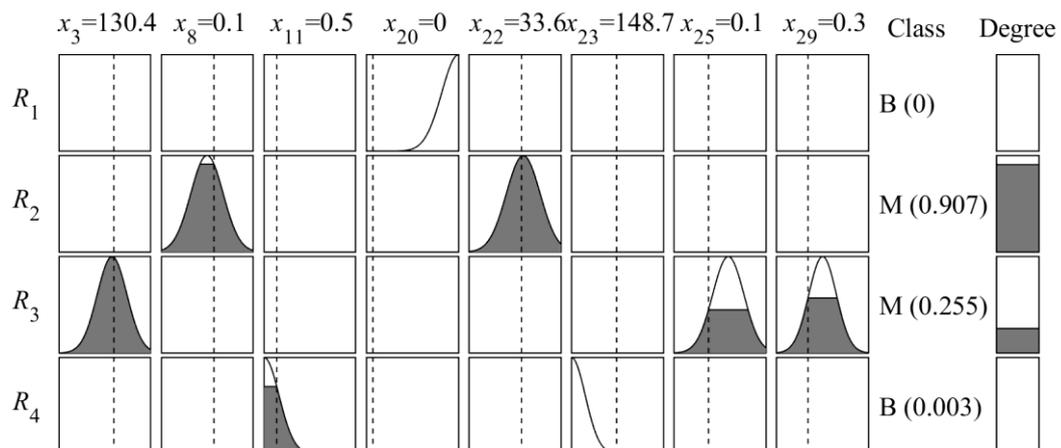
На рисунке 3.12 показана иллюстрация к объяснению результата классификации для классификаторов с правилами рисунка 3.10. Один и тот же образец обработан классификаторами, представленными на рисунках 3.10а и 3.10б. Объяснения полученных результатов классификации образца продемонстрированы на соответствующих изображениях рисунка 3.12. В столбце с информацией о классе правила добавлено значение степени соответствия образца правилу, а в последнем столбце (Degree) приведено графическое изображение этого значения в виде столбчатой диаграммы. Степень соответствия принимает значение из диапазона [0, 1]. Чтобы объяснить решение о полученном

классе образца, нужно показать на какое правило опиралось это решение, а это определяется максимальным значением степени соответствия.

Выводы. Сравнение точности классификации таблиц 3.25 и 3.26 с помощью парного статистического теста Уилкоксона представлено в таблице 3.28. Нулевая гипотеза H_0 теста утверждает об отсутствии значимых различий в оценках эффективности сравниваемых классификаторов, а альтернативная гипотеза H_1 утверждает об обратном. Тест на уровне значимости $\alpha = 0,05$ не выявил значимых различий в точности классификации между классификаторами, построенными с помощью предлагаемой методики, и нечеткими классификаторами FARC-HD. Кроме того, различий в точности не выявлено и с классификаторами на основе деревьев решений, построенных с помощью алгоритма CART.



a)



б)

Рисунок 3.12 – Иллюстрация к объяснению результата принятого решения нечетких классификаторов

Таблица 3.28 – Статистическое сравнение точности классификации

Агрегация	Пары сравнений	Сумма рангов		p-value	Гипотеза H_0 ($\alpha=0,05$)
		+	-		
Максимум	Методика и FARC-HD	331	372	0,7566	Принимается
	Методика и CART	334	406	0,6031	Принимается
Сумма	Методика и FARC-HD	368,5	334,5	0,7949	Принимается
	Методика и CART	375	366	0,9442	Принимается

Статистическое сравнение по критерию интерпретируемости представлено в таблице 3.29. Применялся непараметрический тест Уилкоксона с уровнем значимости $\alpha = 0,05$. Количество правил в классификаторах, построенных предлагаемой методикой, оказалось значительно меньше по сравнению с альтернативами. В среднем меньше на 39,2% по сравнению с FARC-HD и на 81,9% по сравнению с деревьями решений CART. Количество используемых признаков по сравнению с FARC-HD оказалось значительно меньше в среднем на 8%. По сравнению с деревьями решений CART количество используемых признаков оказалось значительно больше в случае агрегацией правил функцией максимума и неразлично в случае агрегации правил функцией суммы. Количество условий в правиле оказалось неразличимым по сравнению с классификаторами FARC-HD, а по сравнению с деревьями решений CART получилось значительно меньше в среднем на 54,5%. Общее количество нечетких терминов в базе правил оказалось значительно меньше по сравнению с FARC-HD в среднем на 49,8%.

В таблице 3.30 представлено статистическое сравнение деревьев решений CART с глубиной не более трех уровней и нечетких классификаторов, построенных с помощью предлагаемой методики. Количество правил и количество используемых признаков у деревьев решений оказалось значительно меньше, чем в нечетких классификаторах. Такое повышение интерпретируемости оказало влияние на понижение точности классификации – она оказалась значимо меньше, чем у нечетких классификаторов.

Таблица 3.29 – Статистическое сравнение показателей интерпретируемости

Агрегация	Пары сравнений	Показатель	Сумма рангов		<i>p</i> -value	Гипотеза H_0 ($\alpha=0,05$)
			+	-		
Максимум	Методика и FARC-HD	#R	78	625	<0,001	Отвергается
		#T	55	648	<0,001	Отвергается
		#F	132	463	0,0047	Отвергается
		#D	269	438	0,1936	Принимается
	Методика и CART	#R	82	659	<0,001	Отвергается
		#F	491	138	0,0063	Отвергается
#D		24	717	<0,001	Отвергается	
Сумма	Методика и FARC-HD	#R	49	654	<0,001	Отвергается
		#T	19	684	<0,001	Отвергается
		#F	86	475	<0,001	Отвергается
		#D	252,5	449,5	0,2077	Принимается
	Методика и CART	#R	30,5	710,5	<0,001	Отвергается
		#F	399,5	195,5	0,0819	Принимается
		#D	24	717	<0,001	Отвергается

Таблица 3.30 – Статистическое сравнение эффективности с деревьями решений CART глубиной не более трех уровней

Агрегация	Пары сравнений	Показатель	Сумма рангов		<i>p</i> -value	Гипотеза H_0 ($\alpha=0,05$)
			+	-		
Максимум	Методика и CART	<i>acc</i>	668	73	<0,001	Отвергается
		#R	544,5	196,5	0,0117	Отвергается
		#F	664	3	<0,001	Отвергается
		#D	45	658	<0,001	Отвергается
Сумма	Методика и CART	<i>acc</i>	672,5	68,5	<0,001	Отвергается
		#R	415,5	325,3	0,5157	Принимается
		#F	618	12	<0,001	Отвергается
		#D	57,5	683,5	<0,001	Отвергается

Результаты эксперимента показали неразличимость точности классификации с рассматриваемыми моделями классификации, а такие показатели интерпретируемости, как количество правил, общее количество нечетких терминов и количество признаков, у классификаторов предлагаемой методики имеют статистически значимое преимущество по сравнению с классификаторами FARC-HD. Сравнение с деревьями решений CART также показало сопоставимую точность и выявило статистически значимое преимущество классификаторов предлагаемой методики по таким показателям интерпретируемости как количество правил и количество условий в правиле. Достигнутые результаты говорят о высоком уровне интерпретируемости нечетких классификаторов, построенных с помощью предлагаемой методики.

3.6 Краткие выводы по разделу

В разделе представлена экспериментальная часть исследования, где проведена оценка эффективности предлагаемых алгоритмов оптимизации, метода разделения данных и методики построения нечетких классификаторов. Оценка алгоритмов оптимизации проводилась с помощью общепринятых тестов, разработанных для нахождения оптимума унимодальных и мультимодальных тестовых функций. Построение нечетких классификаторов для оценки эффективности предлагаемой методики осуществлялась на 38 общедоступных наборах данных из репозитория KEEL (<http://keel.es>). Наборы данных собраны из различных проблемных областей, чтобы оценить независимость методики от решения определенной проблемы. Представлены сравнения с аналогичными алгоритмами оптимизации и методами построения интерпретируемых моделей классификации. Ниже приведены основные выводы экспериментального исследования.

1. Проведен вычислительный эксперимент по нахождению оптимума разработанным алгоритмом смешанной многокритериальной оптимизации и генетическим алгоритмом с недоминируемой сортировкой на общепринятых тестах ZDT и DTZL. В тестах для нахождения оптимума задействованы 10 функций, среди которых пять двухкритериальных и пять трехкритериальных. В качестве критериев эффективности работы алгоритма применялись обратное расстояние поколений и степень доминирования. Результаты оценок подверглись сравнению парным статистическим тестом Уилкоксона, где у разработанного алгоритма обнаружено статистически значимое преимущество (на уровне $\alpha=0,05$) по сравнению с генетическим. Значение обратного расстояния поколений оказалось меньше в среднем на 95,47%, а степень доминирования больше в среднем на 79,92%.

2. Проведен вычислительный эксперимент по нахождению оптимума разработанным алгоритмом дискретной оптимизации на основе распределения

вероятностей с трансформацией целевых значений, а также генетическим алгоритмом и алгоритмом роящихся частиц. Эксперимент проводился для поиска в бинарном пространстве с помощью общепринятых унимодальных и мультимодальных тестовых функций, специально разработанных для проверки алгоритмов однокритериальной оптимизации. В тестах было задействовано 18 эталонных тестовых функций. Для оценки эффективности алгоритмов применялись критерий сходимости в виде показателя отклонения от оптимального значения целевой функции и критерий стабильности в виде показателя среднеквадратичного отклонения значения целевой функции. Статистический тест Уилкоксона показал значимое преимущество (на уровне $\alpha=0,05$) предлагаемого алгоритма по обоим критериям. В среднем отклонение от оптимального значения целевой функции уменьшилось в 4,3 раза по сравнению с генетическим алгоритмом и в 10,1 раза по сравнению с алгоритмом роящихся частиц.

3. Проведен вычислительный эксперимент по настройке параметров функций принадлежности нечетких классификаторов с помощью адаптированного алгоритма градиентного спуска. Алгоритм применялся к классификаторам, предварительно построенным алгоритмом смешанной многокритериальной оптимизации «кукушкин поиск» с генетическим оператором скрещивания на 38 общедоступных наборах данных. Результаты эксперимента показали статистически значимое (на уровне $\alpha=0,05$) повышение точности. Максимальное увеличение точности классификации составило 5,1%.

4. Проведен вычислительный эксперимент для оценки эффективности применения разработанного метода разделения данных при построении нечетких классификаторов с помощью многокритериального метаэвристического алгоритма. В реализации метода для разделения данных применялся предлагаемый в работе дискретный алгоритм оптимизации. В ходе эксперимента установлены вид целевой функции для разделения данных и вариант формирования выборки данных для выбора модели на фронте Парето, которые позволяют получить нечеткие классификаторы со статистически значимым

улучшением точности классификации. Парный статистический тест Уилкоксона показал значимое (на уровне $\alpha=0,05$) повышение точности классификации по сравнению с построением классификаторов без применения разделения данных. Максимальное увеличение точности составило 6,5%. Также получено значимое (на уровне $\alpha=0,05$) повышение интерпретируемости по показателям количества правил, количества признаков и общего количества нечетких терминов в базе правил.

5. Проведен эксперимент по построению нечетких классификаторов с помощью предлагаемой методики построения, оценены точность и интерпретируемость. Выполнено сравнение оценок с известными интерпретируемыми моделями машинного обучения, такими как нечеткие классификаторы FARC-HD и деревья решений CART. Результаты парного статистического сравнения тестом Уилкоксона показали сопоставимую точность классификации полученных методикой моделей по отношению к альтернативам. Тест выявил статистически значимое преимущество (на уровне $\alpha=0,05$) интерпретируемости классификаторов предлагаемой методики. По сравнению с классификаторами FARC-HD результаты таких показателей как количество правил, общее количество нечетких терминов и количество признаков оказались значимо меньше. Количество правил меньше в среднем на 39,2%, общее количество нечетких терминов меньше в среднем на 49,8%, количество признаков меньше в среднем на 8%. По сравнению с деревьями решений CART значимое уменьшение показали количество правил, в среднем на 81,9%, и количество условий в правиле, в среднем на 54.5%. Достигнутые результаты говорят о высоком уровне интерпретируемости полученных нечетких классификаторов, построенных с помощью предлагаемой методики.

4 Внедрение результатов диссертационной работы

В разделе представлено внедрение результатов диссертационной работы. Разработанная методика построения интерпретируемых нечетких классификаторов применена для создания классификатора и программных средств прогнозирования эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19. Результаты используются в Томском научно-исследовательском институте курортологии и физиотерапии (Томский НИИКиФ). Разработанный алгоритм оптимизации MICS применяется в программных средствах для синтеза согласующих цепей в научно-исследовательском институте микроэлектронных систем (НИИ МЭС) и ООО «Микроэлектронные системы». Разработанный дискретный алгоритм оптимизации PDT внедрен в экспериментальную версию интеллектуальной системы автоматизированного проектирования в ООО «50ом Тех.» и испытан для синтеза секции ступенчатого аттенюатора. Соответствующие акты о внедрении (использовании) результатов диссертационной работы приведены в Приложении Б.

4.1 Прогнозирование эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19

Инфекционное заболевание COVID-19, вызываемое вирусом SARS-CoV-2, оказывает пагубное воздействие на организм человека. Согласно опубликованным данным, после выписки из стационара у переболевших COVID-19, могут длительно сохраняться такие клинические симптомы, как кашель, одышка, в том числе при незначительной физической активности, гипо- или атрофия мышц, включая дыхательные мышцы и мышцы туловища и конечностей, артериальная гипертензия или гипотензия, астенический синдром, определяющие

снижение толерантности к физической нагрузке и качества жизни в целом. Такие последствия не оставляют сомнений в необходимости проведения лицам, перенесшим COVID-19, реабилитационных мероприятий. Мероприятия должны быть направлены на предотвращение развития негативных последствий и улучшение прогноза больных, а также восстановление физического и психического здоровья, профессиональной пригодности, с учетом известных на сегодняшний день представлений о патогенезе органических повреждений, ассоциированных с SARS-CoV-2.

Томский НИИКиФ ведет разработку персонализированных технологий комплексной медицинской реабилитации лиц, перенесших короновиральную инфекцию COVID-19. Одним из результатов этой работы является реабилитационный комплекс, основная цель которого заключается в восстановлении здоровья организма пациента после болезни. Комплекс включает в себя следующие компоненты: 1) диета, 2) лечебная физическая культура, 3) психологическое тестирование и коррекция психологических нарушений, 4) классический ручной массаж грудной клетки, 5) сухие углекислые ванны, 6) спелеотерапия, 7) воздействие пульсирующим электростатическим полем на область грудной клетки, 8) вазоактивная электростимуляция, 9) внутренний прием пробиотического препарата.

Для оценки состояния здоровья в Томском НИИКиФ используют тест шестиминутной ходьбы (ТШХ). При проведении теста пациенту ставится задача пройти как можно большую дистанцию за 6 минут. Пройденное расстояние регистрируется. ТШХ считается комплексным показателем здоровья человека, поскольку на итог его выполнения влияет состояние всего организма. Результаты ТШХ позволяют судить о наличии или отсутствии положительной динамики с точки зрения восстановления здоровья организма пациента. Реабилитация пациента считается эффективной, если после нее дистанция ТШХ увеличилась на 15% по сравнению с измерениями до реабилитации.

Важной задачей в ходе проведения исследования являлось прогнозирование эффективности реабилитации пациента до начала прохождения комплекса, а

также определение правил, которые объясняют связь между характеристиками состояния здоровья пациента и достижением эффекта от реабилитации. Для решения данной задачи была предложена прогностическая модель на основе нечеткого классификатора. Помимо прогностического результата классификатор предоставляет базу правил, объясняющую причины найденного решения. Формирование классификатора проводилось с помощью методики построения нечетких систем, предложенной в подразделе 2.7. Далее представлены описания процессов сбора обучающих данных и построение нечеткого классификатора.

Сбор обучающих данных. Для оценки эффективности предлагаемого комплекса в Томском НИИКиФ были проведены клинические исследования, в которых приняли участие 64 пациента мужского (28,13%) и женского (71,87%) пола, поступившие на стационарный этап медицинской реабилитации спустя 30-40 дней после перенесенной коронавирусной инфекции COVID-19. Средний возраст участников составил $56,92 \pm 9,29$ лет. В исследовании приняли участие пациенты с подтвержденным диагнозом «коронавирусная инфекция COVID-19» и осложнением в виде двусторонней полисегментарной вирусной пневмонии, вызванной SARS-CoV-2. Собраны анамнестические данные о заболевании (возраст, рост, вес, сопутствующие заболевания и др.), а также клинические, лабораторные, функциональные, инструментальные, психологические, микробиологические характеристики участников исследования в формате пар показателей «до реабилитации/после реабилитации».

Был сформирован набор данных из 58 признаков и 64 образцов данных о пациентах [117]. В набор были включены антропометрические показатели, а также показатели клинических, лабораторных и функциональных методов исследования до начала реабилитации. Выходная метка для задачи классификации была сформирована следующим образом. После вычисления процента улучшения результата ТШХ в метке класса выставлялась 1, если этот процент был равен или превышал 15%, и 0 в ином случае. В Приложении В приведено описание признаков, включенных в набор данных.

Построение классификатора. На основе полученного набора данных с применением методики построения нечетких систем (подраздел 2.7) был построен классификатор для прогнозирования эффективности реабилитации. Параметры алгоритмов оптимизации и метода выбора методики были установлены в значения, представленные в таблице 3.24. Каждый признак разделялся на три лингвистических термина: «Малое», «Среднее», «Большое». Агрегация правил в вычислении разделяющих функций определялась функцией максимума согласно выражению (1.1). С помощью метода десятикратной перекрестной проверки оценена точность классификации, которая составила 74,67%. База правил классификатора показана на рисунке 4.1 и содержит три правила и семь признаков. Последний столбец определяет заключение правила и содержит метку класса: «1» – реабилитация будет эффективной, «0» – реабилитация будет не эффективной.

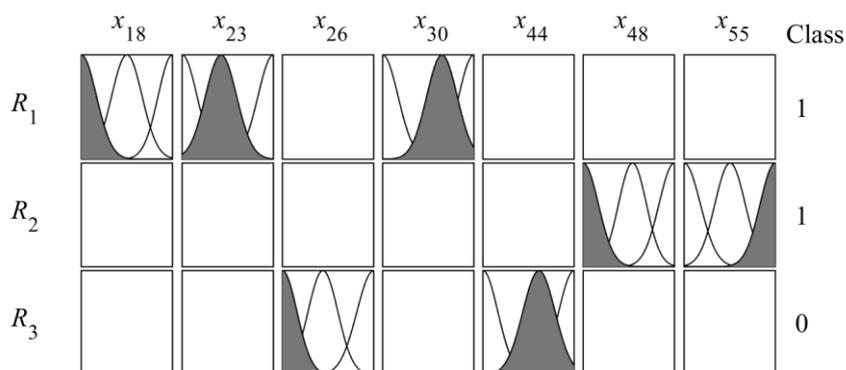


Рисунок 4.1 – База правил нечеткого классификатора для прогнозирования эффективности реабилитации

База правил нечеткого классификатора на естественном языке с учетом лингвистических терминов и названий признаков представлена ниже.

R_1 : ЕСЛИ отсутствуют жалобы на ЖКТ И результат ТШХ средний И пиковая скорость выдоха средняя ТО реабилитация будет эффективной

R_2 : ЕСЛИ наличие триглицеридов в крови низкое И показатель анализа крови «международное нормализованное отношение» высокий ТО реабилитация будет эффективной

R_3 : ЕСЛИ частота сердечных сокращений низкая И количество фибриногена в крови среднее ТО реабилитация будет не эффективной

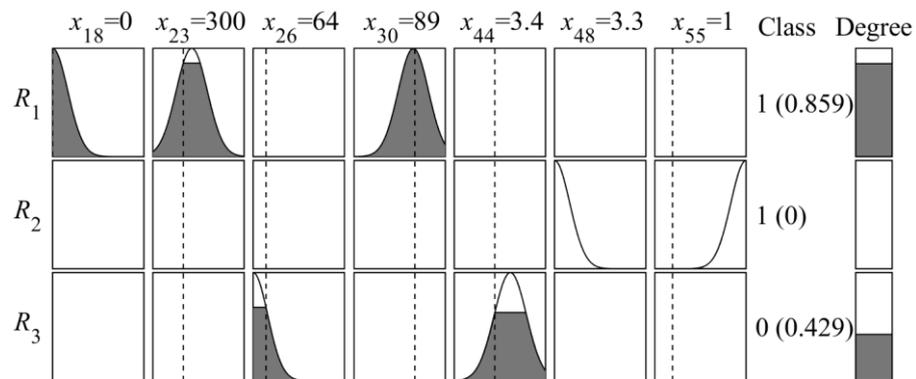
Параметры среднего значения (s) и среднеквадратичного отклонения (σ) функций принадлежности нечетких множеств, определяющих лингвистические термины, показаны в таблице 4.1 для нормированных значений признаков.

Таблица 4.1 – Параметры функций принадлежности нечеткого классификатора для прогнозирования эффективности реабилитации

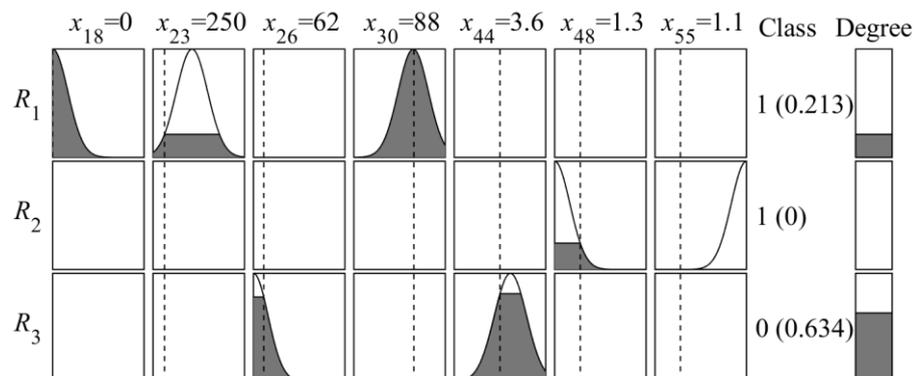
Признак	Описание	Термин «Малое»		Термин «Среднее»		Термин «Большое»	
		s	σ	s	σ	s	σ
x_{18}	Наличие жалоб на желудочно-кишечный тракт	0	0,1668	0,5000	0,1668	1	0,1573
x_{23}	Результат теста шестиминутной ходьбы	0	0,1629	0,4260	0,1711	1	0,1681
x_{26}	Частота сердечных сокращений	0	0,1602	0,4442	0,1635	1	0,1833
x_{30}	Пиковая скорость выдоха	0	0,1682	0,6500	0,1655	1	0,1660
x_{44}	Фибриноген	0	0,1637	0,6144	0,1772	1	0,1693
x_{48}	Триглицериды	0	0,1666	0,5407	0,1527	1	0,1681
x_{55}	Международное нормализованное отношение	0	0,1785	0,5041	0,1659	1	0,1667

Объяснить результат работы классификатора для образца данных можно с помощью вычисления степеней соответствия образца правилам, которые определяется видом t -нормальной функции, и в настоящей работе это функция произведения. На рисунке 4.2 продемонстрированы два примера образцов данных и принадлежащие им степени соответствия правилам в последнем столбце (Degree). На рисунке 4.2а указаны степени соответствия для образца данных со значениями признаков $x_{18} = 0$, $x_{23} = 300$, $x_{26} = 64$, $x_{30} = 89$, $x_{44} = 3,4$, $x_{48} = 3,3$, $x_{55} = 1$. Наибольшее значение соответствует правилу R_1 . Заключение этого правила будет классом образца данных, т.е. «1», реабилитация будет эффективной. На рисунке 4.2б указаны степени соответствия для образца данных со значениями признаков $x_{18} = 0$, $x_{23} = 250$, $x_{26} = 62$, $x_{30} = 88$, $x_{44} = 3,6$, $x_{48} = 1,3$, $x_{55} = 1,1$. Для данного образца класс определяется заключением правила R_3 , реабилитация не будет эффективной, поскольку наибольшее значение степени соответствия у данного образца именно этому правилу.

Результаты и выводы. Применение нечеткого классификатора позволило прогнозировать исход реабилитации пациентов, перенесших коронавирусную инфекцию COVID-19, с точностью 74,67%. Из 58 исходных показателей здоровья в классификаторе используется 7. Благодаря полученной базе из трех правил определена закономерность, связывающая значения семи показателей здоровья пациента и эффективности реабилитации. Значения признаков в базе правил оцениваются нечеткими терминами «Малое», «Среднее» и «Большое». Результат прогнозирования можно объяснить, выделив правило, по отношению к которому образец данных имеет наибольшую степень соответствия. Заключение правила определяют класс входного образца.



а)



б)

Рисунок 4.2 – Иллюстрация к объяснению результата принятого решения при прогнозировании эффективности реабилитации

4.2 Синтез согласующих цепей на основе смешанной многокритериальной оптимизации

Согласующие (СЦ) и корректирующие (КЦ) цепи широко применяются в составе радиоэлектронных устройств и систем для согласования входа\выхода устройства с трактом передачи сигнала и обеспечения максимального коэффициента передачи мощности сигнала от генератора в нагрузку в некоторой полосе рабочих частот. При этом сами СЦ и КЦ, как правило, реализуют на основе реактивных элементов, чтобы они не поглощали мощность передаваемого сигнала [272-274]. Задача широкополосного согласования состоит в том, чтобы обеспечить максимальный (или заданный) уровень коэффициента передачи сигнала в заданной полосе частот при некотором заданном (или минимальном) уровне неравномерности передаточной характеристики для заданных комплексных импедансов генератора $Z_r(f)$ и нагрузки $Z_n(f)$. Если дополнительно требуется получить специальную форму амплитудно-частотной характеристики коэффициента передачи (АЧХ), отличную от прямоугольной, (как правило, с некоторым подъемом или спадом) то такая задача носит название согласования-выравнивания.

Синтез широкополосных согласующих цепей является важной задачей при построении радиотехнических и телекоммуникационных систем и устройств. Обычно требования к согласующим цепям являются противоречивыми по показателям качества полученного результата, что усложняет автоматизацию процесса синтеза.

В подразделе представлен метод структурно-параметрического синтеза широкополосных согласующих цепей на основе алгоритма смешанной многокритериальной оптимизации «кукушкин поиск» с генетическим оператором скрещивания (см. подраздел 2.3). Проведены эксперименты и сравнения эффективности его работы с аналогичными методами.

Основная идея рассматриваемого метода заключается в сведении задачи синтеза реактивной СЦ к проблеме смешанной многокритериальной оптимизации, в которой присутствуют дискретные и непрерывные переменные [170]. Использование смешанной оптимизации позволяет проводить одновременно как структурный, так и параметрический синтез СЦ путем кодирования структуры цепи и ее компонентов с помощью дискретных переменных, а значения их параметров – с помощью непрерывных. Применение многокритериальности дает возможность учитывать несколько целей в процессе синтеза. В настоящей работе рассматриваются две цели: с одной стороны необходимо получить цепь, удовлетворяющую заданной форме АЧХ, а с другой стороны цепь должна содержать минимально возможное число компонентов, или содержать компоненты, которые обуславливают минимальные затраты на изготовление цепи. Таким образом, результат решения задачи синтеза должен представлять множество вариантов (структур) цепей, недоминируемых по данным двум целям, т.е. образующих фронт Парето [170]. Разработчик, основываясь на требованиях решаемой проблемы, выбирает из этого множества подходящий вариант цепи, например с точки зрения физической реализуемости.

Для того чтобы свести задачу синтеза СЦ к проблеме оптимизации, необходимо задать кодирование цепи с помощью переменных вектора решения, определить целевые функции и задать ограничения на значения переменных. Ниже описаны данные этапы метода.

Кодирование цепи. Цепь кодируется в виде смешанного вектора решения (т.е. содержащего целочисленные и вещественные переменные). Схема кодирования представлена на рисунке 4.3.

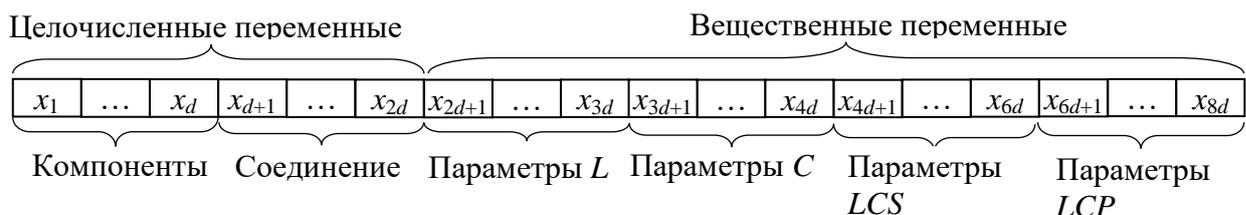


Рисунок 4.3 – Схема кодирования цепи согласования вектором решения

Максимальное количество компонентов в цепи ограничено значением d , которое является параметром и задается до выполнения синтеза. Вектор разделен на целочисленные и вещественные переменные. Целочисленные переменные x_1, \dots, x_{2d} кодируют структуру цепи. Переменные x_1, \dots, x_d содержат информацию о типе компонента цепи или его отсутствии. Значение переменной «1» соответствует катушке индуктивности L , значение «2» – конденсатору C , значение «3» – LC -контур с последовательным соединением элементов (LCS), значение «4» – LC -контур с параллельным соединением элементов (LCP), значение «0» показывает отсутствие элемента, привязанного к данной переменной. Переменные x_{d+1}, \dots, x_{2d} кодируют тип соединения соответствующего компонента, значение «1» показывает последовательное соединение, а значение «2» – параллельное. Переменные x_{2d+1}, \dots, x_{8d} содержат значения параметров компонентов, соответствующих переменным x_1, \dots, x_d . Для катушек индуктивности это переменные x_{2d+1}, \dots, x_{3d} ; для конденсаторов – x_{3d+1}, \dots, x_{4d} ; для LCS контуров – x_{4d+1}, \dots, x_{6d} ; для LCP контуров – x_{6d+1}, \dots, x_{8d} . Если переменная x_i , ($i = 1, \dots, d$), кодирует катушку, то ее индуктивность находится в переменной x_{2d+i} . Если кодирует конденсатор, то его емкость содержится в переменной x_{3d+i} . Если переменная кодирует LCS -контур, то индуктивность катушки находится в x_{4d+i} , а емкость конденсатора в x_{5d+i} . В случае LCP -контура индуктивность содержится в x_{6d+i} , а емкость в x_{7d+i} . Пространство поиска (число переменных в векторе решения) ограничено значением $8d$.

На рисунке 4.4а показана схема двухэлементной цепи, а на рисунке 4.4б показан вектор решения, который кодирует данную структуру. Цепь состоит из двух компонентов C и LCP с типом соединения параллельное включение (p) и последовательное включение (s). Соответствующие параметры имеют значения 0,2 для емкости C , 1,8 и 9,3 для элементов L и C параллельного контура. На рисунке 4.4б переменные, кодирующие компоненты, тип соединения и параметры цепи, выделены темным цветом. При этом вектор решения рассчитан на максимальное значение количества компонентов – 3, т.е. содержит один «пустой» элемент.

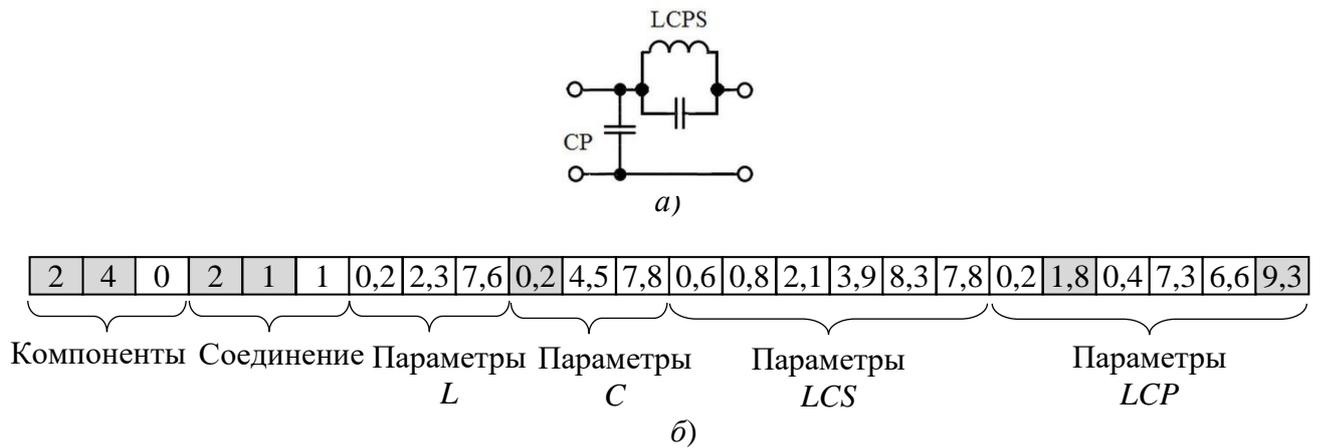


Рисунок 4.4 – Схема цепи (а) и пример ее кодирования (б)

Следует также отметить, что одна и та же цепь может быть закодирована несколькими вариантами. На рисунке 4.5 показан вариант вектора решения, кодирующий цепь на рисунке 4.4а, но имеющий другие значения некоторых переменных в отличие от вектора на рисунке 4.4б.

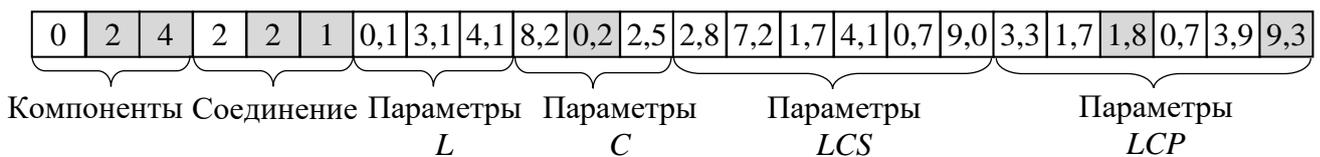


Рисунок 4.5 – Пример варианта альтернативного кодирования цепи

Целевые функции. Как отмечалась выше, синтез цепи в настоящей работе преследует две цели. Первая цель – удовлетворение требуемой форме АЧХ коэффициента передачи. В этом случае в качестве целевой функции предлагается использовать максимальное отклонение коэффициента передачи от заданного значения в диапазоне частот работы цепи. При этом целевая функция должна быть минимизирована:

$$F_1(\mathbf{X}) = \max_{i=1, \dots, n} \left(\left| G(f_i) - G^*(\mathbf{X}, f_i) \right| \right),$$

где $\mathbf{X} = (x_1, \dots, x_{8d})$ – вектор решения, определяющий цепь; f_1, f_2, \dots, f_n – заданные частоты для проверки коэффициента передачи цепи; $G(f_i), G^*(\mathbf{X}, f_i)$ – требуемое и реальное значения коэффициента передачи цепи \mathbf{X} на частоте $f_i, (i = 1, \dots, n)$.

Вторая цель – минимизация числа компонентов в цепи или затрат на ее изготовление. В этом случае предлагается использовать сумму стоимостных коэффициентов компонентов цепи, которая должна быть минимизирована:

$$F_2(\mathbf{X}) = \sum_{i=1}^d \begin{cases} k_L, \text{ если } x_i = 1 \\ k_C, \text{ если } x_i = 2 \\ k_{LCS}, \text{ если } x_i = 3, \\ k_{LCP}, \text{ если } x_i = 4 \\ 0, \text{ иначе} \end{cases}$$

где k_L , k_C , k_{LCS} , k_{LCP} , – величины стоимости компонентов L , C , LCS и LCP соответственно. В случае минимизации количества компонентов данные величины устанавливаются в значение «1».

Ограничения на значения переменных. Оптимизация для синтеза цепей в настоящей работе является безусловной. Ограничения накладываются только на диапазон значений параметров компонентов. Индуктивность для катушек изменяется в диапазоне от L_{\min} до L_{\max} нГн, а емкость конденсаторов от C_{\min} до C_{\max} пФ. Минимальное (L_{\min} и C_{\min}) и максимальное (L_{\max} и C_{\max}) значения элементов зависят от диапазона частот согласования. Таким образом, синтез согласующей цепи сводится к следующей задаче смешанной многокритериальной оптимизации:

$$\begin{cases} F_1(\mathbf{X}) \rightarrow \min \\ F_2(\mathbf{X}) \rightarrow \min \\ x_i \in \{0, 1, 2, 3, 4\}, i = 1, \dots, d \\ x_i \in \{1, 2\}, i = d + 1, \dots, 2d \\ x_i \in [L_{\min}, L_{\max}], i = 2d + 1, \dots, 3d, 4d + 1, \dots, 5d, 6d + 1, \dots, 7d \\ x_i \in [C_{\min}, C_{\max}], i = 3d + 1, \dots, 4d, 5d + 1, \dots, 6d, 7d + 1, \dots, 8d \end{cases}$$

Решить задачу оптимизации предлагается алгоритмом смешанной многокритериальной оптимизации «кукушкин поиск» с генетическим оператором скрещивания, представленным в подразделе 2.3.

Вычислительный эксперимент. Проведено тестирование синтеза цепей на типовых задачах широкополосного согласования, представленных в литературе. Проведено сравнение с альтернативными методами синтеза. Параметры алгоритма «кукушкин поиск» с генетическим оператором скрещивания, которые были заданы при решении тестовых задач, представлены в таблице 4.2. Диапазоны варьирования параметров катушек индуктивностей и конденсаторов изменялись в зависимости от решаемой задачи.

Таблица 4.2 – Параметры алгоритма смешанной многокритериальной оптимизации для синтеза согласующих цепей

Название параметра	Обозначение	Значение параметра
Максимальное число компонентов в цепи	d	30
Количество переменных в векторе решений	n	240
Размер популяции	N	50
Число итераций	$MaxIter$	60000
Вероятность удаления решения	p_a	0,05
Коэффициент прыжка полетов Леви	γ	Равномерно уменьшался с 0,4 до 0,05

Первый пример демонстрирует решение классической задачи одностороннего согласования резистивного сопротивления генератора с комплексной RLC -нагрузкой, предложенный Фано в [275] и представленной на рисунке 4.6а. Задача состоит в том, чтобы рассчитать СЦ, которая в диапазоне частот от 0 до 1 ГГц обеспечивает величину модуля коэффициента отражения $|\Gamma(f)| < 0,346$ или коэффициента передачи $G > 0,88$, где $\Gamma = \sqrt{1 - G^2}$. Сопротивления нагрузки составляет 50 Ом. При этом необходимо получить цепь, у которой АЧХ имеет минимальную неравномерность в полосе согласования, т.е. $\Delta G \rightarrow \min$.

Для оценки эффективности синтеза проведено сравнение с аналогами. На рисунке 4.6б показаны графики АЧХ коэффициента передачи для цепей 1, 2, 5 и 9. Значения элементов СЦ и результаты расчетов представлены в таблице 4.3. Цепь 1 рассчитана на основе аналитической методики Фано-Юлы [275, 276]. Цепь 2 была синтезирована методом «реальной частоты» на основе кусочно-линейной аппроксимации вещественной части в [276]. Цепь 3 была получена на основе применения метода наименьших квадратов [277]. Цепи 4 и 5 были рассчитаны на

основе применения методики «визуального» синтеза [278-280]. Цепи 6 и 7 были рассчитаны с помощью программы GeneSyn [281], в которой реализован структурно-параметрический синтез на основе применения генетического алгоритма. Цепи 8 и 9 получены с помощью метода на основе смешанной многокритериальной оптимизации, предлагаемого в данном исследовании.

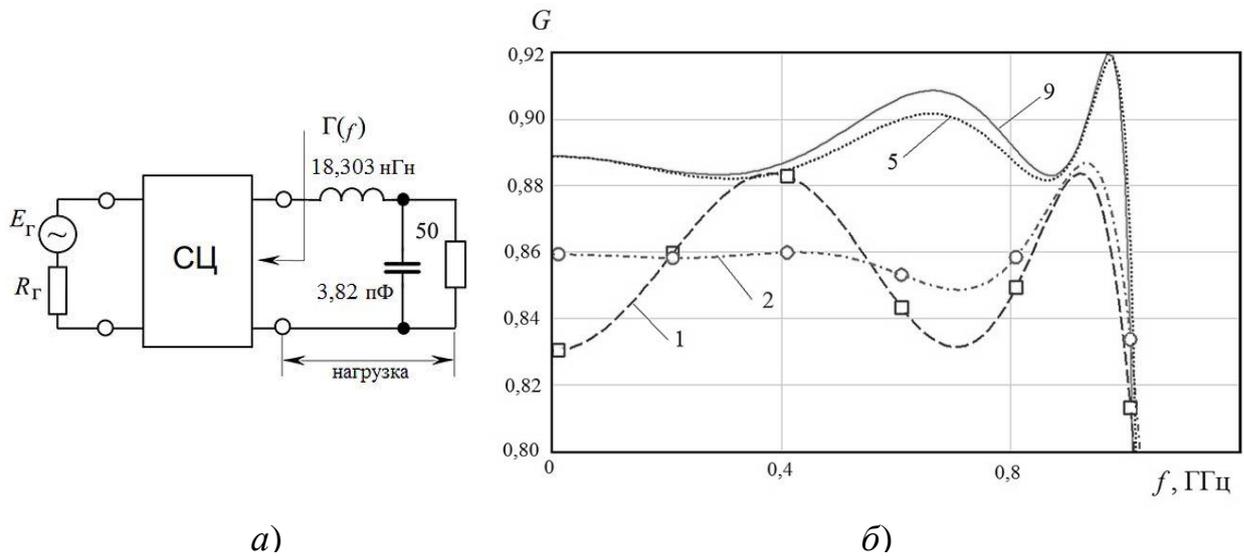


Рисунок 4.6 – Задача одностороннего согласования Фано (а), АЧХ синтезированных СЦ для задачи 1 (б)

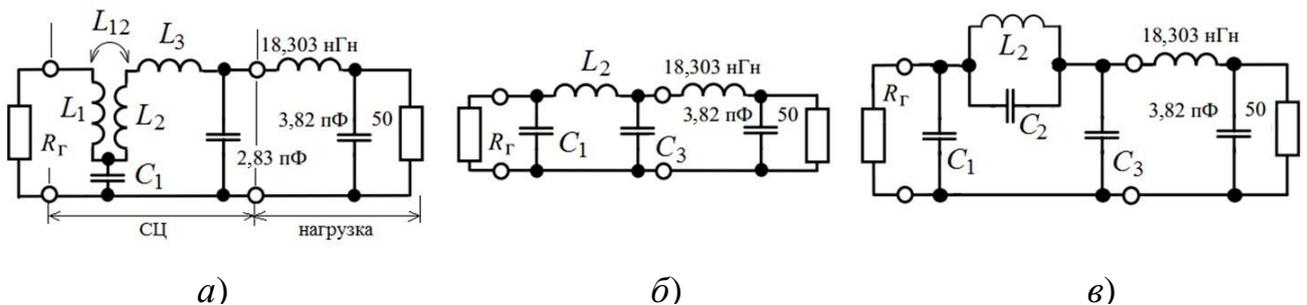


Рисунок 4.7 – Структуры СЦ для задачи 1

Сравнение результатов расчетов для первой задачи показали, что предлагаемый метод выполняет синтез СЦ лучше, чем методы классического синтеза, нелинейной оптимизации и «реальной» частоты, и примерно на том же уровне, что и методы на основе визуального подхода и генетического алгоритма. Так же как и программы LOCUS и GeneSyn, предлагаемый метод для цепи, состоящей из 3-х элементов, находит два возможных решения – СЦ в виде

фильтра низких частот (цепь 8) и цепь с параллельным LC -контуром (СЦ с эллиптической формой АЧХ) в (цепь 9). По сравнению с визуальным подходом предлагаемый метод позволил получить бóльший уровень G_{\min} (0,882 против 0,8814), но неравномерность АЧХ ΔG (0,0405 против 0,0362) стала несколько хуже (цепи 7 и 9 в таблице 4.3). Цепь 7, полученная с помощью генетического алгоритма, имеет лучшие предельные значения G_{\min} и ΔG среди представленных в таблице 4.3.

Таблица 4.3 – Результаты синтеза задачи 1

№ цепи	Рис. 4.7	Метод синтеза	Элементы (Ом, пФ, нГн)	G_{\min}	ΔG	$ \Gamma _{\max}$	Число элементов
1	<i>a</i>	Классический синтез на основе методики Фано-Юлы [275]	$R_r = 120$ $C_1 = 6,18$ $L_1 = 16,85$ $L_2 = 13,57$ $L_{12} = 17,22$ $L_3 = 16,85$	0,8261	0,0588	0,417	4
2	<i>б</i>	Метод «реальной частоты» [282]	$R_r = 110$ $C_1 = 1,12$ $L_2 = 23,15$ $C_3 = 2,936$	0,8487	0,0381	0,389	3
3	<i>в</i>	Алгоритм Левенберга-Марквардта [283]	$R_r = 100$ $C_1 = 0,252$ $L_2 = 6,9$ $C_2 = 2,526$ $C_3 = 2,762$	0,8662	0,0496	0,367	4
4	<i>б</i>	Визуальное проектирование – программа LOCUS [279]	$R_r = 110$ $C_1 = 1,215$ $L_2 = 23,63$ $C_3 = 3,05$	0,8499	0,0508	0,387	3
5	<i>в</i>	Визуальное проектирование – программа LOCUS [279]	$R_r = 100$ $C_1 = 0$ $L_2 = 4,58$ $C_2 = 4,33$ $C_3 = 2,69$	0,8814	0,0362	0,344	3
6	<i>б</i>	Программа GeneSyn [281]	$R_r = 110$ $C_1 = 1,28$ $L_2 = 24,0$ $C_3 = 3,08$	0,854	0,0593	0,382	3
7	<i>в</i>	Программа GeneSyn [281]	$R_r = 100$ $C_1 = 0$ $L_2 = 4,27$ $C_2 = 4,75$ $C_3 = 2,72$	0,8835	0,0252	0,3413	3
8	<i>б</i>	Предлагаемый метод синтеза	$R_r = 110$ $C_1 = 1,296$ $L_2 = 24,02$ $C_3 = 3,096$	0,854	0,0682	0,382	3
9	<i>в</i>	Предлагаемый метод синтеза	$R_r = 100$ $C_1 = 0$ $L_2 = 4,25$ $C_2 = 4,82$ $C_3 = 2,71$	0,882	0,0405	0,3435	3

Второй пример демонстрирует решение задачи синтеза реактивной СЦ для согласования двух комплексных нагрузок [282, 284-286], заданных в виде

эквивалентных схем (рисунок 4.8). Реактивная СЦ должна обеспечить согласование частотно-зависимых импедансов генератора $Z_r(f)$ и нагрузки $Z_n(f)$ в полосе частот от 47 МГц до 157 МГц при уровне коэффициента передачи по мощности не хуже, чем $-1,25$ дБ (0,75 раз). Результаты расчетов аналогичных и предлагаемого методов показаны на рисунке 4.9 и сведены в таблицу 4.4, АЧХ некоторых синтезированных СЦ приведены на рисунке 4.10.

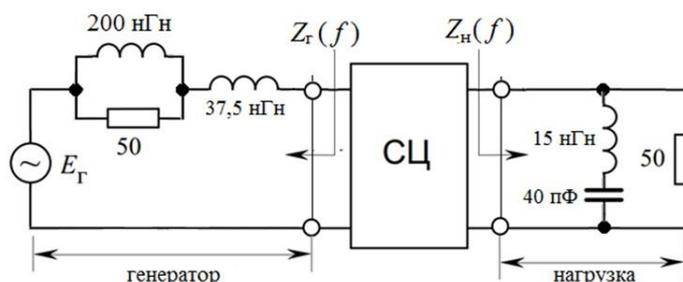


Рисунок 4.8 – Задача согласования двух комплексных нагрузок

Таблица 4.4 – Результаты синтеза задачи 2

№ цепи	Рис. 4.9	Метод синтеза	G_{\min}	ΔG	$ \Gamma _{\max}$	Число элементов
1	<i>a</i>	Метод «реальной» частоты [284]	0,7423	0,109	0,5076	6
2	<i>б</i>	Метод «реальной» частоты [285]	0,7474	0,098	0,5026	4
3	<i>в</i>	Сочетание алгоритмов Гаусса-Ньютона и случайного поиска [287]	0,7503	0,09	0,4997	6
4	<i>г</i>	Сочетание генетического алгоритма и алгоритма наискорейшего спуска [286]	0,7362	0,157	0,5136	5
5	<i>д</i>	Визуальное проектирование – программа LOCUS INDESYS [280]	0,7488	0,063	0,5012	4
6	<i>e</i>	Предлагаемый метод синтеза	0,7522	0,075	0,4978	4
7	<i>ж</i>	Предлагаемый метод синтеза	0,7615	0,075	0,4884	5
8	<i>з</i>	Предлагаемый метод синтеза	0,7876	0,076	0,4609	5
9	<i>и</i>	Предлагаемый метод синтеза	0,7654	0,089	0,4844	5

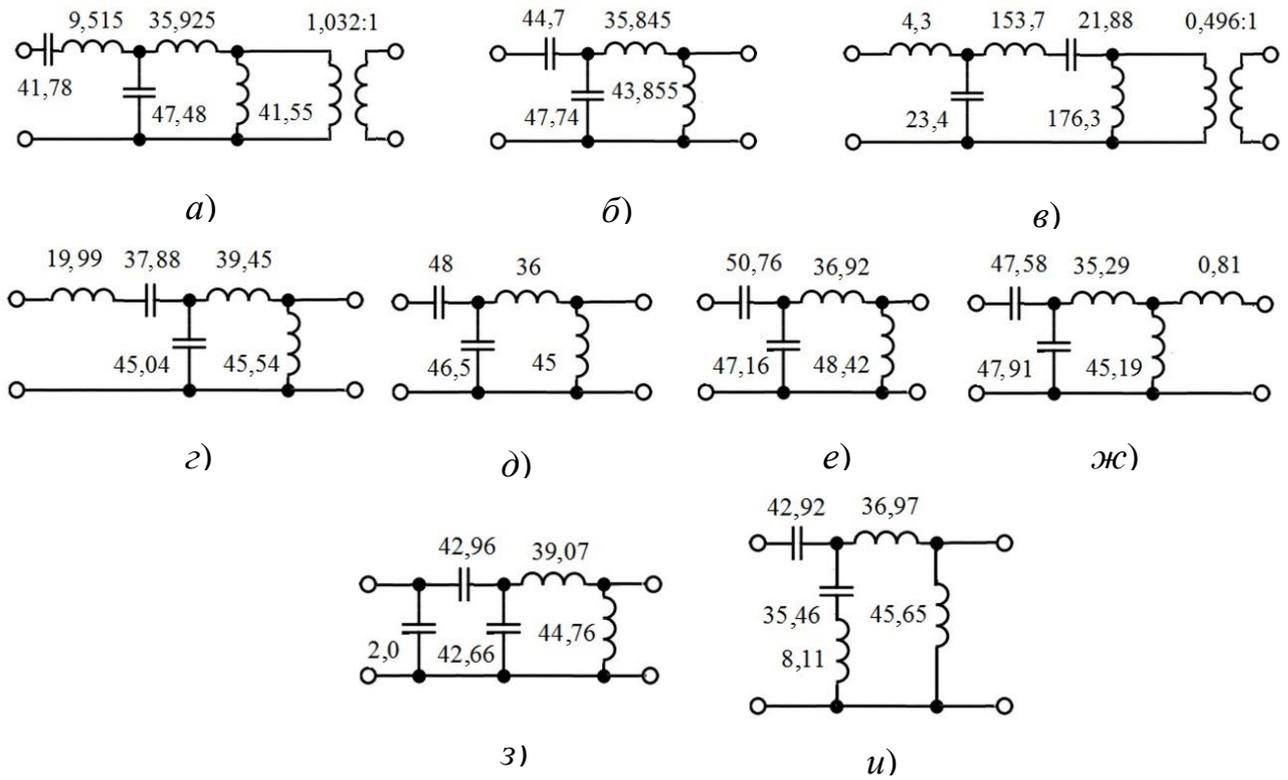


Рисунок 4.9 – Синтезированные СЦ для примера 2 (индуктивность L в нГн, емкость C в пФ)

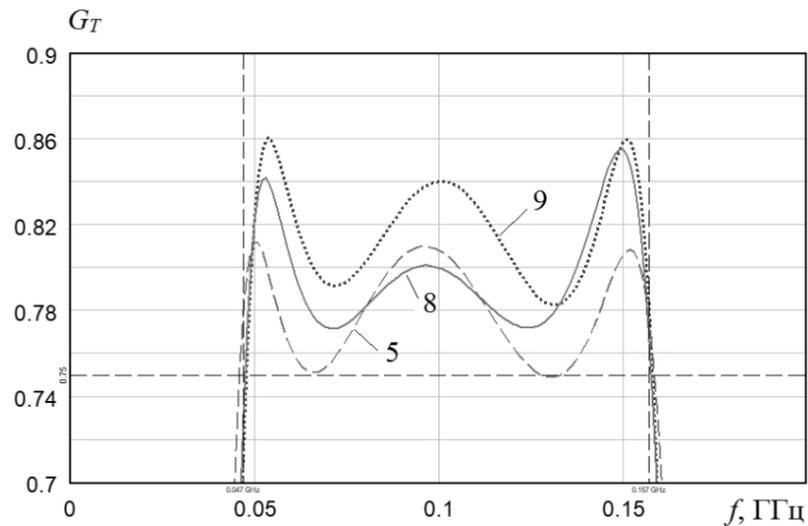


Рисунок 4.10 – АЧХ синтезированных СЦ для задачи 2

Цепи 1 и 2 были получены методом «реальной» частоты» в [284] и [285] соответственно. Цепь 3 получена в [287] с помощью метода, комбинирующего оптимизационный стохастический алгоритм Гаусса-Ньютона и алгоритм случайного поиска. Цепь 4 была рассчитана в [286] на основе применения генетического алгоритма и метода наискорейшего спуска. Цепь 5 получена на

основе применения метода «визуального» синтеза и программы LOCUS [280]. В [280] отмечается, что применение «визуальной» методики позволяет получить лучшие результаты по сравнению с методами «реальной частоты» и методами численной оптимизации. Предлагаемый метод нашел несколько решений данной задачи с разным количеством элементов (от 4 до 8), из которых в таблицу 4.4 вынесено четыре – это цепь 6 с минимальным количеством элементов ($n = 4$) и три пятиэлементные цепи 7, 8 и 9 с разными структурами, которые имеют лучшие показатели по уровню G_{\min} .

Из сравнения результатов расчетов (таблица 4.4) можно сделать вывод о том, что полученные решения (цепи 6-9) имеют лучшие параметры, чем СЦ, полученные на основе методов «реальной» частоты (цепи 1 и 2) и оптимизационных подходов (цепь 3 и 4), и сопоставимы с параметрами СЦ, которая была получена на основе визуальной методики (цепь 5). Рассчитанная предлагаемым методом цепь 8 имеет лучшее значение G_{\min} (0,7876 против 0,7488), чем цепь 5, но неравномерность ΔG у нее стала немного больше (0,076 против 0,063).

Третьим примером является расчет СЦ с наклонной формой АЧХ. В данном примере решается задача синтеза межкаскадной СЦ для двухкаскадного СВЧ усилителя диапазона частот 2-6 ГГц, выполненного на полевых GaAs транзисторах (схема на рисунке 4.11). Эта задача была рассмотрена в [284], а также использовалась в [277, 288] в качестве теста для сравнения различных методов синтеза. В качестве импеданса источника сигнала $Z_T(f)$ для межкаскадной цепи СЦ₂ выступает выходной импеданс транзистора первого каскада, а в качестве нагрузки $Z_H(f)$ – входной импеданс транзистора второго каскада. При этом АЧХ СЦ₂ должна иметь подъем для компенсации спада частотной зависимости коэффициента усиления транзисторов.

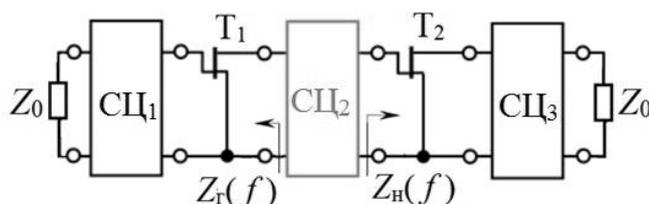


Рисунок 4.11 – Структурная схема двухкаскадного СВЧ усилителя

Исходные данные для проектирования межкаскадной цепи представлены в таблице 4.5, для фиксированных частот диапазона 2-6 ГГц заданы импедансы генератора ($\text{Re}Z_r$, $\text{Im}Z_r$) и нагрузки ($\text{Re}Z_n$, $\text{Im}Z_n$), а также желаемая форма АЧХ коэффициента передачи СЦ₂ (G_0). Здесь $G_0(f_k) = (G^+(f_k) - G^-(f_k))/2$, где G^+ и G^- – верхняя и нижняя границы уровня допуска коэффициента передачи соответственно. Результаты расчетов представлены на рисунке 4.12 и сведены в таблицу 4.6. В таблице $\Delta G_{\max} = \max(|G(f_k) - G_0(f_k)|)$ – максимальное отклонение АЧХ СЦ от желаемого значения; $G_{\max} = \max(G(f_k))$ – максимальное значение АЧХ в полосе рабочих частот. На рисунке 4.13 приведены АЧХ для СЦ с номерами 2, 5, 7 и 8 (номера СЦ указаны в таблице 4.6), желаемое значение G_0 показано серой непрерывной линией, а заданные допуски на АЧХ G^- , G^+ – пунктирными линиями.

Таблица 4.5 – Исходные данные к задаче синтеза межкаскадной СЦ.

f , ГГц	G_0 , дБ	G , дБ	G^+ , дБ	$\text{Re}Z_r$, Ом	$\text{Im}Z_r$, Ом	$\text{Re}Z_n$, Ом	$\text{Im}Z_n$, Ом
2	-1,271	-1,471	-1,071	75,08	0,84	83,16	-135,9
3	-0,5188	-0,7188	-0,3188	81,22	2,98	53,02	-102,9
4	-0,5542	-0,7542	-0,3542	81,94	-1,52	35,56	-77,55
5	0,0	-0,20	0,20	85,15	-1,40	39,93	-68,64
6	-0,6525	-0,8525	-0,4525	81,44	-1,19	22,69	-46,11

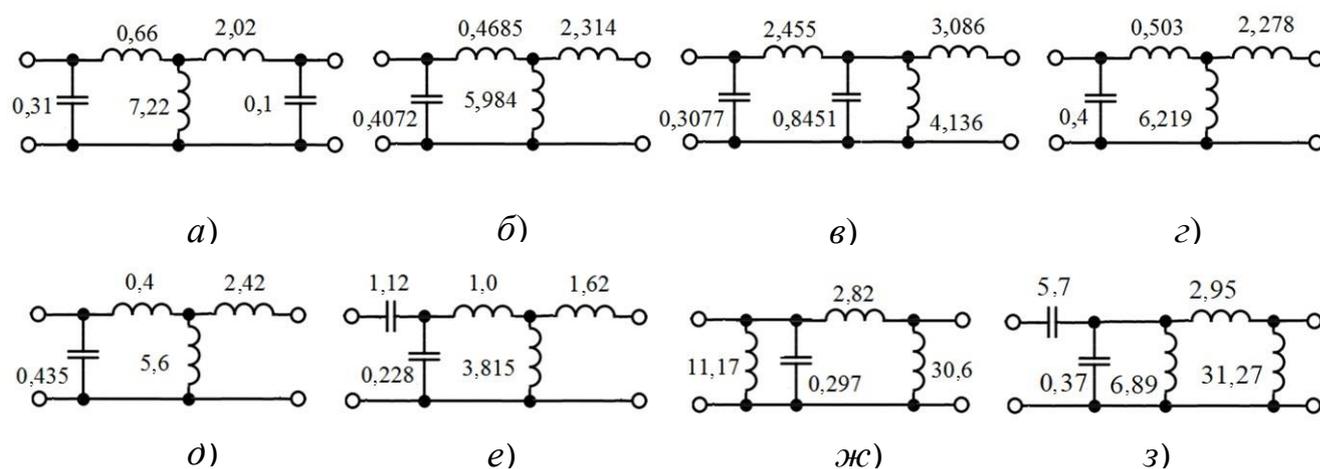


Рисунок 4.12 – Синтезированные межкаскадные СЦ₂ (индуктивность L в нГн, емкость C в пФ)

Таблица 4.6 – Результаты синтеза задачи 3

№ цепи	Рис. 4.12	Метод синтеза	ΔG_{\max} , дБ	G_{\max} , дБ	Число элементов	Элемент CS
1	a	Метод систематического поиска [288]	1,15	– 0,597	5	нет
2	б	Метод «реальной» частоты [284]	0,20	– 0,059	4	нет
3	в	Сочетание генетического алгоритма и метода наискорейшего спуска [286]	0,08	– 0,055	5	нет
4	г	Генетический алгоритм – программа GeneSyn [281]	0,19	– 0,059	4	нет
5	д	«Визуальное» проектирование – программа LOCUS INDESYS [280, 289]	0,23	– 0,041	4	нет
6	е	«Визуальное» проектирование – программа LOCUS INDESYS [280, 289]	0,17	– 0,094	5	да
7	ж	Предлагаемый метод синтеза	0,19	– 0,145	4	нет
8	з	Предлагаемый метод синтеза	0,22	– 0,063	5	да

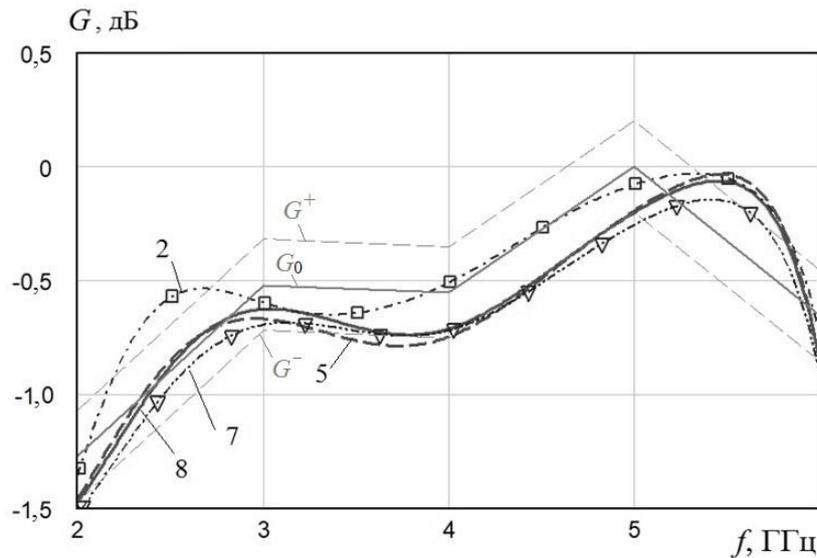


Рисунок 4.13 – АЧХ синтезированных СЦ для задачи 3

Цепь 1 была рассчитана на основе метода систематического поиска [288], цепь 2 – численным методом «реальной» частоты [284]. Цепь 3 – на основе применения генетического алгоритма и градиентного метода нелинейной оптимизации [286]. Цепь 4 – с помощью генетического алгоритма [281]. Цепи 5 и 6 были найдены с помощью «визуального» подхода [280, 289]. При этом для цепи 6 первый элемент был специально задан в виде последовательно включенного конденсатора. Цепи 7 и 8 были получены предлагаемым методом.

Сравнение результатов расчетов межкаскадной СЦ для примера 3 (таблица 4.6) позволяет сделать вывод, что разработанный метод нашел решения, не уступающие альтернативным методам, в том числе и на основе применения генетического алгоритма (GeneSyn) и визуального подхода (LOCUS). Но последовательно включенный конденсатор, наличие которого требуется в составе СЦ₂, есть только в цепях 5 и 8. Причем в случае цепи 5 этот элемент был жестко задан перед началом процедуры синтеза. Для сравнения АЧХ коэффициента передачи для цепей 2, 5, 7 и 8 приведены на рисунке 4.13. Видно, что АЧХ для цепи 7 лучше попадает в желаемый диапазон значений между G^- и G^+ , т.е. имеет меньшее значение ΔG_{\max} , чем цепь 8, но при этом значение G_{\max} для нее стало немного меньше. Следует также отметить, так как импедансы генератора и

нагрузки были заданы в виде дискретных точек (таблица 4.5), для получения «гладких» графиков АЧХ использовалась сплайн-интерполяция.

Результаты и выводы. Тестирование разработанного метода и реализованной на его основе программных средств на типовых задачах широкополосного согласования показало его работоспособность и эффективность, найденные решения не уступают результатам, которые были получены на основе альтернативных подходов, а в некоторых случаях и превосходят их.

В задаче одностороннего согласования цепь предлагаемого метода показала второй результат из девяти вариантов по показателю минимального значения коэффициента передачи, который от лучшего решения отличается на 0,0015дБ, а от худшего на 0,0559 дБ. В задаче двухстороннего согласования цепь предлагаемого метода продемонстрировала лучший результат данного показателя из девяти вариантов цепей. В задаче согласования-выравнивания цепь предлагаемого метода показала третий результат из восьми вариантов по показателю максимального отклонения коэффициента передачи от желаемого значения. Отклонение данного показателя от лучшего варианта составило 0,011дБ, а от худшего – 0,096 дБ.

В отличие от генетического алгоритма, реализованного в программе GeneSyn [281], предлагаемый метод позволяет за один запуск получить набор возможных решений (структур СЦ) с разным количеством элементов, близких к оптимальным. В программе GeneSyn для этого потребуется выполнить несколько стартов с разными настройками размерности синтезируемой цепи. Визуальный подход [278-280] тоже позволяет получать решения близкие к оптимальным, так как дает возможность исследовать решаемую задачу за счет ее визуализации, но он является интерактивным, т.е. требует активного участия человека в процессе синтеза.

4.3 Синтез секции ступенчатого аттенюатора с помощью дискретного алгоритма оптимизации

Организация ООО «50ом Тех.» ведет разработку интеллектуальной системы автоматизированного проектирования «50ohm Tech Circuit Studio». Система позволяет автоматически синтезировать структуры и параметры СВЧ интегральных схем. Разработанный в данном исследовании дискретный алгоритм оптимизации (подраздел 2.4) внедрен в экспериментальную версию данной системы и испытан в рамках опытной эксплуатации для синтеза секции ступенчатого СВЧ-аттенюатора.

Аттенюатор – это устройство, которое снижает мощность сигнала без существенного изменения его формы. Данное устройство используется в широком диапазоне частот и выполняет такие функции как ограничение уровня сигнала, согласование сопротивления и регулировки мощности сигнала в передающих и приемных устройствах. Для синтеза секции аттенюатора в системе используются следующие компоненты: катушки индуктивности, конденсаторы, транзисторы, линии передачи, резисторы, сквозные отверстия.

Использование дискретного алгоритма оптимизации для синтеза предполагает кодирование секции аттенюатора дискретным вектором решений. Схема кодирования представлена на рисунке 4.14. Максимальное количество компонентов равно значению n и задается до запуска алгоритма. Каждый компонент устройства кодируется с помощью восьми элементов вектора решений. Элемент вектора с обозначением « L » соответствует катушки индуктивности. Значения принимаются из множества $\{1, 2, \dots, nL\}$, где nL – число вариантов параметров для катушки. Элемент « C » соответствует конденсатору. Значения принимаются из множества $\{1, 2, \dots, nC\}$, nC – число вариантов параметров для конденсатора. Элемент « T » соответствует транзистору. Возможные значения принадлежат множеству $\{1, 2, \dots, nT\}$, где nT – число вариантов параметров для транзистора. Элемент « R » соответствует резистору и

принимает значения их множества $\{1, 2, \dots, nR\}$, где nR – число возможных вариантов параметров для резистора. Элемент «ЛП» соответствует линии передачи и принимает значение из множества $\{1, 2, \dots, nЛП\}$, где $nЛП$ – число возможных параметров для линии передачи. Элемент «СО» соответствует сквозному отверстию и принимает значение из множества $\{1, 2, \dots, nСО\}$, где $nСО$ – число возможных параметров для сквозного отверстия. Элемент «ТС» соответствует типу соединения компонента устройства и принимает значения из множества $\{1, 2\}$, где значение 1 – последовательное соединение, а 2 – параллельное соединение. Элемент «ТЭ» соответствуют типу выбранного компонента устройства, принимает значение из множества $\{1, 2, \dots, 7\}$, где 1 – катушка индуктивности, 2 – конденсатор, 3 – транзистор, 4 – резистор, 5 – линия передачи, 6 – сквозное отверстие, 7 – компонент отсутствует.



Рисунок 4.14 – Кодирование секции аттенюатора вектором решений

Целевая функция оптимизации представляет собой взвешенную сумму максимального отклонения коэффициента передачи сигнала в состоянии ослабления (K_1), максимального отклонения коэффициента отражения сигнала по входу и выходу в состоянии ослабления (K_2), максимального отклонения коэффициента отражения сигнала по входу и выходу в опорном состоянии (K_3) и максимального отклонения фазы сигнала в состоянии ослабления от фазы сигнала в опорном состоянии (F):

$$E = w_1 K_1 + w_2 K_2 + w_3 K_3 + w_4 F ,$$

где w_1, \dots, w_4 – весовые коэффициенты отклонений, принимающие значения из диапазона $[0, 10]$ и задающиеся пользователем для соблюдения компромисса между отклонениями в найденных решениях.

Проведен вычислительный эксперимент на задаче синтеза секции ступенчатого аттенюатора с ослаблением 4 дБ и диапазоном частот 0,1–20 ГГц. В ходе проведения эксперимента проведено 1000 запусков алгоритма с максимальным количеством итераций 10000. В 56,1% запусках алгоритм нашел приемлемое решение. Среднее время синтеза до получения первого приемлемого решения составило 3 минуты 40 секунд.

4.4 Краткие выводы по разделу

В разделе представлены решения прикладных задач, в которых использовались результаты диссертационного исследования. Ниже приведены краткие выводы о внедрении результатов.

1. С помощью методики построения нечетких систем создан классификатор и на его основе программные средства для прогнозирования эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19. По показателям здоровья пациента классификатор определяет эффект от реабилитационного комплекса, основная цель которого заключается в восстановлении здоровья организма после болезни. Получена база правил, определяющая зависимость между значениями семи показателей здоровья пациента и эффективности реабилитации. Результат решения классификатора может быть объяснен с помощью выделения правила, к которому образец данных имеет наибольшую степень соответствия. Используется в Томском научно-исследовательском институте курортологии и физиотерапии.

2. Разработанный алгоритм смешанной многокритериальной оптимизации применен в программных средствах синтеза широкополосных согласующих цепей. Задача синтеза может быть сведена к смешанной многокритериальной оптимизации, где целевыми значениями являются минимизация отклонения коэффициента передачи АЧХ от заданного значения и минимизация числа компонентов цепи. Целочисленная часть вектора решения кодирует структуру, а

вещественная – параметры компонентов. С помощью алгоритм MISC проводится решение данной оптимизационной задачи, результатом которого являются множество согласующих цепей. Эксперименты на типовых задачах широкополосного согласования показали работоспособность и эффективность синтеза по сравнению с аналогичными методами синтеза. В задаче одностороннего согласования полученная цепь показала второй результат из девяти вариантов по показателю минимального значения коэффициента передачи, который от лучшего решения отличается на 0,0015дБ, а от худшего на 0,0559 дБ. В задаче двухстороннего синтезированной цепь показала лучший результат данного показателя из девяти вариантов. В задаче согласования-выравнивания предложенный метод показал третий результат из восьми вариантов по показателю максимального отклонения коэффициента передачи от желаемого значения с отклонением 0,011дБ от лучшего варианта и 0,096 дБ от худшего. Результаты внедрены в научно-исследовательском институте микроэлектронных систем и ООО «Микроэлектронные системы».

3. Разработанный алгоритм дискретной оптимизации на основе распределения вероятностей с трансформацией целевых значений внедрен в экспериментальную версию интеллектуальной системы автоматизированного проектирования «50ohm Tech Circuit Studio». Испытание в рамках опытной эксплуатации выявило, что в 56 запусках алгоритма из 100 для задачи синтеза ступенчатого аттенюатора находится приемлемое решение с максимальным числом итерации 10000 и средним временем работы 3 минуты 40 секунд. Результаты внедрены в ООО «50ом Тех.».

Заключение

В диссертационном исследовании разработана методология построения интерпретируемых нечетких классификаторов, основанных на правилах, которую можно квалифицировать как научное достижение в области построения систем объяснимого искусственного интеллекта. В ходе исследования получены следующие научные результаты.

1. Обоснована необходимость разработки систем искусственного интеллекта, объясняющих человеку результат своего решения и называемых системами объяснимого искусственного интеллекта. Способность объяснить результат таких систем повышает уровень доверия человека к полученному результату и тем самым обеспечивает возможность их применения в критически важных проблемных областях.

2. Определена проблема построения интерпретируемых моделей машинного обучения для систем объяснимого искусственного интеллекта. Общественное признание данной проблемы подтверждено в Национальной стратегии развития искусственного интеллекта на период до 2030 года и национальном стандарте Российской Федерации «Системы искусственного интеллекта. Способы обеспечения доверия»,

3. Показана возможность достижения интерпретируемости нечеткими классификаторами и их способность объяснить результат на естественном языке. База правил классификаторов состоит из продукционных правил, содержащих лингвистические термины для оценивания значений признаков классифицируемого объекта. Данные правила описывают связь между значениями признаков объекта и его классом, что позволяет объяснить моделируемые классификатором процесс или явление. Применение теории нечеткой логики позволяет находить численные решения для выводов в понятной человеку лингвистической форме. Объяснение полученного результата основывается на правилах, которые определили найденный класс.

4. На основе аналитического обзора выявлены недостатки нечетких классификаторов, связанные с низким уровнем интерпретируемости в современных методах построения. Нечеткие классификаторы типа Min-Max содержат большое количество правил, которое доходит до тысяч, и трудноинтерпретируемые в лингвистические термины функции принадлежности, что фактически делает невозможным перевести данные правила на естественный язык. Генетические нечеткие системы помимо большого количества правил содержат весовые коэффициенты правил, что также усложняет интерпретируемость. Метод трехэтапного построения нечетких классификаторов формирует правила с функциями принадлежности, форма которых может сильно отличаться от формы функций принадлежности при равномерном разбиении пространства признака. Это затрудняет интерпретацию нечетких множеств в лингвистические термины.

5. Предложена методология построения интерпретируемых нечетких классификаторов, основанных на правилах. Методология определяет совокупность методов анализа данных, процессы, в которых применяются данные методы, и их взаимодействие для построения нечетких классификаторов. В методологии используются методы предобработки данных, методы разделения данных, методы смешанной многокритериальной оптимизации, стратегии выбора решений на фронте Парето и методы оптимизации, основанные на производных.

6. Разработан, реализован и исследован алгоритм многокритериальной оптимизации для поиска оптимума в смешанном, дискретно-непрерывном пространстве, основанный на метаэвристике «кукушкин поиск» и генетическом операторе скрещивания. Генерация новых значений дискретных переменных осуществляется с помощью генетического оператора, а значения непрерывных переменных генерируются метаэвристикой. Алгоритм показал высокую эффективность на комплексных оценках сходимости и многообразия.

7. Разработан, реализован и исследован алгоритм оптимизации для поиска решения в дискретном пространстве, основанный на распределениях вероятностей. Распределения вероятностей формируются с помощью весовых

коэффициентов решений, которые определяются трансформационными функциями из целевых значений. Алгоритм показал высокую эффективность на оценках сходимости и стабильности.

8. Разработан метод разделения данных на обучающую и валидационную выборки для построения прогностических моделей машинного обучения, с целью сохранения близости полученных выборок к исходным данным. Проблема разделения данных в методе сводится к задаче бинарной оптимизации. Введен критерий близости, основанный на среднем значении и среднеквадратичном отклонении признаков данных. Применение метода для построения нечетких классификаторов с помощью многокритериального метаэвристического алгоритма показало повышение точности классификации и интерпретируемости моделей по сравнению со случайным разделением и без разделения исходных данных.

9. Адаптирован, реализован и исследован алгоритм градиентного спуска для настройки параметров среднего значения и среднеквадратичного отклонения функций принадлежности гауссова типа в нечетких классификаторах. Для этого классификатор представлен в виде системы разделяющих функций. Для оптимизации используется целевая функция в виде суммы квадратичных отклонений нормированных значений разделяющих функций от ожидаемых значений. Алгоритм продемонстрировал увеличение точности классификаторов, построенных многокритериальным метаэвристическим алгоритмом.

10. Предложена методика построения интерпретируемых нечетких классификаторов, основанная на методологии построения и использующая разработанные алгоритмы оптимизации и метод разделения данных. Для построения множества недоминируемых по критериям точности и интерпретируемости классификаторов применяется алгоритм смешанной многокритериальной оптимизации «кукушкин поиск» с генетическим оператором скрещивания. Для разделения данных на обучающую и валидационную выборки применяется метод разделения данных с дискретным алгоритмом оптимизации на основе распределения вероятностей. Для дообучения классификатора путем

настройки параметров функций принадлежности используется адаптированный алгоритм градиентного спуска. Построенные с помощью методики классификаторы продемонстрировали сопоставимую точность классификации и высокий уровень интерпретируемости по сравнению с известными интерпретируемыми моделями, такими как генетические нечеткие системы FARC-HD и деревья решений CART.

11. Проведены вычислительный эксперимент разработанного алгоритма смешанной многокритериальной оптимизации и сравнение его эффективности с известным генетическим алгоритмом с недоминируемой сортировкой NSGAII на общепринятых тестах ZDT и DTZL. Выявлено статистически значимое преимущество (на уровне $\alpha=0,05$) разработанного алгоритма на комплексных оценках сходимости и многообразия по сравнению с NSGAII. Значение обратного расстояния поколений оказалось меньше в среднем на 95,47%, а степень доминирования больше в среднем на 79,92%.

12. Проведены вычислительный эксперимент разработанного дискретного алгоритма оптимизации и сравнение его эффективности с известными генетическим алгоритмом и алгоритмом роящихся частиц. Эксперимент проводился на общепринятых тестах по поиску оптимума унимодальных и мультимодальных функций в бинарном пространстве поиска. Статистическое сравнение показало значимое преимущество (на уровне $\alpha=0,05$) предлагаемого алгоритма по критериям сходимости и стабильности. В среднем отклонение от оптимального значения целевой функции уменьшилось в 4,3 раза по сравнению с генетическим алгоритмом и в 10,1 раза по сравнению с алгоритмом роящихся частиц.

13. Проведен вычислительный эксперимент по настройке параметров функций принадлежности нечетких классификаторов с помощью адаптированного алгоритма градиентного спуска. Классификаторы были предварительно построены разработанным метаэвристическим алгоритмом смешанной многокритериальной оптимизации на 38 общедоступных наборах данных. Результаты эксперимента показали статистически значимое (на уровне

$\alpha=0,05$) повышение точности. Максимальное увеличение точности классификации составило 5,1%.

14. Проведен вычислительный эксперимент для оценки эффективности применения разработанного метода разделения данных при построении нечетких классификаторов с помощью алгоритма смешанной многокритериальной оптимизации на 38 общедоступных наборах данных. Статистический тест выявил значимое повышение точности (на уровне $\alpha=0,05$) классификации по сравнению с построением классификаторов со случайным разделением данных и без применения разделения данных. Максимальное увеличение точности составило 6,5%. Также выявлено значимое (на уровне $\alpha=0,05$) повышение интерпретируемости по показателям количества правил, количества признаков и общего количества нечетких терминов в базе правил.

15. Проведен вычислительный эксперимент по построению нечетких классификаторов с помощью предлагаемой методики построения на 38 общедоступных наборах данных. Проведено статистическое сравнение показателей точности и интерпретируемости с известными интерпретируемыми моделями машинного обучения, такими как генетические нечеткие системы FARC-ND и деревья решений CART. Сравнение показало сопоставимую точность классификации (на уровне $\alpha=0,05$) и выявило статистически значимое преимущество (на уровне $\alpha=0,05$) по критерию интерпретируемости классификаторов, построенных с помощью предлагаемой методики. По сравнению с классификаторами FARC-ND значения таких показателей как количество правил, общее количество нечетких терминов и количество признаков оказались значимо меньше. Количество правил меньше в среднем на 39,2%, общее количество нечетких терминов на 49,8%, количество признаков на 8%. По сравнению с деревьями решений CART значимое уменьшение показали количество правил, в среднем на 81,9%, и количество условий в правиле, в среднем на 54,5%. Результаты сравнения подтверждают высокий уровень интерпретируемости нечетких классификаторов, построенных с помощью предлагаемой методики.

16. С помощью разработанной методики построен нечеткий классификатор для прогнозирования эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19. Получена база из трех продукционных правил, объясняющая связь между показателями здоровья пациента и эффектом от реабилитации. Из 58 исходных признаков в классификаторе используется 7. На основе классификатора созданы программные средства, осуществляющие прогнозирование эффективности реабилитации с точностью 74,67%. Результаты внедрены в Томском НИИКиФ.

17. Алгоритм смешанной многокритериальной оптимизации «кукушкин поиск» с генетическим оператором скрещивания, разработанный в диссертационной работе, применяется в программных средствах для синтеза цепей согласования. Тестирование на типовых задачах широкополосного согласования показало работоспособность и эффективность синтеза по сравнению с альтернативными методами. В задаче одностороннего согласования полученная цепь показала второй результат из девяти вариантов по показателю минимального значения коэффициента передачи, который от лучшего решения отличается на 0,0015дБ, а от худшего на 0,0559 дБ. В задаче двухстороннего согласования найденная цепь продемонстрировала лучший результат данного показателя из девяти вариантов цепей. В задаче согласования-выравнивания синтезированная цепь показала третий результат из восьми вариантов по показателю максимального отклонения коэффициента передачи от желаемого значения с отклонением 0,011дБ от лучшего варианта и 0,096 дБ от худшего. Результаты внедрены в НИИ МЭС и ООО «Микроэлектронные системы» для проектирования СВЧ усилителей различного назначения.

18. Дискретный алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений, разработанный в диссертационной работе, внедрен в экспериментальную версию интеллектуальной САПР «50ohm Tech Circuit Studio» и испытан в рамках опытной эксплуатации для синтеза секции ступенчатого СВЧ аттенюатора. В 56 запусках алгоритма из 100 находится применимое решение с максимальным

числом итераций 10000. Среднее время синтеза находится в приемлемых для решения поставленной практической задачи пределах и составляет 3 мин. 40 сек. Результаты внедрены в ООО «50ом Тех.».

Таким образом, цель диссертационной работы, заключающаяся в повышении эффективности обнаружения закономерностей в данных с помощью разработки и применения методологии построения интерпретируемых нечетких классификаторов, основанных на правилах, была достигнута, а все поставленные задачи были выполнены.

Перспективы дальнейших исследований автор видит в разработке методов объяснения для систем искусственного интеллекта, основанных на трудноинтерпретируемых моделях «черного ящика», таких как нейронные сети глубокого обучения или ансамбли. Перспективным решением в данной области является построение моделей-двойников на основе нечетких систем, имитирующих прогнозы оригинальных моделей. Такое решение позволит получить базу продукционных правил, объясняющую человеку связь между признаками объекта и целевым значением, а также обосновать прогнозируемый результат.

Список литературы

1. Wiese T. Predictive Maintenance Using Artificial Intelligence in Critical Infrastructure: A Decision-Making Framework / T. Wiese // International Journal of Engineering, Business and Management. – 2024. – Vol. 8, No. 4. – P. 1–4.

2. Goodman B. European union regulations on algorithmic decision making and a «right to explanation» / B. Goodman, S. Flaxman // AI Magazine. – 2017. – Vol. 38, No. 3. – P. 50–57.

3. Zhu J. Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation / J. Zhu, A. Liapis, S. Risi, R. Bidarra, G. M. Youngblood // 2018 IEEE Conference on Computational Intelligence and Games (CIG). – 2018. – P. 1–8.

4. A qualitative exploration of perceptions of algorithmic fairness / A. Woodruff, S. Fox, S. Rousso-Schindler, J. Warshaw // Proceedings Conference on Human Factors in Computing Systems. – 2018. – P. 1–14.

5. Ходашинский И.А. Этические проблемы искусственного интеллекта / И.А. Ходашинский, Ю.А. Шурыгин, К.С. Сарин, Е.А. Кузьмина // Современное образование: интеграция образования, науки, бизнеса и власти. Трансформация образования, науки и производства - основа технологического прорыва: материалы международной научно-методической конференции. В 2 ч., Томск, 26–27 января 2023 года. Том Часть 2. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2023. – С. 125–129.

6. Borisov V. Deep Neural Networks and Tabular Data: A Survey / V. Borisov, T. Leemann, K. Sesler, J. Haug, M. Pawelczyk, J. Kasneci // IEEE Transactions on Neural Networks and Learning Systems. – 2024. – Vol. 35, No. 6. – P. 7499–7519.

7. Castelvechi D. Can we open the black box of AI? / D. Castelvechi // Nature. – 2016. – Vol. 538. – P. 20–23.

8. Lipton Z.C. The mythos of model interpretability / Z.C. Lipton // Communications of the ACM. – 2016. – Vol. 61. – P. 36–43.

9. Аверкин А.Н. Объяснимый искусственный интеллект как часть искусственного интеллекта третьего поколения / А.Н. Аверкин // Речевые технологии. – 2023. – № 1. – С. 4–10.
10. Gunning D. DARPA's Explainable Artificial Intelligence (XAI) Program / D. Gunning, W. Aha // AI Magazine. – 2019. – Vol. 40. – P. 44–58.
11. Ahmed I. From Artificial Intelligence to Explainable Artificial Intelligence in Industry 4.0: A Survey on What, How, and Where / I. Ahmed, G. Jeon, F. Piccialli // IEEE Transactions on Industrial Informatics. – 2022. – Vol. 18. – P. 5031–5042.
12. Moral J.M.A. Explainable Fuzzy Systems / J.M.A. Moral, C. Castiello, L. Magdalena, C. Mencar. – Cham: Springer, 2021. – 232 p.
13. Carvalho D.V. Machine Learning Interpretability: A Survey on Methods and Metrics / D.V. Carvalho, E.M. Pereira, J.S. Cardoso // Electronics. – 2019. – Vol. 8. – P. 832.
14. Zadeh L.A. From computing with numbers to computing with words. From manipulation of measurements to manipulation of perception / Zadeh L.A. // IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. – 1999. – Vol. 46, No. 1. – P. 105–119.
15. Zadeh L.A. Toward human-level machine intelligence: is it achievable? The need for a paradigm shift / Zadeh L.A. // IEEE Computational Intelligence Magazine. – 2008. – Vol.3, No. 3. – P. 11–22.
16. Casalino G. Effect of fuzziness in fuzzy rule-based classifiers defined by strong fuzzy partitions and winner-takes-all inference / G. Casalino, G. Castellano, C. Castiello, C. Mencar // Soft Computing. – 2022. – Vol. 26. – P. 6519–6527.
17. Дуда Р.О. Распознавание образов и анализ сцен / Р. Дуда, П. Харт; Пер. с англ. Г.Г. Вайнштейна и А.М. Васьковского; Под ред. В.Л. Стефанюка. – М.: Мир, 1976. – 511 с.
18. Ту Д.Т. Принципы распознавания образов / Д.Т. Ту, Р.К. Гонсалес; Пер. с англ. И. Б. Гуревича; Под ред. Ю. И. Журавлева. – Москва: Мир, 1978. – 411 с.
19. Mendel J.M. Fuzzy logic systems for engineering: a tutorial / J.M. Mendel // Proceedings IEEE. – 1995. – Vol. 83. – P. 345–377.

20. Zadeh L.A. Fuzzy Sets / L.A. Zadeh // Information and Control. – 1965. – Vol. 8. – P. 338–353.

21. Zadeh L.A. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes / L.A. Zadeh // IEEE Transactions on Systems, Man and Cybernetics. – Vol. SMC-3, No. 1. – P. 28–44.

22. Горбачев С.В. Метод синтеза нейро-нечеткого классификатора на основе дерева решений / С.В. Горбачев, В.И. Сырякин, Т.В. Абрамова // Телекоммуникации. – 2018. – № 9. – С. 2–7.

23. Mamdani E.H. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller / E.H. Mamdani, S. Assilian // International Journal of Man-Machine Studies. – 1975. – Vol. 7, No. 1. – P. 1–13.

24. Yager R. An Introduction to Fuzzy Logic Applications in Intelligent Systems // R. Yager, L. Zadeh. – Cham: Springer, 1992. – 357 p.

25. Ковалев С.М. Гибридный метод обучения стохастических моделей упреждения аномалий на основе нечетких продукций / С.М. Ковалев, А.Н. Гуда, А.В. Суханов // Вестник Ростовского государственного университета путей сообщения. – 2015. – Т. 59, № 3. – С. 40–47.

26. Кудинов Ю.И. Оптимизация параметров интеллектуальных нечетких регуляторов / Ю.И. Кудинов, А.Ю. Келина, А.Ф. Пащенко, Ф.Ф. Пащенко // XII всероссийское совещание по проблемам управления ВСПУ-2014, Москва, 16–19 июля 2014 года / Институт проблем управления им. В.А. Трапезникова РАН. – Москва: Институт проблем управления им. В.А. Трапезникова РАН, 2014. – С. 659–664.

27. Герасименко Е.М. Применение нечеткой логики для принятия решений об эвакуации при наводнении / Е.М. Герасименко, В.В. Курейчик, С.И. Родзин, А.П. Кухаренко // Известия ЮФУ. Технические науки. – 2022. – Т. 228, № 4. – С. 15–29.

28. Леденева Т.М. Нечеткое моделирование медицинских экспертных систем / Т.М. Леденева, С.Л. Подвальный, Р.К. Стрюков, С.В. Дегтярев // Биомедицинская радиоэлектроника. – 2016. – № 9. – С. 16–24.

29. Катасёв А.С. Классификация типа стекла на основе нейронечеткой модели / А.С. Катасёв, А.В. Каляшина, Ю.Н. Смирнов // Вестник Технологического университета. – 2025. – Т. 28, № 6. – С. 102–105.

30. Батыршин И.З. Нечеткие гибридные системы. Теория и практика / И.З. Батыршин, А.О. Недосекин, А.А. Стецко, В.Б. Тарасов, А.В. Язенин, Н.Г. Ярушкина. – Москва: ООО Издательская фирма «Физико-математическая литература», 2007. – 208 с.

31. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л.А. Заде. – М.: Мир, 1976. – 165 с.

32. Dubois D. Fuzzy sets and systems: theory and applications / D. Dubois, H. Pride. – New York: Academic Press, 1980. – 393 p.

33. Батыршин И.З. Основные операции нечеткой логики / И.З. Батыршин // Новости искусственного интеллекта. – 2001. – № 4. – С. 18–22.

34. Kruse F. Foundations of fuzzy systems / F. Kruse, J. Gebhardt, F. Klawonn. – Chichester, West Sussex, England: John Wiley and Sons, 1994. – 265 p.

35. Jezewski M. Introduction to Fuzzy Systems / M. Jezewski, R. Czabanski, J. Leski // Theory and Applications of Ordered Fuzzy Numbers / P. Prokopowicz, J. Czerniak, D. Mikołajewski, Ł. Apiecionek, D. Slezak (eds), 2021. – Studies in Fuzziness and Soft Computing. – Vol. 356. – P. 23–42.

36. Leski J. Neuro-fuzzy system with learning tolerant to imprecision / J. Leski // Fuzzy Sets and Systems. – 2003. – Vol. 138, No. 2. – P. 427–439.

37. Mendel J.M. Explainable Uncertain Rule-Based Fuzzy Systems / J.M. Mendel. – Cham, Switzerland: Springer, 2024. – 580 p.

38. Берштейн Л.С. Использование нечеткого дедуктивного вывода в системах управления / Л.С. Берштейн, А.В. Боженюк // Известия ТРТУ. – 2001. – Т. 22, № 4. – С. 58–64.

39. Kuncheva L.I. Fuzzy classifier design. – Berlin: Springer, 2000. – 315 p.

40. Takagi T. Fuzzy identification of systems and its application to modeling and control / T. Takagi, Sugeno M. // IEEE Transaction Systems, Man and Cybernetics. – 1985. – Vol. 15, No. 1. – P. 116–132.

41. Sugeno M. A fuzzy-logic-based approach to qualitative modeling / M. Sugeno, T. Yasukawa // IEEE Transaction Fuzzy System. – 1993. – Vol. 1, No. 1. – P. 7–31.
42. Ishibuchi H. Distributed representation of fuzzy rules and its application to pattern classification / H. Ishibuchi, K. Nozaki, H. Tanaka // Fuzzy Sets and Systems. – 1992. – Vol. 52, No. 1. – P. 2–132.
43. Ishibuchi H. Selecting fuzzy if-then rules for classification problems using genetic algorithms / H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka // IEEE Transactions on Fuzzy Systems. – 1995. – Vol. 3, No. 3. – P. 260–270.
44. Ishibuchi H. Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes / H. Ishibuchi, T. Nakaskima // IEEE Transactions on Industrial Electronics. – 1999. – Vol. 46. – P. 1057–1068.
45. Kosko B. Fuzzy systems as universal approximators / B. Kosko // Proceedings first IEEE International Conference Fuzzy System, San Diego, CA, Mar. – 1992. – P. 1153–1162.
46. Wang L. Generating Fuzzy Rules by Learning from Examples / L. Wang, J.M. Mendel // IEEE Transaction Systems, Man and Cybernetics. – 1992. – Vol. 22, No. 6. – P. 1414–1427.
47. Simpson P.K. Fuzzy Min-Max neural networks – Part 1: Classification / P.K. Simpson // IEEE Transaction on Neural Networks. – 1992. – Vol. 3, No. 5. – P. 776–786.
48. Сарин К. С. Нечеткий классификатор типа Min-Max: обзор / К. С. Сарин // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2023. – Т. 26, № 1. – С. 65-75.
49. Gheibi O. Lifelong Self-Adaptation: Self-Adaptation Meets Lifelong Machine Learning / O. Gheibi, D. Weyns // SEAMS '22: Proceedings of the 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems, May 2022. – 2022. – P. 1–12.

50. Angelov P. An Approach to On-line Identification of Takagi-Sugeno Fuzzy Models / P. Angelov, D. Filev // IEEE Transactions on System, Man, and Cybernetics, part B – Cybernetics. – 2004. – Vol. 34, No. 1. – P. 484–498.

51. Bargiela A. An inclusion/exclusion fuzzy hyperbox classifier / A. Bargiela, W. Pedrycz, M. Tanaka // International Journal of Knowledge-based and Intelligent Engineering Systems. – 2004. – Vol. 8. – P. 91–98.

52. Nandedkar A.V. A Fuzzy Min-Max Neural Network Classifier With Compensatory Neuron Architecture / A.V. Nandedkar, P.K. Biswas // IEEE Transactions on Neural Networks. – 2007. – Vol. 18, No. 1. – P. 42–54.

53. Zhang H. Data-core-based fuzzy min-max neural network for pattern classification / H. Zhang, J. Liu, D. Ma, Z. Wang // IEEE Transactions on Neural Networks. – 2011. – Vol. 22, No. 12. – P. 2339–2352.

54. Davtalab R. Multi-level fuzzy min-max neural network classifier / R. Davtalab, M.H. Dezfoulian, M. Mansoorizadeh // IEEE Transactions on Neural Networks Learning System. – 2014. – Vol. 25, No. 3. – P. 470–482.

55. Gabrys B. General fuzzy min-max neural network for clustering and classification / B. Gabrys, A. Bargiela // IEEE Transactions on Neural Networks. – 2000. – Vol. 11, No. 3. – P. 769–783.

56. Nandedkar A.V. A general reflex fuzzy min-max neural network / A.V. Nandedkar, P.K. Biswas // Engineering Letters. – 2007. – Vol. 14, No. 1. – P. 195–205.

57. Mohammed M.F. An enhanced fuzzy min-max neural network for pattern classification / M.F. Mohammed, C.P. Lim // IEEE Transactions on Neural Networks and Learning Systems. – 2017. – Vol. 26. – P. 417–429.

58. Mohammed M.F. Improving the fuzzy min-max neural network with a K-nearest hyperbox expansion rule for pattern classification / M.F. Mohammed, C.P. Lim // Applied Soft Computing – 2017.– No. 52. – P. 135–145.

59. Kumar S. A compact fuzzy min max network with novel trimming strategy for pattern classification / S. Kumar, A. Kumar, V. Bajaj, G.K. Singh // Knowledge-Based Systems. – 2022. – No. 246. – P. 108620.

60. Alhroob E. Fuzzy min-max classifier based on new membership function for pattern classification: A conceptual solution / E. Alhroob, N.A. Ghani // Proceedings - 8th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2018. – 2018. – P. 131–135.

61. Сарин К.С. Нечеткий классификатор инкрементного обучения типа Min-Max для решения задач кибербезопасности / К.С. Сарин, Р.Е. Коломников, М.О. Светлаков, И.А. Ходашинский // Научно-техническая информация. Серия 2: Информационные процессы и системы. – 2024. – № 9. – С. 11–21.

62. Sarin K.S. Fuzzy Min-Max Classifier in Cybersecurity Applications / K.S. Sarin, R.E. Kolomnikov, M.O. Svetlakov, I.A. Hodashinsky // Automatic Documentation and Mathematical Linguistics. – 2024. – Vol. 58, No. 5. – P. 299–309.

63. Коломников Р. Е. Нечеткий классификатор инкрементного обучения для распознавания динамической рукописной подписи / Р. Е. Коломников, К. С. Сарин // Электронные средства и системы управления. Материалы докладов Международной научно-практической конференции. – 2023. – № 1-2. – С. 286–291.

64. Коломников Р.Е. Нечеткий классификатор инкрементного обучения для обнаружения фишинговых сайтов / Р.Е. Коломников, К.С. Сарин // Математические методы в технологиях и технике. – 2024. – № 9. – С. 52–55.

65. Коломников Р.Е. Регрессионная модель для прогнозирования размера гипербокса нечеткого классификатора типа Min-Max / Р.Е. Коломников, К.С. Сарин // Электронные средства и системы управления. Материалы докладов Международной научно-практической конференции. – 2024. – № 1-2. – С. 253–256.

66. Azad C. A novel fuzzy min-max neural network and genetic algorithm-based intrusion detection system. / C. Azad, V.K. Jha // Proceedings of the second international conference on computer and communication technologies. – 2016. – P. 429–439.

67. Azad C. Fuzzy min–max neural network and particle swarm optimization based intrusion detection system / C. Azad, V.K. Jha // *Microsystem Technology*. – 2017. – Vol. 23, No. 4. – P. 907–918.

68. Khuat T.T. Random Hyperboxes / T.T. Khuat, B. Gabrys // *IEEE Transactions on Neural Networks and Learning Systems*. – 2023. – Vol. 34, No. 2. – P. 1008–1022.

69. Huang W. Deep Fuzzy Min–Max Neural Network: Analysis and Design / W. Huang, M. Sun, L. Zhu, S. Oh, W. Pedrycz // *IEEE Transactions on Neural Networks and Learning Systems*. – 2024. – Vol. 35, No. 6. – P. 8229–8240.

70. Khuat T.T. A comparative study of general fuzzy min-max neural networks for pattern classification problems / T.T. Khuat, B. Gabrys // *Neurocomputing*. – 2020. – Vol. 386. – P. 110–125.

71. Herrera F. Genetic fuzzy systems: taxonomy, current research trends and prospects // *Evolution Intelligence*. – 2008. – Vol. 1, No. 1. – P. 27–46.

72. Yang X.-S. Genetic Algorithms / X.-S. Yang // *Nature-Inspired Optimization Algorithms (Second Edition)*. – Academic Press, 2021. – P. 91–100.

73. Cordon O. Ten years of genetic fuzzy systems: current framework and new trends / O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena // *Fuzzy Sets System*. – 2004. – Vol. 141. – P. 5–31.

74. Castillo O. Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review / O. Castillo, P. Melin // *Information Sciences*. – 2012. – Vol. 205. – P. 1–19.

75. Au W-H. A fuzzy approach to partitioning continuous attributes for classification / W-H Au, K.C.C. Chan, A.K.C. Wong // *IEEE Transactions on Knowledge and Data Engineering*. – 2006. – Vol. 18, No. 5. – P. 715–719.

76. Ishibuchi H. Comparison of the Michigan and Pittsburgh approaches to the design of fuzzy classification systems / H. Ishibuchi, T. Nakashima, T. Murata // *Electronics and Communications in Japan*. – 1997. – Vol. 80, No. 12. – P. 10–19.

77. Schaefer G. Michigan vs. Pittsburgh Style GA Optimisation of Fuzzy Rule Bases for Gene Expression Analysis / G. Schaefer, T. Nakashima // *International Journal of Fuzzy System Applications*. – 2013. – Vol.3, No. 4. – P. 60–72.

78. Ishibuchi H. Performance Evaluation of Fuzzy Classifier Systems for Multi-Dimensional Pattern Classification Problems / H. Ishibuchi, T. Nakashima, T. Murata // *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*. – 1999. – Vol. 29. – P. 601–618.

79. Nojima Y. Michigan-style fuzzy GBML with (1+1)-ES generation update and multi-pattern rule generation / Y. Nojima, S. Takemura, K. Watanabe, H. Ishibuchi // *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, Otsu, Japan. – 2017. – P. 1–6.

80. Ishibuchi H. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining / H. Ishibuchi, T. Yamamoto // *Fuzzy Sets and Systems*. – 2004. – Vol. 141, No. 1. – P. 59–88.

81. Ishibuchi H. Multiobjective optimization in linguistic rule extraction from numerical data / H. Ishibuchi, T. Nakashima, T. Murata // *Evolutionary Multi-Criterion Optimization*, Berlin: Springer. – 2001. – P. 588–602.

82. Cordon O. Villar Analysis and guidelines to obtain a good uniform fuzzy partition granularity for Fuzzy Rule-Based Systems using simulated annealing / O. Cordon, F. Herrera, P. Villar // *International Journal of Approximate Reasoning*. – 2000. – Vol.2, No. 3. – P. 187–215.

83. Alcalá R. Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions / R. Alcalá, Y. Nojima, F. Herrera, H. Ishibuchi // *Soft Computing*. – 2011. – Vol. 15. – P. 2303–2318.

84. Fazzolari M. Multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm / M. Fazzolari, R. Alcalá, F. Herrera // *Applied Soft Computing*. – 2014. – Vol. 24. – P. 470–481.

85. Kurgan L.A. Caim discretization algorithm / L.A. Kurgan, K.J. Cios // IEEE Transactions on Knowledge and Data Engineering. – 2004. – Vol. 16, No. 2. – P. 145–153.
86. Alcalá-Fdez J. A Fuzzy Association Rule-Based Classification Model for High-Dimensional Problems With Genetic Rule Selection and Lateral Tuning / J. Alcalá-Fdez, R. Alcalá, F. Herrera // IEEE Transactions on Fuzzy Systems. – 2011. – Vol. 19, No. 5. – P. 857–872.
87. Elkano M. Improving the Performance of FARC-HD in Multi-class Classification Problems Using the One-Versus-One Strategy and an Adaptation of the Inference System / M. Elkano, M. Galar, J. Sanz, E. Barrenechea, F. Herrera, H. Bustince // Communications in Computer and Information Science. – 2014. – Vol. 444. – P. 296–306.
88. Alcalá R. A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection / R. Alcalá, J. Alcalá-Fdez, F. Herrera // IEEE Transactions of Fuzzy System. – 2007. – Vol. 15, No. 4. – P. 616–635.
89. Gonzalez A. Selection of relevant features in a fuzzy genetic learning algorithm / A. Gonzalez, R. Perez // IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). – 2001. – Vol. 31, No. 3. – P. 417–425.
90. Ishibuchi H. Hybridization of fuzzy GBML approaches for pattern classification problems / H. Ishibuchi, T. Yamamoto, T. Nakashima // IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). – 2005. – Vol. 35, No. 2. – P. 359–365.
91. Mansoori E.G. SGERD: A Steady-State Genetic Algorithm for Extracting Fuzzy Classification Rules From Data / E.G. Mansoori, M.J. Zolghadri, S.D. Katebi // IEEE Transactions on Fuzzy Systems. – 2008. – Vol. 16, No. 4. – P. 1061–1071.
92. Gonzalez A. SLAVE: A genetic learning system based on an iterative approach / A. Gonzalez, R. Perez // IEEE Transactions on Fuzzy Systems. – 1999. – Vol. 7, No. 2. – P. 176–191.
93. Quinlan J. C4.5: Programs for Machine Learning / J. Quinlan. – San Mateo, CA: Morgan Kauffman, 1993. – 304 p.

94. Liu B. Classification using association rules: Weaknesses and enhancements / B. Liu, Y. Ma, C. Wong, // Data Mining for Scientific and Engineering. – Norwell, MA: Kluwer, 2001. – P. 591–601.

95. Li W. CMAR: Accurate and efficient classification based on multiple class-association rules / W. Li, J. Han, J. Pei // Proceedings of IEEE International Conference on Data Mining, San Jose, CA, 2001. – P. 369–376.

96. Yin X. CPAR: Classification based on predictive association rules / X. Yin, J. Han, // Proceedings of the 3rd SIAM International Conference on Data Mining, San Francisco, CA, 2003. – P. 331–335.

97. Sarin K. A three-stage fuzzy classifier method for Parkinson's disease diagnosis using dynamic handwriting analysis / K. Sarin, M. Bardamova, M. Svetlakov, N. Koryshev, R. Ostapenko, A. Hodashinskaya, I. Hodashinsky // Decision Analytics Journal. – 2023. – Vol. 5. – P. 100274.

98. Ходашинский И.А. Алгоритмы идентификации интерпретируемых и точных нечетких классификаторов / И.А. Ходашинский, И.В. Горбунов // XII всероссийское совещание по проблемам управления ВСПУ-2014, Москва, 16–19 июля 2014 года / Институт проблем управления им. В.А. Трапезникова РАН. – Москва: Институт проблем управления им. В.А. Трапезникова РАН, 2014. – С. 3269–3280.

99. Hodashinsky I. Feature selection based on swallow swarm optimization for fuzzy classification / I. Hodashinsky, K. Sarin, A. Shelupanov, A. Slezkin // Symmetry. – 2019. – Vol. 11, No. 11. – P. 1423.

100. Ходашинский И.А. Отбор классифицирующих признаков с помощью популяционного случайного поиска с памятью / И.А. Ходашинский, К.С. Сарин // Автоматика и телемеханика. – 2019. – № 2. – С. 161–172.

101. Sarin K.S. Identification of fuzzy classifiers based on the mountain clustering and cuckoo search algorithms / K.S. Sarin, I.A. Hodashinsky // 2017 International Siberian Conference on Control and Communications, SIBCON 2017 - Proceedings, Astana. – Astana, 2017. – P. 1–6.

102. Сарин К.С. Метод Чиу для отбора информативных признаков нечетких классификаторов / К.С. Сарин, И.А. Ходашинский // Информатика и системы управления. – 2017. – № 3(53). – С. 84–95.

103. Ходашинский И.А. Методика построения компактных и точных нечетких систем типа Такаги-Сугено / И.А. Ходашинский, К.С. Сарин // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2016. – Т. 19, № 1. – С. 50–56.

104. Ходашинский И.А. Об одном методе инициализации нечетких систем типа Такаги-Сугено / И.А. Ходашинский, К.С. Сарин, С.А. Черепанов // Автометрия. – 2016. – Т. 52, № 2. – С. 61–70.

105. Hodashinsky I.A. Initialization method for fuzzy Takagi–Sugeno systems / I.A. Hodashinsky, K.S. Sarin, S.A. Cherepanov // Optoelectronics, Instrumentation and Data Processing. – 2016. – Vol. 52, No. 3. – P. 266–273.

106. Ходашинский И.А. Идентификация параметров нечетких аппроксиматоров и классификаторов на основе алгоритма «кукушкин поиск» / И.А. Ходашинский, Д.Ю. Минина, К.С. Сарин // Автометрия. – 2015. – Т. 51, № 3. – С. 27–34.

107. Hodashinsky I.A. Identification of the parameters of fuzzy approximators and classifiers based on the cuckoo search algorithm / I.A. Hodashinsky, D.Y. Minina, K.S. Sarin // Optoelectronics, Instrumentation and Data Processing. – 2015. – Vol. 51, No. 3. – P. 234–240.

108. Сарин К.С. Нечеткий аппроксиматор значения плотности тока в пучке электронов форвакуумного плазменного источника / К.С. Сарин, А.В. Медовник, И.А. Ходашинский // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2015. – № 4(38). – С. 161–165.

109. Hodashinsky I.A. Feature Selection for Classification through Population Random Search with Memory / I.A. Hodashinsky, K.S. Sarin // Automation and Remote Control. – 2019. – Vol. 80, No. 2. – P. 324–333.

110. Сарин К. С. Формирование оптимального подмножества входных признаков нечеткого классификатора методом Чиу / К. С. Сарин // Перспективы

развития фундаментальных наук: сборник научных трудов XIV Международной конференции студентов, аспирантов и молодых ученых, Томск, 25–28 апреля 2017 года. Том 7. – Томск: Национальный исследовательский Томский политехнический университет, 2017. – С. 102–104.

111. Ходашинский И. А. Отбор классифицирующих признаков: сравнительный анализ бинарных метаэвристик и популяционного алгоритма с адаптивной памятью / И.А. Ходашинский, К.С. Сарин // Программирование. – 2019. – № 5. – С. 3–9.

112. Hodashinsky I.A. Feature selection: Comparative Analysis of Binary Metaheuristics and Population Based Algorithm with Adaptive Memory / I.A. Hodashinsky, K.S. Sarin // Programming and Computer Software. – 2019. – Vol. 45, No. 5. – P. 221–227.

113. Hancer E. A wrapper metaheuristic framework for handwritten signature verification / E. Hancer, I. Hodashinsky, K. Sarin, A. Slezkin // Soft Computing - A Fusion of Foundations, Methodologies and Applications. – 2021. – Vol. 25, No. 13. – P. 8665–8681.

114. Анфилофьев А.Е. Метаэвристические методы отбора информативных классифицирующих признаков / А.Е. Анфилофьев, И.А. Ходашинский, М.Б. Бардамова, К.С. Сарин // Информационные и математические технологии в науке и управлении. – 2017. – № 2(6). – С. 11–20.

115. Сарин К.С. Метод баггинга и отбор признаков в построении нечетких классификаторов для распознавания рукописной подписи / К.С. Сарин, И.А. Ходашинский // Компьютерная оптика. – 2019. – Т. 43, № 5. – С. 833–845.

116. Cai J. Feature selection in machine learning: A new perspective / J. Cai, J. Luo, S. Wang, S. Yang // Neurocomputing. – 2018. – Vol. 300. – P. 70–79.

117. Ходашинский И.А. Метод нахождения подмножеств согласованных признаков при прогнозировании эффективности реабилитации пациентов после перенесенной коронавирусной инфекции / И.А. Ходашинский, И.Н. Смирнова, М.Б. Бардамова, К.С. Сарин, М.О. Светлаков, А.А. Зайцев, Е.В. Тицкая, А.В. Тонкошкурова, И.И. Антипова, А.И. Ходашинская, Т.Н. Зарипова // Сибирский

журнал клинической и экспериментальной медицины. –2023. – Т. 38, № 4. – С. 270–279.

118. Kononenko I. Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF / I. Kononenko, E. Simec, M. Robnik-Sikonja // Applied Intelligence. – 1997. – Vol.7, No. 1. – P.39–55.

119. Robnik-Sikonja M. Theoretical and Empirical Analysis of ReliefF and RReliefF / M. Robnik-Sikonja, I. Kononenko // Machine Learning. – 2003. – Vol 53. – P.23–69.

120. Ding C. Minimum redundancy feature selection from microarray gene expression data / C. Ding, H. Peng // Journal of Bioinformatics and Computational Biology. – 2005. – Vol. 3, No. 2. – P. 185-205.

121. Darbellay G.A. Estimation of the information by an adaptive partitioning of the observation space / G.A. Darbellay, I. Vajda // IEEE Transactions on Information Theory. – 1999. – Vol. 45, No. 4. – P. 1315–1321.

122. Chiu S.L. Selecting input variables for fuzzy model / S.L. Chiu // Journal of Intelligent and Fuzzy Systems. – 1996. – Vol. 4, No. 4. – P. 267–278.

123. Bardamova M. Methods for Binarization of Metaheuristic Algorithms for Constructing Compact Fuzzy Classifiers of Medical Data / M. Bardamova, M. Svetlakov, K. Sarin, A. Hodashinskaya, Y. Shurygin, I. Hodashinsky // Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications. – 2024. – Vol. 34, No. 4. – P. 1303–1322.

124. Hancer E. Binary PSO Variants for Feature Selection in Handwritten Signature Authentication / E. Hancer, M. Bardamova, I. Hodashinsky, K. Sarin, A. Slezkin, M. Svetlakov // Informatica. – 2022. – Vol. 33, No. 3. – P. 523–543.

125. Бардамова М.Б. Способы адаптации алгоритма прыгающих лягушек к бинарному пространству поиска при решении задачи отбора признаков / М. Б. Бардамова, А. Г. Буймов, В. Ф. Тарасенко // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2020. – Т. 23, № 4. – С. 57–62.

126. Sarin K. Feature selection and identification of fuzzy classifiers based on the cuckoo search algorithm / K. Sarin, I. Hodashinsky, A. Slezkin // *Communications in Computer and Information Science*. – 2018. – Vol. 934. – P. 22–34.

127. Hodashinsky I. Efficient feature selection algorithm based on population random search with adaptive memory strategies / I. Hodashinsky, K. Sarin, A. Slezkin // *Advances in Intelligent Systems and Computing*. – 2019. – Vol. 874. – P. 321–330.

128. Ходашинский И.А. Построение нечеткого классификатора на основе методов гармонического поиска / И.А. Ходашинский, М.А. Мех // *Программирование*. – 2017. – № 1. – С. 54–65.

129. Bardamova M.B. Generation of a Fuzzy Classifier Rule Base for Diagnosing Parkinson's Disease from Handwritten Data / M.B. Bardamova, I.A. Hodashinsky, Yu.A. Shurygin, K.S. Sarin, M.O. Svetlakov // *Scientific and Technical Information Processing*. – 2024. – Vol. 51, No. 6. – P. 609–619.

130. Бардамова М.Б. Генерация базы правил нечеткого классификатора для диагностики болезни Паркинсона по рукописным данным / М.Б. Бардамова, И.А. Ходашинский, Ю.А. Шурыгин, К.С. Сарин, М.О., Светлаков // *Искусственный интеллект и принятие решений*. – 2023. – № 2. – С. 31–44.

131. Chiu S.L. Fuzzy model identification based on cluster estimation / S.L. Chiu // *Journal of Intelligent and Fuzzy System*. – 1994. – Vol. 2, No. 3. – P.267–278.

132. Sarin K. Linear regression to determine the cluster radius for fuzzy rule base generation / K. Sarin, I. Hodashinsky, I. Filimonenko // *2019 International Siberian Conference on Control and Communications, SIBCON 2019 – Proceedings, Tomsk, 2019*. – P. 1–4.

133. Bezdek J.C. FCM: the fuzzy c-means clustering algorithm / J.C. Bezdek, R. Ehrlich, W. Full // *Computers and Geosciences*. – 1984. – Vol. 10, No. 2-3. – P. 191–203.

134. Комарцова Л.Г. Эволюционные методы формирования нечетких баз правил / Л.Г. Комарцова // *Открытые семантические технологии проектирования интеллектуальных систем*. – 2011. – № 1. – С. 181–184.

135. Мех М.А. Сравнительный анализ применения методов дифференциальной эволюции для оптимизации параметров нечетких классификаторов / М.А. Мех, И.А. Ходашинский // Известия Российской академии наук. Теория и системы управления. – 2017. – № 4. – С. 65–75.

136. Ходашинский И.А. Алгоритмы «стадо криля» и кусочно-линейной инициализации для построения систем типа Такаги-Сугено / И.А. Ходашинский, К.С. Сарин, И.В. Филимоненко // Автометрия. – 2017. – Т. 53, № 4. – С. 84–94.

137. Hodashinsky I.A. Krill herd and piecewise-linear initialization algorithms for designing Takagi–Sugeno systems / I.A. Hodashinsky, I.V. Filimonenko, K.S. Sarin // Optoelectronics, Instrumentation and Data Processing. – 2017. – Vol. 53, No. 4. – P. 379–387.

138. Сарин К.С. Построение ансамблей нечетких классификаторов на основе метаэвристики «кукушкин поиск» и горной кластеризации / К.С. Сарин, С.А. Ворожцов, С.Н. Аримпилов // Электронные средства и системы управления. Материалы докладов Международной научно-практической конференции. – 2017. – № 1-2. – С. 26–29.

139. Ходашинский И.А. Аутентификация пользователя по динамике подписи на основе нечеткого классификатора / И.А. Ходашинский, Е.Ю. Костюченко, К.С. Сарин, А.Е. Анфилофьев, М.Б. Бардамова, С.С. Самсонов, И.В. Филимоненко // Компьютерная оптика. – 2018. – Т. 42, № 4. – С. 657–666.

140. Синюк В.Г. Алгоритмическое и программное обеспечение средств создания интеллектуальных проблемно-ориентированных систем, основанных на нечеткой логике / В. Г. Синюк, В.М. Поляков, М.В. Панченко // Вестник Белгородского государственного технологического университета им. В.Г. Шухова. – 2013. – № 3. – С. 159–161.

141. Bardamova M. Population Generation Methods for Metaheuristic Algorithms Used to Construct Compact Fuzzy Classifiers of Medical Data / M. Bardamova, M. Svetlakov, K. Sarin, A. Hodashinskaya, Y. Shurygin, I. Hodashinsky // Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications. – 2024. – Vol. 34, No. 3. – P. 396–411.

142. Fayyad U. From data mining to knowledge discovery in databases / U. Fayyad, G. Piatetsky-Shapiro, P. Smyth // *AI Magazine*. – 1996. – Vol. 17, No. 3. – P. 37–53.

143. Горбунов И.В. Методы построения трехкритериальных Парето-оптимальных нечетких классификаторов / И.В. Горбунов, И.А. Ходашинский // *Искусственный интеллект и принятие решений*. – 2015. – № 2. – С. 75–87.

144. Magdalena L. Fuzzy systems interpretability: what, why and how / L. Magdalena // *Fuzzy Approaches for Soft Computing and Approximate Reasoning* / M.-J. Lesot, C. Marsala (eds.), 2021. – *Studies in Fuzziness and Soft Computing*. – Vol. 394. – P. 111–122.

145. Guillaume S. Designing inference systems from data: an interpretability-oriented review // *IEEE Transactions on Fuzzy Systems*. – 2001. – Vol. 9, No. 3. – P. 426–443.

146. Mencar C. Some fundamental interpretability issues in fuzzy modeling / C. Mencar, G. Castellano, A.M. Fanelli // *Proceedings 4th Conference of the European Society for Fuzzy Logic and Technology and 11th French Days on Fuzzy Logic and Applications, EUSFLAT-LFA 2005 Joint Conference*. – 2005 – P. 100–105.

147. Sarin K. Extracting Knowledge from Images of Meanders and Spirals in the Diagnosis of Patients with Parkinson's Disease / K. Sarin, I. Hodashinsky, M. Svetlakov // *Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications*. – 2022. – Vol. 32, No. 3. – P. 658–664.

148. Ходашинский И.А. Биометрические данные и методы машинного обучения в диагностике и мониторинге нейродегенеративных заболеваний: обзор / И.А. Ходашинский, К.С. Сарин, М.Б. Бардамова, М.О. Светлаков, А.О. Слэзкин, Н.П. Корышев // *Компьютерная оптика*. – 2022. – Т. 46, № 6. – С. 988–1020.

149. Ходашинский И.А. Нечеткие классификаторы для диагностики болезни Паркинсона на основе статических рукописных данных / И.А. Ходашинский, Ю.А. Шурыгин, К.С. Сарин, М.Б. Бардамова, А.О. Слэзкин, М.О. Светлаков, Н.П. Корышев // *Автометрия*. – 2023. – Т. 59, № 3. – С. 72–85.

150. Hodashinsky I.A. Fuzzy Classifiers for Diagnosing of Parkinson's Disease Based on Static Handwritten Data / I.A. Hodashinsky, Yu.A. Shurygin, K.S. Sarin, M.B. Bardamova, A.O. Slezkin, M.O. Svetlakov, N.P. Korishev // *Optoelectronics, Instrumentation and Data Processing*. – 2023. – Vol. 59, No. 3. – P. 346–357.

151. Бардамова М.Б. Генерация базы правил нечеткого классификатора для диагностики болезни Паркинсона по рукописным данным / М.Б. Бардамова, И.А. Ходашинский, Ю.А. Шурыгин, К.С. Сарин, М.О. Светлаков // *Искусственный интеллект и принятие решений*. – 2023. – № 2. – С. 31–44.

152. Fisher R. The use of multiple measurements in taxonomic problems / F. Fisher // *Annals of Eugenics*. – 1936. – Vol. 7, No. 2. – P. 179–188.

153. Gacto M.J. Interpretability of linguistic fuzzy rule-based system: An overview of interpretability measures / M.J. Gacto, R. Alcalá, F. Herrera // *Information Science*. – 2011. – Vol. 181. – P. 4340–4360.

154. Garcia S. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability / S. Garcia, A. Fernandez, J. Luengo, F. Herrera // *Soft Computing*. – 2009. – Vol. 13. – P. 959–977.

155. Ishibuchi H. Voting in fuzzy rule-based systems for pattern classification problem / H. Ishibuchi, T. Narashima, T. Morisawa // *Fuzzy sets and systems*. – 1999. – Vol. 103, No. 1. – P. 223–238.

156. Hastie T. *The Elements of Statistical Learning* / T. Hastie, R. Tibshirani, J. H. Friedman. – New York: Springer, 2009. – 745 p.

157. Garcia S. *Data preprocessing in data mining* / S. Garcia, J. Luengo, F. Herrera. – New-York: Springer, 2015. – 320 p.

158. Pyle D. *Data preparation for data mining* / D. Pyle. – San Francisco: Morgan Kaufman, 1999. – 540 p.

159. He H. Learning from imbalanced data / H. He, E.A. Garcia // *IEEE Transactions on Knowledge and Data Engineering*. – 2009. – Vol. 21. – P. 263–284.

160. Рыков А.С. Системный анализ: модели и методы принятия решений и поисковой оптимизации / А.С. Рыков. – М.: Издательский Дом МИСиС, 2009. – 608 с.

161. Cavazzuti M. Optimization methods: from theory to design / Cavazzuti M. – Berlin: Springer, 2013. – 262 p.
162. Corriou J.-P. Numerical methods and optimizations / J.-P. Corriou. – Cham: Springer, 2021. – 715 p.
163. Shrestha A. Review of Deep Learning Algorithms and Architectures / A. Shrestha, A. Mahmood // IEEE Access. – 2019. – Vol. 7. – P. 53040–53065.
164. Alzubaidi L. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions / L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. Fadhel, M. Al-Amidie, L. Farhan. // Journal of Big Data. – Vol. 8, No. 53. – 2021 – P. 1–74.
165. Cerulli M. Aircraft Conflict Resolution / M. Cerulli, M. Pelegrin, S. Cafieri, C. D'Ambrosio, D. Rey // Encyclopedia of Optimization. Pardalos P.M., Prokopyev O.A. (eds). – Cham: Springer, 2023. – P. 1–8.
166. Characklis G. Developing portfolios of water supply transfers / G. Characklis, B. Kirsch, J. Ramsey, K. Dillard, C. Kelley // Water Resources Research. – 2006. – Vol. 42, No. 5. – P. 1–14.
167. Chen X. A smoothing direct search method for Monte Carlo-based bound constrained composite nonsmooth optimization / X. Chen, C. Kelley, F. Xu, Z. Zhang // SIAM Journal on Scientific Computing. – 2018. – Vol. 40, No. 4. – P. A2174-A2199.
168. Habib M. Multi-objective Particle Swarm Optimization: Theory, Literature Review, and Application in Feature Selection for Medical Diagnosis / M. Habib, I. Aljarah, H. Faris, S. Mirjalili // Evolutionary Machine Learning Techniques, Mirjalili S., Faris H., Aljarah I. (eds). Algorithms for Intelligent Systems. Springer, Singapore. – 2020. – P. 175–201.
169. Rani J.A.E. Supervised Hybrid Particle Swarm Optimization with Entropy (PSO-ER) for Feature Selection in Health Care Domain / J.A.E. Rani, E. Kirubakaran, S. Juliet, B.S.E. Zoraida // International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing. Springer, Singapore. – 2022. – Vol. 1387. – P. 797–805.

170. Burachik R.S. Algorithms for generating Pareto fronts of multi-objective integer and mixed-integer programming problems / R.S. Burachik, C.Y. Kaya, M.M. Rizvi // *Engineering optimization*. – 2022. – Vol. 54, No. 8. – P. 1413–1425.

171. Wang F. A particle swarm optimization algorithm for mixed-variable optimization problems / F. Wang, H. Zhang, A. Zhou // *Swarm and evolutionary computation*. – 2021. – Vol. 60. – P. 100808.

172. Ходашинский И.А. Методы повышения эффективности роевых алгоритмов оптимизации / И.А. Ходашинский // *Автоматика и телемеханика*. – 2021. – № 6. – С. 3–45.

173. *Handbook of metaheuristics* / Editors: M. Gendreau, J.-Yv. Potvin. – Cham: Springer, 2019. – 604 p.

174. Coello C. Handling multiple objectives with particle swarm optimization / C. Coello, G. Pulido, M. Lechuga // *IEEE Transactions on Evolutionary Computation*. – 2004. – Vol. 8, No. 3. – P.256–279.

175. Mirjalili S. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization / S. Mirjalili, S. Saremi, S. Mirjalili, L. Coelho // *Expert Systems with Applications*. – 2016. – Vol. 47. – P. 106–119.

176. Premkumar M. Multi-Objective Grey Wolf Optimization Algorithm for Solving Real-World BLDC Motor Design Problem / M. Premkumar, P. Jangir, B. Kumar M. Alqudah, K. Nisar // *Computers, Materials & Continua*. – 2022. – Vol.70, No. 2. – P. 2435–2452.

177. Mirjalili S. Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems / S. Mirjalili, P. Jangir, S. Saremi // *Applied Intelligence* – 2017. – Vol. 46. – P. 79–95.

178. Eichfelder G. A Solver for multiobjective mixed-integer convex and nonconvex optimization / G. Eichfelder, O. Stein, L. Warnow // *Journal of optimization theory and applications*. – 2023. – Vol. 200. – P. 1–31.

179. *Black box optimization, machine learning, and no-free lunch theorems* / Editors: P. M. Pardalos, V. Rasskazova, M. N. Vrahatis. – Cham: Springer, 2021. – 388 p.

180. Deb K. A fast and elitist multiobjective genetic algorithm: NSGA-II / K. Deb, S. Agrawal, A. Pratap, T. Meyarivan // IEEE Transactions on Evolutionary Computation. – 2002. – Vol. 6. – P. 182–197.

181. Murata T. Multi-objective genetic algorithm and its applications to flowshop scheduling / T. Murata, H. Ishibuchi, H. Tanaka // Computers and industrial engineering. – 1996. – Vol. 30, No. 4. – P. 957–968.

182. Wang W. An improved estimation of distribution algorithm for multi-objective optimization problems with mixed-variable / W. Wang, K. Li, H. Jalil, H. Wang // Neural Computing and Applications. – 2022. – Vol. 34. – P. 19703–19721.

183. Estimation of distribution algorithms: genetic algorithms and evolutionary computation / Editors: P. Larranaga, J. A. Lozano. – New York: Springer, 2002. – 383 p.

184. Bengoetxea E. Estimation of distribution algorithms: A new evolutionary computation approach for graph matching problems / E. Bengoetxea // Energy minimization methods in computer vision and pattern recognition. – Berlin: Springer, 2001. – P. 454–469.

185. Zhang Q. MOEA/D: A multiobjective evolutionary algorithm based on decomposition / Q. Zhang, H. Li // IEEE Transactions on evolutionary computation. – 2007. – Vol. 11, No. 6. – P. 712–731.

186. Tong W. A multi-objective mixed-discrete particle swarm optimization with multi-domain diversity preservation / W. Tong, S. Chowdhury, A. Messac // Structural and multidisciplinary optimization. – 2016. – Vol. 53. – P. 471–488.

187. Chowdhury S. A mixed-discrete particle swarm optimization algorithm with explicit diversity-preservation / S. Chowdhury, W. Tong, A. Messac, J. Zhang // Structural and multidisciplinary optimization. – 2013. – Vol. 47. – P. 367–388.

188. Mokarram V. A new PSO-based algorithm for multi-objective optimization with continuous and discrete design variables / V. Mokarram, M.R. Banan // Structural and multidisciplinary optimization. – 2018. – Vol. 57. – P. 509–533.

189. Coello Coello C.A. MOPSO: A proposal for multiple objective particle swarm optimization / C.A. Coello Coello, M.S. Lechuga // Proceedings of the 2002 Congress on Evolutionary Computation, CEC IEEE. – 2002. – P. 1051–1056.

190. Kennedy J. A discrete binary version of the particle swarm algorithm / J. Kennedy, R.C. Eberhart // Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, FL, USA, 1997. – Vol. 5. – P. 4104–4108.

191. Сарин К. С. Гибридный алгоритм смешанной многокритериальной оптимизации «кукушкин поиск» с генетическим оператором скрещивания / К. С. Сарин // Искусственный интеллект и принятие решений. – 2024. – № 2. – С. 87–105.

192. Свидетельство о государственной регистрации программы для ЭВМ № 2023684020 Российская Федерация. Программа смешанной многокритериальной оптимизации на основе метаэвристики "кукушкин поиск" и генетического оператора скрещивания : № 2023682275 : заявл. 27.10.2023 : опубл. 13.11.2023 / К. С. Сарин ; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники».

193. Yang X.-S., Deb S. Engineering optimisation by cuckoo search // International Journal Mathematical Modelling and Numerical Optimisation. – 2010. – Vol. 1, No. 4. – P. 330–343.

194. Yang X.-S., Deb S. Cuckoo search: recent advances and applications // Neural Computing and Applications. – 2014. – Vol. 24, No. 1. – P. 169–174.

195. Tang J. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends / J. Tang, G. Liu, Q. T. Pan // IEEE/CAA Journal of Automatica Sinica. – 2021. – Vol. 8, No. 10. – P. 1627–1643.

196. Viswanathan G. Levy flights in random searches / G. Viswanathan, V. Afanasyev, S. Buldyrev, S. Havlin, M. da Luz, E. Raposo, H. Stanley // Physica A: Statistical Mechanics and its Applications. – 2000. – Vol. 282. – P. 1–12.

197. Viswanathan G. Levy flight random searches in biological phenomena / G. Viswanathan, F. Bartumeus, S. Buldyrev, J. Catalan, U. Fulcoa, S. Havlind, M. da Luz, M.L. Lyra, E. Raposo, H. Stanley // *Physica A: Statistical Mechanics and its Applications*. – 2002. – Vol. 314. – P. 208–213.

198. Mantegna R. N. Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes // *Physical Review E*. – 1994. – Vol. 49, No. 5. – P. 4677–4683.

199. Сарин К.С. Дискретный алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений / К. С. Сарин // *Программирование*. – 2024. – № 6. – С. 35–47.

200. Sarin K.S. Discrete Optimization Algorithm Based on Probability Distribution with Transformation of Target Values / K.S. Sarin // *Programming and Computer Software*. – 2024. – Vol. 50, No. 6. – P. 445–456.

201. Свидетельство о государственной регистрации программы для ЭВМ № 2024668012 Российская Федерация. Программа дискретной оптимизации на основе распределения вероятностей с трансформацией целевых значений : № 2024667180 : заявл. 23.07.2024 : опубл. 01.08.2024 / К. С. Сарин ; заявитель Федеральное государственное автономное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники».

202. Aly R.H.M. Brain Tumors Diagnosis and Prediction Based on Applying the Learning Metaheuristic Optimization Techniques of Particle Swarm, Ant Colony and Bee Colony / R.H.M. Aly, K.H. Rahouma, H.F. Hamed // *Procedia Computer Science*. – 2019. – Vol. 163. – P. 165–179.

203. Phogat M. Classification of complex diseases using an improved binary cuckoo search and conditional mutual information maximization / M. Phogat, D. Kumar // *Computacion y Sistemas*. – 2020. – Vol. 24 – P. 1121–1129.

204. Houssein E.H. An efficient ECG arrhythmia classification method based on Manta ray foraging optimization / E.H. Houssein, I.E. Ibrahim, N. Neggaz, M.

Hassaballah, Y.M. Wazery // *Expert Systems with Applications*. – 2021 – Vol. 181 – P. 115131.

205. Aytimur A. Binary Artificial Bee Colony Algorithms for {0-1} Advertisement Problem / A. Aytimur, B. Babayigit // *Proceedings of the 2019 6th International Conference on Electrical and Electronics Engineering (ICEEE)*, Istanbul, Turkey, 2019. – P. 91–95.

206. Mohammadzadeh A. Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing / A. Mohammadzadeh, M. Masdari, F.S. Gharehchopogh, A. Jafarian // *Evolutionary Intelligence*. – 2021. – Vol. 14. – P. 1997–2025.

207. Pirozmand P. A novel approach for the next software release using a binary artificial algae algorithm / P. Pirozmand, A. Ebrahimnejad, H. Alrezaamiri, H. Motameni // *Journal of Intelligent and Fuzzy Systems*. – 2021. – Vol. 40. – P. 5027–5041.

208. Almonacid, B.; Aspee, F.; Soto, R.; Crawford, B.; Lama, J. Solving the manufacturing cell design problem using the modified binary firefly algorithm and the egyptian vulture optimisation algorithm / B. Almonacid, F. Aspee, R. Soto, B. Crawford, J. Lama // *IET Software*. – 2017. – Vol. 11. – P. 105–115.

209. El-Dakroury H.E.D.M. Load Restoration in Primary Distribution Networks Using the Binary Particle Swarm Optimization / H.E.D.M. El-Dakroury, A. Gad, A.Y. Abdelaziz // *Proceedings of the IEEE Electrical Power and Energy Conference (EPEC)*, Ottawa, ON, Canada, 2019. – P. 1–6.

210. Xiong G. A binary coded brain storm optimization for fault section diagnosis of power systems / G. Xiong, D. Shi, J. Zhang, Y. Zhang // *Electric Power Systems Research* – 2018. – Vol. 163. – P. 441–451.

211. Dahi Z.A.E.M. A 0-1 bat algorithm for cellular network optimisation: A systematic study on mapping techniques / Z.A.E.M. Dahi, C. Mezioud, A. Draa // *International Journal of Reasoning-based Intelligent Systems*. – 2017. – Vol. 9. – P. 22–42.

212. Hussien A.G. New binary whale optimization algorithm for discrete optimization problems / A.G. Hussien, A.E. Hassanien, E.H. Houssein, M. Amin, A.T. Azar // *Engineering Optimization*. – 2020. – Vol. 52. – P. 945–959.

213. Mourad K. A modified binary firefly algorithm to solve hardware/software partitioning problem / K. Mourad, R. Boudour // *Informatica*. – 2021. – Vol. 45. – P. 1–12.

214. Serigne G. A linearization framework for unconstrained quadratic (0–1) problems / G. Serigne, M. Philippe // *Discrete Applied Mathematics*. – 2009. – Vol. 157. – P. 1255–1266.

215. Sherali H.D. Evolution, and state-of-the-art in integer programming / H.D. Sherali, P.J. Driscoll // *Journal of Computational and Applied Mathematics*. – 2000. – Vol. 124. – P. 319–340.

216. Wolpert D.H., Macready W.G. No free lunch theorems for optimization. *IEEE* / D.H. Wolpert, W.G. Macready // *Transactions on Evolutionary Computation*. – 1997. – Vol. 1. – P. 67–82.

217. Becerra-Rozas M., Lemus-Romani J., Cisternas-Caneo F., Crawford B., Soto R., Astorga G., Castro C., Garcia J. Continuous Metaheuristics for Binary Optimization Problems: An Updated Systematic Literature Review // *Mathematics*. – 2022. – Vol. 11, No. 1. – P. 129.

218. Mirjalili S. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization / S. Mirjalili, A. Lewis // *Swarm and Evolutionary Computation*. – 2013. – Vol. 9. – P. 1–14.

219. Turkoglu B. Binary Artificial Algae Algorithm for feature selection / B. Turkoglu, S.A. Uymaz, E. Kaya // *Applied Soft Computing*. – 2022. – Vol. 120. – P. 108630.

220. Pashaei E. An efficient binary chimp optimization algorithm for feature selection in biomedical data classification / E. Pashaei, E. Pashaei // *Neural Computing and Applications*. – 2022. – Vol. 34. – P. 6427–6451.

221. Jain S. Memetic salp swarm optimization algorithm based feature selection approach for crop disease detection system / S. Jain, R. Dharavath // *Journal of Ambient Intelligence and Humanized Computing*. – 2023. – Vol. 14. – P. 1817–1835.

222. Mohd Yusof N. A novel nonlinear time-varying sigmoid transfer function in binary whale optimization algorithm for descriptors selection in drug classification / N. Mohd Yusof, A.K. Muda, S.F. Pratama, A. Abraham // *Molecular Diversity*. – 2023. – Vol. 27, No. 1. – P. 71–80.

223. Merikhi B. Automatic data clustering framework using nature-inspired binary optimization algorithms / B. Merikhi, M. Soleymani // *IEEE Access*. – 2021. – Vol. 9. – P. 93703–93722.

224. Zhong, C. Feature selection based on a novel improved tree growth algorithm / C. Zhong, Y. Chen, J. Peng // *International Journal of Computational Intelligence Systems*. – 2020. – Vol. 13. – P. 247.

225. Yepes V. Black hole algorithm for sustainable design of counterfort retaining walls / V. Yepes, J.V. Marti, J. Garcia // *Sustainability*. – 2020. – Vol. 12. – P. 2767.

226. Pandey A.C. Feature selection method based on hybrid data transformation and binary binomial cuckoo search. / A.C. Pandey, D.S. Rajpoot; M. Saraswat // *Journal of Ambient Intelligence and Humanized Computing*. – 2020. – Vol. 11. – P. 719–738.

227. Lai X. Diversity-preserving quantum particle swarm optimization for the multidimensional knapsack problem. / X. Lai, J.K. Hao, Z.H. Fu, D. Yue // *Expert Systems with Applications*. – 2020. – Vol. 149. – P. 113310.

228. Barani F. Application of binary quantum-inspired gravitational search algorithm in feature subset selection / F. Barani, M. Mirhosseini, H. Nezamabadi-Pour // *Applied Intelligence*. – 2017. – Vol. 47. – P. 304–318.

229. Ross O.H.M. A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers / O.H.M. Ross // *IEEE Access*. – 2019. – Vol. 8. – P. 814–838.

230. Shreem S.S. Enhanced binary genetic algorithm as a feature selection to predict student performance / S. S, Shreem, H. Turabieh, S. Al Azwari, F. Baothman // *Soft Computing*. – 2022. – Vol. 26. – P. 1811–1823.

231. Nicolau M. Application of a simple binary genetic algorithm to a noiseless testbed benchmark / M. Nicloau // *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. – 2009. – P. 2473–2478.

232. Haupt R.L., Haupt S.E. Practical genetic algorithms / R.L. Haupt, S.E. Haupt. – Hoboken, New Jersey: John Wiley & Sons, 2004. – 253 p.

233. Ghosh M. Binary Genetic Swarm Optimization: A Combination of GA and PSO for Feature Selection / M. Ghosh, R. Guha, I. Alam, P. Lohariwal, D. Jalan, R. Sarkar // *Journal of Intelligent Systems*. – 2019. – Vol. 29. – P. 1598–1610.

234. Bas E. A binary social spider algorithm for continuous optimization task / E. Bas, E. Ulker // *Soft Computing*. – 2020. – Vol. 24. – P. 12953–12979.

235. Mirjalili S. Binary bat algorithm / S. Mirjalili, S.M. Mirjalili, X.-S. Yang // *Neural Computing and Applications*. – 2014. – Vol. 25. – P. 663–681.

236. Pan J.-S. Binary fish migration optimization for solving unit commitment / J.-S. Pan, P. Hu, S.-C. Chu // *Energy*. – 2021. – V. 226. – P. 120329.

237. Сарин К.С. Адаптированный алгоритм градиентного спуска для настройки параметров нечеткого классификатора / Сарин К.С. // *Информационные и математические технологии в науке и управлении*. – 2025. – № 2(38). – С. 18–31.

238. Fausett L.V. Fundamentals of Neural Networks: Architectures, Algorithms, and Applications / L.V. Fausett. – Englewood Cliffs: Prentice-Hall, 1994. – 461 p.

239. Сарин К.С. Метод разделения данных для построения прогностических моделей машинного обучения / К.С. Сарин // *Доклады Томского государственного университета систем управления и радиоэлектроники*. – 2024. – Т. 27, № 4. – С. 88–96.

240. Hu X. Model complexity of deep learning: a survey / X. Hu, L. Chu, J. Pei, W. Liu, J. Bian // Knowledge and Information Systems. – 2021. – Vol. 63. – P. 2585–2619.

241. Wen W. Constructing Complexity Metrics for Measuring Generalization Ability of Deep Learning Models / W. Wen, W. Ke, J. Feng, S. Liu, Z. Xu, X. Sheng // 2024 10th International Conference on Big Data and Information Analytics (BigDIA), Chiang Mai, Thailand. – 2024. – P. 9–16.

242. Bulso N. On the complexity of logistic regression models / N. Bulso, M. Marsili, Y. Roudi // Neural Computation. – 2019. – Vol. 31, No. 8. – P. 1592–1623.

243. Hu X.B. Measuring Model Complexity of Neural Networks with Curve Activation Functions / X.B. Hu, W. Liu, J. Bian, J. Pei // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, 2020. – P. 1521–1531.

244. Buhrman H. Complexity measures and decision tree complexity: a survey / H. Buhrman, R. De Wolf // Theoretical computer science. – 2002. – Vol. 288, No. 1. – P. 21–43.

245. Hanin B. Complexity of linear regions in deep networks / B. Hanin, D. Rolnick // International conference on machine learning, PMLR. Long Beach, California, USA, 2019. – P. 2596–2604.

246. Ortigossa E.S. EXplainable Artificial Intelligence (XAI) – From Theory to Methods and Applications / E.S. Ortigossa, T. Goncalves, L.G. Nonato // IEEE Access. – 2024. – Vol. 12. – P. 80799–80846.

247. Chehreghani M.H. A Review on the Impact of Data Representation on Model Explainability / M.H. Chehreghani // ACM Computing Surveys. – 2024. – Vol. 56. – P. 1–21.

248. Ying X. An Overview of Overfitting and its Solutions // Journal of Physics: Conference Series. – 2019. – Vol. 1168, No. 2. – P. 1–6.

249. Ma C. Understanding and Mitigating Overfitting in Prompt Tuning for Vision-Language Models / C. Ma, Y. Liu, J. Deng, L. Xie, W. Dong, C. Xu // IEEE

Transactions on Circuits and Systems for Video Technology. – 2022. – Vol. 33. – P. 4616–4629.

250. Monica A Survey on Hyperparameter Optimization of Machine Learning Models / Monica, P. Agrawal // 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India. – 2024. – P. 11–15.

251. Yang L. On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice / L. Yang, A. Shami // Neurocomputing. – 2020. – Vol. 415. – P. 295–316.

252. Raji I.D. Simple Deterministic Selection-Based Genetic Algorithm for Hyperparameter Tuning of Machine Learning Models / I.D. Raji, H. Bello-Salau, I.J. Umoh, A.J. Onumanyi, M.A. Adegboye, A.T. Salawudeen // Applied Sciences. – 2022. – Vol. 12, No. 3. – P. 1186.

253. Bergstra J. Algorithms for hyper-parameter optimization / J. Bergstra, R. Bardenet, Y. Bengio, B. Kegl // Proceedings Advances in Neural Information Processing Systems, 2011. – P. 2546–2554.

254. James B. Random search for hyper-parameter optimization / B. James, B. Yoshua // Journal of Machine Learning Research. – 2012. – Vol. 13, No. 1. – P. 281–305.

255. Snoek J. Practical Bayesian optimization of machine learning algorithms / J. Snoek, H. Larochelle, R. Adams // Advances in Neural Information Processing Systems. – 2012. – Vol. 4. – P. 2951–2959.

256. Сарин К.С. Методика построения интерпретируемых нечетких классификаторов для систем объяснимого искусственного интеллекта / Сарин К.С. // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2025. – Т. 28, № 2. – С. 73–87.

257. Herrera F. A 2-tuple fuzzy linguistic representation model for computing with words / F. Herrera, L. Martinez-Lopez // IEEE Transactions on Fuzzy Systems. – 2000. – Vol. 8. – P. 746–752.

258. Ishibuchi H. Classification and Modeling with Linguistic Information Granules / H. Ishibuchi, T. Nakashima, M. Nii. – Berlin: Springer, 2005. – 308 p.

259. Derrac J. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms / J. Derrac, S. Garcia, D. Molina, F. Herrera // *Swarm and Evolutionary Computation*. – 2011. – Vol. 1. – P. 3–18.

260. Garcia S. A study on the use of nonparametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization / S. Garcia, D. Molina, M. Lozano, F. Herrera // *Journal of Heuristics*. – 2009. – Vol. 15, No. 6. – P. 617–644.

261. Demsar J. Statistical Comparisons of Classifiers over Multiple Data Sets / J. Demsar // *Journal of Machine Learning Research*. – 2006. – Vol. 7. – P. 1–30.

262. Garsia S. An Extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for all Pairwise Comparisons / S. Garsia, F. Herrera // *Journal of Machine Learning Research*. – 2008. – Vol. 9. – P. 2677–2694.

263. Garsia S. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power / S. Garsia, A. Fernandez, J. Luengo, F. Herrera // *Information Sciences*. – 2010. – Vol. 180. – P. 2044–2064.

264. Zitzler E. Comparison of multiobjective evolutionary algorithms: empirical results / E. Zitzler, K. Deb, L. Thiele // *Evolutionary Computation*. – 2000. – Vol. 2. – P. 173–195.

265. Deb. K. Scalable test problems for evolutionary multiobjective optimization / Deb K., Thiele L., Laumanns M., Zitzler E. // *Evolutionary Multiobjective Optimization, Theoretical Advances and Applications*. – London: Springer, 2005. – P. 105–145.

266. Zhang Q. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm / Q. Zhang, A. Zhou, Y. Jin // *IEEE Transactions on Evolutionary Computation*. – 2008. – Vol. 12, No. 1. – P. 41–63.

267. Herrera F. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis / F. Herrera, M. Lozano, J. L. Verdegay // *Artificial Intelligence Review*. – 1998. – Vol. 12. – P. 265–312.

268. Гланц С. Медико-биологическая статистика / С. Гланц. – М.: Практика, 1998. – 459 с.

269. Coppersmith D. Partitioning Nominal Attributes in Decision Trees / D. Coppersmith, S. J. Hong, J. R. M. Hosking // Data Mining and Knowledge Discovery. – 1999. – Vol. 3. – P. 197–217.

270. Triguero I. KEEL 3.0: An Open Source Software for Multi-Stage Analysis in Data Mining / I. Triguero, S. Gonzalez, J. M. Moyano, S. Garcia, J. Alcala-Fdez, J. Luengo, A. Fernandez, M. J. del Jesus, L. Sanchez, F. Herrera // International Journal of Computational Intelligence Systems. – 2017. – No. 10. – P. 1238–1249.

271. Wu J. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization / J. Wu, X. Chen, H. Zhang, L-D. Xiong, H. Lei, S. Deng // Journal of Electronic Science and Technology. – 2019. – Vol. 17. – P. 26–40.

272. Vendelin G.D. Microwave circuit design using linear and nonlinear techniques / G.D.Vendelin, A.M. Pavio, U.L. Rohde, M. Rudolp. – London: John Wiley & Sons, 2021. – 1180 p.

273. Besser L. Practical RF circuit design for modern wireless systems: Active circuits and systems. Volume 1 / L. Besser, R. Gilmore. – London-Boston: Artech House, 2003. – 539 p.

274. Besser L. Practical RF circuit design for modern wireless systems: Passive circuits and systems. Volume 2 / L. Besser, R. Gilmore. – London-Boston: Artech House, 2003. – 569 p.

275. Фано Р. Теоретические ограничения полосы согласования произвольных импедансов / пер. с англ. под ред. Г.И. Слободенюка. – М.: Советское радио, 1965. – 72 с.

276. Youla D. C. A new theory of broad-band matching / D. C. Youla // IEEE Transactions on Circuit Theory. –1964. – Vol. CT-11. – P. 30–50.

277. Carlin H. J. The double matching problem: Analytic and real frequency solutions / H. J. Carlin, B.S. Yarman // IEEE Transactions on Circuits and Systems. – 1983, Vol. CAS-30. – P. 15–28.

278. Бабак Л.И. «Визуальное» проектирование корректирующих и согласующих цепей полупроводниковых СВЧ-устройств. Часть 1. Описание процедуры проектирования / Л.И. Бабак, М.В. Черкашин, Д.А. Зайцев // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2007. – Т.14, № 6. – С. 11–23.

279. Бабак Л.И. «Визуальное» проектирование корректирующих и согласующих цепей полупроводниковых СВЧ - устройств Часть 2. Программная реализация и примеры / Л.И. Бабак, М.В. Черкашин, Д.А. Зайцев // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2007. – Т. 15, № 1. – С. 10–19.

280. Самуилов А.А. Методика «визуального» проектирования цепей на сосредоточенных элементах для широкополосного согласования двух комплексных нагрузок / А.А. Самуилов, М.В. Черкашин, Л.И. Бабак // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2013. – Т. 28, № 2. – С. 30–39.

281. Дорофеев С.Ю. Синтез согласующих цепей в распределенном и смешанном элементных базисах / С.Ю. Дорофеев, Л.И. Бабак // Современные проблемы радиоэлектроники. – Красноярск: Сибирский федеральный университет, 2008. – С. 98–101.

282. Yarman B.S. A simplified real frequency technique applied to broadband multistage microwave amplifiers / B.S. Yarman, H.J. Carlin // IEEE Transactions on Microwave Theory and Techniques. – 1982. – Vol. MTT-30, No. 12. – P. 2216–2222.

283. Пегасин Д.В. Синтез согласующих цепей с характеристиками передачи мощности заданного уровня на основе алгоритма Левенберга–Марквардта / Д.В. Пегасин // Доклады Белорусского государственного университета информатики и радиоэлектроники. – 2010. – № 3(49). – С. 17–23.

284. Sussman-Fort S.E. The computer-aided design of microwave matching networks / S.E. Sussman-Fort // International Journal of Microwave and Millimeter-Wave Computer-Aided Engineering. – 1991. – Vol. 1, No. 3. – P. 288–305.

285. Carlin H.J. On optimum broad-band matching / H. J. Carlin, P. Amstutz // IEEE Transactions on Circuits and Systems / H.J. Carlin. – 1981 – Vol. CAS-28, No. 5. – P. 401–405.

286. Du Plessis W.P. A genetic algorithm for impedance matching network design: Dissertation for the degree of Master of Engineering (Electronic) / Du Plessis W.P. – Pretoria: University of Pretoria, 2003. – 195 p.

287. Dedieu H. A New Method for Solving Broadband Matching Problems / H. Dedieu, C. Dehollain, J. Dehollain, G. Rhodes // IEEE Transactions on Circuits and Systems. – 1994. – Vol. 41, No. 9. – P. 561–571.

288. Abrie P.L. D. Design of RF and microwave amplifiers and oscillators / P.L. Abrie. – London-Boston: Artech House, 2000. – 480 p.

289. Самуилов А. А. Программа «визуального» проектирования корректирующих и согласующих цепей LOCUS на базе среды INDESYS / А. А. Самуилов // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2012. – Т. 26, №. 2-2. – С. 119–126.

ПРИЛОЖЕНИЕ А

Свидетельства о государственной регистрации программ для ЭВМ



Рисунок А.1 – Свидетельство о государственной регистрации программы смешанной многокритериальной оптимизации на основе метаэвристики «кукушкин поиск» и генетического оператора скрещивания



Рисунок А.2 – Свидетельство о государственной регистрации программы дискретной оптимизации на основе распределения вероятностей с трансформацией целевых значений

ПРИЛОЖЕНИЕ Б

Акты о внедрении (использовании) результатов диссертационного исследования

<p>ФМБА РОССИИ</p> <p>ФГБУ ФНКЦ МРиК ФМБА России</p> <p>«Томский научно-исследовательский институт курортологии и физиотерапии» филиал Федерального государственного бюджетного учреждения «Федеральный научно-клинический центр медицинской реабилитации и курортологии Федерального медико-биологического агентства» (Томский НИИКиФ)</p> <p>ФГБУ ФНКЦ МРиК ФМБА России) 634009, г. Томск, ул. Р. Люксембург, д. 1 Тел.: (3822) 512-005, Факс: (3822) 512-115 e-mail: niikf@niikf.tomsk.ru ОКПО, ОГРН, 42294702, 1035008852944 ИНН/КПП 5044013246/701743001</p>	<p>УТВЕРЖДАЮ</p> <p>И.о. директора Томского НИИКиФ ФГБУ ФНКЦ МРиК ФМБА России Н.М. Юрьева</p> <p>_____ 2025 г.</p> 
---	---

АКТ

об использовании результатов научных исследований, полученных в рамках диссертационной работы Сарина Константина Сергеевича

Комиссия в составе: заместителя директора по науке ТНИИКиФ ФГБУ ФНКЦ МРиК ФМБА России, канд. мед. наук А.В. Тонкошкурова, заведующего лабораторией планирования медицинских технологий ТНИИКиФ ФГБУ ФНКЦ МРиК ФМБА России, д-ра мед. наук И.Н. Смирновой, главного научного сотрудник лаборатории планирования медицинских технологий ТНИИКиФ ФГБУ ФНКЦ МРиК ФМБА России, д-ра мед. наук Е.В. Тицкой

составила настоящий акт, подтверждающий факт использования в Томском НИИКиФ ФГБУ ФНКЦ МРиК ФМБА России результатов диссертационного исследования К.С. Сарина. В рамках диссертационного исследования К. С. Сариным разработана методика построения интерпретируемых нечетких систем. Методика применена в НИР, выполняемой в рамках Государственного задания «Разработка персонализированных технологий комплексной реабилитации лиц, перенесших COVID-19» (шифр 84.005.22.800). С помощью методики создан нечеткий классификатор для прогнозирования эффективности реабилитации пациентов после перенесенной коронавирусной инфекции COVID-19 с объяснением полученного результата. Прогнозирование осуществляется на основе медицинских показателей здоровья. Исходными данными для построения классификатора являлись результаты клинического исследования пациентов с подтвержденным диагнозом коронавирусная инфекция COVID-19 и осложнением в виде двусторонней полисегментарной вирусной пневмонии.

Применение методики позволило добиться следующих результатов:

1. Осуществить прогноз эффективности реабилитации с точностью 74,67%.
2. Получить базу знаний из трех правил, объясняющую связь между показателями здоровья пациента и эффекта от реабилитации.
3. Понизить количество предикторов для прогноза эффективности реабилитации с 58 до 7.

	<p>Зам. директора по науке, канд. мед. наук _____ А.В. Тонкошкурова Зав. лабораторией ПРМТ, д-р мед. наук _____ И.Н. Смирнова Г.н.с. лаборатории ПРМТ, д-р мед. наук _____ Е.В. Тицкая</p>
--	--

Рисунок Б.1 – Акт об использовании результатов диссертационного исследования в Томском НИИКиФ

УТВЕРЖДАЮ

ректор ТУСУР

д.т.н., профессор



В.М. Рулевский
«15» июля 2025 г.

АКТ

о внедрении (использовании) результатов научных исследований
в рамках диссертационной работы Сарина Константина Сергеевича

Комиссия в составе: директора НИИ МЭС, д.т.н., Л.И. Бабака,
старшего научного сотрудник НИИ МЭС, к.т.н. М.В. Черкашина

составила настоящий акт, подтверждающий факт использования в НИИ МЭС ТУСУР результатов диссертационного исследования К.С. Сарина.

При выполнении диссертационного исследования К.С. Сариним разработан гибридный алгоритм смешанной многокритериальной оптимизации «кукушкин поиск» с генетическим оператором скрещивания и на его основе реализовано программное обеспечение (ПО) для автоматического структурно-параметрического синтеза фильтров и широкополосных согласующих цепей (СЦ).

ПО разработано в рамках выполнения проекта «Разработка технологий изготовления, методов и средств проектирования СВЧ и фотонных интегральных схем на базе полупроводниковых материалов GaN, GaAs, InP, Si и SiGe, а также радиочастотных и оптоэлектронных устройств и модулей на этой основе» в рамках программы «Приоритет-2030». Оно находится в стадии опытной эксплуатации в НИИ МЭС ТУСУР с декабря 2024 г. В настоящее время ПО используется при выполнении НИР «Разработка комплекта интегральных электронных компонентов и конструкций оптических приемопередающих модулей со скоростями не менее 5, 12 и 20 Гбит/с», реализуемой в рамках соглашения №23-91-006207 от 18.12.2-23 г. с Российским научным фондом. В данной НИР ПО применяется для синтеза фильтров и согласующих цепей с заданной формой частотной характеристики (ЧХ) коэффициента передачи по мощности в составе интегральных трансимпедансных усилителей и драйверов лазерных диодов для оптических приемопередатчиков. При этом пользователь может задавать требуемую форму ЧХ, произвольные импедансы генератора и нагрузки, количество, тип и способ включения элементов в составе СЦ. Алгоритм автоматически синтезирует несколько вариантов структурных схем СЦ, которые удовлетворяют заданным требованиям, ранжируя их по степени сложности (количеству элементов).

Использование данного ПО позволяет:

- 1) выполнить автоматический синтез фильтров или СЦ по заданным требованиям к форме ЧХ коэффициента передачи по мощности и произвольных импедансах генератора и нагрузки, которые могут быть заданы в табличной форме или в виде эквивалентной цепи;
- 2) получить множество вариантов структур СЦ с различной сложностью (количеством элементов), что дает возможность пользователю сделать выбор на основе компромисса между сложностью цепи, возможностью ее физической реализуемости и точностью воспроизведения требуемой формы АЧХ;
- 3) решить задачу структурно-параметрического синтеза фильтров и СЦ в автоматическом режиме, что позволяет ускорить процесс разработки радиоэлектронных устройств;
- 4) выполнить проектирование радиоэлектронных устройств с высокими техническими характеристиками.

Разработанный алгоритм и ПО использовались в НИИ МЭС ТУСУР при проектировании СВЧ усилителей различного назначения.

От разработчиков:

доцент каф КСУП ТУСУР, к.т.н.



К.С. Сарин

«15» июля 2025 г.

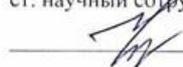
Представители НИИ МЭС ТУСУР:

директор НИИ МЭС ТУСУР, д.т.н.



Л.И. Бабак

ст. научный сотрудник НИИ МЭС, к.т.н.



М.В. Черкашин

Рисунок Б.2 – Акт о внедрении (использовании) результатов диссертационного исследования в НИИ МЭС



АКТ
о внедрении (использовании) результатов научных исследований
Сарина Константина Сергеевича

Комиссия в составе:
зам. директора ООО «Микроэлектронные системы», канд. техн. наук, А.А. Коколова
директора ООО «Микроэлектронные системы», М.П. Сухорукова

составила настоящий акт, подтверждающий факт использования в ООО «Микроэлектронные системы» программного продукта, разработанного К.С. Сариным в ходе выполнения диссертационного исследования.

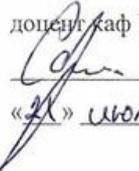
Программа для автоматического структурно-параметрического синтеза фильтров и широкополосных согласующих цепей (СЦ), в основе которой лежит предложенный К.С. Сариным гибридный алгоритм смешанной многокритериальной оптимизации с генетическим оператором скрещивания, применялся в ООО «Микроэлектронные системы» для проектирования СВЧ радиоэлектронных устройств, выполняемых в виде монолитных интегральных схем.

Применение данного программного продукта позволяет в автоматическом режиме выполнять структурно-параметрический синтез пассивных фильтров и цепей согласования в составе многокаскадных радиоэлектронных устройств и получать схемы, близкие к оптимальным с точки зрения воспроизведения заданной формы частотной зависимости коэффициента передачи по мощности, а также сложности синтезируемой цепи.

В частности, разработанная К.С. Сариным программа использовалась в ООО «Микроэлектронные системы» при проектировании малошумящего СВЧ усилителя диапазона частот 12-17 ГГц, выполненного на основе отечественной 0,5 мкм GaAs технологии АО «Светлана-Рост», (свидетельство о регистрации ТИМС № 2024630197 от 14 октября 2024 г.)

От разработчиков:

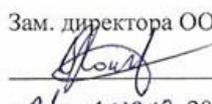
доцент каф КСУП ТУСУР, к.т.н.

 К.С. Сарин

«21» июля 2025 г.

От ООО «Микроэлектронные системы»:

Зам. директора ООО «Микроэлектронные системы»

 А.А. Коколов

«21» июля 2025 г.

Рисунок Б.3 – Акт о внедрении (использовании) результатов диссертационного исследования в ООО «Микроэлектронные системы»

УТВЕРЖДАЮ
Директор ООО «50ом Тех.»
_____ Калентьев А.А.



« 09 » _____ 2025 г.

АКТ

о внедрении результатов диссертационной работы Сарина Константина Сергеевича

Комиссия в составе:

ведущего программиста, к.т.н. Горяинова А.Е.,
ведущего инженера, к.т.н. Добуша И.М.,
ведущего научного сотрудника, к.т.н. Сальникова А.С.,

составила настоящий акт, подтверждающий факт внедрения в компании ООО «50ом Тех.» результатов диссертационного исследования К.С. Сарина. Настоящий акт подтверждает факт использования алгоритма в информационной среде компании, без передачи исключительных прав и обязательств сторон.

В диссертационном исследовании Сариним К.С. разработан дискретный алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений. Алгоритм осуществляет поиск оптимума в пространстве, где переменные принимают дискретные значения.

Разработанный алгоритм внедрен в экспериментальную версию интеллектуальную САПР «50ohm Tech Circuit Studio» и испытан в рамках опытной эксплуатации для синтеза секции ступенчатого СВЧ аттенюатора.

Испытание в рамках опытной эксплуатации позволило добиться следующих результатов:

1. В 56 запусках алгоритма из 100 для задачи синтеза ступенчатого аттенюатора находится приемлемое решение с максимальным числом итераций 10000.
2. Среднее время синтеза составило 3 мин 40 сек, что находится в приемлемых для поставленной практической задачи пределах.

Тем самым, подтверждено, что разработанный алгоритм работоспособен и может применяться для решения практических задач синтеза СВЧ устройств.

Ведущий программист, к.т.н.

_____ Горяинов А.Е.

Ведущий инженер, к.т.н.

_____ Добуш И.М.

Ведущий научный сотрудник, к.т.н.

_____ Сальников А.С.

Рисунок Б.4 – Акт о внедрении результатов диссертационного исследования в ООО «50ом Тех.»

ТУСУР

Министерство науки и высшего образования Российской Федерации
 Федеральное государственное автономное образовательное
 учреждение высшего образования
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ
 УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
 И РАДИОЭЛЕКТРОНИКИ**

ОКПО 02069326, ОГРН 1027000867068,
 ИНН 7021000043, КПП 701750001

пр. Ленина, д. 40, г. Томск, 634050

тел: (382 2) 510-530
 факс: (382 2) 513-262, 526-365
 e-mail: office@tusur.ru
 http://www.tusur.ru



УТВЕРЖДАЮ

Проректора по учебной работе
 и международной деятельности

И.Л. Ларионова

«5» августа 2025 г.

АКТ

о внедрении результатов диссертационной работы Сарина Константина Сергеевича
 в учебный процесс

Комиссия в составе:

председателя

Шурыгина Ю.А., заведующего кафедрой КСУИ, д-ра техн. наук,

членов комиссии

Хабибулиной П.Ю., декана факультета вычислительных систем, канд. техн. наук,

Копубинского В.П., доцента каф. КСУИ ТУСУР, канд. техн. наук

составила настоящий акт о нижеследующем.

Результаты диссертационной работы Сарина К.С. на тему «Методология построения интерпретируемых нечетких классификаторов, основанных на правилах» используются в учебном процессе на факультете вычислительных систем Томского государственного университета систем управления и радиоэлектроники при чтении курса лекций и проведении лабораторных работ для подготовки студентов, обучающихся по направлению «09.03.01 – Информатика и вычислительная техника».

Разработанная Сариним К.С. методика построения нечетких классификаторов включена в лекционный материал по теме «Нечеткие системы». Программные средства, реализующие разработанные Сариним К.С. алгоритм смешанной многокритериальной оптимизации «кукушкин поиск» с генетическим оператором скрещивания, дискретный алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений, адаптированный алгоритм градиентного спуска, применяются в лабораторных работах, посвященных построению нечетких систем на основе анализа данных. Темы

Рисунок Б.5 – Акт о внедрении результатов диссертационного исследования в учебный процесс (страница 1)

лабораторных работ «Метаэвристические методы оптимизации» и «Построение интерпретируемых моделей машинного обучения».

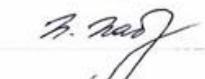
Результаты диссертационной работы включены в одну лекцию и две лабораторные работы по дисциплинам «Искусственный интеллект-1», «Искусственный интеллект-2». «Прикладные методы искусственного интеллекта» на кафедре компьютерных систем в управлении и проектировании Томского государственного университета систем управления и радиоэлектроники.

Заведующий кафедрой КСУИ,
д-р. техн. наук., профессор



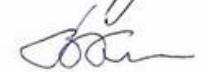
Шурьгин Ю.А.

Декан факультета вычислительных
систем, канд. техн. наук, доцент



Хабидулина Н.Ю.

Доцент каф. КСУИ,
канд. техн. наук, доцент



Козубинский В.И.

Рисунок Б.6 – Акт о внедрении результатов диссертационного исследования в учебный процесс (страница 2)

ПРИЛОЖЕНИЕ В

Признаки набора данных для прогнозирования эффективности реабилитации

Таблица В.1 – Признаки набора данных для прогнозирования эффективности реабилитации пациентов после перенесенной инфекции COVID-19

№	Обозначение в наборе данных	Расшифровка	Единицы	Тип	Границы
1	Probiotic	Пробиотик	-	Бин.	[0; 1]
2	Year	Возраст	годы	Целочисл.	[37; 72]
3	Sex	Пол	-	Бин.	[0; 1]
4	Rost	Рост	См	Целочисл.	[151; 182]
5	Ves	Вес	Кг	Целочисл.	[46; 131]
6	nasled_ssz	Предрасположенность к ССЗ	-	Бин.	[0; 1]
7	Kurenje	Курение	-	Бин.	[0; 1]
8	Alkohol	Злоупотребление алкоголем	-	Бин.	[0; 1]
9	Gipodinamia	Гиподинамия	-	Бин.	[0; 1]
10	Kt	Степень поражения легких	баллы	Целочисл.	[1; 3]
11	Day	День поступления после перенесенного заболевания	дни	Целочисл.	[10; 60]
12	Soputstv	Наличие сопутствующих заболеваний	-	Бин.	[0; 1]
13	Slabost	Степень выраженности жалоб на слабость	баллы	Целочисл.	[1; 3]
14	heart_new	Наличие жалоб на сердечно-сосудистую систему	-	Бин.	[0; 1]
15	Breath	Одышка	баллы	Целочисл.	[1; 3]
16	Cough	Кашель	-	Бин.	[0; 1]
17	nevrologia_new	Наличие жалоб по неврологии	-	Бин.	[0; 1]
18	gkt_new	Наличие жалоб на желудочно-кишечный тракт	-	Бин.	[0; 1]
19	Genchi	Проба Генчи	с	Целочисл.	[10; 35]
20	Shtange	Проба Штанге	С	Целочисл.	[7; 50]
21	Mmrs	Шкала mMRS	баллы	Целочисл.	[1; 3]
22	spo2	Сатурация крови	%	Целочисл.	[96; 99]
23	t6	Результат теста шестиминутной ходьбы	метры	Целочисл.	[220; 460]
24	ad_up	Артериальное давление систолическое	мм рт. ст.	Целочисл.	[100; 165]
25	ad_down	Артериальное давление диастолическое	мм рт. ст.	Целочисл.	[60; 100]
26	heart_rate	ЧСС	уд. в мин	Целочисл.	[54; 125]
27	Gel	Жизненная емкость легких	%	Целочисл.	[62; 151]

Продолжение Таблицы В.1

№	Обозначение в наборе данных	Расшифровка	Единицы	Тип	Границы
28	Ofv	Объем форсированного выдоха	%	Целочисл.	[52; 123]
29	Tiffo	Проба Тиффно	%	Целочисл.	[37; 133]
30	Psv	Пиковая скорость выдоха	%	Целочисл.	[21; 123]
31	mos50	Объемная форсированная скорость выдоха при выдохе 50% объема	%	Целочисл.	[20; 152]
32	mos25	Объемная форсированная скорость выдоха при выдохе 25% объема	%	Целочисл.	[20; 133]
33	Sos	Средняя объемная скорость форсированного выдоха при выдохе 25-75% объема	%	Целочисл.	[17; 232]
34	Hb	Гемоглобин	г/л	Целочисл.	[118; 168]
35	Erythrocytes	Эритроциты	10^{12} литр	Веществ.	[4,0; 5,3]
36	Leukocytes	Лейкоциты	10^9 литр	Веществ.	[3,1; 12,0]
37	Lymphocytes	Лимфоциты	%	Целочисл.	[18; 59]
38	Eosinophils	Эозинофилы	%	Целочисл.	[1; 12]
39	Soe	Скорость оседания эритроцитов	мм в час	Целочисл.	[3; 40]
40	palochki_neutrophils	Палочкоядерные нейтрофилы	%	Целочисл.	[0; 7]
41	segm_neutrophils	Сегментоядерные нейтрофилы	%	Целочисл.	[33; 70]
42	Monocytes	Моноциты	%	Целочисл.	[4; 15]
43	Glucose	Глюкоза	ммоль/л	Веществ.	[4,25; 13]
44	Fibrinogen	Фибриноген	г/л	Веществ.	[1,8; 5,4]
45	Cholesterol	Холестерин	ммоль/л	Веществ.	[2,63; 7,13]
46	cholesterol_vp	Холестерин липопротеидов высокой плотности	ммоль/л	Веществ.	[0,75; 2,48]
47	cholesterol_np	Холестерин липопротеидов низкой плотности	ммоль/л	Веществ.	[1,18; 5,9]
48	Tag	Триглицериды	ммоль/л	Веществ.	[0,5; 3,32]
49	not_ch_vp	Не ХС ЛПВП	ммоль/л	Веществ.	[1,3; 5,76]
50	creatinine	Креатинин	мкмоль/л	Целочисл.	[46; 125]
51	Skf	Скорость клубочковой фильтрации	мл/мин/ $1,73 \text{ м}^2$	Целочисл.	[42; 102]
52	Ast	Аспартатаминотрансфераза	ед/л	Веществ.	[12; 128]
53	Alt	Аланинаминотрансфераза	ед/л	Веществ.	[8,9; 188]
54	Aptv	Активированное частичное тромбопластиновое время	С	Веществ.	[26; 43]

Продолжение Таблицы В.1

№	Обозначение в наборе данных	Расшифровка	Единицы	Тип	Границы
55	Mno	Международное нормализованное отношение	-	Веществ.	[0,87; 1,59]
56	moch_c	Мочевая кислота	мкмоль/л	Целочисл.	[160; 637]
57	Dd	D-димеры	нг/мл	Целочисл.	[0; 2366]
58	Borg	Шкала Борга	баллы	Целочисл.	[0; 6]